



NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

**EVOLVING A SIMULATION MODEL PRODUCT LINE
SOFTWARE ARCHITECTURE FROM HETEROGENEOUS
MODEL REPRESENTATIONS**

by

Kevin J. Greaney

September 2003

Dissertation Supervisors:

Luqi
James B. Michael

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2003	3. REPORT TYPE AND DATES COVERED Dissertation	
4. TITLE AND SUBTITLE: Evolving a Simulation Model Product Line Software Architecture from Heterogeneous Model Representations			5. FUNDING NUMBERS	
6. AUTHOR(S) Kevin J. Greaney				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Missile Defense Agency Washington, DC 20301-7100			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
ABSTRACT (maximum 200 words) National- and Department-level decision-makers expect credible Department of Defense models and simulations (M&S) to provide them confidence in the simulation results, especially for mission-critical and high-risk decisions supporting National Security. Many of these large-scale, software-intensive simulation systems were autonomously developed over time, and subject to varying degrees of funding, maintenance, and life-cycle management practices, resulting in heterogeneous model representations and data. Systemic problems with distributed interoperability of these non-trivial simulations in federations' persist, and current techniques, procedures, and tools have not achieved the desired results. The Software Architecture-Based Product Line for simulation model representations, employing Architecture Readiness Levels presented in this dissertation provides an alternative methodology. The proposed four-layered M&S software architecture-based product line model enables the development of model representations supported by readiness levels. Each layer reflects a division of the software architecture-based product line. The layer represents a horizontal slice through the architecture for organizing viewpoints or views at the same level of abstraction while the software architecture-based product line represents a vertical slice. A layer may maintain multiple views and viewpoints of a software architecture-based product line. A Domain Metadata Repository prescribes the interaction between layers. We introduce the Domain Integrated Product Development Team concept.				
14. SUBJECT TERMS Models and Simulations, software architecture, product lines, architecture description languages (ADL), Extensible Markup Language (XML), verification, validation, readiness levels, interoperability, heterogeneous model representations and data, credibility, confidence, distributed development			15. NUMBER OF PAGES 558	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

EVOLVING A SIMULATION MODEL PRODUCT LINE SOFTWARE ARCHITECTURE FROM HETEROGENEOUS MODEL REPRESENTATIONS

Kevin J. Greaney
Colonel, United States Army
B.A., Northeastern University, 1974
M.S., Shippensburg University, 1977
M.A., Webster University, 1992

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN SOFTWARE ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
September 2003**

Author:

Kevin J. Greaney

Approved by:

Luqi
Professor of Computer Science
Dissertation Committee Chair*
Dissertation Co-Advisor

Valdis Berzins
Professor of Computer Science

James B. Michael
Professor of Computer Science
Dissertation Co-Advisor

John A. Hamilton, Jr.,
Professor of Computer Science
Auburn University

Patricia A. Sanders
System Executive Officer
Missile Defense Agency

Approved by:

Peter Denning, Chair, Department of Computer Science

Approved by:

Carson K. Eoyang, Associate Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

National- and Department-level decision-makers expect credible Department of Defense models and simulations (M&S) to provide them confidence in the simulation results, especially for mission-critical and high-risk decisions supporting National Security. Many of these large-scale, software-intensive simulation systems were autonomously developed over time, and subject to varying degrees of funding, maintenance, and life-cycle management practices, resulting in heterogeneous model representations and data. Systemic problems with distributed interoperability of these non-trivial simulations in federations' persist, and current techniques, procedures, and tools have not achieved the desired results.

The Software Architecture-Based Product Line for simulation model representations, employing Architecture Readiness Levels presented in this dissertation provides an alternative methodology. The proposed four-layered M&S software architecture-based product line model enables the development of model representations supported by readiness levels. Each layer reflects a division of the software architecture-based product line. The layer represents a horizontal slice through the architecture for organizing viewpoints or views at the same level of abstraction while the software architecture-based product line represents a vertical slice. A layer may maintain multiple views and viewpoints of a software architecture-based product line. A Domain Metadata Repository prescribes the interaction between layers. We introduce the Domain Integrated Product Development Team concept.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	THE NEW STRATEGIC ROLE FOR MODELS AND SIMULATIONS	1
B.	AUTHORITATIVE REPRESENTATIONS AND CREDIBLE M&S.....	3
C.	LIMITATIONS OF CURRENT APPROACHES TO ACHIEVE CREDIBILITY.....	4
D.	RESEARCH QUESTION AND HYPOTHESIS	7
E.	SIMULATION SOFTWARE ARCHITECTURE (SSA) OVERVIEW	8
F.	RESEARCH CONTRIBUTIONS	10
G.	SIGNIFICANCE OF DISSERTATIONS CONTRIBUTIONS.....	12
H.	DISSERTATION VISION	13
I.	DISSERTATION ORGANIZATION	15
II.	LITERATURE SEARCH AND DOMAIN M&S BACKGROUND	19
A.	INTRODUCTION.....	19
B.	LITERATURE SEARCH	19
C.	THE DEPARTMENT'S MODEL AND SIMULATION FRAMEWORK.....	22
1.	Background	22
2.	Classes of Simulations.....	24
3.	Simulation Model Hierarchies	25
4.	Distributed Simulations.....	28
5.	The Growing Need for Credible Models and Simulations	29
a.	<i>The Department's Acquisition Domain</i>	<i>29</i>
b.	<i>The Department's Training Domain.....</i>	<i>34</i>
c.	<i>The Department's Operations Domain</i>	<i>34</i>
d.	<i>The Department's Analytical Domain</i>	<i>36</i>
e.	<i>The Department's Experimental Domain.....</i>	<i>37</i>
D.	SUMMARY	38
III.	CREDIBILITY OF DEPARTMENT SYSTEM REPRESENTATIONS	41
A.	INTRODUCTION.....	41
B.	THE GROWTH ERA: M&S CREDIBILITY ISSUES IDENTIFIED.....	41
C.	THE MANAGED ERA FOR CREDIBLE DEPARTMENT M&S	42
D.	ESTABLISHING DOD M&S CREDIBILITY AND USERS CONFIDENCE.....	47
E.	VERIFICATION, VALIDATION & ACCREDITATION PROCESSES	48
1.	The Department's Verification Process	51
2.	The Department's Validation Process	52
3.	The Department's Accreditation Process	54

4.	The Role of Data in the VV&A Process	55
5.	Verification and Validation (V&V) Techniques	59
6.	Shortfalls of the Department's VV&A Process.....	64
7.	Secondary Credibility Contributing Factors.....	66
a.	<i>Reuse</i>	67
b.	<i>Documentation</i>	69
c.	<i>Risk Management</i>	70
d.	<i>Cost of Verification and Validation</i>	73
e.	<i>Return on Investment</i>	74
F.	SUMMARY	75
IV.	HETEROGENEOUS SYSTEM REPRESENTATION ANOMALIES ..	79
A.	INTRODUCTION.....	79
B.	ELUSIVE AUTHORITATIVE REPRESENTATIONS	79
C.	DATA MODEL HETEROGENITY AND INTEROPERABILITY	80
D.	HETEROGENEOUS SYNTACTIC DATA MODELS.....	82
E.	HETEROGENEOUS CONCEPTUAL MODELS.....	86
F.	FIDELITY HETEROGENITY	91
G.	MULTIRESOLUTION MODELING (MRM) HETEROGENITY.....	94
H.	INTEROPERABILITY AND HETEROGENITY IN THE C4ISR DOMAIN	100
I.	SUMMARY	104
V.	THE DEPARTMENT'S FRAMEWORK FOR IMPROVED CREDIBILITY	111
A.	INTRODUCTION.....	111
B.	THE DEPARTMENT'S AS-IS SIMULATION ARCHITECTURE	111
C.	THE DEPARTMENT'S TO-BE ARCHITECTURE	113
1.	High-Level Architecture (HLA)	117
2.	Conceptual Model of the Mission Space (CMMS).....	120
3.	Functional Description of the Mission Space (FDMS)	125
4.	Data Standards.....	125
D.	SUMMARY	128
VI.	MODEL QUALITY ATTRIBUTES FOR IMPROVED CREDIBILITY	131
A.	INTRODUCTION.....	131
B.	SYSTEMIC SIMULATION SOFTWARE QUALITY ISSUES	131
C.	SOFTWARE QUALITY INITIATIVES.....	132
1.	Internal / External Quality	135
2.	Quality in Use	141
3.	Testing for Quality Attributes	143
4.	Model and Simulation Quality Attributes	145
a.	<i>Fidelity</i>	145
b.	<i>Accuracy</i>	150
c.	<i>Precision</i>	152
d.	<i>Timeliness</i>	152

e.	<i>Error</i>	152
f.	<i>Resolution</i>	153
g.	<i>Detail</i>	154
h.	<i>Aggregation / Disaggregation</i>	155
5.	Model and Simulation Quality Approaches	156
D.	SUMMARY	157
VII.	SOFTWARE ENGINEERING EFFORTS SUPPORTING CREDIBILITY	159
A.	INTRODUCTION	159
B.	PROCESS IMPROVEMENT SUPPORTING M&S DEVELOPMENT	159
C.	CONTRACT MANAGEMENT	165
D.	SOFTWARE ENGINEERING OF MODERN DOD M&S	166
E.	SOFTWARE ENGINEERING EDUCATION INITIATIVES	176
F.	SUMMARY	177
VIII.	SOFTWARE ENGINEERING OPTIONS TO IMPROVE CREDIBILITY	181
A.	INTRODUCTION	181
B.	PRODUCT LINES	181
C.	SOFTWARE PRODUCT LINES	181
1.	Potential Benefit of a Software Product Line	184
2.	Challenges for Software Product Lines	185
a.	<i>Developing Software Product Lines in the Department</i>	185
b.	<i>Component Development for Software Product Lines</i>	187
3.	Product Line Practice Areas	193
a.	<i>Software Engineering Practice Area</i>	195
b.	<i>Technical Management Practice Area</i>	198
c.	<i>Organizational Management Practice Area</i>	200
D.	EVOLVING SOFTWARE ARCHITECTURE THEORY	202
1.	The Federal Enterprise Architecture (FEA)	202
2.	Emerging Software Architecture Concepts	203
3.	Software Architecture Views	206
4.	Software Architecture Quality Analysis Methods	208
5.	The eXtensible Markup Language (XML)	211
6.	Software Architecture Description Languages (ADL)	212
E.	SUMMARY	213
IX.	ANALYSIS OF MODEL CREDIBILITY IN FIVE BMDS M&S	217
A.	INTRODUCTION	217
B.	SELECTION CRITERIA OF A RELEVANT DOD DOMAIN	218
C.	ORGANIZATION MANAGEMENT	219
1.	Domain Overview of the Organization Structure	219
2.	A Mission-Critical System in the National Defense	221
3.	A High-Risk, Software-intensive System	223
D.	BMD SYSTEM M&S DOMAIN OVERVIEW	224

	1.	BMDS Legacy Model and Simulation Systems	225
	2.	BMDS System-Level Legacy M&S Systems.....	226
	a.	<i>Commanders Analysis and Planning Simulation (CAPS)</i> ..	227
	b.	<i>Missile Defense Wargame and Analysis Resource (MDWAR)</i>	228
	c.	<i>Extended Air Defense Simulation (EADSIM)</i>	228
	d.	<i>Extended Air Defense Testbed (EADTB)</i>	229
	e.	<i>Missile Defense System Exerciser (MDSE)</i>	229
	f.	<i>BMD System-Level M&S Status</i>	230
E.		BMD SYSTEM-LEVEL SIMULATION MODEL REPRESENTATIONS	230
	1.	BMD System-Level M&S TDS Weapon System Representations	231
	2.	BMDS System-Level M&S MDS Weapon System Representations	233
	3.	BMD System-Level M&S BDS Weapon System Representations	235
	4.	BMD System-Level M&S Weapon System Representation Summary.....	237
	5.	BMD System-Level M&S Sensor System Representations.....	238
	6.	BMD System-Level M&S Sensor System Representations Summary.....	240
	7.	BMD System-Level M&S Battle Management System Representations	241
	8.	BMD System-Level M&S Communication System Representations	243
	9.	BMD System-Level M&S Platform System Representations	244
	10.	BMD System-Level M&S Geodetic Coordinate Representations	244
	11.	BMD System-Level M&S Summary of Representations	247
F.		BMDS INTERNATIONAL COOPERATIVE M&S PROGRAM	249
G.		BMDS ELEMENT-LEVEL MODELS & SIMULATIONS PROGRAM	250
H.		BMDS-LEVEL THREAT, ENVIRONMENT, SIGNATURE & LETHALITY MODELS & SIMULATIONS.....	251
I.		BMDS SYSTEM-LEVEL M&S VV&A PROGRAM	251
	1.	Agency M&S VV&A EFFORTS (1986-2000)	251
	2.	Agency M&S VV&A EFFORTS (2001-2003)	256
J.		BMDS SYSTEM M&S FIDELITY	257
	1.	Agency M&S Fidelity Background 1986 - 2000	257
	2.	An Evolving M&S Fidelity Framework 2000 - 2003	259
	3.	BMD System-Level M&S Radar Detection Models	261
K.		BMD SYSTEM-LEVEL M&S REPRESENTATION HETEROGENITY.....	265
L.		BMDS SYSTEM-LEVEL M&S DEVELOPMENT DEMOGRAPHICS	266
M.		ANALYSIS OF BMDS M&S SOFTWARE STATE FACTORS.....	268

N.	ANALYSIS OF BMDS M&S SOFTWARE DEVELOPMENT PORTFOLIO.....	269
O.	BMDS M&S SOFTWARE INTEROPERABILITY	271
P.	EDUCATION ENABLERS SUPPORTING M&S CREDIBILITY	272
Q.	SUMMARY	274
X.	A SOFTWARE ARCHITECTURE-BASED PRODUCT LINE MODEL FOR SIMULATION MODEL REPRESENTATIONS	277
A.	INTRODUCTION.....	277
B.	THE INFLUENCE OF SOFTWARE ARCHITECTURE IN DEPARTMENT M&S.....	278
	1. The Department's <i>De Facto</i> M&S Software Architecture.....	279
	2. A Layered Architecture Model Approach.....	279
C.	THE HORIZONTALLY-LAYERED SIMULATION SOFTWARE ARCHITECTURE	282
	1. The Software Architecture Layered Style	282
	2. Viewpoints and Views.....	284
	3. The Referent Layer	285
	4. The Conceptual Layer	287
	5. The Component Layer.....	288
	6. The Implementation Layer	289
	7. The Simulation Software Architecture Framework (SSAF)	290
	8. Software Architecture Reconstruction	293
D.	THE VERTICALLY-SLICED SOFTWARE PRODUCT LINE ARCHITECTURE	295
	1. Core Architecture Asset Development.....	300
	a. <i>Mining for Product Line Core Architecture Assets</i>	300
	b. <i>Reverse Engineering for Product Line Core Architecture Assets</i>	301
	2. Product Development of the Software Product Line Architecture.....	303
	a. <i>Reengineering Architecture Assets for Software Product Lines</i>	303
	b. <i>Reengineering for Software Product Line Architecture Variability</i>	306
E.	THE SIMULATION SOFTWARE ARCHITECTURE-BASED PRODUCT LINE MODEL DOMAIN METADATA REPOSITORY ...	310
	1. The Simulation Software Architecture-Based Product Line Model Domain Metadata Repository	310
	2. The Simulation Software Architecture-Based Product Line Model Domain Metadata Repository Registry.....	312
F.	ARCHITECTURE READINESS LEVELS (ARL)	314
G.	DOMAIN INTEGRATED PRODUCT DEVELOPMENT TEAMS (DIPDT)	320
H.	SUMMARY	324

XI.	CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK	327
A.	REVIEW OF MAJOR RESEARCH AREAS RELEVANT TO THIS WORK	327
B.	EVALUATION OF A SIMULATION SOFTWARE ARCHITECTURE METHOD AGAINST THE CURRENT M&S TECHNIQUES.....	329
C.	RECOMMENDATIONS FOR FUTURE RESEARCH.....	334
1.	Extension of the Simulation Software Architecture (SSA)- Based Model into Different Domains	334
2.	Simulation Software Architectural Framework	335
3.	Simulation Product line Architecture	336
4.	SSA Domain Metadata Repository Model.....	336
5.	Architecture Readiness Levels.....	337
D.	CONCLUDING REMARKS	338
 APPENDIX A – BALLISTIC MISSILE DEFENSE ORGANIZATION		
	(1996-2001)	339
A.	INTRODUCTION.....	339
B.	THEATER MISSILE DEFENSE (TMD) ORGANIZATION – (1996-2001)	339
1.	PATRIOT – (1996-2001)	341
2.	Theater Area Air Defense (THAAD) Program – (1996-2001)	341
3.	Navy Area Defense (NAD) Program– (1996-2001)	341
4.	Navy Theater-Wide (NTW) Defense Program– (1996-2001).....	342
5.	Airborne Laser Defense Program (ABL)– (1996-2001)	342
6.	Medium Extended Air Defense System (MEADS)– (1996-2001) 342	
7.	Arrow Weapon System– (1996-2001).....	343
C.	NATIONAL MISSILE DEFENSE (NMD) ORGANIZATION – (1996-2001)	343
1.	NMD Ground–Based Sensors- (1996-2001).....	343
2.	NMD Space–Based Sensors- (1996-2001)	343
3.	NMD Ground–Based Interceptor (GBI) - (1996-2001)	344
 APPENDIX B – BALLISTIC MISSILE DEFENSE SYSTEM (BMDS)		
	PROGRAM 2002-PRESENT.....	345
A.	BMD SYSTEM OBJECTIVES	345
B.	BMD SYSTEM-LEVEL CAPABILITY.....	346
C.	BMD SYSTEM-LEVEL LAYERED DEFENSE CAPABILITY.....	347
1.	Boost Phase Defense Segment (BDS).....	347
2.	Midcourse Defense Segment (MDS).....	347
3.	Terminal Defense Segment (TDS)	348
4.	Sensor Segment	348
5.	Battle Management / Command and Control (BM/C2).....	349
D.	BMD SYSTEM ACQUISITION STRATEGY	350
E.	BMD SYSTEM TECHNOLOGY PLANNING	351

APPENDIX C– COMMANDERS ANALYSIS AND PLANNING SIMULATION (CAPS).....	353
APPENDIX D –MISSILE DEFENSE WARGAME AND ANALYSIS RESOURCE (MDWAR).....	359
APPENDIX E – EXTENDED AIR DEFENSE SIMULATION (EADSIM) ..	363
APPENDIX F – EXTENDED AIR DEFENSE TESTBED (EADTB).....	365
APPENDIX G – MISSILE DEFENSE SYSTEM EXERCISER (MDSE).....	369
APPENDIX H – ELEMENT LEVEL M&S	373
APPENDIX I – THREAT, ENVIRONMENT, SIGNATURE, and LETHALITY (TSEL) MODELS	375
A. BMDS SYSTEM-LEVEL THREAT M&S	375
1. TM93	375
2. Strategic and Theater Attack Modeling Process (STAMP).....	376
B. BMDS SYSTEM-LEVEL SIGNATURE AND ENVIRONMENT M&S	378
1. Synthetic Scene Generation Model (SSGM).....	378
2. The Battlespace Environment Signature Toolkit (BEST).....	381
C. BMDS SYSTEM-LEVEL LETHALITY M&S.....	384
1. Post Engagement Ground Effects Model (PEGEM)	384
2. Parametric Endo- Exo-atmospheric Lethality Simulation (PEELS)	385
APPENDIX J – THE DOMAIN METADATA REPOSITORY REGISTRY (DMRR).....	387
A. COMMON DMRR METADATA FACILITIES	387
B. DMRR TYPES OF ADMINISTERED	388
C. DMRR HIGH-LEVEL METAMODEL	389
D. DMRR ADMINISTRATION AND IDENTIFICATION METAMODEL REGION	389
E. DMRR CLASSES USED AS COMPOSITE DATATYPES.....	390
F. DMRR NAMING AND DEFINITION METAMODEL	391
G. DMRR CLASSIFICATION METAMODEL REGION	392
H. DMRR DATA ELEMENT CONCEPT METAMODEL REGION.....	393
I. DMRR CONCEPTUAL AND VALUE DOMAIN METAMODEL REGION	394
J. DMRR DATA ELEMENT METAMODEL REGION	396
K. DMRR CONSOLIDATED METAMODEL.....	397
LIST OF REFERENCES	399
INITIAL DISTRIBUTION LIST	535

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 2-1.	Major Research Areas Relevant to this Work.....	20
Figure 2-2.	Department M&S Hierarchy (After [PMR97]).....	28
Figure 3-1.	Synthesized M&S Credibility Taxonomy.....	48
Figure 3-2.	VV&A and Legacy M&S (From [RPG00]).....	50
Figure 3-3.	Data VV&A & Legacy M&S (From [RPG00]).....	58
Figure 3-4.	V&V Techniques (After [GMS+96, RPG00]).....	60
Figure 4-1.	Resolution Dimensions (From [Dav98])	96
Figure 4-2.	Distinction between Normal and MRM Design (From [Dav98]).....	97
Figure 4-3.	RASSP Taxonomy Model (From [HMB+00])	98
Figure 4-4.	LISI Model (From [TLW+00]).....	102
Figure 5-1.	JTA Hierarchy Model (From [JTA02a]).....	115
Figure 5-2.	The FEDEP Process (From [DMS99])	120
Figure 6-1.	The ISO 9126-1 Quality Model (From [ISO01]).....	134
Figure 6-2.	Overview of Fidelity Context (From [RPG00]).....	150
Figure 7-1.	The SWEBOK (From [BCC+01b])	174
Figure 8-1.	Model of an Architectural Description (From [IEE00b]).....	208
Figure 9-1.	JTA Domains Included in this Study (After [JTA02a]).....	218
Figure 9-2.	BMDS Defense Phase Segments and Required Capabilities (After [Ray01c])	223
Figure 9-3.	BMD M&S Addressed in this Study (After [PMR97])	225
Figure 9-4.	MDS System-Level Model and Simulation Capability (After [TEM03]).....	259
Figure 9-5.	Example of Radar Fidelity in Core Models (From [TEM03]).....	260
Figure 10-1.	Department's <i>De Facto</i> Two-Layer M&S Architecture.....	281
Figure 10-2.	Simulation Software Architecture (SSA) Model	284
Figure 10-3.	SSA Referent Layer Model.....	286
Figure 10-4.	SSA Conceptual Layer Model	287
Figure 10-5.	SSA Component Layer Model.....	288
Figure 10-6.	SSA Implementation Layer Model	290
Figure 10-7.	Simulation Software Architecture Framework (SSAF).....	292
Figure 10-8.	The Horseshoe Model Underlying OAR (From [BSW+99])	294
Figure 10-9.	Software Product Line Architecture (SPLA).....	297
Figure 10-10.	Software Architecture Classification (From [ISO79c]).....	299
Figure 10-11.	Software Architecture Reconstruction.....	306
Figure 10-12.	Future BMDS Requirements (From [Gre03b]).....	308
Figure 10-13.	SSA-Based Product Line Model Domain Metadata Repository.....	310
Figure 10-14.	BMDS Domain Metadata Repository (DMR)	312
Figure 10-15.	BMDS Domain Metadata Repository Registry (DMRR).....	313
Figure 10-16.	Current Readiness Level Methods.....	315
Figure 10-17.	MDA Contract Complexity (Notional)	321
Figure 11-1.	Major Areas Researched for this Work	327
Figure 11-2.	Missile Defense Domain SSA Status (After [Gre03b]).....	335
Figure 11-3.	Future Areas of Research (From [Ste03b]).....	337

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 7-1.	Department Institutional Constraints Affecting M&S Credibility.....	176
Table 8-1.	Product Line Practice Areas (From [Nor03])	195
Table 8-2.	Software Architectural Views / Viewpoints / Viewtypes (From [CBB+03]).....	206
Table 9-1.	Terminal Defense Segment Missile Representations	233
Table 9-2.	International Terminal Defense Segment Missile Representations	233
Table 9-3.	BMDS Midcourse Defense Segment Missile Representations.....	235
Table 9-4.	Boost Defense Segment Laser Representations.....	236
Table 9-5.	BMDS Boost Defense Segment Kinetic Energy Representations.....	237
Table 9-6.	BMDS TDS Radar Sensor System Representations.....	238
Table 9-7.	BMDS International TDS Sensor System Representations	239
Table 9-8.	BMDS MDS Radar Sensor System Representations.....	239
Table 9-9.	BMDS BDS Satellite / Infrared (IR) Sensor System Representations	240
Table 9-10.	BMDS Airborne Laser Sensor System Representations.....	240
Table 9-11.	BMDS Battle Manager System Representations	242
Table 9-12.	BMDS International BM System Representations	243
Table 9-13.	BMDS Communication System Representations	243
Table 9-14.	BMDS Platform System Representations	244
Table 9-15.	BMDS Geodetic Coordinate Representations	246
Table 9-16.	BMD System-Level M&S Representation Summaries	248
Table 9-17.	BMD International Cooperative M&S Program.....	250
Table 9-18.	BMD System-Level M&S Demographics.....	267
Table 9-19.	BMD System-Level M&S SLOC – 2002 (After [TEM03]).....	271
Table 9-20.	BMD System-Level M&S Interoperability Status.....	272
Table 10-1.	Architecture Readiness Level Summary.....	316
Table 10-2.	Domain Integrated Product Development Team (DIPDT).....	323

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank the people whose many contributions made this research possible. Dr. Luqi and Dr. Valdis Berzins pioneered and implemented the DoD's first distance-learning Ph.D. program in software engineering at the NPS, and developed the impetus for this research. Dr. Bret Michael set the highest standards, and challenged me to exceed them, while also supporting several Missile Defense Agency (MDA) software engineering research projects. Dr. Drew Hamilton's mentoring and collaboration on simulation vulnerability assessment research significantly broadened the experience.

Dr. Patricia Sanders sponsored the MDA Software Engineering Improvement Initiative and deserves special thanks. She had the vision and understood the need to champion and advance many of the cited software engineering, simulation, and VV&A concepts for the Department. Her support went even further when she approved many recommendations to improve the MDA model and simulation program based on this research. I also acknowledge and thank the MDA Model and Simulation Directorate and MDA Software Engineering Team, Mr. Butch Caffall and Major (P) Tom Cook for their constant support, collaboration, and encouragement. I also add my kudos to the NPS SEAC staff and MDA BIRC staff for cheerfully supporting every request. Thank you.

I especially want to thank Debbie, my wife, and Meaghan, my daughter. No words can ever express my gratitude for your love, patience, support and sacrifice. Your constant encouragement was more than anyone could have expected. Debbie, you have been everything for me since college. Meaghan, always remember that I am glad to be your Dad. You are my girls, my soul and my heart's inspiration, always special, and beyond the meaning of insufficient words. I dedicate this dissertation to you.

This dissertation also Honors the memory of my Mom and Dad. They were members of the "Great Generation" who experienced the harsh realities of the Depression, survived the Second World War horrors, and rebuilt the world. My Dad, a first-generation Irish-American, went to War as an American Soldier at age eighteen. He returned as Tech Sergeant after the "duration plus six" to grow a family, work two jobs, and go to night school for law. My Mom always sacrificed for her family. They gave me everything and all the opportunities I enjoyed. I can never repay that debt.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. THE NEW STRATEGIC ROLE FOR MODELS AND SIMULATIONS

Software¹ enables the political [Bus02d], economic [RTI02], psychological [Bus02b, Bus02c], and military [Rum01] centers of United States national power. The United States military center of national power, manifested in the Department of Defense² (DoD), employs software-intensive systems³ in a wide variety of missions, tasks, and functions, including models and simulations⁴ (M&S), supporting the Department's national defense mission. The Department requires credible⁵ M&S supporting confidence⁶ in the simulation results provided to Department- and National-level decision-makers, especially in high-risk, mission-critical decisions [DoD94]. Establishing credibility in Department simulations involves many disciplines and knowledge areas including software engineering⁷, processes⁸, quality⁹, product management¹⁰ and architecture¹¹. The Department's complex organizational dynamics, and complicated acquisition procedures [BSK+00, SAG+00, Wol02a, Wol02b] also impact the level of M&S credibility, at times adversely.

There is a new sense of urgency in the Department for credible M&S, and an increased emphasis on improved confidence in simulation results [DoD03a]. In the past, models and simulations played secondary and tertiary roles supporting the Department's decision-making process. Today, models and simulations are Department corporate assets [DoD94], evolving into primary enablers directly supporting an information age national

¹ Software includes the computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system [IEE90].

² Department of Defense will be referred to by the terms Department or DoD, used interchangeably throughout this document.

³ A software-intensive system is any system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole [IEE00b].

⁴ Models and simulations (M&S) – In this research models and simulations (M&S) refers to mathematical abstractions and software implementations, as opposed to the term modeling and simulations used to identify an analytical problem solving approach [GMS+96].

⁵ Credible means 1. Capable of being believed; believable; plausible; 2. Worthy of confidence; reliable. Plausible is defined as 1. Seemingly or apparently valid, likely, or acceptable. Reliable is further defined as 1. That can be relied upon; dependable [Web95].

⁶ Confidence is defined as 1. Trust or faith. 2. A trusting relationship. 3. Something confided. 4. A feeling of assurance, especially of self-assurance. 5. The assurance that someone keeps a secret [Web95].

⁷ Software engineering is the application of a systematic, disciplined, quantifiable approach to the development; operation, and maintenance of software; that is, the application of engineering to software [IEE90].

⁸ Process is a sequence of steps performed for a given purpose: for example, the software, development process [IEE90].

⁹ Quality is 1). The degree to which a system, component, or process meets specified requirements; 2). The degree to which a system, component, or process meets customer or user needs and expectations [IEE90].

¹⁰ Product management includes the definition, coordination, and control of the product characteristics during its development cycle [IEE90].

¹¹ Architecture means the organizational structure of a system or component [IEE90].

security strategy [Bus02d]. The Department's support for this strategy and supporting objectives requires confidence, credibility, security and interoperability in M&S, which are "broadly applicable to the full range and scope of missions and activities across the breadth of DoD operations" [DoD01a] and enable the Department's execution of critical missions, tasks, and functions [Sha97].

National-level decision-makers require credible mission-critical M&S software¹² supporting confidence in national security policy decisions including the emerging Homeland Defense security policy [FOP+01, Bre02b, Bus02a, Bus02b, Bus02c, CS02] and complex, mission-critical, high-risk, defensive systems such as ballistic missile defense, designed to counter asymmetric twenty-first century threats against the American Homeland, including global terrorism and weapons of mass destruction [Bus02e]. President Bush recently signed the *National Strategy For Homeland Security* [Bus02c] emphasizing six mission-critical areas supporting the Nation's defense including the protection of critical infrastructure and key assets. Within the critical infrastructure mission area, the President identified eight major initiatives including the development of a national infrastructure protection plan, securing cyberspace, and harnessing "the best analytic and modeling tools to develop effective protective solutions" [Bus02c].

From a strategic perspective, [DoD02d] noted, "Modeling and simulation are simply a means to an end" [DoD02d]. Confidence in the high-risk, mission-critical results produced from the Department's simulations employed as a means for National Security necessitate the highest levels of credibility. The Department's draft *Modeling and Simulation Strategic Plan* [DoD03a] reflects this new emphasis on the warfighter with the first Department M&S mission statement: "Exploit Modeling and Simulation to Transform the Warfighting Capabilities of the United States" [DoD03a].

¹² Critical software is software whose failure could have on safety, or could cause large financial or social loss [IEE90].

B. AUTHORITATIVE REPRESENTATIONS AND CREDIBLE M&S

Credible M&S requires authoritative representations¹³, according to the Department's M&S policy directives, *DoD Modeling and Simulation Management* [DoD94] and the Department's *Modeling and Simulation Master Plan* [DoD95]. A Department M&S baseline assessment [DoD95] identified authoritative representations as one of the “many shortfalls that must be corrected to realize the DoD M&S vision”¹⁴ [DoD95] and support the Department's readiness, modernization, force structure, and sustainability requirements. In this research we employed the concept of computer-based simulations defined by [CS97] as a wide range of computationally based activities employing digital computers. In addition, we view the Department's large-scale, legacy¹⁵ modeling and simulation systems as software-intensive systems.

Attesting to the importance of credible M&S representations in Department M&S, three of the six major [DoD95] objectives addressed authoritative representations. This research focused on [DoD95] objective three, authoritative system¹⁶ representations. Other major [DoD95] objectives included the natural environment authoritative representations (objective two), and human behavior authoritative representations (objective four). Enabling tasks and timelines established to achieve this [DoD95] objective included:

- Build prototype classes of objects as part of the architecture¹⁷ prototype effort in fiscal year 1995,
- Identify common object classes in fiscal year 1996,
- Assign DoD executive agents to develop common object classes in fiscal year 1996,
- Identify and assign responsibilities for developing system M&S for all DoD mission areas by fiscal year 1997,

¹³ Authoritative representations are models, algorithms, and data that have been developed or approved by a source which have accurate technical knowledge of the entity or phenomenon to be modeled and its effects [DoD95].

¹⁴ Department M&S Vision (1995) A. Defense modeling and simulation will provide readily available, operationally valid environments for use by the DoD Components: (1) To train jointly, develop doctrine and tactics, formulate operational plans, and assess warfighting situations. (2) To support technology assessment, system upgrade, prototype and full-scale development, and force structuring. B. Furthermore, common use of these environments will promote a closer interaction between the operations and acquisition community in carrying out their respective duties. To allow maximum utility and flexibility, these modeling and simulation environments will be constructed from affordable, reusable components interoperating through an open-system architecture [DoD95].

¹⁵ Legacy M&S – M&S systems developed in the past without the benefit of modern standards, and still in use [PMR94].

¹⁶ DoD M&S system representations include weapons, sensors, units, command, control, communications, computers and intelligence, surveillance, reconnaissance (C4ISR) platforms, and logistic support for a wide scope of U.S., Allied, Coalition, and major threat platforms [DoD95].

¹⁷ Architecture in this context is defined as the structure of components in a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time [DoD95]. Other definitions are introduced later.

- Develop aggregation/disaggregation¹⁸ methodologies by a date to be determined [DoD95].

C. LIMITATIONS OF CURRENT APPROACHES TO ACHIEVE CREDIBILITY

The Defense Modeling and Simulation Office (DMSO) initiated several studies in 2000 to assess the progress of M&S within the Department since the 1995 publication of the M&S Master Plan [DoD95]. The DMSO methodology employed multiple techniques including written surveys and interviews [Wri01b] with key members of the Department's M&S community, sponsorship of the Integration Task Force (ITF) effort, and completion of a Warfighter M&S needs assessment. These initiatives produced three major product deliverables, a *Master Plan Baseline Assessment* [Mos00, Cra01a, Cra01b], a final assessment *Report of Warfighter M&S Needs* [MWD+00], and the *Report of the Integration Task Force* [HCG+01a, HCB+01b]. The survey results indicated that most of the [DoD95] objectives experienced some progress, with the greatest progress noted on the development of a common technical framework, and the establishment of the High Level Architecture (HLA), an Institute of Electrical and Electronic Engineers (IEEE) standard [IEE98e, IEE00c, IEE00d, IEE00e], and the Department's technical architecture for simulation interoperability [Gan00].

However, most objectives for authoritative representations remained incomplete, with the least progress noted on the representation of human behavior [Mos00, MT01, Flo02a]. The three DMSO study initiatives [HCG+01a, HCG+01b, MWD+00, and Mos00] collectively provided direction to the Department's 2001 draft revision of the *DoD Modeling and Simulation Master Plan (Draft)* [DoD01a]. Many of the persistent systemic problems with authoritative representations require long-term executable solutions to improve quality and evidence that M&S provides a return on investment [Mye99]. The true scope of the issues, challenges, and difficulties faced by the Department's M&S community to develop authoritative representations is evident by a comparison of the system authoritative representation sub-objectives of the approved [DoD95] and the draft [DoD01a] plan. Simi-

¹⁸ Aggregation is the ability to group entities while preserving the effects of entity behavior and interaction while grouped. Disaggregation is the ability to represent the behavior of an aggregated unit in terms of its component entities based on maintained state representations [DoD98].

lar [DoD95] objectives originally scheduled for implementation in the 1995-1997 time-frame, slipped almost a decade to 2004-2006 [DoD01a].

There were several understandable reasons for the [DoD95] schedule slip. A few of the reasons included a very ambitious schedule negatively impacted by Department budget cuts during the 1990s; the sheer scope of the work across the Department, Service and M&S domains¹⁹; the breadth of the technical challenges, including improved fidelity²⁰ of authoritative representations [Gro99, RGH00]; and the overall immaturity of software engineering processes and practices [HHK93, HNC+00] supporting M&S development. Other contributing factors to the [DoD95] schedule slip included significant interoperability issues, external forces including the Department's balky acquisition process [Mos02], changing Department priorities [Fer01]; and the unplanned allocation of resources to Year 2000 problem remediation [Gre97, WG99]. The draft [DoD01a] again reiterated the need for "credible authoritative representations" (sub-objective 3.3), which are interoperable, consistent, and promote "access to the underlying algorithms and data, which define the representation within M&S by other applications" [DoD01a].

The Department also continues to experience persistent, systemic M&S architecture problems. The *C4ISR Information Superiority Master Plan* [DoD02d] identified continuing concerns with the Department's M&S technical architecture noting the HLA for Department M&S lacked seamless interoperability with the Common Operating Environment (COE) [JTA02a, JTA02b] supporting the development of the Global Command and Control System (GCCS) and the Department's Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) architecture framework [AWG97]. The proposed [DoD02d] solution is to develop a common Technical Reference Model (TRM) for the C4ISR and the M&S community, evolving over the next twenty

¹⁹ Domains are the physical or abstract space in which the entities and processes operate. The domain can be the land, sea, air, space, undersea, a combination of any these, or an abstract domain such as an n-dimensional mathematics space, or economic or psychological domains [DoD98].

²⁰ Fidelity: 1. The degree to which a model or simulation reproduces the state and behavior of a real world object or the perception of a real world object, feature, condition, or chosen standard in a measurable or perceivable manner; a measure of the realism of a model or simulation; faithfulness. Fidelity should generally be described with respect to the measures, standards or perceptions used in assessing or stating it. 2. The methods, metrics, and descriptions of models or simulations used to compare those models or simulations to their real world referents or to other simulations in such terms as accuracy, scope, resolution, level of detail, level of abstraction and repeatability. Fidelity can characterize the representations of a model, a simulation, the data used by a simulation (e.g., input, characteristic or parametric), or an exercise. Each of these fidelity types has different implications for the applications that employ these representations [Gro99, RGH00, RPG00]

years. Based on the actual completion dates of previous Department M&S projects, this time-to-completion estimate for a TRM may be accurate. However, it is not clear if this timeline is responsive to the Department’s operational requirements and transformation initiatives [MAK01], since the goal for the Department is to have legacy systems interoperable, in terms of command and control functions, by the end of fiscal year 2008 [Wol01, FBK+01].

Lessons-learned from incorporating M&S into the Department’s test and evaluation (T&E) process [DOT01a, DOT01b] confirmed the inherent limits of M&S cited twenty years earlier [PAD78, PAD80]. [DOT01a, DOT01b] revalidated the lack of quality, authoritative system representations as the Department’s chronic systemic M&S shortfall with the “underlying issues of the credibility of the tools [being] a major constraint” [PHP+96]. Improving the fidelity²¹, accuracy²² and precision²³ quality attributes²⁴ of authoritative representations supports credibility. However, M&S quality has also been an elusive objective for the Department. Many prior systemic deficiencies continue to exist, including the need for authoritative representations and data, improved information assurance measures [HGE03], and interoperability standards [HCG+01a, HCG+01b, You02c]. We refer to the entire class of problems impeding the development of authoritative representations as heterogeneous system representation anomalies [IEE93], discussed in Chapter IV.

Heterogeneous system representation anomalies, a major focus of this study, include two broad categories adversely affecting simulation federation credibility: technical interoperability and substantive interoperability [DSB+99, YPH01]. Technical interoperability [YPH01] is the capability of federates to physically connect and exchange data, while substantive interoperability [YPH01] adds the requirement for the federation to provide

²¹ Fidelity is the accuracy of the representation when compared to the real world [DoD95].

²² Accuracy: The degree to which a parameter or variable or set of parameters or variables within a model or simulation conform exactly to reality or to some chosen standard or referent. See resolution, fidelity, precision [RPG00].

²³ Precision: 1. The quality or state of being clearly depicted, definite, measured or calculated. 2. A quality associated with the spread of data obtained in repetitions of an experiment as measured by variance; the lower the variance, the higher the precision. 3. A measure of how meticulously or rigorously computational processes are described or performed by a model or simulation [RPG00].

²⁴ An attribute is a property or characteristic of one or more entities, a property inherent in an entity, or associated with that entity for database purposes [DoD98].

consistent, accurate simulated representations, adhering to “fair fight” principles²⁵ to adequately meet federation accreditation (e.g., intended use) objectives. Substantive interoperability [Dav98, DSB+99, YPH01] involves the representation validity supporting the federations’ intended use, interoperability among entities²⁶, and the contextual effects on interoperability²⁷. The logical interaction of federation entities involves:

- The level of representation for the entity to include the necessary level of detail and goodness-of-fit,
- The key entity model attributes reflecting critical real-world characteristics salient to the federation’s purpose,
- The entity models exhibit the needed behaviors, work together logically, and employ consistent algorithms across the federation to compute effects [YPH01].

D. RESEARCH QUESTION AND HYPOTHESIS

The primary focus of this dissertation was to determine if system representations in five selected Missile Defense Agency’s (MDA)²⁸ software-intensive M&S systems were authoritative according to current Department standards, and credible enough to support confidence by Department- and National-level decision-makers in missile defense decisions critical to national security.

The second objective included the identification of pre-conditions and alternatives to improve and enhance the simulation credibility and confidence in the results produced by the Agency’s M&S software. The basic research problem investigated during this effort was:

How may the Agency better define, develop, evolve, manage, and maintain authoritative model representations in the five selected large-scale, software-intensive legacy simulations²⁹, and effectively address simulation interoperability issues (e.g., heterogeneous system representation anomalies) within existing constraints and pre-conditions (e.g.,

²⁵ Two or more simulations may be considered to be in a fair fight when differences’ in the simulations’ performance characteristics have significantly less effect on the outcome of the conflict than actions taken by the simulation participants [DoD98].

²⁶ Interoperability among entities includes accurate temporal resolution, spatial resolution and the logical interaction among entities modeled in different federates [YPH01].

²⁷ Contextual effects on interoperability address the accurate, coherent relationship required between components of the physical environment to include ocean, ground, space, atmosphere, weather, infrared and electromagnetic spectrums [HY01a].

²⁸ Hereafter, the Missile Defense Agency may be referred to as the Agency or the Ballistic Missile Defense System (BMDS).

²⁹ Chapter IX provides the analysis of the five selected Agency simulations under study.

technical, organizational, managerial) affecting credibility in the five selected simulations systems?

In response to this question I submit the following hypothesis: Employing a domain³⁰-managed, four-layered simulation software architecture-based³¹, product line³² model with a referent³³ layer, developed in a distributed domain integrated software engineering environment supporting the evolution of five selected Agency legacy simulations can improve the authority of representations in the five selected missile defense simulations. The Software Architecture-Based Product Line Model developed in a distributed development environment by a Domain Integrated Product Development Team employing Architecture Readiness Levels (ARL) to measure quality provides such a methodology. The methodology includes software development practice areas [JS02]: software engineering³⁴, organizational management³⁵ and technical management³⁶ [CN02, Nor03] components of the Department's system engineering process [DAU01]. Hypothesis testing includes concurrent implementation of the techniques and procedures in the Agency's M&S domain.

E. SIMULATION SOFTWARE ARCHITECTURE (SSA) OVERVIEW

This dissertation introduces a four-layer, simulation software architecture-based³⁷ product line model, and architecture readiness levels supporting distributed simulation software development by a domain integrated product development teams employing prod-

³⁰ Domain for the purpose of this research is an area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area [Nor03]. A more M&S domain-unique definition for domain is provided in [DoD98].

³¹ Software architecture is the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution [IEE00b].

³² A product line is a group of products sharing a common, managed set of features that satisfy specific needs of selected market or mission [BFG+00].

³³ 1. A referent is a codified body of knowledge about a thing being simulated [RPG00]. 2. Something referenced or singled out for attention, a designated object, real or imaginary or any class of such objects [Gro99, RGH00].

³⁴ Software engineering practice areas are those practices necessary to apply the appropriate technology to create and evolve both core assets and products [JS02].

³⁵ Organizational management refers to the management of the business issues that are visible at the enterprise level, as opposed to the project level [JS02].

³⁶ Technical management practices areas include those management practices necessary to engineer the development and evolution of core assets and products [JS02].

³⁷ Architecture-based development is a process that utilizes the software architecture as the primary tool for the design, evolution, implementation, management, migration, and understanding of a software system. It involves organizing the work products around the architecture; implementing the software system based on the architecture; and maintaining the implementation to reflect changes in, and to ensure conformance to, the architecture [CN96].

uct line practice areas to evolve Agency's the five legacy simulations heterogeneously organized in a joint M&S³⁸ hierarchy³⁹ today. The proposed four-layered simulation software architecture-based product line model supported by a domain metadata repository enables the development of authoritative representations supported by readiness levels⁴⁰.

The four-level layered software architecture-based product line model replaces the level of abstraction⁴¹ prescribed previously in the traditional Department M&S hierarchy⁴² (Figure 2-2), with each layer (e.g., referent, conceptual, component, implementation) reflecting a division of the software architecture-based product line model into units. The layer represents a horizontal slice through the architecture for organizing classifiers or packages at the same level of abstraction [UML99, MB02, Mil02b], while the software architecture-based product line represents a vertical slice. Vertical slice components of the model include a Simulation Software Architectural Framework (SSAF) and the Simulation Product Line Architecture (SPLA). A layering scheme prescribes the interaction between layers. A Referent layer maintains multiple real-world views⁴³ and viewpoints⁴⁴ for establishing the foundation of authoritative model representations.

The methodology supports development of software architecture-based product line readiness levels or Architecture Readiness Levels (ARL) and a domain metadata repository, addressed in Chapter X. The ARL methodology supports conceptual model validation, data management, the verification, validation process, and the implementation of quality attributes instantiated in the software implementation. The ARL methodology also sup-

³⁸ Joint M&S includes representations of joint and service forces, capabilities, equipment, materiel, and services used by the joint community or by two, or more, Military Services [DoD95].

³⁹ Hierarchy is a ranking or ordering of abstractions [DoD98].

⁴⁰ Readiness Levels are based on Technology Readiness Levels – a systemic metric/measurement system developed by NASA that supports assessments of the maturity of a particular technology and the consistent comparison between different types of technology [Man95].

⁴¹ Abstraction is a view of an object that focuses in the information relevant to a particular purpose and ignores the reminder of the information [IEE90].

⁴² We address simulation hierarchies in Chapter II. Originally devised to manage simulation complexity, multiple viewpoints of simulation hierarchies evolved in many forms. The Department selected a version of the hierarchy cited in Figure 2-2. Systemic issues persist with this simulation hierarchy model, and the lack of integration between the different layers of the simulation hierarchy. [WGB+03] most recently addressed this systemic shortcoming of the Department's M&S architecture.

⁴³ In the conceptual framework of IEEE Std 1471-2000, an architectural description (AD) is organized into one or more constituents identified as architectural *views*. A view addresses one or more concerns identified by system stakeholders. The term *view* is used to refer to the expression of the systems architecture with respect to a particular viewpoint. A view may consist of one or more architectural model [IEE00b].

⁴⁴ The viewpoint is a specification of the conventions for constructing and using a view, a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis. The viewpoint determines the language (including notation, model, or product types) to be used to describe the view, and any associated modeling methods or analysis techniques applied to representations of the view [IEE00b].

ports the development of quantitative assessments of system representations by the developer and enables the measurement of the production process. The domain metadata repository supports the domain metadata registry supporting interoperability with other Government agencies and private sector metadata registries.

The centrally-managed, distributed, domain integrated product development team supports the second study objective: a distributed M&S software engineering management methodology integrating the “production-related” approach to attain high levels of quality by defining specific characteristics of a product with precise measurements, supported by the “process-related”⁴⁵ approach to prevent faults and minimize rework. The production related approach employs a distributed architecture-based software development model based on product lines.

The target audiences for the software architecture-based product line family include Departmental- and National-level decision-makers. Other key stakeholders supported by this concept include the Department’s M&S community; simulation software developers; supported analysis, training, acquisition, experimentation, and operational communities; developmental and operational testers; and the private sector supporting M&S software engineering, transformation initiatives, and simulation-based acquisition (SBA).

F. RESEARCH CONTRIBUTIONS

The Department seeks improved credibility in M&S supporting Departmental- and National-level decision-makers’ confidence in simulation results. Major issues with the implementation of the Department’s verification, validation, and accreditation (VV&A) process, conceptual model development practices, software development maturity, process improvement, and M&S quality have caused Department decision-makers to question the amount of confidence they should place in results produced by the Department’s M&S. As a result it has been difficult for the Department to identify and justify a return on invest-

⁴⁵ Acquisition of software-intensive systems shall use process improvement and performance measures [DoD03b].

ment for M&S, especially for large-scale M&S costing hundreds of million dollars or even exceeding a billion dollars⁴⁶, to develop and maintain [BGK+00].

Conversely, the demand for improved M&S capability, interoperability, reusability, and affordability continue to grow [DoD03a], driven by the need to transform United States warfighting capability. The main contributions of this work add to the model and simulation body of knowledge and address a causal factor of the credibility problem with the definition and development of a flexible, extensible methodology for creating software architecture-based product models supporting the development of authoritative and consistent M&S representations in very large-scale software-intensive M&S systems.

[RGH00] noted that researchers study fidelity and conceptual models separately and independently, although they share many similarities and have a common source of difficulties with a lack of formal, fundamental simulation framework. [RGH00] also suggested a cooperative research effort to understand the mutual relationship between fidelity and conceptual models to resolve the common problems. The software architecture-based product line⁴⁷ referent view incorporates the characteristics and attributes of a conceptual model domain architecture framework. This approach provides a method for the evolution of a domain conceptual model referent from an existing heterogeneous Department M&S hierarchy, and establishing a quantifiable methodology for establishing M&S credibility not accomplished by previous methods. Key elements include:

- Development of a layered M&S software architecture supporting architecture-based systems for improved interoperability, involving reverse-engineering, reuse, and re-engineering techniques,
- Identification of a method to evolve an existing heterogeneous M&S domain hierarchy into a software architecture-based product line model,
- Contributing to the development of scalable solutions for building credible M&S software solutions,
- Supporting the development of an alternative M&S framework supporting verification and validation employing Architecture Readiness Levels,

⁴⁶ The Department's Selected Acquisition Report (SAR) summary as of June 2002 identified the cost estimate for the JSIMS program at \$1.29 billion.

⁴⁷ A product line is a group of products sharing a common, managed set of features that satisfy specific needs of selected market or mission [BFG+00].

- Contributing to the development of data quality⁴⁸ measures, improved M&S meta-data⁴⁹, and the body of knowledge to standardize data in support of common, consistent authoritative representations;
- Adding to the Department's Modeling and Simulation Body of Knowledge.

G. SIGNIFICANCE OF DISSERTATIONS CONTRIBUTIONS

There are significant Department M&S issues to resolve, including heterogeneous system representation anomalies. The Department has yet to achieve the full potential of M&S technology due to a lack of credibility in the current M&S practices including conceptual models, the VV&A process, and model quality attributes such as fidelity [YPE00]. Simulation interoperability has also posed significant challenges. All previous M&S interface protocols⁵⁰ and supporting distributed interoperability (e.g., ALSP, DIS, HLA) encountered heterogeneous (e.g., substantive interoperability) anomalies and failed to resolve the problem.

Interim solutions such as translators [You02c] may provide near-term relief, however, they introduce other maintenance, cost, performance, and accuracy risks. In addition, integrating translators into simulation federations presents new challenges including excessive latency, or incorrect translations due to syntactical or semantic errors. [TH03] contends that improving several different aspects such as standards, architectures, components, data models, and processes may improve interoperability discussed in this research.

The strategy to replace many of the large-scale legacy M&S with new, better, software engineered M&S solutions such as the Joint Simulation System (JSIMS), Joint Warfare Simulation (JWARS) [BBG+99d], and Joint Model and Simulation System (JMASS), discussed in Chapter VII, have not achieved the desired results. All three major Department simulation software development efforts to date exceeded initial cost estimates, failed to meet schedule, and produced initial results well short of the design specifications. The

⁴⁸ Data quality is the correctness, timeliness, accuracy, completeness, relevance and accessibility that make data appropriate for use. Quality statements are required for source, accuracy (positional and attribute), currency, logical consistency, completeness (feature and attribute), clipping indicator, security classification, and releasability [DoD98].

⁴⁹ Metadata is data that describes data and may include definitions, classification, accuracy, data type, precision, and source [DoD95].

⁵⁰ Protocols are a set of rules and formats, semantic and syntactic, which determine the communication behavior of simulation applications [DoD95].

introduction of a software architecture-based product line model into the Department's M&S domain has the potential to support:

- A Department M&S Software Product Line Family Strategy including the development of an integrated architectural description⁵¹ containing architecture views, components, and data models, supporting improved software development of simulations with authoritative representations,
- An extension to the Department's architectural framework to support a simulation software architecture-based product line model,
- The development of a software architecture-based product-line simulation metadata registries,
- Additions to the Department's M&S Body of Knowledge (MSBOK).
- The evolution and maturation of product line practices in the Department supporting improved M&S software engineering processes.
- Follow-on efforts to develop a composable M&S capability in the Department.

H. DISSERTATION VISION

A critical early decision in this dissertation effort involved whether to follow a broad or narrow dissertation vision. The objectives of the NPS distance learning Ph.D. program⁵² [NPS02] and the scope of the literature search cited in Chapter II established the framework for our decision process. There were pros and cons for each approach. A narrow dissertation vision supported a research area concerning some specific aspect of the Department's simulation software practice affecting simulation credibility. The narrow vision study approach potentially provides more focus, clarity, and precision. However, there are many past and present research efforts employing the narrow vision study-scope efforts cited in Chapter II. Conversely, there is a dearth of broad vision research efforts "related to the development and evolution of complex [Department simulation] software systems" [NPS02], indicating a need for a broad vision or strategic level dissertation vision.

We chose a broad dissertation vision. Two historical precedents reinforce this decision, the first being the concept of a "silver bullet" solution to the software industry's woes, which languished on for over twenty years before private sector and government executive

⁵¹ Every system has an architecture, and an architecture can be recorded by an architectural description (AD). IEEE Std 1471-2000 distinguishes the architecture of a system, which is conceptual, from particular descriptions of that architecture, which are artifacts or concrete products [IEEE00b].

⁵² The Ph.D. program in Software Engineering is specifically designed for DoD software practitioners who want to acquire the skill and knowledge to perform state-of-the art research on issues related to the development of large complex software systems, and to intelligently manage the research of other software practitioners [NPS02]

management realized the software quality problem was multi-dimensional. The lack of credibility in Department M&S software is also multi-dimensional.

The second historical precedent involves the industry's learning curve over the past fifteen years from the less-than-successful "small-grain" reuse programs through today's "large-grain" reuse initiatives, including software product lines, a central focus of this research. A broad dissertation vision requires an understanding of the primary and contributing factors eroding credibility in Department simulations including organizational issues, quality models, software engineering, life-cycle models and processes, software architecture, interoperability, and information assurance, to name a few.

Another factor contributing to the decision to pursue a broad dissertation vision is a personal objective to add to the software engineering and software architecture body of knowledge to develop the needed skill sets for the future Department Domain and Enterprise Software Engineers and Architects. It is an interesting note that many models in different domains have five levels of maturity: Maslow's Hierarchy of Needs, the Capability Maturity Model family, the Cognitive Hierarchy, and the Information Superiority Correlation Model [Gre97]. An additional model, the Levels of Information Systems Interoperability (LISI) [TLW+00] model also has five levels identifying the levels of interoperability maturity (e.g., isolated, connected, functional, domain, enterprise).

A basic premise of each of these models is that the next higher layer builds on contributions of the previous layer(s). This suggests that if the Department requires future Domain and Enterprise Level Software Engineers and Architects, a maturation process and a multi-dimensional view and understanding of the contributing knowledge bases (e.g. software engineering or simulation body of knowledge) is in order.

The last factor supporting this view is time. The Department's Transformation process has substance, and the pace quickens daily in all areas of the Department to discard non-productive practices or systems, better integrate warfighting capabilities, introduce effective and efficient business practices, reduce major headquarter overhead, and better allocate the Department's human resources. Credible and *timely* modeling and simulation capabilities must quickly evolve to support these many initiatives or risk becoming irrelevant. Three recent case studies reinforce this assessment:

- The Department completely revamped and revised the non-responsive acquisition procedures three times in approximately one year: 5 April 2002 [ASC02b, ASC02c, DoD02b, DoD02f], 30 October 2002 [Wol02a, Wol02b, DoD02g, DoD02h], and 12 May 2003 [DoD03b, DoD03c]. This pace of change is unheard of in the Department's acquisition bureaucracy. Periodic Department-level studies, Defense Science Boards, and Blue Ribbon panels studied the systemic acquisition problems many times over the past thirty years and generally provided similar recommendations each time. For many reasons the Department never enacted most of the far-reaching recommendations.
- The recent war in Iraq reflected this new Transformation thought process in many ways including vastly improved sensor-to-shooter decision cycle times, improved management and visibility of logistics, and a much closer integration of all Service warfighting efforts enabled by information technology [OBB+03}.
- A recent Department manpower transformation initiative proposes radical changes in both the military and civilian management practices of the Department. The proposed civilian management changes affect bureaucratic Civil Service rules, an area historically immune to change. [Hay03a].

Time is a critical factor. The Nation's strategic decision cycle time against foreign and domestic threats continues to decrease, while new challenges appear every day to test all aspects of national power, including the economic, military, political, and psychological pillars. A broad vision, strategic view of simulation credibility supports the Department's transformation initiatives, focusing on M&S as a critically needed corporate asset which has yet to achieve its full potential. The "silver-bullet" solution languished nearly twenty years until discredited as a flawed concept. The challenge involves improving simulation credibility supporting confidence in national defense decisions, while concurrently reducing the development *time* and *complexity* for *new* portable and distributed simulations to meet unknown future exigencies. The problem definition suggests a need for broad research approach and evaluation of all germane factors.

I. DISSERTATION ORGANIZATION

We adopted a dual-track research methodology, conducting an exhaustive secondary source search to determine the scope of the problem, complemented by primary source research of simulation model representations in a specific Department domain. Chapter II provides a literature survey on the current research areas relevant to Department simulation credibility.

The literature survey clearly reveals two major eras affecting Department simulation credibility: (1) the Growth Era before 1994, and (2) the Managed Era, or post-1994 period, when the Department began to plan, develop policy and procedures, budget, and corporately manage the M&S portfolio developed during the Growth Era and plan for new joint-simulations to replace many of the legacy Growth Era M&S. This chapter also introduces the Department's M&S management framework for establishing credible simulations, including classes of simulation, simulation hierarchies, initiatives to establish distributed simulation capabilities, the expanding need of credibility by the major stakeholders influencing the Department's M&S management priorities: acquisition, training, operations, analysis, and experimental domains.

Research indicates that M&S credibility is a multi-dimensioned problem. Chapter III through Chapter VII review different dimensions of the Department's M&S credibility challenge. A major objective of this work is to provide a contribution to the next period of Department M&S development—the Software Architecture-based Composable era.

Chapter III introduces M&S credibility issues and identifies concerns that decision-makers have with Department simulation results, followed by an overview of the Departments VV&A processes, and several factors contributing to the implementation of, or shortfalls of the VV&A processes in the Department. Two major periods, the Growth and the Managed Eras characterize the efforts to improve credibility in the Department's simulation results.

Chapter IV reviews several heterogeneous challenges to Department M&S credibility. Improving the credibility of large-scale, legacy Department M&S requires an understanding of the underlying systemic causes for heterogeneous system representation anomalies, especially in federation interoperability. These diverse challenges include syntactic and semantic heterogeneity, data complexity, interoperability, technical and substantive interoperability, fidelity, and multiresolution modeling.

Chapter V provides the Department's current architectural framework for resolving the underlying systemic causes generating the pre-conditions for heterogeneous system representation anomalies, especially in federation interoperability. The Department's initiatives to improve the credibility of large-scale, legacy Department M&S supporting distrib-

uted interoperability include the “as-is” architecture and the evolving “to-be” architecture: the Joint Technical Architecture, and the Common Technical Framework. The Common Technical Framework components include the High-Level Architecture, conceptual model requirements, and data standards supporting credible simulation development.

Chapter VI addresses software and simulation quality attributes affecting heterogeneous system representation anomalies and credibility in Department M&S, especially in federation interoperability. The chapter reviews the internal and external attributes of the ISO 9126-1 quality model [ISO01], testing for quality attributes, and M&S quality attributes addressed in much of the current literature, including an overview of aggregation and disaggregation. The chapter concludes with a discussion on M&S quality approaches.

Chapter VII reviews three selected areas influencing credibility of large-scale, legacy simulations in the Department including process improvement, the status of new major simulation software engineering initiatives (e.g., JWARS, JSIMS, JMASS) and the growing software engineering body of knowledge. The chapter also includes a discussion of the challenges generated by Department contract management oversight for achieving quality software and reducing heterogeneous system representation anomalies, Department institutional factors affecting M&S credibility, and recent Department software engineering education initiatives. In addition, research reveals a growing awareness that the Department needs qualified software engineers who understand the foundations of the software-intensive systems upon which we basing our future security, economic prosperity, and military preparedness, including credible M&S.

Chapter VIII discusses traditional product lines, and introduces software product lines and software architectures. The chapter also reviews core asset development, reverse engineering to develop core assets, reengineering core assets, component development supporting a product line methodology, product line practice areas, and an overview of evolving software architecture theory potentially applicable to improved M&S credibility and reducing heterogeneous system representation anomalies.

Chapter IX provides an analysis of heterogeneous system representation anomalies in five selected M&S in a component of the Nation’s ballistic missile defense domain, the Missile Defense Agency. The analysis reviewed the following areas affecting credibility of

the five systems under study: organizational change impacts, the National Security-based need for credible M&S, risk identification, a domain M&S overview to provide a context for the five systems under study, an overview of the five systems under study, and an analysis of selected model representations in the five systems under study. Specific analysis areas affecting heterogeneous system representation anomalies and credibility included VV&A, fidelity, M&S development demographics, and interoperability.

Chapter X introduces the detailed research design for the architecture-based product line model including the method, specification, design, and implementation of a software architecture-based development methodology for a product line referent with multiple views and viewpoints. The chapter also introduces the M&S architecture-based product line model as an abstract software architecture-based foundation, supported by Architecture Readiness Levels (ARL), for resolving heterogeneous system representation anomalies, and improving credibility in federation interoperability.

Chapter XI presents the major conclusions of this research, suggests areas for future research, and provides the significance of a software architecture-based product line model supporting M&S credibility. We identify possible future research opportunities for the software architecture-based product line model methodology not addressed within the scope of this research (e.g., composability, managing product line variability).

II. LITERATURE SEARCH AND DOMAIN M&S BACKGROUND

A. INTRODUCTION

Chapter II provides a literature survey on the current research areas relevant to Department simulation credibility. The literature survey clearly reveals two major eras affecting Department simulation credibility: (1) the Growth Era before 1994, and (2) the Managed Era, or post-1994 period.

This chapter also introduces the Department's M&S management framework for establishing credible simulations, including classes of simulation, simulation hierarchies, initiatives to establish distributed simulation capabilities, the expanding need of credibility by the major stakeholders influencing the Department's M&S management priorities: acquisition, training, operations, analysis, and experimental domains. Research indicated that M&S credibility was a multi-dimensioned problem. Chapter III through Chapter VII review different dimensions of the Department's M&S credibility challenge.

B. LITERATURE SEARCH

A significant body of literature evolved since the mid-1970s focused on the need for major systemic improvements in M&S software credibility as a basis for supporting higher levels of confidence of simulation results by Department- and National-level decision-makers. The taxonomy shown in Figure 2-1 represents a synthesis of the major research areas for models, simulations, verification, validation, and accreditation affecting the credibility of simulations and Department- or National- level decision-makers confidence in the Department's simulation or federation results.

Unfortunately, there is no "silver bullet" for improving simulation software credibility. Research indicates that improving the credibility of the Department's computer-based M&S software supporting enhanced user confidence involves the following multiple interrelated domains, competencies, disciplines, and bodies of knowledge illustrated in Figure 2-1 and discussed in the following chapters:

- Department of Defense M&S Credibility Measures----- Chapter III
- Heterogeneous System Representation Anomalies ----- Chapter IV
- The Architectural Framework -----Chapter V

- Simulation Software Quality Attributes -----Chapter VI
- Process Improvement and Software Engineering -----Chapter VII
- Software Product Lines and Architecture-----Chapter VIII
- Domain Analysis----- Chapter IX

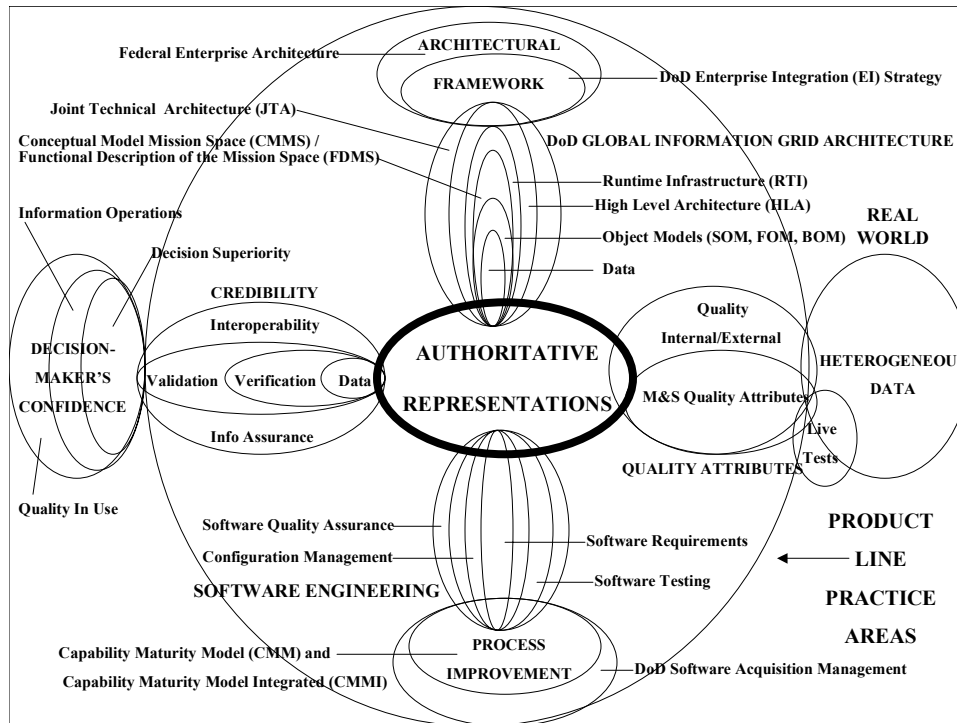


Figure 2-1. Major Research Areas Relevant to this Work

M&S credibility has also taken on additional importance and emphasis as the Department employs large-scale, legacy M&S supporting Department missions and national security initiatives. The literature survey found the use of M&S software pervasive in many domains, expanding rapidly into many new areas, and experiencing a fast-growing demand for even more credible applications and uses:

- ***United States Homeland security and national security strategy*** [Bus02a, Bus02b, Bus02c, Bus02d];
- ***Department policy, memoranda, and strategy documents*** [Kri88, Car88, DOT89, Kri89, Pai97, Gan98, Gan99, DoD00b, GMW00a, GMW00b, Mon00a, Mon00b, She00a, HRA+01, Wol01, Rum01, Rum02a, Rum02b, Ald02a, Ald02b, Ald02c, Ald02d, DoD02d, DoD03a, DoD03b, DoD03c];

- **United States Government Accounting Office (GAO) Reports** [PAD78, PAD79a, PAD79b, PAD80, Mye82a, Mye82b, She82, Con86a, Con86b, Che87, Bre88, Bro88, Bro91, Dav91, HSW+92, Con93, Geb93, Geb94a, Geb94b, Geb95, Dav96, ZHF98, Bro99, HDK+00, HEE+00, McC00, RFH+00, BCP+01];
- **Defense and Army Science Board studies** [WHM+79, JBC+88, BTE+93, HW96, GKB+97, LWF+98, HWB+98, MCC+98, BBB+99, FEE+99, FGW+99, GBB+99, GWB+99, HAC+99, LLD+99, MSI+99, HNC+00, Mor00b, BCC+01a, FOP+01, VFH+01];
- **Private and Government sector studies, analyses, and reports** [Qua64, SCC+88, DOT89, DB91, LA92, Dow92, DH92a, DH92b, ABD+93, CSC94a, CSC94b, Coh94, SG94, Bec94, EBG+94, Par94, PBA+94, PMR94, JAS95, MCU+95, HBM96, PHP+96, Por96, WSM+96, CCK+96, JAS96, JAS97a, JAS97b, JAS97c, Lie97, Gau98, NMY+98, SG99, BCE+99, DOT00, SBA00, BF00, COH00, MWD+00, DMS00b, RGH00, Cra01a, Cra01b, DDH01, DOT01a, DOT01b, Har01a, Har01b, HCG+01a, HCG+01b, Nel01, SBA01a, SBA01b, RTI02]; and
- **Department Inspector General reports** [RSV+93, BSV+97, GGU+97, AMD98, GUS+00].

M&S software credibility areas that affect decision-makers' confidence in simulation results and authoritative model representations:

- **Software engineering and development processes based on best practices** [You89, BL91, You93, Coh94, Dem95a, Som95, HCP96, Hoe96, Wie96b, Pre97, HBL97, Hin97, CS97, BW97, Rub97, RFL+98, Roa98, Woo98, HHK+99, Bas00, RSF+00, BDA+00, HHP00, HNC+00, RFH+00, BCC+01b, KM01a, Rak01, CHS+02, MR02, Li02, LQZ02];
- **Process improvement and reengineering** [Sch91, HC93, Hal95, Ham01b, PWC+95, FC99, MCR+00, BBF+01];
- **Modeling and simulation improvement** [BKR95, Ham96, HNP97, Hug97a, CR98, Koo99, Ham01a, LK00, BCB02, Bee02]; and
- **Emerging bodies of knowledge (BOK) in related knowledge areas** such as software quality measurement [HM96, Sch02a], project management [Fra94, Dun96], and modeling and M&S BOK [ACP+99], the Department's joint technical architectures [JTA02a, JTA02b], sound knowledge engineering practices for conceptual model development [Pac99a, Pac01a] and verification and validation [IEE86, IEE98b] of the conceptual model, data, and simulations [Pac02a, Bor02] conducted during the entire M&S life-cycle.

A top-level literature survey was also conducted for:

- ***Studies in the Department's Missile Defense M&S domain*** [Bra96, Li99, ISE99, Wel00, Bra00a, Bra00b],
- ***Performance recommendations supporting improved information technology investments within the Department and federal government***: [SBD+95, BPC+96, MLW+96, AVL+97, Hin97, Pai97, Dod98, PKM+98, Bro99, Mcc00, Rod00, RSF+00, CLP+01, CHS+02, FNP+02, BBF02].
- ***The scope and magnitude of simulation software credibility***. The problem with simulation credibility appears to be a systemic issue extending well beyond the Department's purview, and found in different domains and diverse, non-Department systems such as economic models [Pos00, KBA+02], transportation [And00], energy [Sta74], supply [PSA79], environmental protection [DMF+95], retirement forecasting [Dod97, Che86a, Che86b], supercomputing [Bro91], air pollution [GMW89, DMP+97], credit subsidies [Cal97], highway management [Sch01a], nuclear weapons [CFM99, HEC91], and manufacturing [AR00].

Finally, the Department M&S software programs must support and comply with applicable Federal Government and Department requirements, software development processes and product security measures including:

- ***Information operations / assurance and critical infrastructure protection requirements*** [EFL+97, LX99, ABB00, Bus01, FOP+01, GPH+01, WFP01, ALL02],
- ***Interoperability standards*** [HKL96, HMD99, MCL+99, HM00, BLB+01, BLR+02], and
- ***Legal requirements for information technology and performance measurements*** [CC96, ATL01, Cos01, Gla01, Man02, AS02a, AS02b].

C. THE DEPARTMENT'S MODEL AND SIMULATION FRAMEWORK

1. Background

National- and Department-level decision-makers need high-quality computer-based simulation models and modeling frameworks, supported by valid qualified and quantifiable levels of process and product credibility. The quest for confidence covers an ever-expanding range of M&S ranging from high-fidelity engineering models to highly aggregated campaign-level M&S. The potential spectrum of Department uses for M&S grows

continuously when one considers that “everything is a simulation except combat,”[BTE+93]. The terms model⁵³ and simulation⁵⁴ have several definitions, including Law and Kelton [LK00]⁵⁵, the IEEE [IEE90]⁵⁶, and the Department [DoD94]. In this research we use the Department’s definitions: a model “is a physical, mathematical, or otherwise logical representation of a system, entity⁵⁷, phenomenon, or process” [DoD94], while a simulation is “a method for implementing a model over time” [DoD94].

John von Neumann first proposed the concept of modern digital simulation as the application of gathering repetitive, statistical data on modeled phenomena, via the Monte Carlo method [Rot83]. [Tho97a] provides an historical account of computer-based simulation from the 1940s wartime combat operations research role and focus on improving the use of weapons, through the 1960s use of M&S for systems analysis. From 1940 to 1997, there have been four major eras in the evolution of software during the computer eras affecting the development of Department M&S software [Pre97]:

- The early years, characterized by custom software, batch operations and limited software distribution,
- The second era, from the mid-1960s to the late 1970s, experienced software products developed by the private sector, multi-user computer systems, on-line databases, and the first real-time systems, and the need for software maintenance grew,

⁵³ A model is a mathematical representation of a system, where a system is any sort of process or structure. A process model normally operates sequentially over time, but may use other dimensions. Sequential processes in time are modeled as an independent variable over a linearly ordered set using real numbers (continuous time) or integers (discrete time). A model has two types of system parameters, or variables characterized by range (the set from which it assumes a value) and type (random or deterministic). A deterministic variable assumes a unique from the range at each point in time, while a random variable is described statistically probability distribution over the range. System variables may be for input or output, and the resulting model may be static or dynamic. In a static model, the outputs at any time are only a function of the inputs at that time, possess no memory and has no recollection of previous events, whereas the outputs of a dynamic system, may depend upon past values, as well as present inputs, thus the system remembers certain information about previous inputs. The remembered information is called the state of the system. A structural model usually involves the representation of relations e.g., airline flight route [Mar83a]

⁵⁴ A simulation is the representation of certain features of the behavior of a physical or abstract system by the behavior of another system. Simulations employ computerized models of certain significant features of some physical or logical system. The object of the simulation process is to provide an experimental model for the accumulation of data on the target system, and is comprised of the following steps: experiment definition, modeling, computer implementation, validation, and data gathering [Rot83].

⁵⁵ The technique to imitate or simulate the real-world operations of facilities, processes, or systems supporting scientific study; which require a set of assumptions about how it works. The assumptions, usually mathematical or logical relationships constitute a model, used to gain an understanding of how the corresponding study behaves. A simulation employs a computer to evaluate a model numerically, collect data to estimate the desired true characteristics of the model [LK00].

⁵⁶ The IEEE defined a simulation as “(1) a model that behaves or operates like a given system when given a set of controlled inputs; (2) the process of developing or using a a model as in (1) [IEE90].

⁵⁷ An entity is a distinguishable person, place, unit, thing, or concept about which information is kept [DoD98].

- The third era, from the mid-1970s lasted over a decade and ushered in a period of low-cost computation power for consumer use, saw the development of distributed systems, and witnessed the initial efforts into artificial intelligence development,
- While the fourth era, from the mid-1990s to the present, included mobile and distributed systems, the Internet, network/parallel computing, expert systems, neural networks, powerful desktop personal computers, object-oriented technologies, and the software to integrate legacy systems [Pre97].

During these four periods the M&S community experienced the development of new computer-based models and simulations, a growing concern with credibility, emerging trends of repeated use of models for different applications and the development of families of models or hierarchies.

2. Classes of Simulations

The current Department M&S framework has three M&S classes: live, virtual, and constructive, supporting five major customer domains: training, operations, analysis, experimentation and acquisition [DoD02d]. Live simulation refers to real people operating real systems including field training exercises involving troops and actual equipment, which allow soldiers to use organizational equipment under actual environmental conditions, simulating combat. The live simulation also provides ground test data on actual hardware and software performance in an operational or development environment. Live test data supports simulation validation (e.g., calibration) using the model-test-model concept employed in the Department's acquisition process.

Virtual M&S involves real people operating simulated systems, digital representations of environments, organizations, systems, other entities, and processes. Virtual M&S put the human in the loop in a central role by exercising motor control skills, decision skills, or communication skills. Current state-of-the-art virtual M&S bring the system or subsystem and its operators together in a synthetic or simulated environment. Virtual M&S run in real time immersing players in synthetic environments or simulators. Virtual M&S may include actual hardware, stimulated by the output of computer simulations.

Constructive M&S involves simulated people operating simulated systems. Constructive simulations employ digital representations of environments, organizations, systems, other entities, and processes, operated with human players, or without human players. Human player interactions are limited to controlling units, and generally referred to as wargames. Human-in-the-loop (HIL) simulations provide the stress and decision-making associated with live simulations, and permit the introduction of multiple types of platforms for evaluation of actual interaction and interoperability.

Categories of constructive M&S consist of computer models, wargames, and analytical tools used across a range of activities. These activities include detailed engineering design [Law02] and cost [Kan01] or subsystem and system performance calculations to support development of technical specifications. Higher-level constructive M&S provides information on the outcomes of battles or major campaigns involving joint or combined forces; identify mission needs; and support operational effectiveness analyses measures of effectiveness (MOE), loss exchange ratios (LER), force exchange ratios (FER), and measures of performance (MOP). Stochastic⁵⁸ or deterministic⁵⁹ simulations approximate the real world. Other uses of constructive M&S include developing life cycle cost estimates, supportability analyses, and force management analysis of alternatives.

3. Simulation Model Hierarchies

The concept of a hierarchy of simulation models evolved in Europe and the United States during the 1970s with the objective of managing M&S complexity⁶⁰. However, the various hierarchy models lacked integration and usually consisted of a high-level model and several lower level M&S. [Ham96, HNP97] suggest that hierarchical modeling supports abstraction⁶¹. A variable resolution M&S concept for integrating heterogeneous M&S into the hierarchy has existed since the early 1980s [DH92a, DH92b].

⁵⁸ The theory of stochastic processes deals with events that develop in time or space and which cannot be described precisely except in terms of probability theory [Mar83b].

⁵⁹ A deterministic variable assumes a unique value from the range of set values at each point in time [Mar83a].

⁶⁰ Complexity refers to time and space resources needed to simulate a model, and is a dimension independent from resolution, although they are often related in a monotonic manner [Zei92b].

⁶¹ Abstraction isolates important aspects of the problem and suppresses the unimportant aspects [HNP97].

[HHP00] suggests the literature includes many other different M&S taxonomies. For example, [Zei92a] addressed model representations in modular hierarchies, while [Sch92] detailed model hierarchies of different resolutions. Providing other viewpoints, [HBF92] focused on maintaining process independence between different levels of resolution; while [BD02] identified hierarchical model framework to support improved design quality of object-oriented systems; whereas [Rei92] adopted a different five-level capabilities assessment viewpoint⁶². Lacking an architectural foundation, many of these hierarchy models depended on broad categorical definitions, heuristics, subject matter experts, and general qualifications to match an existing M&S to the model framework.

[Hug97a, Hug97b] proposes there is not a universally accepted way to categorize M&S, since M&S range over a vast spectrum of fidelity, precision, and resolution from detailed engineering representations to highly aggregated representations of force-on-force engagements, often involving multiple dimensions of structural, functional, temporal, and qualitative abstraction, which are not complete or necessarily independent of each other. [Ham96, HNP97] cite the close relationships between the concepts of aggregation and decomposition with the levels of abstraction concept. [Ham96, HNP97] also note that abstraction (e.g., ignoring behavior or minor differences) indicates a simplification strategy, whereas abstraction by lumping together indicates the use of an aggregation method. [Ham96, HNP97] further suggest the use of an open architecture as a means of developing simulations with the flexibility to support multiple levels of aggregation.

With the Department's growing need to address human behavior, [MT01] proposed a hierarchical model composed of virtual individuals, groups, and crowds. [Sta97] provides a different perspective, suggesting that a hierarchy of linked measures of merit-- including measures of performance, measures of functional performance, and measures of force effectiveness-- may improve acquisition activities. Finally, [PMR94] introduced a Notional Hierarchy of Technologies⁶³ with five categories for assigning M&S enabling

⁶² The capabilities assessment levels built on weapons system level analysis, and added a supporting analyses level, program alternative analyses, mission analyses, and at the top of the pyramid, campaign analysis [Rei92].

⁶³ The enabling technologies included standards and protocols at the lowest level, and added layers for underlying collaborative technologies, utilities and infrastructure, applications, and an integrated acquisition environment at the top [PMR94].

technologies: standards and protocols, collaborative technologies, utilities and infrastructure, applications, and at the top, an integrated acquisition environment.

The Department's simulation hierarchy includes four categories of M&S for communicating the scope or level of model detail: engineering, engagement, mission and campaign, [PMR94, CS97], generally with more abstraction moving up the simulation hierarchy and more refinement, or level of detail moving down the simulation hierarchy. The M&S characterized in the hierarchy represents increasing aggregation from the engineering models at the base of the hierarchy to the highly aggregated representations supporting a theater level view [HM95]. Not every model or simulation fits perfectly into one of the classifications, and M&S fidelity quality, addressed later, can vary at any point in the M&S hierarchy.

The engineering M&S include very detailed simulations, concentrating on individual components and their interaction, with application in design analysis, risk mitigation in component performance and tradeoff, requirements specification, and performance analysis. Engagement M&S usually depict friendly versus enemy engagements, and generally provide some type of system effectiveness outcome supporting analysis of alternatives (AoA), requirements evaluation, system effectiveness, tactics, techniques, and procedures (TTP) development and assessments, tradeoff analysis, and test support. Virtual prototypes usually run as a simulation at the engagement level and above.

Mission/Battle level M&S depict multi-platform, multi-task force packages and usually provide some type of mission effectiveness outcome. Mission/Battle M&S support AoA and requirements evaluation, deployment analysis, weapons integration, TTP development and assessments, training, and wargaming. At the apex of the M&S hierarchy sit the highly aggregated Theater/Campaign models. These M&S also support AoA, requirements evaluation, TTP development and assessment, war gaming, and battle-staff training. Figure 2-2 is an annotated illustration of the current Department M&S hierarchy [PMR94].

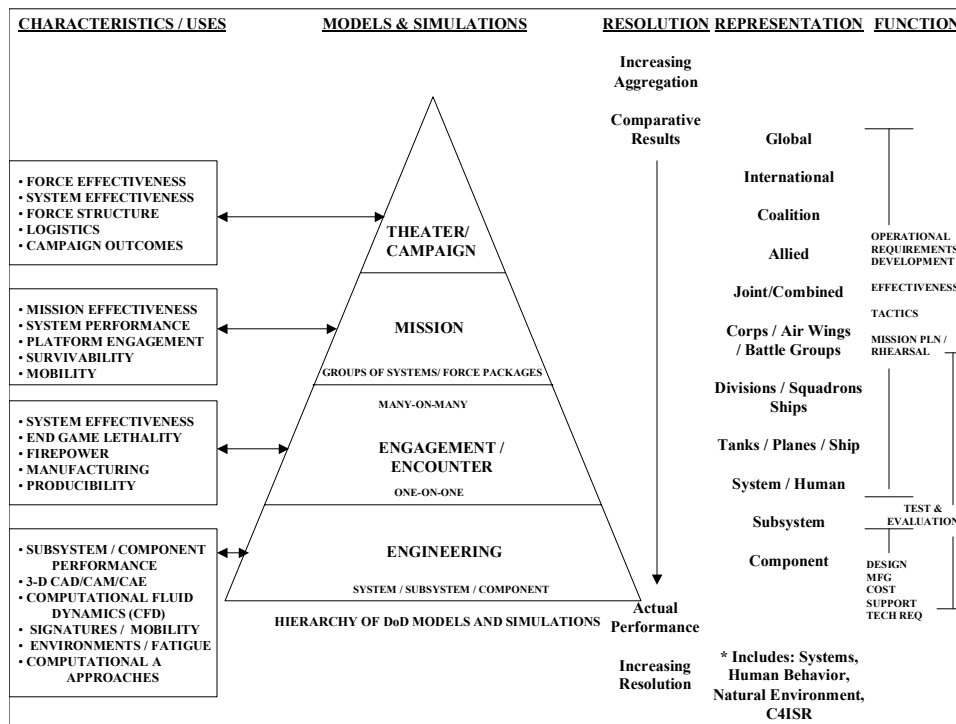


Figure 2-2. Department M&S Hierarchy (After [PMR97])

4. Distributed Simulations

The Department supports three architectural implementations for distributed M&S: the Aggregate-Level Simulation Protocol (ALSP), the Distributed Interactive Simulations (DIS)⁶⁴ application protocols [IEE95, IEE98a], and the most recently approved method, the HLA. For distributed simulations [Ham96, HNP97], an individual simulation called a federate interoperates with other selected federates through a common technical framework, such as the HLA, forming federations.

The IEEE supports the HLA rules, *IEEE 1516/D1* [IEE98e], HLA object model template specification [IEE00c], HLA federate interface specification [IEE00d], and HLA object-model template [IEE00e]. The HLA object-model-template specification [IEE00e] defines the format and syntax of HLA object models, but not the content. With the evolution of the HLA beyond the Department, a growing demand for interoperability, and ap-

⁶⁴ DIS is a synthetic environment within which humans may interact through simulations at multiple sites networked using compliant architecture, modeling, protocols, standards, and databases [DoD95].

proval of the HLA standard by the IEEE, the definitions employed in this research will remain consistent with approved IEEE definitions (e.g., accuracy⁶⁵, resolution⁶⁶) whenever possible. Legacy Department definitions included in the research provide context for decisions, issues, and possible ambiguities faced by Department simulation developers. Interestingly enough, the term *fidelity*, previously defined by the Department [DoD98], and used liberally in the M&S literature, is not included in the IEEE's standard glossary of software engineering terms [IEE90], or the HLA object model template specification [IEE00e].

5. The Growing Need for Credible Models and Simulations

As M&S technology developed in the Department, several functional areas, including the acquisition, training, operations, analysis, and experimental functional domains incorporated simulation technology into their infrastructure with the goal of enhancing performance, reducing overall costs, and attaining improved schedule milestones. In all of these functional domains, M&S is now a key component of the domain toolbox, although the past performance of simulation technology has not always met expectations.

a. The Department's Acquisition Domain

The Department's acquisition domain, including developmental and operational test and evaluation communities, employ M&S in the acquisition of new operational systems [DoD03b, DoD03c]. The Department also incorporated M&S into the acquisition process to reduce cost, improve performance, maintain schedule and mitigate risk. The requirement for credible M&S results are incorporated in the Department acquisition directive [DoD03b] and instructions [DoD03c], which state,

- The conduct of test and evaluation, integrated with *modeling and simulation*, shall facilitate learning, assess technology maturity and interoperability, facilitate integration into fielded forces, and confirm performance against documented capability needs and adversary capabilities as described in the system threat assessment [DoD03b].

⁶⁵ Accuracy according to the IEEE is the measure of the maximum deviation of an attribute or parameter value in the simulation or federation from reality or some other chosen standard or referent [IEE00e].

⁶⁶ The IEEE defines resolution as the smallest resolvable value separating attribute or parameter values that can be discriminated. Resolution may vary with magnitude for certain data types [IEE00e].

- The T&E [test and evaluation] strategy shall provide information about risk and risk mitigation, provide empirical data to validate model [DoD03c]
- Adequate time and resources shall be planned to support pre-test predictions and post-test reconciliation of models and test results, for all major tests. [DoD03c].

In addition, “The PM shall plan for M&S resources throughout the acquisition life cycle. The PM shall identify and fund M&S resources early in the life cycle” [DoD03c].

In the area of acquisition reform [WHA+00], the National Performance Review (NPR) established a Department acquisition reform objective of reducing new system delivery time by twenty-five percent, subsequently increased by the Department to fifty percent [ACP+99], in part by acquisition reform initiatives capitalizing on engineering trade-offs and M&S performance modeling to achieve the NPR goals. Simulation software also supports the evolving Department transformation⁶⁷ objectives [OAH+97, DGH+98, Rum01, Rum02a, Rum02b] to improve the quality, interoperability, effectiveness, cost, performance, and schedule metrics of the Department’s business and warfighting systems supporting the National Military Strategy [HBM96, Rum02b].

In the acquisition arena, transformation includes the replacement of inflexible requirements-based methods with a capabilities-based methodology [Rum01] supported by evolutionary acquisition practices [Ald02d] for major Department acquisitions including the Nation’s missile defense program. The success of M&S to support the Department’s transformation process hinges in part, on the acceptance and successful implementation of new or enhanced M&S methods and techniques such as the Simulation, Test, and Evaluation Process (STEP) [CS97], Simulation-Based Acquisition (SBA) [San97b, San97c, FT98, JMS98, San98b, SBB+98, Bro99, GS99, San99, SBA99, SBB+99a, SBB+99b, AT00, SBA00, BGK+00, KLM+00, KM01b, Zit01, ECP02], Distributed Product Descriptions (DPD) [HH00b], the Army’s Simulation & Modeling for Acquisition, Requirements, and Training (SMART) program [Bia00, EKH00, EKH01], and the development of new collaborative environments [San98a, San98b, San99, Hol01] including the Joint Synthetic Test

⁶⁷ Transformation results from the exploitation of new approaches to operational concepts and capabilities, the use of old and new technologies, and new forms of organization that more effectively anticipate new or still emerging strategic and operational challenges and opportunities and that render previous methods of conducting war obsolete or subordinate [Rum01]

and Evaluation Battlespace [San98b, Bra98, Hol01], the Advanced Distributed Simulation (ADS) [Kec99] capability, and testing system-of-systems interoperability (e.g., the Single Integrated Air Picture) [Dah02].

The Revolution in Business Affairs, a Department acquisition reform imperative, leverages M&S to reduce Total Ownership Cost and improve delivery for new systems by employing SBA processes [Eva96, Lav97, San97b, BSB+98, Hol01] among other best business practices. The vision for the Department's SBA initiative is "an acquisition process in which the Department and industry are enabled by robust, collaborative use of M&S technology integrated across acquisition phases and programs" [San97b]. The objective of SBA is to reduce schedule time and risk in the development and maintenance of a product with an improved quality at a lower cost in a collaborative government/private sector environment employing an Integrated Product and Process Development methodology over the system life cycle. However, Department support for these initiatives requires the development of business-case framework to evaluate the investment opportunity and the potential return-on-investment [BGK+00].

STEP is an iterative model-test-model approach with a goal of improved M&S credibility, test community confidence, and a better understanding of the strengths and limitations of complex models, M&S and data by decision-makers. The STEP paradigm of model-simulate-fix-test-iterate [CS97] embodied in the DoD developmental and operational test and evaluation (T&E) process has the potential to support improved confidence in tests and evaluation results.

Department testers [DOT89, Bra98, Kec99, BCE+99, Pau99, Obr99, Coy00a, Coy00b, DOT01a, MOR02] also seek answers to where M&S may provide credible information and under what conditions M&S can replace testing. Hydrocode⁶⁸ models track shock propagations and materiel distortion to support vulnerability/lethality assessments [TIL+95, MH98] and support Department Live Fire Test & Evaluations (LFT&E). The LFT&E testers characterize the results of fragment and kill-vehicle interaction with the target, which are subsequently incorporated into lethality M&S. Current research into hy-

⁶⁸ Hydrocodes are physics-based, first principle M&S of hyper velocity solids interaction used in vulnerability and lethality M&S to track shock propagation and distortions of multiple materials [IDA01]

drocodes models cited by [IDA01] indicates current hydrocode model technology is limited to the primary effects of a weapon on a system and has major voids in secondary weapons effects, suggesting that M&S can complement, but not replace live-fire tests.

[DOT01b] cites the quality of Department software testing as an ongoing systemic issue, needing major improvements in the area of M&S support for improved test and evaluation readiness capabilities. [Coy00a] supports this assessment and suggests additional emphasis in the area of best software test processes to improve the quality of the Department's operational testing program. However, [BCE+99] notes that simulating complex system interactions, limited by cost, time, personnel, and technology, adversely impacts the accuracy and fidelity of the M&S supporting Departmental test programs.

A Departmental M&S survey [DDH01, HBD+01] sponsored by the Director, Operational Test and Evaluation (OT&E) to gain insight and confidence, performed a limited review of Department M&S including types, characterization, management, management activities, cost, and noted for VV&A that much policy guidance has been developed, but implementation appears minimal. The increasing dependence on M&S noted by [DDH01] added increased emphasis on management activities considered critical to success including:

- M&S Support Plan
- M&S Staff
- VV&A Plan / processes
- M&S reuse
- Collaborative environment
- Performance incentives [DDH01, HBD+01].

The February 2001 Department M&S Workshop for Assuring M&S Credibility for Defense Acquisition and T&E: Survivability, Lethality and Mission Effectiveness [DOT01a] corroborated the results from [DDH01, HBD+01]. The Workshop identified several systemic, ongoing factors hampering M&S and VV&A programs, and concluded that acquisition program managers need more help implementing M&S and VV&A within funding constraints [DOT01a]. Furthermore, [DOT01a] cautioned there is “no definitive or reliable link between following M&S and VV&A policy and technical guidance and having confidence in M&S credibility for acquisition applications” [DOT01a]. As a result of these

systemic issues, [DoD01a] concluded that the lack “of adequate VV&A resulted in uncertain or poor representations of military forces in M&S, thereby causing negative training, inaccurate analyses, and lack of trust in M&S” [DoD01a].

An October 2002 Military Operations Research Society (MORS) M&S Workshop [GF02] addressed relevant topics to this study including: the status and health of the Department’s M&S policy, the capability versus expectations for using M&S in Departmental acquisition, the Department’s VV&A practices, and addressed the state of the Department’s M&S credibility since [DDH01, HBD+01, and DOT01a]. The workshop members concluded that the Departmental M&S policies were adequate and identified several areas needing additional emphasis including risk mitigation, operational assessments, and test-prediction and post-test reconciliation [GF02].

The workshop members also noted that most of the Department’s M&S effort appeared in areas requiring the highest fidelity, although current Departmental M&S policies or funding levels do not support these efforts. Specific areas of concern included: evaluations beyond the bounds of a test, system performance prediction methods, and the expanded use of synthetic test environments in lieu of live tests [GF02]. Numerous VV&A issues included:

- The continuing perception that VV&A is too costly,
- VV&A not adequately integrated in the M&S process,
- Programs lack independent V&V,
- VV&A lacks up-front estimation of the effort, causing unnecessary work,
- VV&A lacks multidisciplinary membership; and
- The model-test-model and model reuse paradigms had not evolved as anticipated [GF02].

The workshop findings according to [GF02] also noted the failure of Department M&S users, including decision-makers to understand the problem or question under review, contributing to the unfocused use of M&S, unbounded VV&A, and the unnecessary expenditures of resources. [GF02] proposed examining options for program office incentives supporting effective M&S use, improved reuse, and development of M&S for cross-program use.

b. The Department's Training Domain

The Department uses M&S across a broad spectrum of training and education from individual platform simulators to Joint Task Force operational- and strategic-level exercises employing multiple simulations replicating a theater of war [DoD03a]. Success of the Department's transformation process also depends upon the transformation of the current Department training paradigm [OPV+02] as directed by the Defense Planning Guidance [DPG02] for fiscal years 2003-2007. Achieving the strategic goal of a training transformation requires a common operational architecture providing interoperability of live, virtual and constructive training and mission-rehearsal systems "to build unparalleled military capabilities, which are knowledge-superior, adaptable, and lethal" [OPV+02].

M&S also has the potential to greatly improve critical Department program initiatives involving joint/coalition doctrine development, training, operations, force management, requirements development, materiel readiness, life cycle/ total ownership costs, human behavior, education, mission planning, research and development, modernization, analysis, contracting, and joint warfare experimentation, although at this point a scientifically definitive conclusion on its effectiveness cannot be drawn.

c. The Department's Operations Domain

Department M&S supports deliberate planning, crisis course of action analysis, and mission rehearsal. The Department's draft *Modeling and Simulation Strategic Plan 2003-2012* [DoD03a] provides a challenging vision statement for Department M&S efforts citing the need to: "Provide the Power to predict, Prepare, and Respond Rapidly to Win Decisively Against Any Threat" [DoD03a]. Software and information support the goal of twenty first century military decision superiority identified by [Rum02b] for U.S. forces to communicate with each other, share information, and see the same, precise, real-time picture of the battlespace, while simultaneously sharing knowledge of friendly and enemy locations.

Furthermore, the Department determined that M&S "has the potential to significantly enhance the military capability and readiness of US forces" [DoD01a]. In or-

der to achieve these goals, “DoD must also undertake high-fidelity transformation exercises and experiments that address the growing challenge of maintaining space control or defending against attacks on the U.S. national information infrastructure” [Rum01]. Improving the Nation’s missile-defense capability is a major pillar of the National Security Strategy [Bus02d] and Homeland Defense [Bus02c]. Capability-based acquisition [DoD03b, DoD03c] places additional requirements on future M&S requirement including the need to model a potential adversary’s limitations and strengths.⁶⁹

There is also an increasing Department demand for by the Department’s warfighting community for credible and interoperable new modeling and simulation projects supporting the development of the four major *Joint Vision 2020* [She00a] operational concepts – dominant maneuver, precision engagement, focused logistics, and full dimension protection⁷⁰ [She00a], linked by three critical interdependent factors: interoperability, innovation and decision superiority. Interoperability, based on open systems architecture, is the keystone of future military operations and “the foundation of effective joint, multinational and interagency operations” [She00a]. Decision superiority is “...superior information converted to superior knowledge to achieve...better decisions arrived at and implemented faster than an opponent can react” [She00a]. The concept of military decision superiority has roots in the very genesis of warfare. From warfare’s inception the world’s military forces continually refined decision superiority to achieve a strategic advantage or victory in battle. In the future, M&S will provide real-time decision support during mission execution [DoD03a]

A recent Department initiative, the *Command, Control, Computer, Communication, Intelligence, Surveillance and Reconnaissance (C4ISR) Information Superiority Modeling and Simulation Master Plan* for improving information and decision superiority, plans for environments “constructed, to the extent possible, from affordable, reusable com-

⁶⁹ The scope of this effort includes the adversary’s Political, Military, Economic, Social, Infrastructure and Information (PMESII) [DoD03a].

⁷⁰ *Dominant Maneuver* is the ability of joint forces to gain positional advantage with decisive speed and overwhelming operational tempo in the achievement of assigned tasks. *Precision Engagement* is the ability of joint forces to locate, surveil, discern, and track objectives or targets; select, organize, and use the correct systems; generate desired effects; assess results; and reengage with decisive speed and overwhelming operational tempos as required, throughout the full range of military operations. *Focused Logistics* is the ability to provide the joint force the right personnel, equipment, and supplies in the right place, at the right time, and in the right quantity, across the full range of military operations. *Full Dimension Protection* is the ability of the joint force to protect its personnel and other assets required to decisively execute assigned tasks [She00a].

ponents interoperating through an open systems architecture” [DoD02d]. [SH96] proposed organizing around information enabling distributed operations with a decentralized decision-making process, as a means to establish the environment for quality and speed. However, issues with M&S reuse and interoperability in the Department indicate that the “value of past investments in M&S is depreciated because the DoD does not have the capability to easily identify algorithms, data sets, models or M&S that can be reused” [DoD01a].

The Department requires additional resolution, level of detail and refinement to existing M&S tools, especially for opposing-force representations of entities, which “must be significantly increased to permit realistic assessments of C4ISR performance [and] the degree of fidelity and detail of M&S of C4ISR functions be commensurate with this importance” [DoD02d]. The Department also requires increased fidelity and M&S with “more accurate representations of all elements of the mission space” [DoD01a]. Although the Department made progress in the area of developing authoritative representations, [DoD01a] notes “many combat, combat support, and combat service support systems abound, making it difficult to determine which are authoritative or best of breed.” [DoD01a].

d. The Department’s Analytical Domain

The military operations research community advanced operational decision-making over the past fifty years [You97 MK98], employing computer-based models and software M&S as “tools supporting an analytical process with better decisions being the objective” [Hug97b]. [Pre97] suggests that software supports a multitude of uses, including M&S, and delivers information, which may be the most critical product of the twenty-first century. Analysis supported by M&S occurs in every Department domain. In addition, the Department uses M&S in analysis or wargames supporting force structure analysis and long-range strategic planning, both major Department activities.

e. The Department's Experimental Domain

The Department also uses M&S to explore new capabilities, portray future operational environments and develop new tactics, techniques, procedures, doctrine, organizational structures, processes, and systems [DoD03a]. The Revolution in Military Affairs initiatives employ large-scale, legacy M&S, future evolutions of legacy M&S software systems, and new generations of live, virtual and constructive DoD M&S, for experimentation, advanced-concept exploration, analysis, and requirements elicitation. These initiatives support expanded joint, combined, or multinational federations, enable M&S-based acquisition, and advance development of distributed system engineering collaborative environments.

Department M&S users require timely simulation applications, improved analysis capabilities, and improved operational support with M&S integrated with go-to-war systems, including international collaborative initiatives and coalition operations [DoD03a]. [Rum01] cited M&S in the 2001 DoD Quadrennial Defense Review as critical for future joint, combined and coalition operations and the dynamic military transformation process to the point where “U.S forces rely heavily on war games and M&S to support this program of field exercises and experiments” [Rum01].

Millennium Challenge 2002 (MC02), the most elaborate war game ever conducted by the Department, involved 13,500 joint service participants, federated forty-two models [Koz02] in seventeen locations and nine live training sites, and concluded three weeks of playing a classified 2007 scenario with specified objectives [MC02], on August 15, 2002, after taking two years and \$250 million to plan and execute [Nur02, Nay02]. Lessons-learned from MC02 impacting M&S credibility and confidence of results identified in [Nay02, Sch02b] includes recommendations for a better Department doctrine defining how to achieve value from the growing portfolio of the Department's M&S, independent reviews of the experiments, a better definition between objectives designed to validate concepts and objectives designed to identify victors of the virtual battles, and an increased emphasis on effectiveness, including smaller games and M&S to test and train.

D. SUMMARY

Chapter II provided a literature survey on the current research areas relevant to Department simulation credibility. The literature survey clearly revealed two major eras affecting Department simulation credibility. The Growth Era before 1994 experienced over forty years of research, experimentation, and development of a wide variety and types of M&S with varying degrees of credibility. The Managed Era, or post-1994 period, introduced policy and management procedures as the Department began to plan, program, develop policy and procedures, budget, and corporately manage the Department Enterprise-wide M&S portfolio developed during the Growth Era and plan for new joint-simulations to replace many of the legacy Growth Era M&S. This chapter also introduced the Department's M&S management framework for establishing credible simulations, including classes of simulation, simulation hierarchies, and categories of M&S (live, virtual, constructive), within a prescribed hierarchy of M&S (engineering, engagement, mission, and campaign).

The current Department M&S hierarchy maintains only general, qualified levels for M&S fidelity and resolution, provides only limited support for the user to ascertain a simulation's attributes. The model provides little if any support for quantifiable measures of fidelity and resolution. In addition, metrics based on this model tend to be qualified, subjective, and lack authority. In this research we analyzed the current Department M&S Hierarchical model as a core asset and reviewed alternatives for mining reverse engineering and reengineering these core assets into product line architecture.

Other Department initiatives included the establishment of new distributed simulation capabilities supporting the expanding need of credibility by the major stakeholders. Major categories of stakeholders influencing the Department's M&S management priorities included the acquisition, training, operations, analysis, and experimental domains.

Research indicated that M&S credibility was a multi-dimensioned problem. Chapter III through Chapter VII review different dimensions of the Department's M&S credibility challenge. A taxonomy provided at Figure 2-1 represents the synthesis of the major areas identified in the literature search to counter these major challenges and support credibil-

ity in M&S and confidence in the results they produce, including the architectural framework, VV&A, software engineering, process improvement, and knowledge acquisition.

THIS PAGE INTENTIONALLY LEFT BLANK

III. CREDIBILITY OF DEPARTMENT SYSTEM REPRESENTATIONS

A. INTRODUCTION

Chapter III introduces M&S credibility issues and identifies concerns that decision-makers have with Department simulation model representation and results, followed by an overview of the Departments VV&A processes, and several factors contributing to the implementation of, or shortfalls of the VV&A processes in the Department. Two major periods, the Growth and the Managed Eras characterize the Department's experiences with M&S. The Growth period included the forty years of M&S development prior to 1994, and the Managed Era includes the Department's post-1994 efforts to improve M&S credibility.

B. THE GROWTH ERA: M&S CREDIBILITY ISSUES IDENTIFIED

In 1978, a United States Comptroller General report [PAD78] identified a number of major systemic quality issues in the simulation software development practices of government agencies and concluded that:

There are no generally accepted guidelines for establishing models credibility. In addition, there is not a generally accepted threshold beyond which a model can be termed 'credible', and the concept of credibility varies greatly between model builders/developers and users/decision-makers as well as among individuals within either of these two broad groups [PAD78].

The need for credible M&S grew in the Nation's private and public sectors. By 1980, information from computer-based simulations played an expanded role in the analysis of public policy issues and supported decisions for systems costing billions of dollars [Sta74, PAD78, PAD79a, PAD79b, PAD80, Che86a, Che86b, Che87]. Computer-based simulations also aided political-military analysis [DBW87, DH92a, DH92b], supported the development of strategic course of action development underpinning national policy [Lie97], assisted elements of the Department's decision-making process [HRA+01], and recently emerged in an expanded role in the United States National Security Strategy [Bus02d].

The report *Models, Data, and War: A Critique of the Foundation for Defense Analysis*, [PAD80] found the same major systemic shortcomings in Department-wide M&S

cited earlier by the General Accounting Office (GAO) in other government agencies including the GAO [PAD78], the Federal Energy Commission [PAD79a], and a survey of other government agencies [PAD79b]:

- Overall poor quality,
- Lack of documentation,
- Inadequate understanding of model assumptions, limitations, and capabilities,
- Insufficient model development practices,
- The need for better coordination between the model developer and the user,
- The requirement for better monitoring of development efforts,
- The need for improved procedures to update and maintain models,
- Improve how the M&S credibility is established,
- Improve how the validity of the model is determined,
- Problems with finding data to make the model function,
- Models supporting credible Department decisions should be transparent, appraised⁷¹, and consistent [PAD78, PAD79a, PAD79b, PAD80].

However, by 1988 the Department realized that the existing ways and means for determining simulation credibility were insufficient and improving simulation credibility and user confidence in simulation results would require new policies, procedures, resources, and organizations. A Secretary of Defense memorandum [Car88] noted that, “as the need to improve the capabilities and credibility of simulation continues to increase, we are redoubling our efforts to develop comprehensive, consistent and credible guidance to the services” [Car88]. The Director, Operational Test and Evaluation, stated that when “results from modeling and simulation are used...special care is necessary to ensure they are credible” [Kri89]. Early evaluations of simulations used in operational tests defined simulation credibility as a, “fragile commodity...[that] as applied to the M/S processes and results, is a combined impression of the inputs, processes, outputs, conclusions, the persons or agencies involved, and the strength of the evidence presented” [DOT89].

C. THE MANAGED ERA FOR CREDIBLE DEPARTMENT M&S

In response to the identified systemic deficiencies, the Department implemented and updated several M&S management initiatives including M&S management policy

⁷¹ The term ‘appraised’ in this context means the model is mathematically correct, matches the real world, and uses empirically valid data [PAD80].

[DoD94], master plans [DoD95, DoD01a], and verification⁷², validation⁷³, and accreditation⁷⁴ instructions and procedures [DoD96, GMS+96, DoD00a, RPG00, DoD01b, DoD02a], including VV&A of conceptual models, and data⁷⁵, in order to resolve these issues. The Department published the first Department M&S management directive, *DoD Modeling and Simulation Management* [DoD94] in January 1994 to resolve systemic issues, provide departmental M&S policy, determine organizational responsibilities, define information requirements⁷⁶, establish an investment program, and develop procedures, including VV&A, to “Strengthen the uses of M&S in the Department of Defense” [DoD94]. [DoD94] also assigned Department M&S Executive Agents (MSEAs) responsibilities to DoD Components for common⁷⁷ and general-use⁷⁸ M&S applications such as terrain or environment.

However, major systemic shortcomings persisted to undermine confidence in simulation results when the Department published the first Department M&S master plan, *DoD Modeling and Simulation Master Plan* [DoD95]. At the time, M&S were still too costly, failed to meet users needs, remained stove-piped within the functional communities, experienced excessive development time, lacked credible VV&A procedures, and needed improvements in many quality characteristics: interoperability⁷⁹, security, maintainability, extensibility, and usability [DoD95].

Concurrently, a 1995 Modeling and Simulation Benefits Task Force [WSM+96] completed a limited review of the Department’s corporate M&S assets to document the quantifiable benefits of the Department’s investment in M&S. Based upon a very limited sample size and meta-analysis, [WSM+96] achieved mixed results and concluded that: “a

⁷² Verification is the process of determining that a model or M&S implementation accurately represents the developer’s conceptual description and specification. Verification also evaluates the extent to which the model or M&S has been developed using sound and established software engineering techniques [DoD95].

⁷³ Validation is the process of determining the extent to which a model or M&S is an accurate representation of the real world from the perspective of the intended use(s) of the model or M&S [DoD95].

⁷⁴ Accreditation is the official certification that a model or M&S is acceptable for use for a specific purpose [DoD95].

⁷⁵ Data is a representation of facts, concepts, or instruction in a formalized manner suitable for communications, interpretation, or processing by humans or by automatic means [DoD98].

⁷⁶ “When possible, existing information systems shall be used or modified rather than creating new systems” [DoD94].

⁷⁷ Common-use M&S are applications, including services or materials, provided by a DoD Component to two or more DoD components [DoD94].

⁷⁸ General-use M&S applications are representations used by or common to many M&S, including the physical environment or environmental effects such as terrain, atmosphere, or hydrographic effects [DoD94].

⁷⁹ Interoperability is the ability of a model or simulation to provide services to, accept services from, other models and simulations, and to use the exchanged services to operate effectively together [DoD95].

formal reporting mechanism does not exist for gathering information, nor do methodologies exist for objectively assessing the value of using M&S” [WSM+96].

A 1996 DoD *Study on the Effectiveness of Modeling and Simulation in the Weapon System Process* [PHP+96] identified three major types of M&S challenges: technical, cultural and managerial, and fifteen specific issues, many previously cited by [PAD78, PAD80, DoD95]. Many of the systemic challenges identified by [PHP+96] addressed credibility concerns with issues including proprietary models and data, security of data, availability of data descriptions, interoperability of M&S tools, consistent variable resolution⁸⁰ descriptions, physics-based models based on empirical data rather than physical principles, and continued VV&A deficiencies.

[PHP+96] also cited proprietary data, and resistance to funding V&V as two M&S management problems in the Department. The Department also acknowledged the limited research and review of V&V methodologies within the community and that the “verification and validation (V&V) process leading to accreditation is expensive and not well understood” [PHP+96]. In addition, the complexity of the software engineering process of non-trivial software-intensive simulations and establishment of M&S credibility [Ros01, WPL02] add difficulty to the validation process and may limit credible validation efforts [KM01a].

These challenges highlighted the lack of “broadly accepted community standards for representing military systems and organizations in M&S” [DoD95]. Consequently, representations of the same system in different models were frequently incompatible according to [DoD95], limiting the effectiveness of the federation objectives and reducing confidence in the results as a consequence of unresolved substantive interoperability issues. To improve M&S credibility, [DoD95] established six Department-wide M&S objectives with seventeen sub-objectives, and one hundred and fifteen supporting action. The six major Department M&S objectives delineated by [DoD95] established requirements to:

⁸⁰ Resolution is the degree of detail and precision used in the representation of real-world aspects in a model or simulation [DoD95].
Resolution: 1. The degree of detail used to represent aspects of the real world or a specified standard or referent by a model or simulation. 2. Separation or reduction of something into its constituent parts; granularity [RPG00].

- Develop a common technical framework (CTF) for M&S with sub-objectives for HLA conceptual models of the mission space⁸¹ (CMMS), and data standardization,
- Develop timely / authoritative representations of the natural terrain,
- Develop authoritative representations of systems,
- Develop authoritative representations of human behavior,
- Establish a M&S infrastructure to meet developer / end-user needs with a VV&A sub-objective,
- Share the benefits of M&S [DoD95].

Acknowledging the criticality of computer-based information [JBC+99] stated: “Software is the physical structure of the information age... It is fundamental to economic success and national security.” However, here may be even more challenges, as the bipolar threats of the late twentieth century fade away and are replaced by the modern twenty-first century world with a multitude of military asymmetric threats, [Lie97] cautions that it is:

Now necessary to address a much more confusing set of problems: unexpected environments, non-traditional forces, new missions, and new concepts of operation. Relevant data are limited. Many of the widely used models of the last 40 years or so may be inappropriate for the new arena [Lie97].

Trying to overcome over forty years of modeling and simulation history without V&V guidance, while maintaining a large inventory of legacy models, and attempting to stay ahead of the learning curve, the Department developed the first VV&A instruction, *DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)* [DoD96], and VV&A best practices guide, *Department of Defense Verification, Validation, and Accreditation (VV&A) Recommended Practices Guide* [GMS+96]. In addition the Department continuously updated M&S policy [Gan98, Gan99, Wol01], verification and validation instructions [DoD00a, DoD01b, DoD02a], master plans [DoD01a, DoD02c, DoD02d], manuals [DoD98], verification and validation technical guidance [GMS+96, RPG00] and developed a strategic plan [DoD03a] to support these objectives. Department

⁸¹ Designated as the Conceptual Model of the Mission Space (CMMS) in the [DoD95]. The term CMMS is in the process of officially being changed to the Functional Description of the Mission Space (FDMS) [RPG00].

components⁸² also established supplemental policies and procedures [AP93, AR97, DON99, AFI00].

The Department's first strategic plan [DoD03a], provides six strategic goals with twenty-nine objectives supporting the new M&S mission and vision statements to help the Department conduct war, transform the force, strengthen, Joint warfighting capabilities, develop and test new concepts to:

- Operationalize M&S as a real-time decision support tool for the warfighter (five supporting objectives),
- Provide rapidly and readily accessible and executable M&S (six supporting objectives),
- Increase M&S awareness, education, training, and collaboration (four objectives),
- Effectively represent the complexity of modern warfare across the full spectrum of operations (five supporting objectives),
- Employ M&S to accelerate acquisition, reduce life-cycle-costs, foster interoperability, and improve quality of systems (five supporting objectives),
- Align and increase science and technology investment to transform M&S for the warfighter (four supporting objectives) [DoD03a].

The Department and the Service components also developed M&S guidelines and best practices [PMR94, GMS+96, Fal97, SAF00] to improve confidence by decision-makers that Department M&S provides reasonable, correct, defensible and credible results for given situations, circumstances and environments. The private and technical sector, including the Simulation Interoperability Standards Organization (SISO), the International Standards Organization (ISO), and the IEEE also contributed to the growing M&S body of knowledge. In addition to the development of policies, plans, procedures, instructions, and guidance the Department established the Defense Modeling and Simulation Office (DMSO) to plan, program, coordinate and budget the Department's M&S program and an Executive Council for Modeling and Simulation (EXCIMS) to provide assistance and advice on major M&S issues [DoD94].

The Department also made significant investments to improve the credibility in M&S during the 1990s. These major investment initiatives includes the common technical framework initiatives for HLA, development of a HLA-compliant Runtime Infrastructure

⁸² DoD Components include the Office of the Secretary of Defense (OSD) Components, the Military Departments, the Chairman of the Joint Chiefs of Staff, the Combatant Commands, the General Counsel of the Department of Defense, the Inspector General of the Department of Defense, the Defense Agencies, and the DoD Field Activities [DoD94].

(RTI), authoritative data sources, including the Environmental Effects for Distributed Interactive Simulation (E²DIS) program [PBG+94], and the Synthetic Environment Data Representation and Interchange Specification (SEDRIS) project, and major joint simulation development efforts such as the JSIMS, JWARS, and JMASS initiatives.

D. ESTABLISHING DOD M&S CREDIBILITY AND USERS CONFIDENCE

The Department established the M&S VV&A process to develop credibility in M&S software and maintain user confidence in the results. The Department and the private sector M&S community developed a body of research, studies, best practices, and analysis focused on improved verification and validation theories, methods, techniques, tools, and procedures to remedy systemic issues and shortcomings identified by reports, studies, and other assessments cited earlier.

Research into the Department's VV&A process identified several patterns between the individual VV&A processes and several other areas affecting the establishment of M&S credibility. The areas identified in Figure 3-1 address topics consistently raised as concerns and issues in the VV&A literature and provides a synthesized M&S credibility taxonomy addressed in the following chapters:

- The Department's verification process,
- The Department's validation process,
- The Department's accreditation process,
- The role of data in the VV&A process,
- Verification and validation techniques,
- Shortfalls of the Department's VV&A process,
- Software development / software engineering practices,
- The Department's M&S framework,
- Heterogeneous data,
- VV&A practices,
- Organizational areas and issues,
- The Department's Common Technical Framework,
- VV&A techniques,
- Software architecture,
- Quality,
- Process Maturity,
- Secondary contributing factors (e.g. reuse, documentation, risk management, cost, return on investment),

- Department institutional constraints.

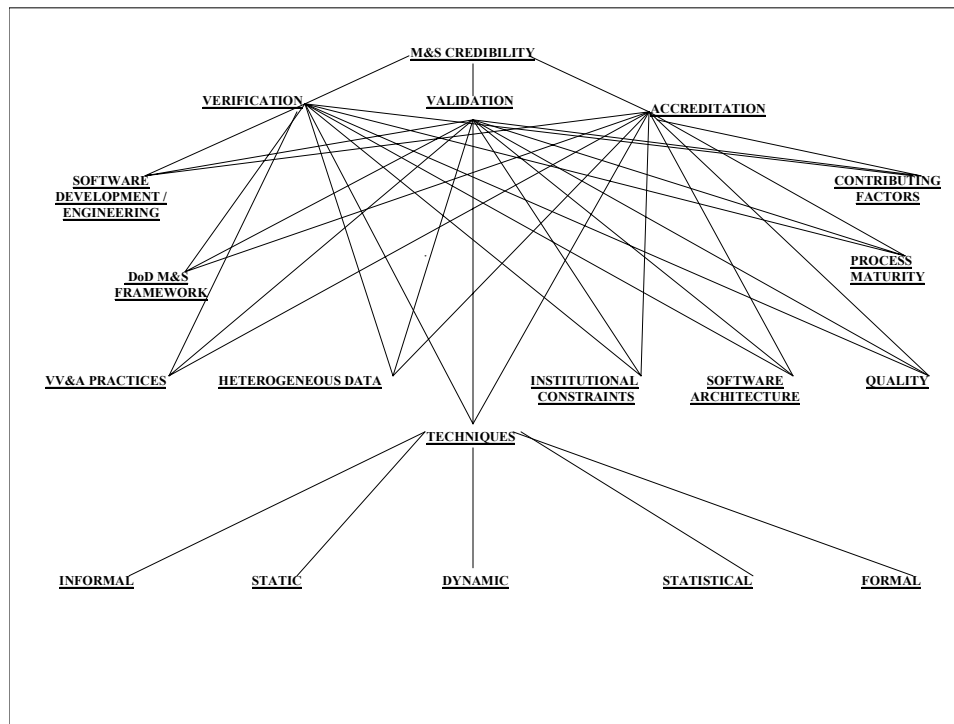


Figure 3-1. Synthesized M&S Credibility Taxonomy

E. VERIFICATION, VALIDATION & ACCREDITATION PROCESSES

Early studies by [Arm64, Mec64, and NF67] identified the lack of verification methods in the M&S process and proposed the concepts of high face validity, conceptual validity, and operational validity. Later, [Mih72, Sha75] addressed the concepts of V&V in conjunction with systems analysis, system synthesis and model analysis. [Sar91, Sar92, Sar98] considers M&S V&V critical and addressed model validation methods and techniques including conceptual model validity, model verification, operational validity, data validity and development of a “model range of accuracy” from techniques employed in M&S output analysis.

[MBZ99] provided a concise chronology of M&S credibility from 1962 through the 1990's and identified the growing emphasis and research on V&V, and the growing awareness of the importance of the underlying software development foundation. In 1988, the

Director, OTE acknowledged the emerging efforts to establish Department M&S verification, validation and credibility assessments, noting that it was “essential that the M&S employed and the results derived from them be credible” [Kri88].

There is a significant body of related literature on improving M&S credibility and improving confidence in simulation results. A large body of simulation-related topics including risk reduction, V&V principles and techniques, M&S quality characteristics, process improvement, software development / engineering procedures, domain implementation uses, and approved architectural standards for achieving higher M&S confidence levels, includes research by [BL91, Sad91, Dow92, Gia92, Hen92, Met92, Bec94, Lew94, Par94, Cau95, Gan95, BCN96, CCK+96, Por96, FY97, ML97, Mue97a, Mue97b, Pre97, Bar98, Che98, Gan98, Gau98, HM98, JDD98, LR98, MT98, NM98, NMY+98, Pac98c, Roa98, ROG98, BSW99, Dah99, DMS99, FGW+99, GB99, Gre99, Mar99, Pac99a, San99, Whi99, BF00, BSL+00, COH00, EKH00, HNC+00, MLU+00, San00, BS01, Co01a, CY01, DDH01, FED01, FH01, Hal01b, GKB01, HNK+01, LS01, Luq01, MJL01, Mor01, Pac01b, Pac01c, Pug01, Ros01, YPH01, BML02, Bre02a, CW02, Dan02, FRC+02, Kil02, Pac02b, Pac02c You02a, You02b].

More recently, [Coy00b and BG01] noted several significant systemic M&S challenges within the DOD Test and Evaluation community affecting confidence in the results. [Coy00b] specifically cautioned that the use of M&S to extrapolate beyond the known bounds almost never works, however, interpolating between two demonstrated test results is an acceptable practice in evolutionary acquisition programs. [Tho97a] also raised the concern that the V&V process lacked adequate definition as recently as 1983.

Current V&V theories, methods, techniques, and procedures cover a wide range of activities cited by [Kri89, RSH90, Rit92, WS92, WF93, ARS+96, JAS97a, JAS97b, Pac98d, RPG00, Roa98, SM00, Rak01, Wri01a, WH02] and techniques proposed by [Gia92, Cau95, FH01, Fri01, GMS+96, KFC98, MM98, RPG00]. Verification and validation for large complex M&S such as mission-level models or physics based models have additional impediments [KM01a] including the dynamic nature of M&S evolution, the cost and constraints of the data, the difficulty of validating the models, and the limited shelf life of previous validation efforts.

The Department published [GMS+96], and established the twelve guiding principles for the VV&A of M&S and federations. [Bal98] also contributed fifteen guiding principles to provide better insights into VV&A activities, supporting the Department's model [GMS+96] or the [Bal98] M&S life cycle models. A follow-on Internet variant of the [GMS+96], the *VV&A Recommended Practice Guide* [RPG00], promotes the current best V&A practices and procedures for legacy M&S (Figure 3-2).

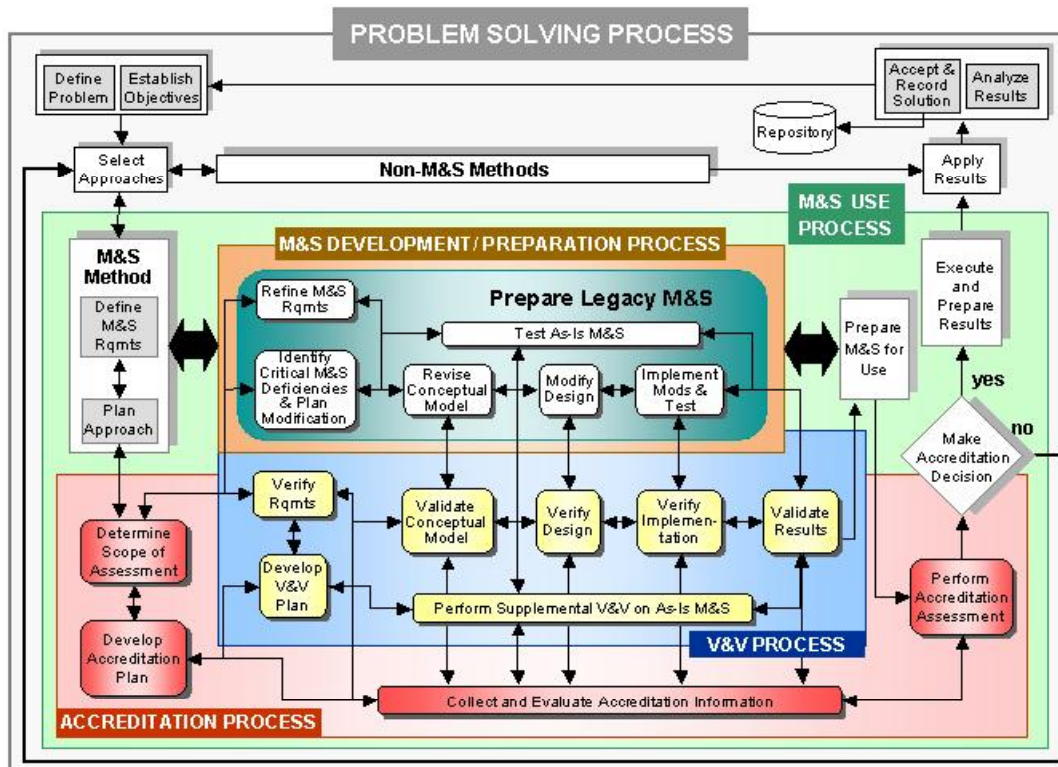


Figure 3-2. VV&A and Legacy M&S (From [RPG00])

Other approaches evolved. [Rak01] identified three basic processes for software validation based on testing, measurement, and software reliability growth. Current VV&A procedures built on these theories and lessons-learned from previous VV&A approaches, including the Aggregate Level Simulation Protocol (ALSP) VV&A process [TP96] and the Distributed Interactive Simulation (DIS) VV&A process [GW96, MBZ99]. [Bal98] advanced an analytic hierarchy process, which supported the measurement and evaluation of both quantitative and qualitative factors, expert knowledge, independent evaluation and

comprehensive assessment to improve user confidence in M&S results. In addition, [Bal98] introduced a model-testing concept called model verification, validation, and test (VV&T) for identifying and ascertaining whether specific inaccuracies or errors exist in the model. [Bal98] also recommended continuous V&V planning, execution, and documentation throughout the entire M&S life cycle.

A Modeling and Simulation Information Analysis Center report on VV&A automated support tools [DMS00b] and a subsequent article by [GFM01] proposed leveraging existing software development industry tools, basically expanding the development and application of V&V automated test support tools analogous to those currently employed in the test and evaluation community. [Lew98] proposed a V&V Managers Toolkit based on a VV&A management model with a metric suite for cost, performance, schedule, and technical performance indicators. Currently, automated tool support for the VV&A process is extremely limited [Jor01]. In addition, [HNP97] cautions that automated tools to assist the complex and difficult validation process may be undesirable from an engineering viewpoint.

Whether using qualitative or quantitative methods, a combination of the two methods to complement each other, with or without automated tools, the V&V process requires disciplined processes and procedures to provide satisfactory levels of confidence in the simulation results. However, at the present time [Rak01] notes, “basic software V&V activities are not well understood and are applied inconsistently” [Rak01].

1. The Department’s Verification Process

The software V&V process entails a broad scope of activities, which support software quality characteristics [DB91, Dav92c, AV98, Hal01a] discussed in Chapter VI. The verification process determines the M&S capability and establishes the foundation for confidence in the model, establishing quality early in the development process by ensuring the developer accurately captured the validated conceptual model in the implementation design. “Verification,” as defined by the Department is “the process of determining that a model implementation accurately represents the developer’s conceptual description and specifications” [DoD94, DoD96, DoD01a] when compared with the first abstraction of the

real world, the conceptual models of the mission space (CMMS). In addition, “verification ensures that a M&S meets all the requirements specified by the user and that it implements those requirements correctly in software” [GMS+96].

There are many other definitions for verification. Verification is also concerned with solving the equations right [Roa98]. The IEEE similarly defines verification as “confirmation by examination and provisions of objective evidence that specified requirements have been fulfilled” [IEE98b]. Verification also supports the implementation of the functional interface, and output requirements and may be defined as “Did I build the thing correctly” [GMS+96]?

The conceptual model in the M&S process is analogous to the keystone in a building archway. A missing, incorrect, incomplete or sub-standard keystone threatens the structural integrity of the archway. Today, [Pac99b] notes conceptual model documentation “for many legacy M&S is limited or non-existent” [Pac99b]. [Ham96, HNP97] further note that confidence in the verification process depends upon the underlying model’s validity. The absence of a valid conceptual model threatens the integrity and credibility of the simulation and confidence in the simulation results.

2. The Department’s Validation Process

Model and simulation validation is performed at two distinct levels: 1) conceptual model validation, and 2) result validation, and satisfies the criteria of how well the functions match the real world and addresses “Did I build the right thing” [GMS+96]? The conceptual model validation is an essential, integral component of the Department’s VV&A process and is the foundation for establishing credibility, defining fidelity requirements and developing accurate model representations. Results validation compares the M&S results with known or projected behavior, tests, and sensitivity analysis or subject matter expert opinion to determine if the results are “sufficiently accurate for the range of intended uses of the M&S” [GMS+96].

The validation process complements and supplements the verification process, and supports development of M&S credibility [IEE86, Sar91, Cyn92, DH92a, Hen92, Sar92, Ham96, Bal97, HNP97, IEE97c, Bal98, IEE98b, Sar98, Pol99, Co01a]. Model and simu-

lation validation “is the process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model” [DoD94]. [Roa98] views the validation process as solving the right equation. The Department’s definition of validation [DoD94] includes the validation of conceptual model, results, and data in the V&V process supporting multiple Department domain requirements including acquisition programs [CS97, COH00, Coy00b]. The IEEE defines validation as the “confirmation by examination and provisions of objective evidence that particular requirements have been fulfilled” [IEE98b].

There are many validation strategies, with several noted in this research. [HNP97] identified two possible strategies to validate a model: the axiomatic approach (e.g., theory-driven) where the existence of a set of assumptions describe the fundamental truths of the problem domain, and the model proven as a theorem; and the empirical approach (e.g., data-driven) in which the model performance is compared to expected values or historical data. Generalized validation (e.g., evaluation) according to a VV&A taxonomy developed by [Dav92b] would include empirical and theoretical evaluation methods in addition to evaluation by other methods. Output accuracy methods outlined by [Kil02] includes benchmarking, face validation, results validation and sensitivity analysis. According to [Kil02] there are three primary ways to determine output accuracy: another M&S or benchmarking, subject matter expert experience employing face validation, results validation with test data, with all methods sometimes supplemented with sensitivity analysis. In order to achieve the highest degree of M&S credibility, [LK00] opine that the “most definitive test of a M&S model’s validity is to establish that it’s output data closely resemble the output data that would be expected from the actual (proposed) system” [LK00].

The degree of confidence in M&S depends heavily on M&S anchoring in physical testing [Wel00]. Although tests and evaluations may develop sufficient confidence levels that the model is valid for its intended purpose, [Bal98] cautions that complete M&S model testing is not possible and that successfully testing each sub-model (module) does not imply overall model credibility. Furthermore, [LK00] acknowledges the timing and relationship of validation is critical for establishing and improving confidence levels that “correct” results are available for credible decisions. Five key quality factors identified by [MLW01]

support M&S credibility: capability, software accuracy, data accuracy, results accuracy and usability.

[BCE+99] addresses another difficult testing challenge of striking a balance between live testing, sound analytical methods and ground testing. M&S is extremely challenging for advanced or unprecedented systems, where “substantial design deficiencies or unanticipated system characteristics requiring correction continue to be revealed during flight tests despite the sometimes substantially and costly applications of M&S...even those subjected to comprehensive verification and validation” [BCE+99].

Even with these concerns, there are still differences of opinion within the M&S community today in the requirement for V&V, the scope of the effort, the high cost, how it is implemented, specific techniques, and determining “how much V&V is enough?” In addition, [Bal98] suggests that some tests are more appropriate to evaluate the behavioral accuracy or validity of the model, while other tests better determine the accuracy and verification of model transformation from one form into another. Consequently, the Department’s M&S domain “lacks a coherent process that links V&V information to application-specific requirements for M&S credibility”[GMS+96].

3. The Department’s Accreditation Process

The final action of the Department’s VV&A process is the decision-makers accreditation⁸³ [SS92, CSC94a, CSC94b, JAS96, San97a, Sta97, Pac01a] of the simulation. M&S accreditation is “the official certification that a model, simulation or federation of simulations is acceptable for use for a specific purpose”, implemented by the Department [DoD95, GMS+96] to provide credibility and confidence in M&S developed for another purpose or agency. This process provides the decision-maker with an understanding of the capabilities and limitations (e.g., caveats) of the simulation. Accreditation in the Department’s M&S domain requires confidence in the V&V process, documentation, data, use history, and known limitations of legacy simulations, especially when developed for other purposes, or maintained by different organizations.

⁸³ Accreditation in terms of information system security has another meaning: The formal declaration by a Designated Approval Authority (DAA) that an information system is approved to operate in a particular security mode using a prescribed set of safeguards at an acceptable level of risk (DITSCAP) [Mon00b]

The challenge of achieving credibility in the Department's M&S was succinctly noted by [San97a] who wrote that it "is not confidence in M&S we seek, but confidence in the decisions we are making" [San97a]. [SS92] propose that all models are wrong in some aspects, and the validation process identifies how and when the model is erroneous, while the accreditation process is the "determination that the decision to be made is not sensitive to these errors and limitations"[SS92]. As emphasis on formal VV&A grew within the Department, two kinds of accreditation were initially proposed according to [Pac01c]: 1) class or domain accreditation, as a form of blessing of the simulation for a kind (e.g., class) of application; and 2) specific application accreditation, as a formal certification that the simulation and its associated inputs is appropriate for a particular use. [Pac01c] cautions against the temptation to use class accreditation, noting that the potential exists today within the Department's M&S domain to "use... almost right, but not quite right software", contributed to the uninsured five billion dollar loss of the Ariane 5 booster in 1996 [LLF+96].

Department component accreditation responsibilities include the accreditation of forces and capabilities employed in joint, general, and common-use M&S, and ensuring the appropriate representation of its forces and capabilities in joint and common use M&S development [DoD94]. Simulation credibility defined by [Kil02] includes the ability of the most important elements to meet the required capability and intended use. Required capabilities explained by [Kil02] include descriptions of the simulation's purpose, represented model entities and fidelity, physical environment description, element relationships, and assumptions and limitations. Addressing accreditation of federations, [Pac01a] identifies the compatibility of individual federates in a federation as a key element in validation assessment to determine if the federation can appropriately support the intended application.

4. The Role of Data in the VV&A Process

Data is critical to the Department goal of improving credibility and confidence in M&S results, and plays a key role in developing simulation credibility and underpinning user confidence in the results. The [RPG00] cites four basic types of data including: 1) reference or metadata, 2) hard-wired data such as constants and set parameters, 3) instance data for input and output functions, and 4) validation data comprised of actual measure-

ments or real-world facts used for comparison with M&S results for validation. The three basic metadata types: administrative, descriptive, and quality, identify databases [Gor96], or individual data elements [RPG00], whereas, metadata quality supports the quality assessment techniques, results, and addresses attributes for accuracy, precision, completeness, portability, consistency⁸⁴, currency and flexibility. An additional fifth type of data includes exchange metadata for federations. Multiple authors including [Atw91, NMY+98, HSB98, Goz00, PFL+00, Del00, PKL01, SH01, Tol01, LWK+02] address data verification, validation and certification issues.

From another viewpoint, [HNP97] suggests three categories of data: discrete data, which may take on one of a finite set of values; continuous data, which may assume any value in an interval of real numbers; and categorical data, which may take on a finite, normally small set of values. Whatever data ontology is chosen, authoritative data⁸⁵ is required for developing credible representations and other critical activities in Department M&S. However the [DoD01a] cites authoritative data as “the single-most pervasive deficiency area identified with M&S use” [DoD01a].

Data is also critical to the Department’s VV&A process, yet widely misunderstood and sub-optimally executed in the Department. [Sar98] cited historical data validation methods such as rationalism and empiricism, as well as a predictive validity approach to improve the process. Acquiring authoritative, certified data supporting sufficient M&S validation processes is also difficult, and exacerbated by the existence of scientific and technical data characterized as complex data⁸⁶. Complex data categories include highly derived data, objects, compositions and artifacts of legacy systems⁸⁷ [DoD95].

A major Department study on the existing data verification, validation and certification (VV&C) methodology, [NMY+98], concluded that data quality results from the integration of the data producer’s V&V process, which supports the user’s data V&V efforts. Both activities are components of the M&S V&V process, conducted during the entire

⁸⁴ Consistency refers to data that is maintained free from variation or contradiction [DoD98].

⁸⁵ Authoritative data comes from authoritative data sources whose products have undergone producer verification, validation, and certification activities [DoD98].

⁸⁶ Complex data cannot be characterized as a single concept or data element [DoD95]

⁸⁷ Highly derived data includes categories such a probability of hit/kill. Objects employ multiple inheritance, multiple root hierarchies and polymorphic attributes. Composition includes images, binary large objects (blobs), and compound documents. Artifacts are normally attributed to data from legacy systems designed to reduce storage requirements by the use of embedded codes or logic [DoD95].

M&S life cycle, with data quality metadata bridging the two activities. However, [NMY+98] identified the lack of discrete, formal VV&C standards supported by data quality metrics, including metadata quality metrics as a systemic Departmental data deficiency. Four major deficiencies noted by the [NMY+98] study included:

- Data V&V methods lacked consistency,
- No linkage between the development of data products and user V&V requirements,
- No central repository for data V&V information,
- Data VV&C not synchronized with the M&S VV&A effort [NMY+98].

The concept that data V&V should be an integral part of the M&S life-cycle process [IEE96, IEE97b] as opposed to a separate distinct procedure has been emphasized by recent research [Sar98, NMY+98], and reflected in Figure 3-3. This process is a variant of the earlier recommended procedure in the [DoD95] guidelines defining a separate process for verification, validation and certification (VV&C) of data by the producer and the user. The separate VV&A and VV&C processes raised the probability that validation of the of data for the intended use was more implied than the result of an explicit, defined, repeatable process.

Several methodologies exist to improve data quality. Data quality [DoD98] is the correctness, timeliness, accuracy, completeness, relevance, and accessibility characteristics, which make data appropriate for use.⁸⁸ Validated data for rapidly evolving M&S is a major challenge, but is essential for M&S tools if they are to achieve their full potential. Data accuracy cited by [Kil02] addresses the appropriateness of the data, including resolution; the quality of the data established by the data producer and user; and the correctness of data transformations. [Cau95] proposes a reduced order metamodel validation methodology to compare M&S results with real-world data, if available. Suggested improvements in the data validation process supported by [HH00a, HS00b] address data interchange formats (DIF), while [HSB98] recommend a CMMS data dictionary for improved interoperability. [DoD01a] adds the requirement for new V&V methodologies and improved technologies, including software quality metrics detailed by [Kan03] to enhance the software quality V&V practices within the Department.

⁸⁸ Quality statements are needed for source, accuracy (positional and attribute), currency, logical consistency, completeness (feature and attribute), security classification, and releasability [DoD95]

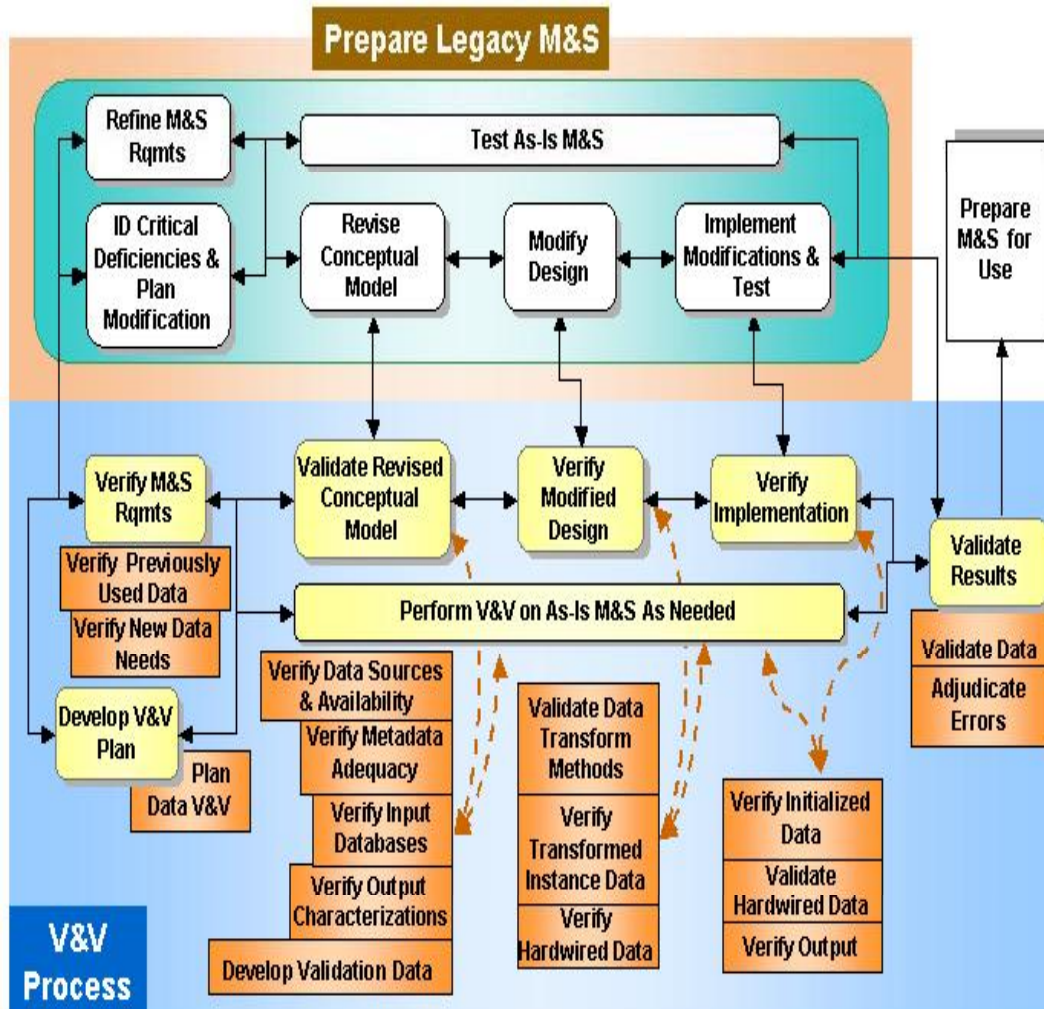


Figure 3-3. Data VV&A & Legacy M&S (From [RPG00])

[Dav92b] reinforced the need for improved life-cycle VV&A practices with recommendations to develop positive incentives including a vision for VV&A, and the development of consistent policy and procedures across the Department. [Dav92b] further recommended early successful implementations, tailoring of VV&A to the corporate culture, establishing independent reviews, providing long and short-range planning processes, and understanding project/program concerns with VV&A. [Jor01] notes that data V&V should be performed as an integral part of the V&V process, stating that “algorithms only work if the required data is available and is accurate for the aggregation portrayed” [Jor01].

Establishing credibility and consistency according to [HBM96] requires the publication of the approach, data, and tests be for peer review to improve the current situation of too many redundant databases, too little testing, too few reviews of data sources, and insufficient analysis of embedded assumptions. Recent works by [MSO01] on missing data techniques and [SC02] on sparse data method have addressed some of the issues of sparse or missing historical data. However, at the present time, the Department's objective for an over-arching coherent program for affordable, timely, verified, validated, and authoritative data on demand is still more a goal than a reality [Cra01b].

5. Verification and Validation (V&V) Techniques

The Department's V&V process for M&S is the approved foundation for achieving quality and developing M&S credibility. A current Department M&S objective includes maintaining and enhancing the V&V program for models, simulations and data to provide users with confidence in the results and knowledge of the limitations. The Department has over seventy-five software engineering and model specific techniques for performing V&V exist today, with methods ranging from informal to formal methodologies [GMS+96, Pol98, RPG00]. Figure 3-4 identifies the informal, static, dynamic and formal V&V techniques.

The mathematical and logical formalism increases from the informal techniques on the left to the formal methods on the right, as does the complexity of the techniques. Informal V&V techniques noted in Figure 3-4 lack rigorous mathematical formalism. These tools and methods employing human reasoning and subjective techniques may be very effective if applied using well-structured approaches, formal guidelines and mature, disciplined processes [GMS+96]. Informal V&V techniques are among the most common, frequently used approaches in the Department [RPG00].

Static V&V techniques cited in Figure 3-4 provide insights into the models structure, modeling techniques, data, control flow and syntax. Different V&V approaches addressed by [Pac02b] depend on whether the M&S source code is available. V&V agents employ these techniques to review the accuracy of the model design and source code.

These techniques are widely used and do not require model execution runs. Quality auto-

mated support tool requirements identified by [AWQ+93, ACC+94, MBH99, GFM01] are also necessary if the Department is to realize the goals of improved results from VV&A activities. Few current automated tools are available to assist in this process, except the language compiler or interpreter [GMS+96, RPG00].

<u>VERIFICATION AND VALIDATION (V&V) TECHNIQUES</u>		
<u>INFORMAL</u>	<u>DYNAMIC</u>	<u>STATISTICAL</u>
AUDIT	ACCEPTANCE TESTING	VARIANCE ANALYSIS
DESK CHECK	ALPHA TESTING	CONFIDENCE LEVELS
FACE VALIDATION	ASSERTION CHECKING	FACTOR ANALYSIS
INSPECTIONS	BETA TESTING	HOTELLING'S T ² TESTS
REVIEWS	BOTTOM-UP TESTING	MULTIVARIATE
TURING TEST	COMPARISON TESTING	ANALYSIS
WALKTHROUGHS	COMPLIANCE TESTING	NONPARAMETRIC
	DEBUGGING	GOODNESS-OF-FIT
	EXECUTION TESTING	NONPARAMETRIC TESTS
	FAULT INSERTION TEST	OF MEANS
	FIELD TEST	REGRESSION ANALYSIS
	BLACK-BOX TESTING	THEIL'S INEQUALITY
	INTERFACE TESTING	COEFFICIENT
	OBJECT-FLOW TESTING	TIME SERIES ANALYSIS
	INTERFACE TESTING	t-TEST
	PARTITION TESTING	
<u>STATIC</u>	PREDICTIVE VALIDATION	<u>FORMAL</u>
CAUSE-EFFECT	PRODUCT TESTING	INDUCTION
GRAPHING	REGRESSION TESTING	INFERENCE
CONTROL ANALYSIS	SENSITIVITY ANALYSIS	LOGICAL
DATA ANALYSIS	SPECIAL INPUT TESTING	DEDUCTION
FAULT/FAILURE	STATISTICAL METHODS	INDUCTIVE
ANALYSIS	STRUCTURAL TESTING	ASSERTIONS
INTERFACE ANALYSIS	SUBMODEL TESTS	LAMBDA
INTERFACE ANALYSIS	SYMBOLIC DEBUGGING	CALCULUS
SEMANTIC ANALYSIS	TOP-DOWN TESTING	PREDICATE
STRUCTURAL ANALYSIS	VISUALIZATION /	CALCULUS
SYMBOLIC EVALUATION	ANIMATION	PREDICATE
SYNTAX EVALUATION	GRAPHICAL	TRANSFORMATION
SYNTAX ANALYSIS	COMPARISONS	PROOF OF
TRACABILITY		CORRECTNESS
ASSESSMENT		

Figure 3-4. V&V Techniques (After [GMS+96, RPG00])

Dynamic V&V Techniques provided in Figure 3-4 evaluate execution behavior of the model during run-time. One technique inserts probes or stubs into the executable code to collect information during the running of the M&S. This method normally employs three steps: inserting the probe or stub instrumentation, executing and collecting the required data, and analyzing the output file for discrepancies or faults. Dynamic methods and statistical techniques have been addressed in a significant body of Department V&V guidelines [GMS+96, RPG00]. [RSH90] submit a propagative approach to sensitivity analysis in large-scale M&S where the sheer numbers of parameters make it impractical to

test more than a small fraction of relevant cases, a common challenge in all large software-intensive systems.

Figure 3-4 lists a number of various statistical V&V techniques available to validate models. The best environments for these techniques include completely observable models and provide for the collection of all data for validation. In this category, [NF67] advanced the technique of Analysis of Variance, while, [MM98] suggest peer-reviewed statistical techniques for effectively managing model fidelity. In addition [MM98] contend that Department models often comprise a set of sub-model requiring additional integration into a coherent representation of a mission space.

The literature also reveals an abundance of additional V&V techniques and procedures. [Ham96, HNP97, Bal98, LK00] discussed simultaneous confidence levels, confidence levels and model range of accuracy to address operational validity. [Ham96, HNP97, LK00] also explained several statistical procedures (inspection, time-series approaches) for comparing real-world data to M&S output. [ROG98] proposed a multistage validation framework consisting of conceptual and operational validation employing two statistical methods, a two-sample t test and a two-dimensioned, two-sample Kolmogorov-Smirnov test [HNP97], concluding that a model valid for one level of detail may be invalid for another use. [Ham96, HNP97] suggests the validation process is ambiguous, complex, and difficult, subject to undesirable subjective approaches, in part because the first part of the validation process builds upon a credible validation of the conceptual model, itself an abstraction.

In addition, [Fri01] explains the application of Fisher's Combined Probability Test and Goodness-of-Fit Tests to support formal statistical comparisons, even when "the data are sparsely distributed across a highly dimensional factor space" [Fri01]. [Hal01a] suggests that the breakdown of the model into testable functional elements supports validation data requirements and makes the point that data "obtained from testing in this manner is at a low enough level in the model to support statistical comparisons between model predictions and test results" [Hal01a]. Classical statistical techniques [LK00] that work well with ideally distributed, independent observations, may prove difficult to apply correctly with all

M&S situations since both real-world systems and M&S output produce non-stationary distributions and auto-correlated observations [RPG00].

Disciplined verification, validation, and accreditation techniques for M&S and data, support interoperability for new large-scale, complex simulations, and federations. A survey of current M&S model VV&T techniques [Bal98] apply throughout the system life cycle. Several additional methods and techniques to improve validity prescribed by [HNP97, LK00] include: collect high-quality data, maintain an assumption documents, conduct structured walk-through, validate model using quantitative techniques, validate the output, and use animation to improve user understanding of the system. [BML02] further suggests a theoretical framework for determining the confidence in a simulation based on the credibility of three factors: the model and its embedded data, runtime data, and the operation or application of the model.

[Sar98] identified operational validity techniques for M&S test programs, which include event validity, comparison to other models, degenerate tests, extreme condition tests, fixed values, face validity, parameter variability-sensitivity analysis, traces and Turing Tests. [Sar98] also proposes hypothesis tests, which support a comparison of means, variances, distributions, and time series of the output variables to determine if the model's output behavior has an acceptable range of accuracy. However, the current state of M&S software development maturity largely lacks objective V&V processes with quantitative methodologies supporting the sufficiency of a model for a specific purpose.

The highest possible confidence-levels achievable in the V&V process include the formal V&V techniques noted in Figure 3-4. Formal methods are also the most complex, least understood, and least implemented V&V techniques in the Department [GMS+96]. The scope of formal method techniques, processes, and procedures include [LG93, LGB94, FMG94, GMS+96, LG97, Ber98a, Ber98b, LB00, RPG00, MLJ01, MJL01, POP00, BC02], including the changes of formal methods [Rob98] employ mathematical proofs for correctness, and require significant effort. Formal methods applied early in the development process, normally achieve the best results. However, formal methods applied to the most critical components, may prove effective at any time during the development process [Mic03].

[Nog00] developed a formal method to automatically assess risk and the duration of a software development project supported by evolutionary software processes. For large-scale software engineering projects, [BL91] proposed a formal methods mathematical approach as a basis for precise communication and the foundation to achieve a high degree of automation directly supporting increased confidence in the software quality. NASA also realized the increasing complexity and criticality of software applications required an emphasis in formal methods. Major factors identified by [NAS97, CVL+97, CKM+98] for increased use of formal techniques included the following:

- Fault-protection and safety functions required software features beyond the hardware level to detect and isolate failures, and initiate recovery procedures,
- Software-failure characteristics were different than those found in hardware,
- Complex systems placed ever-growing demands on verification,
- Disciplined software engineering organizations had developed fine-tuned verification processes, but defects still remained in the product,
- Few existing techniques were rigorous enough for improving the quality of products during the early life-cycle phases of requirements generation and design [CKM+98].

The overlapping tools, techniques, procedures and terms of reference between the M&S domain and emerging software engineering/development domains continually manifest themselves in the literature. [MLU+00] propose an incremental qualification approach to M&S VV&A, noting that M&S “is subject to software development...as well as system engineering practices. [MLU+00]” [ARS+96] also identified V&V as a system engineering discipline, which must be included in the domain engineering process to improve the level of confidence in critical software, specifically safety and mission-critical software.

The Joint Accreditation Support Activity (JASA) developed and continuously refined the Susceptibility Model Assessment and Range Test (SMART) methodology [JAS97a, JAS97b]. The SMART methodology involves a four-phase approach to VV&A for acquisition M&S including: incremental reviews, analysis, evaluation, testing, and documentation of M&S and associated data to support the verification and validation process detailed by [CSC94a, CSC94b]. With over ten years of VV&A related-experience in a wide range of M&S programs, the JASA identified eight major obstacles to VV&A success

similar to the systemic issues adversely impacting Department M&S credibility identified earlier by [PAD78, PAD79a, PAD79b, PAD80]:

- Lack of cooperation from the M&S developer based on concerns that identification of errors could adversely impact future business, intellectual property [IEE99b] issues, and failure to specify V&V in contract tasks,
- Failures to follow disciplined processes, such as configuration management, and develop quality documentation,
- Failure to define intended use and M&S requirements, including functionality and fidelity,
- Failure to define accreditation information requirements,
- Lack of or mismatch of skills,
- Insufficient priority applied to VV&A efforts,
- Lack of interpersonal skills demonstrated by VV&A team members,
- Lack of perseverance [JAS97a, JAS97b].

The private sector contractors supporting Department M&S initiatives experienced the same systemic issues establishing M&S credibility and maintaining users confidence in M&S results. A 1993 Department audit report [RSV+93] of a sample of defense contractor M&S programs identified costly duplication, redundancy and unnecessary proliferation of existing M&S capabilities with the following quality indicators:

- Only twenty percent of the defense contractor M&S had completed a formal VV&A program,
- Five percent of defense contractor M&S had adequate configuration management controls in place,
- Thirty five percent of the defense contractor M&S had adequate documentation,
- The findings indicated “a substantial lack of effective management control over the development and use of M&S by the government contractor community” [RSV+93].

6. Shortfalls of the Department’s VV&A Process

There are also many V&V shortfalls, limitations, and critiques of the Department’s V&V process cited in the literature including [Che87, Dav92b, LA92, Met92, Pay96, TP96, CVL+97, KP97, HM98, ML97, Mue97a, NAS97, JDD98, Lew98, NM98, ROG98, Pac99b, MLU+00, TRW00, Cra01b, HNK+01, Jor01, MT98, HS02, Kil02, Pac02b]. There are also limits to the confidence established by validation with well known validation limits [Kil02] including the limited scope of validation tests, costly validation data which is

difficult to obtain, and limited tools [Lew94, Bar98, DMS00b]. In addition, conventional methods and techniques may not be adequate to validate some highly aggregated M&S (e.g. mission and campaign-level models) [Kil02].

High-fidelity models, HLA federations [HLA98], and legacy M&S issues [LBS01, ZB02] pose additional, significant challenges for the proper execution of V&V. Effective and efficient validation of all potential logic paths in the M&S is probably technically and economically infeasible, since the number of possible system states in a large simulation is very large. The number of states grows exponentially with regard to the number of degrees of freedom in the model simulation or software [Mic03]. [Pac02b] suggests that the utility of M&S is limited if the M&S validity and fidelity cannot be adequately quantified. This includes testing and test support V&V issues identified by [Kri88, HN97, Che98, Kec99, Obr99, Whi00, LLC+02].

A finding of the 1999 *Defense Science Board on Test and Evaluation* [HAC+99] concluded that “Verification, Validation and Accreditation as presently practiced with respect to M&S techniques in general is not sufficiently disciplined to inspire confidence in their use in the T&E process” [HAC+99]. [HAC+99] further proposed better coordinated and more disciplined M&S software development processes supported by improved VV&A to address the following findings on M&S use in the test and evaluation:

- M&S is effective and cost effective in many clearly defined applications where both the expertise and data are available to support a common mission,
- Large-scale constructive force-on-force combat M&S tend to be the least useful when the inputs are largely unknown or uncertain,
- The current Department M&S investment emphasizes model architectures, interfaces, graphical displays, and programming in lieu of conceptual model development and basic data collection,
- No single, authoritative catalog or list of available M&S and quality attributes exists,
- Open air testing is not feasible for some weapon systems, creating a dependency of some degree on M&S for evaluations of weapon systems,
- M&S has potential in the test design and planning process, and the reliability, availability and maintainability (RAM) areas,
- Cost and benefits are difficult to measure, up-front funding is often lacking, and there is a concern that the Department cannot maintain the talent necessary to improve M&S capability [HAC+99].

For all the potential benefits associated with M&S, the actual achievements fall short of predicted performance objectives, simulations lack widespread credibility to instill user confidence, and the community still requires an acceptable method for determining the return on investment for M&S. [PHP+96] identified challenging, systemic, and recurring issues of credibility in the tools and serious doubts about the confidence of M&S results with the “underlying issues of the credibility of the tools being a major constraint” [PHP+96]. Only forty-five percent (20 of 44 systems) of the model summaries reviewed in a 1998 Department medical M&S catalog [Gau98] revealed any V&V activities. The inability to adequately quantify M&S fidelity and validity limits overall M&S utility.

In a commentary of the current state of M&S [Whi99] suggests that VV&A has been an after-thought, a topic of discussion, but not properly implemented. These limitations acknowledges [Pac01b] make it difficult to establish confidence in M&S results, identify the risks associated with decisions based upon M&S results, or determine the appropriate resource levels to increase confidence levels in the M&S. More importantly, these examples are just instances of the many issues identified by [DOT01a] where the inadequate implementation of VV&A policies may result in the use of non-credible M&S to support critical Department program acquisition decisions.

7. Secondary Credibility Contributing Factors

In addition to the primary VV&A procedures, several other mitigating factors surfaced often in the literature and appeared to have an important secondary affect on decisions to initiate or conduct a VV&A program. We call these secondary credibility-contributing factors, and these factors include reuse, documentation or lack of documentation, risk management, cost of V&V, and the affect of V&V on a simulations return on investment (ROI). Employed consistently, the secondary credibility-contributing factors support credibility, may reduce overall life cycle cost, and identify a return on investment. Unfortunately, the Department’s complex organizational dynamics, and complicated acquisition procedures [Wol02a, Wol02b] minimized the potential contributions of these factors. This list of credibility-contributing factors is not all-inclusive, however, we selected them for their potential influence on the development of software-intensive simulation systems.

a. Reuse

Software reuse [IEE99a, IEE99b] is the “practice of using existing software components⁸⁹ to develop new applications” [DCC+93, KDM+94, IEE99a, IEE99b, Jaz02]. Software reuse has the potential to reduce development costs, reduce time to market, improve simulation quality and support the V&V process, according to [Som95] and [Pres97] who identify and explain the salient considerations supporting successful software reuse. In 1991, the Department established the Software Reuse Initiative (SRI) to collaboratively and cooperatively advance the reuse strategy with other government agencies, such as NASA, industry, and academia to make software reuse effective for the Department⁹⁰ [RTC94]. Software reuse faced many early challenges, which included organizational acceptance, management support, legal issues, intellectual property rights, liability concerns and changes to existing acquisition policies [DCC+93, IEE99b].

Reuse evolved according to [Moo01] and the current focus is now on “reusing knowledge itself, not just life-cycle artifacts such as specifications, code, and test data” [Moo01]. Departmental M&S policies also support reuse noting, “When possible, existing information systems shall be used or modified rather than creating new systems” [DoD94]. [Pac02a] suggests that as the ability to automatically generate code from specifications improves, the focus of reuse efforts should be in the earlier activities of the software life cycle, which precede coding, and smaller components may be easier to reuse. [TJ00, Moo01] reinforce this concept and propose domain engineering and analysis as the key to reusing knowledge, capturing common elements across the domain, to further additional domain development.

In the M&S domain, explicit, well-documented conceptual models would be prime candidates for reuse, based on both definitions of the term reuse. Unfortunately, as [DoD01a] notes, past investments in M&S have not resulted in the Department’s capability to identify algorithms, data sets, models or simulations for reuse. Reuse of conceptual

⁸⁹ Reusable software components can be executable programs, code segments, documentation, requirements, design and architectures, test data and test plans, or software tools. They may also be knowledge and information needed to develop, use, or maintain the component. [DCC+93]

⁹⁰ The DoD effort included the ARPA Software Technology for Adaptable, Reliable Systems (STARS) effort, the Army Software Reuse Initiative, the Navy Software Reuse Initiative, the Air Force Central Archive for Reusable Defense Software (CARDS), and the Defense Information Systems (DISA) Software Reuse Program (SRP). [RTC94]

models in the Department is inconsistent since, “Many Defense community simulations”, according to [Pac02a], “lack explicit descriptions of the conceptual models which are implemented in them” [Pac02a]. This is a central theme and key driver of this dissertation.

Recent reuse techniques explored knowledge-based tutoring systems applied to software reuse [NLR99], computer-aided tools employing the relational hydrograph model and computer-aided software evolution system [Har99a], a software reuse reference model [RN99], the extension of existing reuse schemes with systemic software reuse methodologies and reuse success factors (e.g., planning, management support, process control, architecture, and domain focus) [RDK+99], and the employment of adapters to encapsulate, and connect multi-use component interactions [RNJ99]. Other more abstract approaches identified architecture reuse possibilities based on component-based reuse of a domain-specific software architecture [HBL97], architectural mismatches [GAO95], and the architectural style approach [MG96] for classifying, storing, retrieving reusable architectural design components.

[HBL97] cites software component reuse [SLM91, LBG+96] and rapidly changing requirements [RL95, LBL+97] as the two major problems confronting large, complex software development, and suggests software evolution formalization as the best approach to understand the development and evolution process [BL94, DLB94, LG97]. A recent survey by [MET02] of large and small European companies employing reuse, confirmed these problems and found that approximately thirty-three percent of reuse projects in the survey population failed. Although the sampled companies belonged in different business domains, [MET02] found successful reuse projects shared good application commonality, high process maturity levels, and use of standard procedural or object-oriented development methodologies. Successful reuse projects also shared the following attributes:

- High commonality among applications,
- Reuse is basically a technology transfer endeavor and requires management commitment,
- The approach to reuse may be standard, but the deployment approach produced the best results when tailored to the context of the organization [MET02].
-

[RDB+00] suggests the current open source software movement, including Linux, is a logical extension of both the traditional reuse methodologies and other previous

open source standards including UNIX, Sendmail, Simple Mail Transfer Protocol (SMTP) for email, and the gcc compiler. However, since reuse is not part of software engineering curriculums, [MET02] “forecast there will be a long delay before sensible advances are made” [MET02]. Progress has been noted in the reuse of communication protocols. When problems occur, the cause may be the mixing and matching of protocols, and using them for an unintended purpose (e.g., at the MAC layer rather than the session layer) [Mic03].

b. Documentation

Software documentation is an important software artifact. Complexity of large software development projects drives the need for documentation to provide communications between project members, customers, and management. [STS93a, STS93b, AS94] address documentation principles, methods, products, tools and management techniques supporting improved quality and productivity.

Providing users with knowledge of M&S limitations and developing confidence in simulation results is a major Departmental M&S objective. [Sar98] supports model V&V documentation and a detailed basis-of-confidence document. [LK00] proposes maintaining an assumption document to support the conceptual model development. This method also has potential to support a model basis of confidence document.

However, documentation is usually a low priority for the software developer. In addition, [Kil02] confirms that M&S capability⁹¹ requirements, seldom understood by the user and rarely documented with useful detail beyond top-level users manuals continue to exist as major simulation development shortfalls. As a result, documentation for Departmental M&S systems is often incomplete, missing, outdated, or lacking detail [DoD01a].

⁹¹ Simulation capability descriptions should at a minimum include the purpose, modeled elements, environment, relationship, assumptions, and limitations [Kil02a].

c. Risk Management

There are many widely used definitions for the concept of risk, including the IEEE's definition [IEE01]⁹². In similar views, [BCC+00] defines risk as the probability of loss, while [HHG+02] views risk as the probability that a future event may or may not occur, and the consequences. In this research we employ the risk management structure and definitions from the Risk Management Process Model [And01]. Risk is:

A measure of the potential inability to achieve overall program objectives within defined program cost, schedule, and technical constraints and has two components: (1) the probability/likelihood of failing to achieve a particular outcome, and (2) the consequence/impacts of failing to achieve that outcome [And01].

The Risk Management Process Model [And01] includes the following events, processes or procedures, varying with the phases of the system acquisition and system definition, and integrated into the program management function: risk events, risk management, risk planning, risk assessment, risk handling, risk monitoring, and risk documentation.

[HHG+02] also cites heuristic approaches to risk management including the probability of the future event occurring, the consequence of occurrence, and identifies guides, checklists and standard operating procedures to reduce or mitigate risk. [Boe91, Pfl98, Gal99, BCC+00, RPG00, And01, HDK+02, HHG+02] identified software risk and risk management⁹³ approaches in software-intensive information systems including: guidelines, checklists, and taxonomies. These methods also include the probabilistic techniques for *ex post facto* analysis or software reliability and formal statistical approaches detailed by [RPG00].

In decision-making there are two basic types of risk: rejecting correct evidence or accepting incorrect evidence. In M&S software development the risks of accepting incorrect evidence are characterized in the [RPG00] as either development or operational risks. The Department's VV&A process, designed to reduce development risk by reducing errors and defects early in the development process, theoretically reduces the

⁹² Risk is the likelihood of an event, hazard, threat, or situation occurring and its undesirable consequences; a potential problem [IEE01].

⁹³ Software risk management is a software engineering practice with processes, methods, and tools for managing risk in a project. It provides a disciplined environment for pro-active decision-making to assess continuously what can go wrong, determine important risks, and implement actions to deal with the risks [BCC+00].

number of operational risks of incorrect M&S outputs. However, [HHG+02] considers software integration risk as the largest unknown “unknown” risk based on the current software development paradigm of integrating varied and unstable software components such as database management systems, search engines and web browsers, which are constantly evolving to remain competitive in the market place.

The risk of failure normally occurs in three ways and with different combinations of cost, performance, and schedule. These occurrences further decompose into the following risk areas of technical, supportability, programmatic, cost, and schedule risks. [BCC+00] identify several common risk factors including: management, ignoring risks, discipline, training, fallacy of easy solutions, work plans, schedule, delivery, constraints, customer, methods, and tool selection. Common risk-reduction methods, tools and processes exist today to including project management tools such as PERT/CPM/Gantt charts and software estimation methods incorporated in software estimation tools [GAB97].

Multiple organizations in the private, government, and academia sectors continue to develop advanced risk management techniques, procedures, and automated tools, including the International Council on Systems Engineering (INCOSE) Risk Management Working Group, the Project Management Institute (PMI) Risk Management Specific Interest Group, the Carnegie-Mellon Software Engineering Institute (SEI), Software Program Managers Network (SPM), the National Aeronautic and Space Agency (NASA), the Naval Postgraduate School (NPS), and the Defense Systems Management College (DSMC). The INCOSE Risk Management Working Group and the PMI Risk Management Specific Interest Group [HHG+02] collaborated on a joint project for a Universal Risk Project and developed a list and definition of universal risk areas⁹⁴, applicable to any type of project or operation in the private and government sector⁹⁵. A universal risk is defined as “an event or condition that causes a deviation from the plan, with a reasonable chance of affecting the conduct of an analysis and that may occur in any project, operation or system,

⁹⁴ The three major risk groups identified by consensus included: management risk area, external risk area, technology risk, [HHG+02]

⁹⁵ This report used the INCOSE risk management process terms – planning, assessment, handling and monitoring. The PMI terms for the same steps are – risk management planning, risk identification, qualitative and quantitative risk analysis, risk response planning and risk monitoring and control [HHG+02].

regardless of industry, organization or project/system type” [HHG+02]. In addition, [HHG+02] identified and defined specific risks under each major group.

The SEI risk management process approach [GAB97] consists of several functions performed continuously over the project lifecycle: identification, analysis, planning, tracking, control, and communication. [SPM95] explains an additional four areas of risk classified by the Software Program Managers Network: productivity, quality, timeliness, and user satisfaction. Recently, [HDK+02] presented potential improvements to Department risk management of the operation and implementation of information systems supporting vital defense functions by incorporating an approach called value focused thinking. The NASA independent verification and validation (IV&V) methodology [Con01] addresses a risk management approach, which is experiencing renewed emphasis following the less than successful results of recent NASA projects based on risks introduced into the projects by the implementation of the “better, faster, cheaper” management process [OCD+02].

Improving reliability of systems in the early software development stages when systems are especially prone to errors may be key to improving quality and the risk assessment process in software engineering. Recent research at the NPS by [LN00, NJL00, NLB00a, NLB00b, NLB+00, Nog00] focused on improving quality and reducing risk in the earliest phases of the software development project. In addition, [NLB00a, NLB00b] identified the inherent weakness of human dependencies and inconsistent application of the risk assessment process by different experts, and introduced a formal method with three metrics for requirements, personnel, and complexity supporting risk identification.

New automated risk management tools continue to evolve. CASE tools such as the Computer Aided Prototyping System (CAPS) [LK88, Luq89, NLB00a, and NLB00b] support a risk assessment-based software evolution and prototyping approach. A risk analysis template [Whi99] consists of elements deemed to be important to the model. Applied to each element of acceptability for the model, it addresses the consequence of missing, poor or incorrect elements. To manage the risks introduced by these factors, [BK02a] introduced the credibility indicator trees method including fault tree analysis to support defined levels of V&V and credibility. Risk mitigation techniques directly affect

the entire simulation software life cycle, contribute to improved M&S credibility and support the accreditation process.

The objective of the Department's VV&A process is to reduce M&S risk [Kil02]. The verification process reduces the risk of unintended errors and the use of inappropriate data, while the validation process improves confidence that the M&S outputs match the real world, and that the level of data accuracy is sufficient for the intended uses. Accreditation reduces the risk of selecting an inappropriate or unsuitable M&S for use. However, well-known V&V process constraints include the lack of time, budget limitations, an incomplete knowledge of reality, and complex M&S systems. Moreover, [HHG+02] contends that techniques are not the critical issue, but rather the lack of management and engineering discipline ensures we continue to practice poor paradigms for system development.

Mission-critical software systems may present high-risks to the overall system development. [CHK+90] identifies Mission Critical Computer Resources (MCCR) as the totality of computer hardware and software that is integral to a weapon system along with the associated personnel, documentation, supplies, and services. The DSMC developed the MCCR process for four reasons:

- Software for weapon systems is on the critical path of system development,
- Software can experience significant development problems resulting in cost overruns,
- Modern weapon system performance depends on the quality of the computer resources and the system is only as good as the software,
- Once a software development project falls behind, adding more resources (e.g., money, programmers) will not shorten the development times [CHK+90].

d. Cost of Verification and Validation

There are concerns in the Department about the cost and lack of implementation funding for conducting credible VV&A programs. Published VV&A costs range from five to eighteen percent of a projects funds cited by [GMS+96] with a mode between ten and twelve percent of the project cost. The cost of model validation noted by [Sar98] is usually non-trivial, particularly when confidence in the simulation results is important. Furthermore, developing absolute validity of a model over the entire domain for all poten-

tial uses contends [Sar98] is too costly, time consuming and probably impossible. Instead, [Sar98] suggests developing sufficient confidence by executing tests and evaluations as a means for determining a model valid for its intended application.

The availability of documentation from previous VV&A initiatives and unambiguous acceptance criteria also determines the cost of establishing M&S credibility [Mue97b]. Addressing the ever-increasing size of M&S, [MBZ99] conclude that the “increasing of scale and degree of complexity of M&S system[s] add [to] the difficulty and cost of VV&A process in life cycle of simulation” [MBZ99]. However, in a striking contrast to [MBZ99] cost concept, [Whi99] contends that there is no additional cost for VV&A “since V&A activities are integral to normally accepted software development processes” [Whi99].

e. Return on Investment

Major investments in Department M&S have not credibly established a return on investment (ROI) cautions [BGK+00], and the challenge to implement credible and cost-effective V&V is growing. In many advanced technologies, providing confidence in performance to the specified level across the operating envelope depends to an unprecedented degree on confidence in system M&S. For these reasons, [BGK+00] developed a business case framework to evaluate Departmental M&S investment opportunities.

[Sta96, BGK+00] cites the failure to identify a return on investment (ROI) for M&S on the lack of PM staff knowledge, and the lack of Department institutional incentives to build expensive models. In addition [BGK+00] compare and contrast case studies of the commercial sector Boeing 777 aircraft and the Joint Strike Fighter M&S development within the Department M&S environment. Understanding the pressures that program managers are under to complete projects on schedule and within budget, [BGK+00] address several systemic impediments that preclude program managers from expending the funds, staff, or time to investigate the potential benefits of M&S. In order to improve this situation [BGK+00] develop a seven-step procedure to build a M&S business case, and reiterate a recurring theme found in the body of research; recommending a disciplined approach and methodology.

F. SUMMARY

Chapter III introduced M&S credibility issues and identified concerns that decision-makers have with Department simulation results, followed by an overview of the Departments VV&A processes, and several factors contributing to the implementation of, or shortfalls of the VV&A processes in the Department. Two major periods, the Growth and the Managed Eras characterize the initiatives, factors, and efforts to improve credibility in the Department's simulation results.

This chapter also addressed the credibility of M&S in the context of the procedures, policies and programs the Department established to improve decision-makers' level of confidence in the results produced by the M&S. It is clear from the research that Department and the Service Components acted with due diligence to implement the policy, procedures, and guidelines to improve the quality of Department M&S. By the mid-1990s the Department established M&S management organizations, developed policies, implemented plans, allocated significant funding, and executed major new programs with the goals and objectives of improving Department M&S practices. However, notwithstanding these considerable efforts, major concerns still exist about the credibility of the Department simulation process. There are several major reasons for this lack of credibility in Department M&S identified in the chapter.

The new Department M&S management organizations, such as the DMSO, had to overcome Service parochialism, the unpredictability of the Department budget process, the vagaries of the Department acquisition process, and the life cycle management of over forty years of unplanned Department M&S growth. Then in the late 1990s significant funding and manpower resources originally programmed to improve Department M&S were committed to the Year 2000 problem resolution project. Research indicates three major categories of technical, cultural, and managerial challenges collectively hindered the development of improved Department M&S credibility.

The research also identified an expanding role for M&S at the national policy and strategic level, and within the Department five major domains. A synthesized M&S Credibility Taxonomy established at Figure 3-1 identified interrelated factors affecting the De-

partment's objectives for improved credibility, especially in the area of authoritative representations. There are many other factors involved in simulation credibility; however, these factors directly support the development of software-intensive simulations.

The Department established the VV&A process in the mid-1990s as the primary method for establishing M&S credibility, prescribed regulatory requirements and provided best practices and techniques to the community. The Department continually cited a lack of credibility in M&S stemming from inadequate VV&A implementation negating decision-makers' confidence levels in the simulation results as a systemic issue needing major improvements, especially in the Department's developmental and operational test communities. Companion assessments of VV&A process implementation in the private sector and other public sectors indicated that the VV&A process failed to deliver the desired results.

We identified and discussed VV&A principles illustrated in Figure 3-2, including the critical role that data plays in the VV&A process. However, the lack of authoritative data is a major deficiency in the Department's attempt to improve M&S credibility. Activities supporting credence in the M&S included model verification, conceptual model validation, operational validity, data validity (Figure 3-3) and a model range of accuracy. Five key factors supported model credibility: capability, software accuracy, data accuracy, results accuracy, and usability. The most definitive factor supporting credibility is to compare the model output with the actual or proposed system output data. We also identified and summarized common verification and validation techniques in Figure 3-4 employed by the Department, including M&S V&V techniques (e.g., informal, static, dynamic, statistical, formal) identified in the [GMS+96 and RPG00]. We noted that systemic issues identified in by reports, studies, and assessments the early 1980s are still evident.

The Department's foundation for M&S credibility supporting confidence in simulation results is the M&S verification and validation process. Verified and validated conceptual model support credible V & V processes. However, many consider V & V too expensive and time consuming. In this research we reviewed the V&V process in the context of supporting Department- and National-level decisions with improved credibility and confidence, and confirmed previous assessments indicating the quality of Department V&V practices are *ad hoc* and inconsistently applied.

The conceptual model of the mission space (CMMS or FDMS) is a critical component of a rigorous Department verification and validation program. The M&S community developed many conceptual model formats [RPG00], however, there is no consensus on a standard conceptual model at this time. In this research we identified the underlying issues and propose a model supported by the software engineering process and a better understanding of the complexity of the real-world fidelity definition and measurement. This research also expanded on the work accomplished by the SISO Fidelity Experimentation Implementation Study Group and their work on the Fidelity Conceptual Framework. [RGH00].

During the research, several common themes continually emerged from the literature: reuse, the lack of documentation, risk management, cost, and the identification of a quantifiable return on investment (ROI) from M&S. Reuse initiatives provided mostly disappointing results; however, “large-scale” reuse initiatives have the potential to improve this process. Documentation is a consistent problem in many software development projects. In the M&S area in addition to the lack of software documentation, the lack of conceptual model documentation, coupled with the lack of documented V&V activities severely undermined the Department’s efforts to improve credibility. Risk management is a continuous activity applied during the entire life cycle of the M&S, where the early identification and resolution of risks may be critical to improving overall M&S quality and credibility, appears uneven.

The cost to accomplish V&V, constantly cited as being too high, may continue to grow as the M&S become larger and more complex. In addition, identifying a quantifiable ROI has been an elusive goal for the Department, and as a result, Department acquisition program managers may be reluctant to invest the required resources in a tool, which requires significant resources today for questionable results tomorrow.

The Department’s legacy M&S portfolio is a major study factor. The Department developed a significant and expensive portfolio of legacy M&S over the past forty years. Although an operational system has a *de facto* architecture, many of these M&S evolved in an *ad hoc* manner and have become large legacy systems with multiple stakeholders and expensive support infrastructures. A significant number of these systems lack conceptual

models and have an *ad hoc* V&V history. In some cases the developer created conceptual models after the fact, if at all, and adherence to disciplined M&S V&V process was a low priority. Department agencies may also have a tendency to use these systems for studies or tests for which they are ill suited, in order to show a return on investment. As a result the acceptability criteria and caveats developed by the accreditation process for these simulations may be of limited value or counter-productive to the original purpose and intent of the study or test objectives, and fail to establish credibility in the simulations or confidence in the simulation results

IV. HETEROGENEOUS SYSTEM REPRESENTATION ANOMALIES

A. INTRODUCTION

Improving the credibility of large-scale, legacy Department M&S requires an understanding of the underlying systemic causes generating the pre-conditions for heterogeneous system representation anomalies, especially in federation interoperability. Chapter IV reviews several heterogeneous challenges to Department M&S credibility. These diverse challenges include syntactic and semantic heterogeneity, data heterogeneity, complexity, interoperability, technical and substantive interoperability, fidelity heterogeneity, and multiresolution modeling heterogeneity.

B. ELUSIVE AUTHORITATIVE REPRESENTATIONS

Authoritative representations have been consistently identified as major objectives in Department M&S, yet few of the [DoD95, DoD01a] objectives for authoritative representations have been achieved [Mos00, Cra01a, Cra01b]. As a result of the long-standing unresolved systemic issues associated with Department M&S senior civilian and military decision-makers have significant reservations that Department M&S, as currently practiced, can meet the serious future demands generated by national security requirements for M&S in the Department. In 1987, [Che87] placed boundaries on simulation results, later confirmed by [San97a] noting:

Although simulations are useful tools, they are always approximations to reality, and, therefore, the credibility-the level of confidence that a decision-maker should have in their results--is open to question [Che87].

Current Department M&S users still require accurate, interoperable elements of the mission-space and the draft [DoD01a] retains authoritative representations as a Department-wide M&S objective. In addition [DoD01a] lists expanded requirements for credible, authoritative representations including U.S., allied, friendly, coalition, neutral and threat and paramilitary system representations for C4ISR, combat, combat support and combat service support systems and processes, supported by standard taxonomies and common object classes for systems representations by FY2004. Successful attainment of these interoperability objectives by the Department and international M&S communities assumes the

acceptable resolution of many heterogeneity⁹⁶ issues, including abstractions, networks, federations and representations.

Abstraction is an intrinsic human activity employed to reduce or filter the full complexity of the real world to manageable proportions by deleting unnecessary detail in order to provide data in context (e.g., information) supporting the decision-making process [HJS97]. Abstraction mechanisms [Til98] selectively emphasize the level of detail, emphasizing the details pertinent to the intended use, while hiding unneeded detail. [Sow88, Til98] discussed common abstraction mechanisms including: classification, aggregation, and generalization:

- Classification is a form of abstraction establishing an *instance-of* relationship between the object type in the schema and its instance with an object defined as a set of instances with shared common characteristics,
- Aggregation is a form of abstraction establishing a *part-of* relationship between the component objects and the aggregate object, with relationships between objects considered at a higher level aggregate object and specific details of the constituent objects hidden,
- Generalization is a form of abstraction establishing an *is-a* relationship between the specialized and generic object, when similar objects relate to a higher-level generic object and the constituent objects considered specializations of the generic object [Sow88, Til98].

C. DATA MODEL HETEROGENITY AND INTEROPERABILITY

Data models establish the essential properties and well-defined relationships between raw data in a system in a form, which supports efficient storage, timely retrieval of data, and provides the basis for tools and techniques to support data modeling [Rag01]. A data model captures the static⁹⁷ and dynamic⁹⁸ characteristics supporting data-related processes, with static properties defined in a database schema and the dynamic characteristics developed into specifications for reports, queries, and transactions. The schema maintains the object type definitions, attributes, relationships and static constraints of the data repository or database, an instance of the schema.

⁹⁶ Heterogeneous is defined as: consisting of or involving dissimilar elements or parts. Heterogeneous networks are a collection of simulations with partially consistent behaviors and/or partially correlated databases. Examples include simulators of different fidelity, mixed virtual and live simulations, and mixes of virtual and constructive simulations [DoD98].

⁹⁷ Static characteristics include objects, attributes, and relationships among objects [Til98].

⁹⁸ Dynamic characteristics include operations on objects, operation properties, and relationships among operations [Til98].

The most significant record-based logical data models includes the hierarchical data model, organized into simple tree structures; the network data model, a superset of the hierarchical data model, without the requirement for tree structures; and the relational data model, based on a mathematical foundation of a relation (e.g., a set of n-tuples) organized as a related collection of tables [Oll83]. These classic highly machine-oriented, and record-oriented data models, or *syntactic data models* [Til98], are well suited to the computer environment and organized for optimal efficiency supporting storage and retrieval operations, however these models lack semantic relativism⁹⁹ and abstraction mechanisms¹⁰⁰ required for dealing with complexity in large-scale systems [Til98].

[Til98] describes a second category, *semantic data models*, which combine basic knowledge representation techniques with database technology, and represent a transition from the basic record-oriented approach towards a model supporting more human-oriented semantic constructs. While traditional data models store data, and semantic models shift towards a more human knowledge model, [Til98] suggests a third method, the *conceptual modeling*¹⁰¹ activity geared more towards the domain-level knowledge and program understanding, with a focus on the end-user. Program understanding supports development of a fidelity-based product line conceptual model referent domain-architecture, since large legacy systems represent substantial corporate knowledge of an organization's business rules, requirements, and design decisions. [Til98] further suggests that all three techniques may be more suitable than one method if there are different categories of artifacts (e.g., data, knowledge, information), requiring a relational model physical storage of data, a conceptual model for representing domain-level knowledge, or a semantic model for interactive discovery.

⁹⁹ Semantic relativism is the ability to view the elements and concepts representing a modeled system from different perspectives based on the application [Til98].

¹⁰⁰ The most common abstraction mechanisms are classification, a form of abstraction in which an object is defined as a set of instances; aggregation; and generalization, a form of abstraction in which similar objects are related to a higher level generic object [Sow88]

¹⁰¹ Conceptual modeling (e.g., conceptual schemata) in this context is the activity of formally describing aspects of some information space for the purpose of understanding and communicating, with an emphasis on the knowledge organization used by humans, rather than the data organization used by machines [Til98].

D. HETEROGENEOUS SYNTACTIC DATA MODELS

[Wie99] submits the objective of interoperation is to increase the value of information when accessing, relating, and combining information from multiple sources. In addition, [Wie90, Wie93, Wie96a] acknowledged that data comes from many diverse and heterogeneous sources and further suggested that joining heterogeneous data is necessary to generate information or compose large-scale software applications [MBS+99].

[Wie90, Wie93, Wie96a] also addressed several contributing factors for differences among systems in their research on database schema integration including the different perspectives of the developers and users; the use of equivalent constructs (e.g., different uses of the same constructs to create the same representation); and incompatible design specifications. Focusing primarily on database constructs and schema research, [WW90, Wie93, Wie96a, Wie00a, Wie00b] developed eight viewpoints for differences among systems, and [You02c] added a ninth viewpoint of heterogeneity germane to this research:

- Heterogeneity of Hardware and Software systems,
- Heterogeneity of Organizational Models including network, hierarchical, relational, universal database models and object-structured data,
- Heterogeneity in Representation since it causes anomalies at the limits or boundaries (e.g. 5-digit zip code versus 9-digit zip code),
- Heterogeneity of Scope with different databases capturing different views of the same real-world object (e.g., employees paid versus employees available),
- Heterogeneity of Level of Abstraction manifested by different aggregation schemes (e.g., personal income versus family income), implying the existence of an aggregation hierarchy,
- Heterogeneity of Meaning addressing assignment of attribute values to real-world attributes (e.g., postal codes versus town names),
- Heterogeneity of Temporal Validity caused when values have, often implicitly, different temporal ranges (e.g., monthly budget versus weekly production), which may not support aggregation [Wie93],
- Semantic Heterogeneity occurs because the meaning of words depends on the context, and data sources develop within their own context. Types of Semantic Heterogeneity include synonyms, spelling differences, and the use of identically spelled words to refer to different objects [Wie00a, Wie00b],
- Heterogeneity of Structure causing a variation in the structure of the information arrangement between systems employing the same organizational model (e.g., a plane modeled as an attribute in one system and an object in another [You02c]).

In supporting a decision-maker, [Wie02] suggests access to a variety of methods and heterogeneous information sources for predicting the future including extrapolation, interpolation, projections, published projections, spreadsheets, discrete simulations, and continuously executing simulations (e.g., weather predictions). Many methods are computation intensive, and except for desktop models such as spreadsheets, most simulations are too complex, costly, or require too much manpower or time to meet decision-makers requirements for information supporting analysis of alternate future courses of action. In addition, distributed simulations employing HLA, DIS, or ALSP protocols rarely interface with other external types of information systems such as databases, although simulation systems may include internal databases or files to retain past, and near-current data for table lookups and scenario support to the simulation. Relational databases support the Structured Query Language (SQL) for database access; however, SQL currently has no native capability to communicate with a simulation or a federation.

[Wie00a, Wie00b] notes that semantic heterogeneity causes failure to find the desired object, and affects the level of precision in selection, aggregation, and comparison when integrating information. Furthermore, [Wie00a, Wie00b] believes a global solution to semantic mismatch is unfeasible due to the number of participants, terminology changes, and scope, but notes that precise, finely differentiated terms and abbreviations may be suitable for domain efficiency. For instance, [Wie00a, Wie00b] opines that consistency is local (e.g., precision for an on-line commerce requires a consistent structure and a consistent terminology for the same set of objects).

[WW90, Wie93, and MW02] suggest that knowledge of the enterprise operation supports the development of the best possible information from diverse sources, and also note that since heterogeneity is never completely resolved, uncertainty heightens in the decision-making process. In the same vein, [Wie00a] suggests that many small-scale efforts provide superior results over major standardization efforts, and proposes that precision and relevance attributes will be more valuable than completeness and recall attributes. Precision, with its many components (e.g., rule-based matching, metadata, quality attributes, and processing models) is an important aspect of information, and will become increasingly important [Wie00a].

[Wie99] also notes that interface standards are critical for building and maintaining multi-layer systems, citing the role of SQL for structured tables, CORBA and DCOM for middleware; and XML when data cannot be structured well. [Wie99 and DM03] further suggest using XML for specific domains with domain-specific type descriptions or schemas, to help in matching the meaning of shared information.

As a possible solution to the heterogeneity challenge, [WW90, Wie97a, Wie99, Wie00a, Wie00b] proposes a two-dimensioned mediator architecture partitioning resources and services to resolve heterogeneity issues with three horizontal layers (e.g., client applications, intermediate service modules, and base servers), and vertically into any domains, with a limited number of supporting servers. Noting that any implementation for a system presenting past and future information requires the cooperation of specialists in distinct fields and an openness to learn new methods and techniques, [Wie96a, Wie97a, Wie02] identified a three layered hierarchy, with an intelligent middleware mediator layer linking the heterogeneous information sources at the bottom with the application programs at the top to achieve an intelligent integration of information. The mediator performs the following tasks:

- Accessing and retrieving relevant data from multiple heterogeneous information sources,
- Abstracting and transforming retrieved data into a common representation and semantics,
- Integrating the homogenized data according to matching keys,
- Reducing the integrated data by abstraction to increase the information density in the transmitted result [Wie97a].

[Wie97a, WJ98, Wie99, Wie02] demonstrated a capability for supporting integrated predictive capabilities from spreadsheets and other simulation tools into information systems, employing an interface language, SimQL, combined with wrappers. SimQL provides a schema, describing the accessible contents, and a query language to access the information resources, similar to SQL capabilities [Wie99] and has four major components:

- A compiler for the SimQL language, generating code to access wrapped resources,
- A repository containing the schemas for the wrapped resources with input and output parameters identified,
- A wrapper generation tool to bring simulations [WJ98], spreadsheets, and web resources into compliance,

- The source forecasting tools, spreadsheets, discrete simulations, and web sources [Wie99].

Other methods for composing components from heterogeneous, autonomous, and distributed information sources include CPAM [MBS+99], an image indexing and retrieval algorithm, WBIIS, providing a search capability for a large image database [WWF+98], and the Object-Oriented Method for Interoperability (OOMI) [You02c].

The [You02c] solution is a federated interoperability object model (FIOM) and an automatic translator generator. [YBG+01, You02c] developed the FIOM, an interoperability object model defined for a specific federation of systems, and an interoperability object model for interoperability (OOMI) with a corresponding structure. [YBG+01, You02c] also envisions the OOMI as an extension of the contemporary object model with a class structure property and extended attribute and operation properties to capture the different representations. The [YBG+01, You02c] translator function is a software wrapper, component of a message-based architecture, or as part of the data store, with an XML-based message translation currently under study for implementing the proposed model. Where [You02c] provides a necessary near-term, interim solution for interoperability via the Object-Oriented Method for Interoperability (OOMI) translators, this research addresses the root cause issues that effect simulation substantive interoperability in high-risk, mission critical simulation systems, and addresses the issue at the first level of abstraction, the architecture.

Research continues reveal new methods to fuse information from heterogeneous resources including databases, text, semi-structured information bases [WW90, Wie97a] involving intrinsic semantic differences [Wie03], supported by complementing ontology¹⁰² research [Gua98, MW02]. Commercial software vendors also continue to expand basic database systems, adding communication, data mining, and analysis functionality to the basic database capabilities supporting the decision-makers need to plan and schedule future actions, and assess the effects of alternate future courses of action [Wie99]. Data repository holdings of past and near-current data provide only limited background information

¹⁰² The semantics of an information source is captured by its ontology, the collection of domain terms and their relationships used in the discourse [MW02].

for assessing alternate future courses of action, while decision-makers require a spectrum of heterogeneous information including past, present, and likely future situations [Wie99, DoD01a, Wie02, Wie03]. Simulations, coupled with other heterogeneous information resources, may provide a potential new information source to support a decision-makers analysis of alternate future courses of action.

E. HETEROGENEOUS CONCEPTUAL MODELS

The Department addressed many M&S problems in the 1990's to improve overall quality, including technical and substantive interoperability. Although there has been progress with the technical interoperability problem since the mid-1990s, many approaches failed to improve the credibility of authoritative representations and reduce the federation's interoperability problems caused by substantive interoperability anomalies. This results from simulation developers abstracting real-world objects in their model representations from different perspectives, requirements, constraints, and objectives.

Over time developers modeled many of the same objects in different ways, potentially causing problems when they interact in a federation. [DSB+99, HY01a, HY01b, YPH01] identified systemic substantive interoperability issues with legacy simulation software systems and the inconsistent representation of the same entity in multiple systems. There is a very close correlation on the root causes between the technical and substantive interoperability issues cited in the simulation domain, and the previously cited work of [Wie93, Kin01, You02c] on developing information from heterogeneous data sources.

This research revealed significant progress in the area of technical interoperability, including research into multilevel simulation of discrete network models [Ham96], distributed simulations [Ham96, HNP97], and the Department-sponsored development of the HLA. These advances supported the ability of federates to physically connect and exchange data employing a federation object model, the use of common standards, and coordinated data structures. The HLA infrastructure supports hardware and software compatibility, standards compatibility, time management, security, and coordinated use of the RTI.

However, federation interoperation currently requires significant manpower and coordination, and often encounters technical and substantive interoperability issues during

operation. Substantive interoperability has been a systemic issue since the Department's earliest experiences with networking simulations, SIMNET, and persisted through the evolution of the ALSP, DIS and HLA protocols. Today, substantive interoperability, including entity interoperability issues remains a major challenge for achieving truly distributed simulation interoperability [DoD03a]. Entity interoperability includes a logical interaction between entities modeled in different simulations, temporal resolution, spatial resolution, and a coherent relationship between the components of the physical environment (e.g., ground, ocean, atmosphere, weather, and the electro-magnetic spectrum) to support the interoperability context [YPH01]. Entity interoperability also encompasses the level of representation¹⁰³, attributes and behavior including:

- The required federation entities are available at the necessary level of detail,
- The federation entities fit together,
- The required federation entities meet the critical characteristics of the real-world entities,
- The required federation entities possess the salient attributes supporting the federation's intended use,
- The required federation entities accurately model the needed behavior,
- The required federation entities work together logically,
- The required federation entities employ consistent algorithms to compute effects [YPH01, HY01a].

The RTI time management services support federation synchronization of individual federate time and attempts to manage federation temporal resolution assuming:

- Each federate computes change in the entity states at some resolution of simulation time,
- Incremental state changes occur often enough to support logical interaction between entities, and allow the exchange of data to occur [YPH01, HY01a].

The spatial resolution shared among the federates must also support the intended use, although the federates are not required to compute the or store the entity spatial data in the same reference frame assuming:

- Each federate computes the location of entities on a specific geospatial reference system,

¹⁰³ Level of detail – Entities do not require the same resolution, as long as the interactions between entities meet the intended use of the federation [YPH01].

- The computed and shared entity locations allow different federates to adequately and consistently meet required federation geospatial and other environment objectives [YPH01a].

The environment is a critical factor for achieving substantive interoperability, with each federate developing a sufficiently consistent correlated view of the environment, including both visual (e.g., tanks, planes, buildings, water) and non-visual spectrums (e.g., radio frequency (RF), infrared radiation (IR), photons (lasers)). Early efforts enforced homogeneity by developing a mapping between the different simulation environmental representations, while today there are integrated data set solutions including a standard syntax and semantics with the SEDRIS data reference model providing a standard interface specification, data dictionary, and data coding standards [YPH01]. Research continues in representing the effects of the environment on systems and human behavior (e.g., the detection capability of an IR or RF sensor in a perturbed environment), and conversely on the impact of systems and humans on the environment (e.g., the impact of chemical or radiation weapons in a city).

A few analogous examples of the current challenges presented by the lack of technical and substantive interoperability provided below illustrate, compare, and contrast the concepts of technical and substantive interoperability:

- You dial a number for an important business call, but accidentally reach a computer modem and hear the sound of a modem trying to synchronize with a sending modem. The communication network performed as designed. The communication protocols worked up to a point, however, you received no message. This represents a passing grade for technical interoperability, but a lack of substantive interoperability. The result: time lost redialing, maybe lost business since the message did not get through on time.
- In another example, arriving home after work and just as you prepare to relax for a late dinner, the phone rings and you are greeted by a telemarketer or worse yet, no response, indicating an autodial technique advancing ahead of the telemarketer's verbal proposal. In this case you hung up the receiver even though the communication network and protocols worked exactly as designed. This case illustrates successful technical interoperability and a lack of substantive interoperability, however, since the telemarketer's message failed to go through. The result: no message delivered and no sale for the telemarketer.
- In an HLA federation scenario the publisher submits a number 0.506127 for a state variable in a specific instance, and the federation subscription process apparently works. Initial analysis suggests the results are normal. Subsequently following a

major live test event such as a flight test, post-test simulations runs execute, but errors in process truncate the value to three digits, e.g., 0.506 from the 0.506127 number needed for the required precision. Multiple runs of the simulation execute, but the results appear inconsistent. Several factors in the host processing systems including translation, level of precision, or even perturbations due to chaos theory¹⁰⁴ may be the root cause. The result: substantive interoperability anomalies in an HLA federation, inconsistent results, lack of credibility in the simulation and loss of confidence by the user.

Substantive interoperability anomalies occur when one of the following types of representational anomalies manifests themselves: 1) event phase anomalies, 2) event ordering anomalies, 3) state error anomalies or registration anomalies¹⁰⁵ [HY01a, YPH01] and create intolerable representational anomalies. Representational anomalies beyond acceptable tolerances for the federation's intended purpose can manifest themselves from an invalid federate and/or from interactions between valid federates.

Tests may identify the source conditions of possible interoperability-related anomalies: including functional compositions and manifold representations [HY01a]. Functional compositions occur when the computation of one or more object states in one federate depend upon the data provided by another federate and violates one or more of the following interoperability criteria adversely affecting M&S quality:

- Dependency representation,
- Representational accuracy,
- Range consistency,
- Sensitivity consistency,
- Temporal representation,
- Internal sensitivity,
- Error consistency,
- Stochastic consistency [HY01a].

Manifold representations, normally exhibited under very specific conditions, occur when two or more federates: 1) represent the same behavior or state, and 2) interact either directly or indirectly. Both parallel manifold representations and sequential manifold rep-

¹⁰⁴ This phenomenon is common to chaos theory, and is also known as a sensitive dependence on initial conditions, first identified by Edward Lorenz, a meteorologist, in 1960.

¹⁰⁵ Event phase anomalies exhibit a timing or phase error. Event ordering anomalies occur when the simulated object produces the same events as the simuland under identical conditions, but in a different order. State error anomalies are identified by a difference between the state a simulated object assumes and the state that object's referent exhibits under identical conditions and the difference is beyond tolerance levels [HY01a].

representations may cause interoperability problems, including aggregation or disaggregation issues if they fail to meet manifold representation interoperability criteria including: state correspondence, abstraction transform, or state continuity. The simulation conceptual model¹⁰⁶ provides the necessary supporting information to support these tests [Pac01a], while the Federation Development and Execution Processes (FEDEP) [GYL03] include conceptual model development procedures and guidelines for building a federation, supporting credible verification and validation [DSB+99].

The level to which heterogeneous system representation anomalies affect the credibility of a federation is derived from the representational acceptability criteria, based on the federation's purpose and the capabilities the federation or federate must meet to support that purpose [YPH01]. The *DoD Modeling and Simulation Verification, Validation Accreditation Instructions* [DoD96] supports credible M&S development and includes acceptability criteria¹⁰⁷, linking the roles and responsibilities of federation life-cycle development and operation with improved decision-makers confidence in the results when:

- Developers design and implement the software functionality to satisfy the criteria,
- V&V agents evaluate federation capabilities against the criteria,
- Accreditation agents make accreditation recommendations based upon the criteria [YPH01].

Department decision-makers accredit the use of the federation for a specific purpose based on confidence that the acceptability criteria were credibly established and implemented. A contributing factor to the substantive interoperability problem has been an inconsistent implementation of the Department's verification, validation, and accreditation (VV&A) policy identified in Chapter III. The Department's VV&A processes depend on consistent, credible conceptual models. However, Department's conceptual model implementation is inconsistent, and the theoretical debate over conceptual model formats, methods, uses and development processes continue unabated. In addition data to validate Department's M&S remains elusive, expensive and difficult to acquire, adding to the challenges of conducting creditable VV&A.

¹⁰⁶ The simulation conceptual model is the developer's translation of modeling requirements into a detailed design framework, from which the software, hardware, networks and systems/equipment that make up the M&S can be built [HY01a].

¹⁰⁷ Acceptability criteria should be necessary for the purpose, specific, measurable and complete [YPH01]

Substantive interoperability issues also exist due to the level of software engineering discipline and the general lack of process maturity exhibited by Department M&S software developers (see Chapter VII). The Department initiated major new large-scale M&S development efforts, JSIMS, JWARS, and JMASS, to replace many of the aging legacy systems and decrease the issues with substantive interoperability. However, all three major systems are well behind schedule, significantly over-cost, and failed to demonstrate they meet the original requirements. The development and implementation of software architectures, including product lines, in the Department while still very immature as it mirrors progress made in the private sector, has the potential to improve the Department M&S substantive interoperability problem in the future. Compounding the VV&A deficiencies is the use of over-subscribed terms for M&S quality attributes such as fidelity.

F. FIDELITY HETEROGENITY

Chapter VI discusses M&S quality attributes, including fidelity, and Table 6-2 provides the current Department context for M&S fidelity. However, continued concern about M&S fidelity within the SISO led to establishment of two fidelity study groups in the late-1990s with charters to produce a fidelity conceptual framework for M&S fidelity and a glossary of fidelity related terms [Gro99]. The Simulation Standards Interoperability Organization chartered the Fidelity Definition and Metrics Implementation Study Group (ISG-FDM) [Gro99] and a follow-on effort, the Fidelity Experimentation ISG (ISG-FEX) [RGH00] to address the issue of fidelity. These efforts produced collaborative integrated project reports addressing appropriate M&S fidelity definitions, a fidelity conceptual framework, formulas and concepts for the SISO community, and recommendations for future DMSO M&S efforts. However, fidelity, the subject of many scientific papers, studies, analyses, and reports, still lacks an acceptable, quantifiable methodology.

The [Gro99, RGH00] reports, a major SISO collaborative effort by over one hundred researchers, provided a summary of workable contextual frameworks within which M&S fidelity is considered, with an explicit indication of how such frameworks would relate to the larger theoretical context of modeling and M&S theory, including initial identification of methods, approaches, and metrics for usefully defining, estimating, and measur-

ing aspects of M&S fidelity. Fidelity of a model or simulation defines the terms of the relevant referent and the capabilities of the model or simulation. Fidelity also describes the essential characteristics of the model or simulation relative to its referent.

[Gro99, RGH00] also identified the dimensions and attributes of M&S fidelity based on the characteristics of reality, addressed by entities, factors and relationships within the simulation's enumeration of the problem's critical aspects. [Gro99, RGH00] further defined the scope of involved entities and the identifiable depths of entities. [Gro99] identified three dimensions of simulation fidelity, which include a number of the possibilities, or a percentage of factors

The first dimension of simulation fidelity [Gro99] identified is enumeration of the involved entities, addressed in both scope¹⁰⁸ and depth¹⁰⁹. In the second dimension of simulation fidelity [Gro99] included the identification of involved factors¹¹⁰ relating the processes, which influence, impact, or describe entity states and behavior. Specification of the significant relationships among entities is the third dimension of simulation fidelity forwarded by [Gro99] to explicitly articulate the assumptions about dependencies or independence among entities. [Gro99] also describes dependent relationships not explicitly identified as "otherwise"¹¹¹.

Fidelity presents significant challenges for accurately addressing the size, scope, and complexity of the real world and our current understanding of the real world preclude its use as a practical measure. Since the real world is not a good foundation to measure fidelity, research efforts have identified the need for common referent, a commonly understood standard [Gro99, RGH00]. This raises the point of how to assess the fidelity of a simulation against only those aspects of the referent needed to support the simulation; otherwise, if the simulation represented all aspects of the simuland¹¹² it would in fact be the simuland [Gro99, RGH00].

¹⁰⁸ Scope addresses the spectrum of entities represented by the simulation [Gro99].

¹⁰⁹ Depth is the level at which entities in a simulation can be individually identified [Gro99]

¹¹⁰ Factors may include components, materials, internal parameters, algorithms, and parameters related to measures of performance (MOPS)/effectiveness (MOE)/ merit (MOM) [Gro99].

¹¹¹ The "otherwise" category described by [Gro99] as a description for all dependent relationships that cannot be described more precisely.

¹¹² A simuland is the system being simulated by a simulation [DoD98].

The second obstacle [Gro99, RGH00] identified is the specification of the simulation. Defined simulation fidelity has the potential to act as a metric for describing how well the behavior of the unique aspects of the simulation matches selected parts of the simuland. [Gro99, RGH00] suggests the simulation also has many other characteristics describing the nature, behavior and character of the simulation, independent of the simuland, requiring a complex, multidimensional set of measures. The multidimensional set of measures include, but is not limited to, the simulation's intrinsic and extrinsic quality; development costs; resources needed to develop scenarios and execute the simulation; the extent that decomposition, aggregation and interfaces must be described; and the intended computation environment [Gro99, RGH00].

Simulation fidelity attributes address the quality of the parameters within the dimensions of M&S fidelity and include the following characteristics: factor order, accuracy, precision, timeliness, consistency, repeatability, and possible error sources [Gro99, RGH00]. Generally, within simulations the higher the factor order¹¹³, [Gro99, RGH00] suggests the greater the fidelity. However, since factors may be high order, low order, or some other order within the simulation, [Gro99, RGH00] suggests terms of factors provide better definition of simulation fidelity in, in lieu of using the concept of aggregation. Other simulation fidelity attributes include consistency¹¹⁴, repeatability¹¹⁵, and people issues¹¹⁶. A closely related concept including aspects of abstraction and aggregation is multiresolution modeling.

¹¹³ The quality of parameters within a simulation start with the order of the parameter description of the factor in the simulation with a zero order parameter description being a constant value, a first order parameter description would vary in one way according to some statistical distribution, a second order parameter description would vary in two ways, etc [Gro99, RGH00]

¹¹⁴ Consistency addresses whether simulation results are biased and stable in terms of the dispersion of results created by the simulation processes [Gro99, RGH00].

¹¹⁵ Repeatability means that a simulation should produce the same results given the same stimuli [Gro99, RGH00].

¹¹⁶ People issues, often ignored, is a special fidelity concern since as players, operators or analysts can impact accuracy, precision, consistency, and repeatability in simulation results [Gro99, RGH00].

G. MULTIREOLUTION MODELING (MRM) HETEROGENITY

[Zei91, Zei92a, Zei92b, Dav92a, Edd92,] introduced concepts supporting variable-resolution modeling (VRM), a precursor to multiresolution methodology (MRM)¹¹⁷. These early citations provided definitions, introduced basic concepts, identified types of VRM, provided examples, illustrated the importance of VRM for integrating new models or redesigning existing models, including potential cost savings [Sil92]. [Dav92a, Dav93] also discussed three classes of variable resolution they termed: selected viewing, alternative sub-models (or model families), and integrated hierarchical variable resolution (IHVR). The IHVR concept forwarded [Dav92a, Dav93] by defines critical processes, hierarchically composed of subordinate processes.

The study effort also built on previous MRM research by [Ham96, HNP97, Dav98, DB99, SHB00, and DB03], which proved invaluable for establishing a foundation supporting the introduction of software architecture-based M&S product lines. [Dav98] notes that MRM is “still a frontier subject in modeling and simulation” [Dav98], with more work needed in diverse areas including fundamental theory, computational tools, and visualization techniques. Important theoretical predecessor works include object-oriented modeling and discrete-event simulation concepts by [Zei91], modular hierarchical model representations by [Zei92a], a review of stochastic versus deterministic methodologies in combat simulations by [Luc00], and the development of methodologies for structured families with multiple levels of resolution¹¹⁸ [Zei92b].

Furthermore, [DB99] recognized the limitations of programming and operations research and identified a need for in-depth research “that leads to sound theories and designs well before coding even begins,” [DB99] and “should reflect sound principles of software engineering, including the ability to design in an object-oriented modeling framework” [Dav98, DB99]. In lieu of combining several different models at different levels of resolution into a federation or other form of distributed simulation operation, [Ham96, HNP97]

¹¹⁷ Some of the many terms that MRM issues arise under are model abstraction theory, problem decomposition, variable-resolution modeling, variable-fidelity modeling, hierarchical modeling, aggregation and disaggregation, chunking, and building of lumped, reduced, or simplified models [Dav98].

¹¹⁸ [Zei92b] suggests that resolution is the degree to which objects and processes in the real world are resolvable in the model, and furthermore, the resolution level is synonymous with the level of detail [Zei92b].

discuss an alternative strategy of developing and operating at multiple levels of resolution. [DBW87, BM92, Dav92a, DH92a, DH92b, Dav93, Ham96, DC97, HNP97, Dav98, DB99, MDB00A, MDB00b, Dav00, DBM00, BDM00, and Ade01] developed a significant body of knowledge on multiresolution modeling and multi-resolution multi-perspective modeling (MRMPM).

[DB99] further defines multiresolution modeling¹¹⁹. [DH92a] noted that achieving interoperability in diverse, independently developed M&S with different, but overlapping resolutions, based on different assumptions, limitations, and perspectives is a major challenge in contemporary M&S. [Dav92a, Dav93 and DBM99] in Figure 4-1 illustrate that the term resolution represents many different concepts and dimensions.

However, [Dav00] noted MRM may not be sufficient for applications requiring different abstractions or perspectives, that vary by conception of the system or use of variables, and requires a new model construct for both multiple resolution and multiple perspectives (MRMPM). In [MDB00a] models and families of models employed hierarchical trees to remain mutually consistent across multiple resolution and perspectives. [BDM00] employed an exploratory analysis technique using low-resolution models at the macro level of analysis and high-resolution models to support validation efforts.

Adding to the existing body of knowledge, [Dav00, DBM00, BDM00, and Dav93, and HOB95] compiled a significant body of work in multi-resolution, multi-perspective modeling (MRMPM) complementing the existing aggregation / disaggregation research. Taking a hybrid top-down /bottom-up approach to validation, [Dav95a] proposed a family of models optimally designed top-down and bottom-up concurrently, noting that the more detailed models may only have selected details and may lack the information required to develop valid aggregate models. [HNP97] suggest the development of a resolution hierarchy supporting the ability to control the bounds of inputs. [Dav95a] further indicates aggregated models may be able to establish the context and boundary conditions for events required in models that are more detailed. [Dav95b] applied the combat ratio 3:1 rule¹²⁰ to

¹¹⁹ Building a single model, a family of models, or both to describe the same phenomena at different levels of resolution [DB99].

¹²⁰ In combat operations the ratio of attackers to defenders for a successful offensive operation [Dav95b].

the complexities of aggregation and disaggregation and detailed how to achieve reasonably accurate and higher-aggregated levels.

[TNM01] later addressed the multi-resolution question in federations, and identified the problems that arise from federations or interaction between simulations with different levels of detail or abstraction requiring aggregation and disaggregation of information required by an HLA federation. [TNM01] submit two new approaches: the middleware, and the federate-based approaches to replace the module-based approach to implement aggregation/disaggregation in HLA environments.

Lacking integration, most model hierarchies are just a collection of models with different resolution and some type of procedures for calibrating selected input parameters of the lower-resolution with higher resolution models. [Dav98] noted the different design philosophy between the normal bottoms-up approach favored by the Department, which assumes truth is inherent in the most detailed models, whereas MRM, as noted in Table 4-1 supports any approach in achieving mutually calibrated models in a family. Today, with the end of the Cold War added increased uncertainty and added significant complexity to possible analytical issues and M&S development arising from the increasing instability, new variables, and unfamiliar post-Cold War geographical-political-military environments.

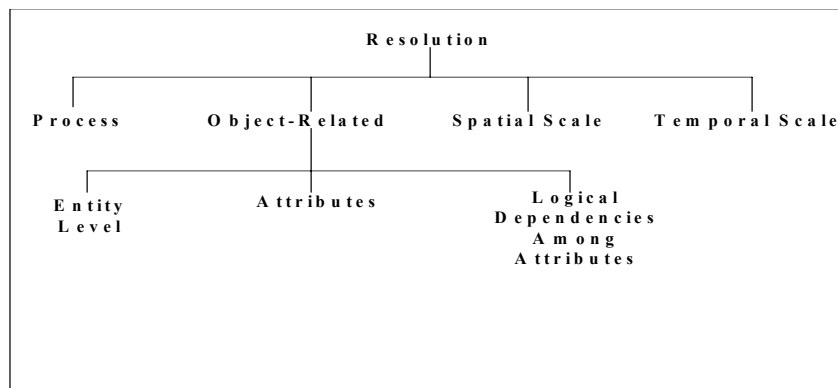


Figure 4-1. Resolution Dimensions (From [Dav98])

Different design methodologies listed in Figure 4-2, supported by several MRM tools exist. [DBM00] described the implementation of a fast-running, stochastic, mul-

tiresolution, stochastic model PGM¹²¹ Effectiveness Modifier (PEM) for exploratory analysis into the complex situational and tactical factors of long-range precision fires. In another effort to conduct insight-oriented exploratory analysis, [MDB00b] described the effort to employ the PC-based EXHALT model, based on the analytical methods identified in [DC97], to conduct a study of interdiction capabilities to quickly stop an invading army.

<u>NORMAL DESIGN</u>	<u>MRM DESIGN</u>
• DESIGN BOTTOM-UP AND MAINTAIN GENERALITY • DESIGN TO REFLECT SINGLE BEST "REALITY" • USE OBJECT-ORIENTED MODELING AND PROGRAMMING AROUND PHYSICALLY AROUND PHYSICALLY SIMILAR ENTITIES (MISSILES) • AVOID STRUCTURAL APPROXIMATIONS TO MAXIMIZE DEGREES-OF-FREEDOM FLEXIBILITY • DISPLAY AGGREGATE VARIABLE ONLY AS NEEDED • FOCUS ON SIMULATION RATHER THAN SENSITIVITY • FOCUS ON SIMULATION RATHER THAN NEEDS FOR SENSITIVITY PROGRAM • DESIGN WITH BOTTOM-UP ETHIC	• EXPLOIT "AGGREGATION FRAGMENTS" IN ALGORITHMS TO PROVIDE MULTIPLE LEVELS OF OPERATION • DESIGN TO HONOR MULTIPLE REPRESENTATIONS, INCLUDING REDUCTIONIST REPRESENTATIONS. DESIGN ALTERNATIVE USER MODES AND CORRESPONDING MODEL REPRESENTATIONS. • EXTEND "OBJECT" CONCEPT TO "FUNCTIONALLY" SIMILAR ENTITIES (E.G., SHOOTERS, WHETHER AIRCRAFT OR MISSILES). • SEEK APPROXIMATIONS THAT SIMPLIFY MODEL STRUCTURE SENSIBLY, THEREBY INCREASING ANALYTICAL FLEXIBILITY FOR EXPLORATION AND QUICK CHANGES OF ASSUMPTIONS. • CHOOSE PERSPECTIVE UP FRONT FROM POSSIBLE MODULARIZED COMBINATIONS OF AGGREGATE VARIABLES. • INSERT MULTIPLIERS, STRETCHERS, AND OTHER "ABSTRACTIONS OF CONVENIENCE". • DESIGN WITH ETHIC OF USING BOTTOM-UP, TOP-DOWN, AND SIDEWAYS APPROACHES IN MUTUALLY SUPPORTIVE WAYS.

Figure 4-2. Distinction between Normal and MRM Design (From [Dav98])

In a related effort, the ARPA/Tri-Services sponsored the Rapid-prototyping of Application Specific Signal Processors (RASSP) program research by [HMB+00] reviewed and compared simulation terminology, interoperability challenges, and previous modeling taxonomies; as a basis for designing a multi-axis taxonomy to describe the information content of computer model types; and define abstraction levels supporting development of interoperable models. The RASSP modeling taxonomy provides a framework to categorize models based on a set of attributes, which may distinguish models for different intended uses, and establish formal, concise, unambiguous definitions for various model types which

- Represent model attributes relevant to model designer and users,
- Provide a readily understandable common terminology,
- Identifies five distinct model characteristics: temporal detail, data value detail, functional detail, structural detail, and programming level,

¹²¹ PGM is an acronym for precision-guided munitions

- Consists of a set of attributes characterizing a model's relative resolution of details for important model aspects shown in Figure 4-3 [HMB+00].

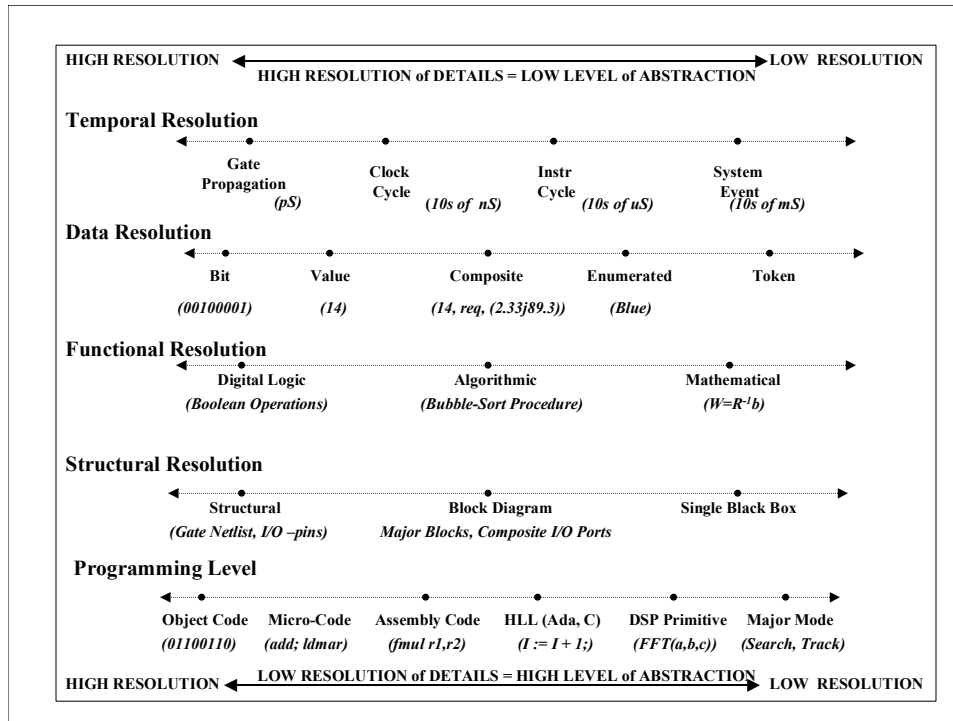


Figure 4-3. RASSP Taxonomy Model (From [HMB+00])

Within the RASSP Taxonomy Model at Figure 4-3, a given model instance describes information, presented graphically at one specific level within the model, even though some terms may span a range of abstraction levels¹²². The RASSP Taxonomy Model provides five categories of resolution in which:

- Temporal Resolution Axis represents the resolution of events presented on a time scale,
- Data Resolution Axis represents the resolution of the format of values specified in the model,
- Functional Resolution Axis represents the level of detail of a model describing the functionality of a component or system,

¹²² The abstraction is an indication of the level of detail specified about how a function is to be implemented and is adversely related to the level of detail. If there is much detail, or high resolution, the abstraction is considered low, and the abstraction levels form a hierarchy [HMB+00].

- Structural Resolution Axis represents the level of detail of a model describing how a component is constructed from its constituent part,
- Software Programming Resolution Axis represents the level of granularity of software instructions that the model of a hardware component interpret when executing target software [HMB+00].

The level of abstraction does not indicate accuracy; in the same way precision is different from accuracy, since two different models of a given function with different abstraction may be equally as accurate. In addition the RASSP Taxonomy Model provides two views of the attributes, the internal view as viewed from inside the model, and the external view as viewed from the model's interface boundary, each view supporting the temporal, value, structure, and function resolutions for a total of eight attributes enabling clarity and precision, unlike many other taxonomies which may mix attributes [HMB+00]. The internal resolution references how a model describes the timing of events, functions, values, and structures of the elements contained within the models boundary, whereas the external resolution describes how a model defines the interface of the modeled device to other devices [HMB+00].

The RASSP Taxonomy Model supports several hierarchies¹²³: the functional or logical hierarchy, and the structural or physical hierarchy. The functional hierarchy decomposes a system according to functions (e.g., receiver, register, transmitter), and the structural hierarchy decomposes a system according to physical structure (e.g., frames, racks, chassis) or logical relationships (e.g., data structures) [HMB+00].

Newer MRM theories include possible modifications to the HLA RTI and possible use of components. [SKO+01] completed additional work on MRM methods and proposed the use of state variables of simulation objects or environments in an MRM simulation system to determine the resolution of the models. [AS01] submitted a new HLA service, the multiresolution management service, using existing HLA services as a suggested addition to the RTI. [HBM96] proposes building multiple levels of resolution with defined ranges, into multiple independent models, as opposed to one supermodel, with the ability to remove representations, which are not required to complete comparative analysis. In a simi-

¹²³ Hierarchies in RASSP are a multi-level system supporting aggregation and decomposition, in which a node at a given level of the hierarchy may be represented by the set of its descendent nodes on the next lower level [HMB+00].

lar concept, [CGG98] explained the JSIMS Military Modeling Framework initiative, a Department enterprise-wide Tiger Team initiative to develop Mission Space Objects formed with interoperable, multiresolution models from reusable and dynamically re-configurable components to achieve a common interpretation within a synthetic battle space.

H. INTEROPERABILITY AND HETEROGENITY IN THE C4ISR DOMAIN

Interoperability poses significant challenges in many areas including the Command, Control, Computers, Communications, Intelligence, Surveillance, and Reconnaissance (C4ISR) dimension. [DoD02d] defines critical Department M&S requirements for improved C4ISR. However, there are currently severe M&S-to-C4ISR shortcomings identified by [SFJ+98, RHS99, Sut99, Tol99, DA00, TLW+00, ML01, BBN+01, HMT+01, WHL01, CWS02, DoD02c, DoD02d, ST02] for modeling information superiority.

[DoD02d] summarizes these shortcomings and notes:

- Good information superiority models do not exist,
- Pre-scripted battlefield entities are non-responsive and lower study fidelity,
- Existing models are simplistic, speculative, and too narrowly focused, lacking the ability to do end-to-end analysis,
- Current force-on-force models do not account for C4ISR or make broad assumptions,
- A lack of system performance data,
- The Department needs integrated treatment of threat signatures and sensor mission management,
- Suitable Opposing Force (OPFOR) representations are lacking [DoD02d].

A SISO M&S-to-C4I Interoperability Working Group report [TLW+00] identified that uncoordinated M&S and JTA standards have been, and continue to be developed by both communities, and several factors inhibit a completely seamless M&S-to-C4ISR interface including:

- Applicable standards (HLA, JTA) in both domains continue to evolve quickly,
- Interoperability between C4ISR systems or components may span multiple levels,
- Each domain must be able to retain authority and responsibility for data V&V and security,
- Each domain has access point responsibilities for data management, metadata, and conversions,
- Both domains must support active interfaces at the domain boundaries,

- Both architectures require a translation capability until future architectures employ the same data elements based on a common design [TLW+00].

The *C4ISR Architecture Framework* [AWG97] provides the strategic direction for C4ISR architecture development throughout the Department, and provides guidance on the development of a broad set of products used to document the operational, system and technical architecture views. An M&S-to-C4ISR interoperability framework proposed by [TLW+00] includes a three-phase plan with near-, mid-, and long-term solutions evolving from a limited message-based as-is architecture. The to-be architecture shares common databases, implicit interoperability, “plug and play” capabilities, and full duplex information exchange [TLW+00].

[TLW+00] identifies success in the mid-term architecture with common components, improved interfaces, and common standards, with simulations updating C4I systems over two-way communications initialized by either the simulation or the C4I system. Currently translators establish this interface, although translators represent an interim, expensive, error-prone, single-point-of-failure software solution until common, credible, interoperable components become available. [TLW+00] proposes solutions including a C4I/M&S technical reference model (TRM), a broad data class (metadata), a reference FOM, and SISO guides to linking C4I to M&S. The Levels of Information Systems Interoperability (LISI) model [AWG98], shown in Figure 4-4 illustrates components or systems spanning multiple levels of sophistication, supporting various system-to-system information exchanges.

Current department data quality initiatives support the object-oriented approach. The U.S. Army developed the Army Object Model Standards Category (OMSC) [JHB98, HB98a, HB98b, MG98] for M&S domain objects, and the Army Integrated Core Data Model (AICDM) a standard reference data model [WHL+01], which includes the Joint Common Data Base (JCDB), the Army Land C2 Information Exchange Data Model, and the C2 Core Data Model [MHL+02]. [HB99, LFW00, WHL+01, WHL+02] assessed interoperability between M&S and C4ISR data models, and evaluated the degree of alignment to determine common/data object interoperability.

LEVEL (Environment)			Interoperability Attributes			
			P rocedures	A pplications	I nfrastructure	D ata
Enterprise Level (Universal)	4	c	Multi-National Enterprises	Interactive (cross applications)	Multi-Dimensional Topologies	Cross-Enterprise Models
		b	Federal Enterprise			Enterprise Model
		a	DoD Enterprise	Full Object Cut & Paste		
Domain Level (Integrated)	3	c	Domain Service/Agency Doctrine, Procedures, Training, etc.	Shared Data (Situation Displays Direct DB Exchanges)	WAN	DBMS
		b		Group Collaboration (White Boards, VTC)		Domain Models
		a		Full Text Cut and Paste		
Functional Level (Distributed)	2	c	Common Operating Environment (DII-COE Level 5) Compliance	Web Browser	LAN	Program Models and Advanced Data Formats
		b		Basic Operations (Documents, Maps, Briefings, Pictures Spreadsheets, Data)		
		a	Program Standard Procedures, Training, etc.	Advanced Messaging (Parsers, E-Mail+)	Network	
Connected Level (Peer-to-Peer)	1	d	Standards Compliant (JTA, IEEE)	Basic Messaging (Plain Text, E-mail w/o attachments)	Two Way	Basic Data Formats
		c		Data File Transfer		
		b	Security Profile	Simple Interaction Text Chatter, Voice, Fax, Remote Access, Telemetry)	One Way	
		a				
Isolated Level (Manual)	0	d	Media Exchange Procedures	N/A	Removable Media	Media Formats
		c	Manual Access Controls		Manual Re-entry	Private Data
		b				
		a				
		o	No Known Interoperability			

Figure 4-4. LISI Model (From [TLW+00])

An Army M&S Interoperability Working Group [Bun00], reviewed the technical challenges, complexity, cost, and criticality of M&S interoperability, and provided a definition for M&S interoperability¹²⁴ and proposed four types of interoperability within a generic 3-tier architecture for simulations including:

- HLA Federation supporting federation runtime interoperability,
- Scenario data sharing among different M&S (e.g., environment, weapons effectiveness data) including automated tools, and reduced cost,
- Pedigreed data supporting the requirement for a lower-level M&S (e.g., higher resolution, precision) to develop data for a higher-level M&S in the Department's M&S hierarchy,
- Model / Algorithm sharing in M&S with different architectures and fidelity requirements [Bun00].

Related initiatives include the Department's Data Architecture [JTA02a], interfacing with C4I systems [Tol01], the development of a DII COE M&S infrastructure [HS00a], interoperability certification [BKW+02], object architectures supporting semantically interoperable systems [BCC+00], and weapons effectiveness [LPA+02]. [TH03] noted major discrepancies between the current OMSC and the JCDB standard reference data models with implications that the simulations currently under development do not support the Army Battle Command Systems, and require custom interface software or translators.

A follow-on SISO M&S / C4I study group [BCC+02] confirmed the need for a common technical reference model (TRM) [TLW00] to improve interoperability based on a review of M&S reference models, a prototype C4I / M&S interoperability TRM, information exchange activities, a general unified model (GUM), and related reference models. Proposed future initiatives included continued analysis of interoperability theories, a C4I / M&S TRM use-case study and development of TRM user guides. [TH03] proposed additional challenges to the current C4ISR-to-M&S incompatibility issue with six theses he framed as grand-challenges.

In addition, [TH03] propose to foster the development of information techniques and systems to support the war fighter based on the same views of the world, a common ontology, a common technical framework, and a common understanding of the constituent

¹²⁴ M&S interoperability is the ability of a model or simulation to provide services from other models and simulations, and to use the services so exchanged to enable them to operate effectively together [Bun00].

dynamics. [TH03] also express concern with the perception that the M&S community has not provided enough emphasis on the interoperability issues between C4ISR systems and M&S, beyond the development of representations of units, equipment, and behavior.

As a result, [TH03] identified a need for a common architecture, a common ontology, a common set of algorithms and methods supporting a common overarching concept bridging the different methodologies. The Department's alleged M&S research void cited by [TH03] provides a major impetus for this dissertation and the development of product line domain architecture. The six theses proposed by [TH03] to improve the C4ISR systems and military simulation systems ability to support future military operations include:

- Complementary techniques needed for future military operations,
- Use of the same commercial standards,
- Use of the same common architecture,
- Use of data and object models based on the same common ontology,
- Use of the same common set of algorithms and methods,
- A common overarching concept for C4ISR systems, and military simulation systems development and evolution [TH03].

Finally, [Pac01a] contends that although the Joint Technical Architecture (JTA) provides the Department architecture for exchanging information at the current time, and the HLA defines interoperability standards for M&S, no standards currently exist for:

- Decomposing M&S into entities and processes,
- Representation abstraction¹²⁵ of the simulated subject,
- Documenting authoritative representations in the M&S conceptual model [Pac01a].

I. SUMMARY

Chapter IV reviewed several heterogeneity challenges to Department M&S credibility. Improving the credibility of large-scale, legacy Department M&S requires an understanding of the underlying systemic causes generating the pre-conditions for heterogeneous system representation anomalies, especially in federation interoperability. These diverse challenges include syntactic and semantic heterogeneity, data heterogeneity, complexity, interoperability, technical and substantive interoperability, fidelity heterogeneity, and multiresolution modeling heterogeneity.

¹²⁵ Abstraction is a general model mapping process that takes a base model into a lumped mode where the models may be expressed in different formalisms, and subsumes aggregation as a special case [Zei92b].

Authoritative representations have been consistently identified as major objectives in Department M&S, yet few of the [DoD95, DoD01a] objectives for authoritative representations have been achieved [Mos00, Cra01a, Cra01b]. Senior civilian and military decision-makers have significant reservations that Department M&S, as currently practiced, can meet the serious future demands generated by national security requirements for M&S in the Department. In addition, current Department M&S users still require accurate, interoperable elements of the mission-space and the draft [DoD01a] retains authoritative representations as a Department-wide M&S objective including U.S., allied, friendly, coalition, neutral and threat and paramilitary system representations for C4ISR, combat, combat support and combat service support systems and processes. Successful attainment of these interoperability objectives by the Department and international M&S communities assumes the acceptable resolution of many heterogeneity issues, including abstractions, networks, federations and representations.

Abstraction is an intrinsic human activity employed to reduce or filter the full complexity of the real world to manageable proportions by deleting unnecessary detail in order to provide data in context (e.g., information) supporting the decision-making process [HJS97]. Abstraction mechanisms [Til98] selectively emphasize the level of detail, emphasizing the details pertinent to the intended use, while hiding unneeded detail. [Sow88] discussed the three most common abstraction mechanisms classification, aggregation, and generalization [Sow88, Til98].

Data models establish the essential properties and well-defined relationships between raw data in a system in a form, which supports efficient storage, timely retrieval of data, and provides the basis for tools and techniques to support data modeling. A data model captures the static and dynamic characteristics supporting data-related processes, with static properties defined in a database schema and the dynamic characteristics developed into specifications for reports, queries, and transactions. The schema maintains the object type definitions, attributes, relationships and static constraints of the data repository or database, an instance of the schema.

The most significant record-based logical data models include the hierarchical data model and the relational data model. These classic highly machine-oriented, and record-

oriented data models, or *syntactic data models* are well suited to the computer environment and organized for optimal efficiency supporting storage and retrieval operations, however these models lack semantic relativism and abstraction mechanisms required for dealing with complexity in large-scale systems [BCE+00].

[Til98] describes a second category, *semantic data models*, which combine basic knowledge representation techniques with database technology, and represent a transition from the basic record-oriented approach towards a model supporting more human-oriented semantic constructs. While traditional data models store data, and semantic models shift towards a more human knowledge model, [Til98] suggests a third method, the *conceptual modeling* activity geared more towards the domain-level knowledge and program understanding with a focus on the end-user. [Til98] further suggests that all three techniques may be more suitable than one method if there are different categories of artifacts (e.g., data, knowledge, information), requiring a relational model physical storage of data, a conceptual model for representing domain-level knowledge, or a semantic model for interactive discovery.

[Wie93] acknowledged that data comes from many diverse and heterogeneous sources and further suggested that joining heterogeneous data is necessary to generate information. Focusing primarily on database constructs and schema research, [WW90, Wie93, Wie00a, Wie00b, Wie01] developed eight viewpoints for differences among systems, and [You02c] added a ninth viewpoint of heterogeneity germane to this research:

- Heterogeneity of Hardware and Software systems,
- Heterogeneity of Organizational Models
- Heterogeneity in Representation
- Heterogeneity of Scope,
- Heterogeneity of Level of Abstraction,
- Heterogeneity of Meaning,
- Heterogeneity of Temporal Validity [Wie93],
- Semantic Heterogeneity [Wie00a, Wie00b],
- Heterogeneity of Structure [You02c].

This research revealed progress in the area of technical interoperability, including research into multilevel simulation of discrete network models [Ham96, BKG+02], distributed simulations [Ham96, HNP97], and the Department-sponsored development of the

HLA. Although there has been progress with the technical interoperability problem since the mid-1990s, many approaches failed to improve the credibility of authoritative representations and reduce the federation's interoperability problems caused by substantive interoperability anomalies. Over time developers modeled many of the same objects in different ways, potentially causing problems when they interact in a federation.

[DSB+99, HY01a, HY01b, YPH01] identified systemic substantive interoperability issues with legacy simulation software systems and the inconsistent representation of the same entity in multiple systems. There is a very close correlation on the root causes between the technical and substantive interoperability issues cited in the simulation domain, and the previously cited work of [Wie93, You02c] on information from heterogeneous sources. However, substantive interoperability remains a systemic issue persisting through the evolution of the ALSP, DIS and HLA protocols, and remains a major challenge for achieving truly distributed simulation interoperability.

[DB99] defines multiresolution modeling as “building a single model, a family of models, or both to describe the same phenomena at different levels of resolution” [DB99]. [DH92a] noted that achieving interoperability in diverse, independently developed M&S with different, but overlapping resolutions, based on different assumptions, limitations, and perspectives is a major challenge in contemporary M&S. [Dav92a, Dav93 and DBM99] in Figure 4-1 illustrate that the term resolution represents many different concepts and dimensions. However, [Dav00] noted MRM may not be sufficient for applications requiring different abstractions or perspectives, that vary by conception of the system or use of variables, and requires a new model construct for both multiple resolution and multiple perspectives (MRMPM).

In [MDB00a] models and families of models employed hierarchical trees to remain mutually consistent across multiple resolution and perspectives. [BDM00] employed an exploratory analysis technique using low-resolution models at the macro level of analysis and high-resolution models to support calibration and validation efforts. [TNM01] later addressed the multi-resolution question in federations, and identified the problems that arise from federations or interaction between simulations with different levels of detail or abstraction requiring aggregation and disaggregation of information required by an HLA

federation. [TNM01] submit two new approaches: the middleware, and the federate-based approaches to replace the module-based approach to implement aggregation/disaggregation in HLA environments. In a related effort, the ARPA/Tri-Services sponsored the Rapid-prototyping of Application Specific Signal Processors (RASSP) program to review and compare simulation terminology, interoperability challenges, and previous modeling taxonomies as a basis for designing a multi-axis taxonomy to describe the information content of computer model types and define abstraction levels supporting development of interoperable models [HMB+00].

Interoperability poses significant challenges in many areas including the Command, Control, Computers, Communications, Intelligence, Surveillance, and Reconnaissance (C4ISR) dimension. [DoD02d] defines critical Department M&S requirements for improved C4ISR. However, there are currently severe M&S to C4ISR shortcomings and:

- Good information superiority models do not exist,
- Pre-scripted battlefield entities are non-responsive and lower study fidelity,
- Existing models are simplistic, speculative, and too narrowly focused, lacking the ability to do end-to-end analysis,
- Current force-on-force models do not account for C4ISR or make broad assumptions,
- Models lack of system performance data,
- The Department needs integrated treatment of threat signatures and sensor mission management,
- Suitable Opposing Force (OPFOR) representations are lacking [DoD02d].

A SISO M&S-to-C4I Interoperability Working Group report [TLW+00] identified that uncoordinated M&S and JTA standards have been, and continue to be developed by both communities, and several factors inhibit a completely seamless M&S-to-C4ISR interface including:

- Applicable standards (HLA, JTA) in both domains continue to evolve quickly,
- Interoperability between C4ISR systems or components may span multiple levels,
- Each domain must be able to retain authority and responsibility for data V&V and security,
- Each domain has access point responsibilities for data management, metadata, and conversions,
- Both domains must support active interfaces at the domain boundaries,
- Both architectures require a translation capability until future architectures employ the same data elements based on a common design [TLW+00].

An M&S-to-C4ISR interoperability framework proposed by [TLW+00] includes a three-phase plan with near-, mid-, and long-term solutions evolving from a limited message-based as-is architecture. The projected to-be architecture shares common databases, implicit interoperability, “plug and play” capabilities and full duplex information exchange [TLW+00].

THIS PAGE INTENTIONALLY LEFT BLANK

V. THE DEPARTMENT'S FRAMEWORK FOR IMPROVED CREDIBILITY

A. INTRODUCTION

Chapter V provides the Department's current architectural framework for resolving the underlying systemic causes generating the pre-conditions for heterogeneous system representation anomalies, especially in federation interoperability. The Department's initiatives to improve the credibility of large-scale, legacy Department M&S supporting distributed interoperability include the "as-is" architecture and the evolving "to-be" architecture: the Joint Technical Architecture, and the Common Technical Framework. The Common Technical Framework components include the High-Level Architecture, conceptual model requirements, and data standards supporting credible simulation development.

B. THE DEPARTMENT'S AS-IS SIMULATION ARCHITECTURE

The Department established terms of reference for an M&S framework to improve the communication of concepts, interoperability, and development of authoritative representations including architectural implementations (e.g., SIMNET, ALSP, DIS, HLA). The Common Technical Framework (CTF) for M&S contains three components: the HLA, CMMS, and data. The Federation Development and Execution Process (FEDEP) support the implementation of the HLA. The HLA is the current Department and IEEE standard software architecture for distributed M&S interoperability, although it has not fully resolved long-standing technical and substantive interoperability issues. Consequently, representations of the same system in different models are frequently incompatible.

Conceptual Models of the Mission Space (CMMS) / Functional Description of the Mission Space (FDMS) are the developer's method for developing the M&S requirements into a detailed designed framework to develop the software. The verified and validated CMMS / FDMS subsequently supports the critical role of verifying the software development effort. Authoritative data, the final component of the CTF is critical to the Department's goal of improving credibility and confidence in simulation results, however authoritative data may be the single most pervasive M&S deficiency area.

Software architecture theory continues to evolve in academia, the private sector, and the public sector including the Department. Prior to the development of the HLA, the Department's M&S as-is distributed architecture framework consisted of the Simulation Network (SIMNET), the Aggregate Level Simulation Protocol (ALSP)¹²⁶, and Distributed Interactive Simulation (DIS) environments. SIMNET [MCU+95, Gre99] was a pioneering collaborative effort from 1983 until 1989 between the Defense Advanced Research Projects Agency (DARPA) and U.S. Army, to develop distributed simulations operating on several interconnected computers and determine the feasibility for distributed simulations to provide a training capability for the U.S. Army.

The ALSP distributed network protocols patterned after the SIMNET effort [WSW91, WWG93] permits multiple, pre-existing, large-scale aggregated-level, constructive warfare simulations to interact in logical time over local or wide area networks, control its own objects and share information with other simulations. ALSP, originally developed by DARPA in 1992, and currently managed by the U.S. Army Simulation, Training and Instrumentation Command (STRICOM) through a multi-service agreement, was based on four principles,¹²⁷ and a distributed architecture composed of a three-part, two-layer protocol component and two software components,¹²⁸ permitting interoperation of dissimilar Service and Joint constructive M&S [WSW91, WWG93].

Early ALSP confederations of Service simulations (e.g., Corps Battle Simulation, Air Warfare Simulation) supported several joint and combined training exercises (e.g., Atlantic Resolve, Unified Endeavor, Ulchi Focus Lens) [DoD95]. The Joint Training Confederation, initially deployed for major training exercises in 1992, evolved into the largest ALSP confederation with eight interacting simulations¹²⁹ [Log02]. The ALSP confederation simulations, developed originally as stand-alone systems, support only limited interoperability, require lengthy set-up times, lack interfaces to real-world systems, and need large

¹²⁶ ALSP is a family of M&S interface protocols and supporting infrastructure software that permits the integration of distinct M&S and war games. Combined, the interaction protocols and software enable large-scale, distributed M&S and war games of different domains to interact at the combat object and event level [DoD98].

¹²⁷ ALSP principles: 1) Distributed computation based on combat entity ownership, 2) avoidance of single critical resources, 3) reliance on broadcast communications, and 4) replication of a limited set of combat entity attributes among all M&S [WSW91].

¹²⁸ The ALSP protocol component exists of 1) a peer level protocol joins translators and object attribute management, 2) a connection protocol manages the translator to gateway, and 3) a second peer level protocol that joins gateways and deals with timing issues. The ALSP software components include the translators and the gateways [WSW91].

¹²⁹ ALSP interacting simulations: AWSIM, CBS, RESA, MTWS, CSSTSS, JQUAD, TACSIM, and PSM [Log02].

amounts of manpower. The Department plans to replace ALSP confederations with the Joint Simulation System.

The Distributed Interactive Simulation (DIS) protocol [MCU+95, IEE98a] provides a synthetic interactive networked environment in which humans may interact real time, through the connection of different platform-level virtual M&S, simulators, or instrumented live exercises. The DIS protocols and standards [IEE98a] establish a common data exchange environment or common messaging environment, using Protocol Data Units, supporting the interoperability of heterogeneous, geographically-distributed live, virtual, and constructive simulations [DoD95]. The DIS synthetic environment uses verified scenarios, tactics, techniques, and procedures to train testers on new hardware or software [AWQ+93], and conduct trial test runs before executing costly field tests.

The DIS protocol development effort also experienced broad participation from an open forum of industry, government, and academia, and plays a central role in the Department's M&S portfolio. [LBV96, Gre99] cited key DIS techniques including a "dead reckoning" methodology based on a world and body coordinate system to minimize communication requirements between simulators in order to produce a realistic and credible virtual battlefield. However, the computation-intensive DIS protocol requires high bandwidth levels for its broadcast messaging technique, and lacks support for different time management methods, realistic command and control representations, and dynamic changes in the environment [DoD95].

C. THE DEPARTMENT'S TO-BE ARCHITECTURE

The Department's future "to-be" architecture development includes the evolving Joint Technical Architecture (JTA), common Operating Environment (COE) and C4ISR Framework supported by the Technical Reference Model (TRM) with a major focus to improve interoperability and information exchange. The Department's COE, JTA, and C4ISR framework support the foundation of the software-intensive "to-be" architecture. The Department's "to-be" architecture or target architecture will evolve to meet the performance requirements codified by the Government Performance and Results Act of 1993 and Clinger-Cohen Act of 1996 [CC96], the legal frameworks for developing United States

Government enterprise-level information technology architecture standards and policy [LSF+01]. The Department's "to-be"-architecture will also reflect the emerging Federal Enterprise Architecture standards identified by [LLT+99] and approved by the Chief Information Officer (CIO) council [Ham00a] under the direction of Executive Order 13011, for Federal Information Technology.

The Department's Enterprise Integration (EI) Implementing Strategy and Enterprise Model described by [DoD00c] identifies near- and long-term strategies including the existing Corporate Information Management strategy, an integrated technical architecture framework for information management, data standards, and shared databases. The long-term EI strategy provided by [DoD00c] "ensures consistency, quality, timeliness, availability, and security of shared, corporate data by implementing corporate databases using standard data elements as soon as possible" [DoD01a].

The *Defense Information Infrastructure (DII) Common Operating Environment (COE)* [Har97, Har98a, DoD00d] emerged in the mid-1990s as a foundation for developing open systems with a "plug and play" open-architecture capability based on the open technical architecture developed for the Global Command and Control System [HPS98]. The DII COE, an architectural approach for building interoperable systems, includes a collection of reusable components, a software infrastructure, and a set of standards and guidelines for an open architecture designed around a client / server model [BPM+95]. The DII COE also contains products to meet operational requirements, which are not fully JTA compliant [HB00, JTA01a]; however, the goal is to evolve to full compliance with the appropriate JTA standard. In support of this goal, [HS00a, CH01, CM03] presented M&S infrastructure alternatives to better align the current M&S architecture with the DII COE.

The *Joint Technical Architecture (JTA)* [JTA02a, JTA02b] and the *C4ISR Architecture Framework* [AWG97] provides the Department with the basis for developing systems with the required seamless interoperability, defining service areas, interfaces, terminology, and standards applicable to all systems and mandated for the management and development of new or improved Department systems [Ste02a]. The Department implemented the JTA by defining an interrelated set of operational, technical, and system views. The JTA also builds on service areas identified in the Technical Reference Model [BK01a, TRM01a,

TRM01b], supported by technically stable, mature commercial standards and guidelines. The JTA [JTA02a] consists of two main parts: the JTA Core and the JTA Domains with sub-domains shown in Figure 5-1, the JTA Hierarchy Model.

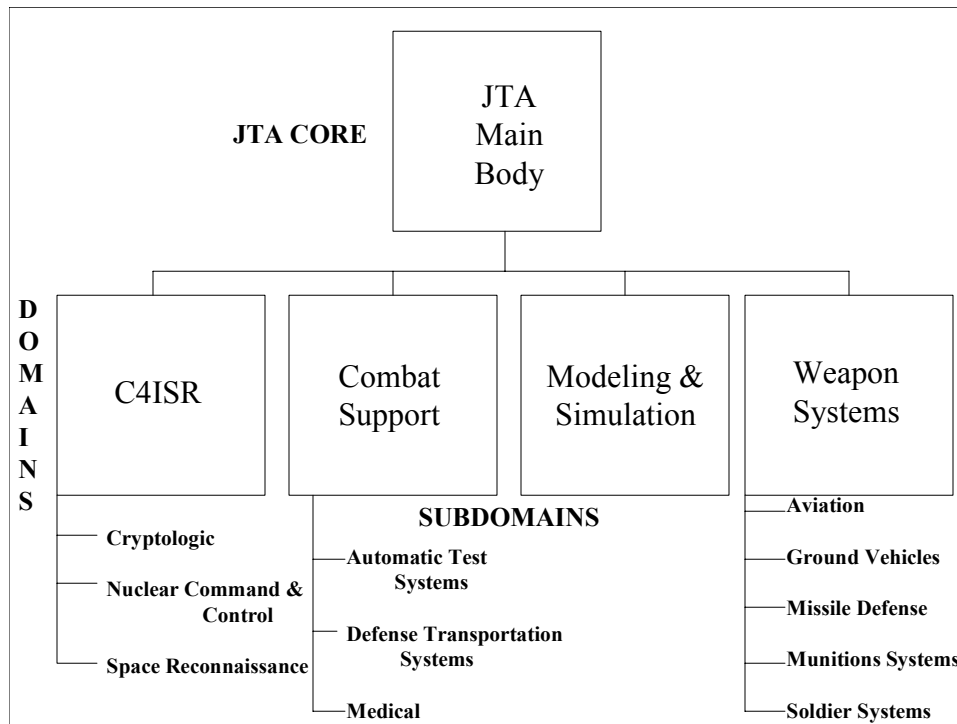


Figure 5-1. JTA Hierarchy Model (From [JTA02a])

The JTA [JTA02a] core information technology categories include information processing standards, information transfer standards, information modeling, metadata, information exchange standards, human-computer interface, and information security standards. The JTA also identifies the minimum set of JTA elements applicable to all Department systems to achieve interoperability and domain-specific elements to ensure interoperability within the domain (e.g., C4ISR, Combat Support, Modeling and Simulation, and Weapon Systems), but not necessarily for inter-domain interoperability between the four JTA domains or their subordinate domain elements [JTA02a].

The Department's Technical Reference Model (TRM) [TRM01a, GMW00b] and TRM User Guide [TRM01b] support the requirements of increasingly complex and diverse

systems by integrating the service and interface views, and identifying interfaces and content. The [TRM01a] evolved from the Department's Technical Architecture Framework for Information Management (TAFIM), and includes three major model elements (1) the services, (2) the interfaces and (3) the entities. The three major model elements specified by the [TRM01a] include the following characteristics: the ability to support system architectures [Dem95b, MR02], model degree of freedom to select or expand on services and interfaces, and the ability to support new services.

The [TRM01a] also provides interface definitions, and environment configurations, supports a model's ability for different views, and provides methods of mapping the model to other known reference models. In a related effort within the M&S domain, the SISO C4ISR/ Simulation Technical Reference Model study group [BBM+01, GLT+02] developed a standard frame of reference for interoperability between C4ISR systems and M&S systems. [RHS99] also developed a supporting conceptual model with an information exchange model focused on the information exchange between C4ISR and M&S components.

The Department developed the JTA [JTA02a, JTA02b] and the C4ISR Architecture Framework [AWG97] to improve the system architecture. However, [CBB+03] caution that the C4ISR Architecture Framework [AWG97] almost exclusively documents the system architecture, and that none of the three [AWG97] views (e.g., operational, systems, technical), or the essential or supporting products, "prescribe anything that remotely resembles software architecture" [CBB+03]. This assessment of the maturity of the Department's software architecture including [AWG97] is a systemic issue, since [IEE00b] notes there is no single, accepted framework codifying software architectures, despite significant research in the area.

The Modeling and Simulation Domain prescribed by [TRM01a] includes the Department's Common Technical Framework (CTF) [DoD95, GMS+96] for facilitating M&S interoperability and reuse. The CTF has three components: the High-Level Architecture (HLA) to which the Department's which M&S must conform; Conceptual Model of the Mission Space (CMMS) to provide a basis for the development of consistent and authoritative representations; and data standards to provide common representations of data across M&S and C4I systems [DoD95, GMS+96]. The Department's M&S architecture efforts

cited by [DoD95, GMS+96] focused on interoperability and improving the shortfalls of the previous interoperability protocols (e.g., ALSP, DIS).

1. High-Level Architecture (HLA)

The [DoD95] defined the requirement for a common high-level simulation architecture to facilitate interoperability of all types of simulations, interoperability with C4I systems, and improved reuse of M&S components. In the Department's M&S domain, the current standard software architecture for distributed simulation interoperability is the HLA. Under development since 1995 and designated the technical architecture for all Department M&S [Gan00], the HLA technical architecture includes three major components: the simulation member of the federations, or federate, the runtime infrastructure (RTI), and the runtime interface. The HLA, is also an IEEE standard [IEE00c] with a reusable common distributed framework, composed of an HLA rule set [IEE98e, IEE00c], the HLA interface specification [IEE00d,] and the HLA Object Model Template (OMT) [IEE00e, Lut00], supporting a wide range of M&S application areas at different resolutions.

The HLA rules [IEE00c] are divided into two major categories: federate and federation rules. An HLA federate may be another computer-based simulation, a manned simulator, an interface to a live or instrumented range, or a simulation utility. An HLA federation is a set of interacting simulations or federates represented by a federation object model (FOM) based on the OMT format [IEE00c, BRA02]. The HLA OMT [IEE00e] specifies the object model and the essential objects, attributes, and interactions, shared across the federation, supporting interoperability. An HLA federate requires a simulation object model (SOM) based on the OMT, to identify all public information shared across the federation [IEE00c].

A FOM identifies the essential classes of objects, object attributes, and object interactions [IEE00e], supported by the HLA federation. At runtime in an HLA federation, 1) the federate manages all object representations, and 2) only one federate owns any specified attribute of an instance of an object at one time. The HLA interface specification describes the run time services provided by the RTI to a federate, and prescribes the interface from the federate to the RTI, with six classes of management services: federation, declara-

tion, object, ownership, time (continuous, time-stepped, discrete event), and data distribution [IEE00d].

In addition, the HLA supports the passive collection of data and the interface to live participants using the HLA interface to interact with the RTI [Dah99]. The HLA runtime interface specification supports the following RTI activities: 1) a standard for federation interaction with the RTI, 2) the method to invoke RTI services supporting federation runtime interactions, and 3) the ability of a federate to interact with the RTI [IEE00d]. Several HLA interoperability capabilities and lessons-learned identified by [Tol02] have been addressed in federation calibration experiments [BLR+02], federating time-stepped and discrete event simulations [Bee02], and object-oriented architectures [CA02]. [Dah99, Har01c] described the current technical status of the HLA and identified future challenges. Other initiatives include component-based extensions [RPK02]; employing the Extensible Markup Language (XML) [SB00] and Unified Modeling Language (UML) methods suggested by [BRJ99, OMG99, SB01, Oes02, and MB02] to improve HLA documentation; and distributed test and evaluation implementations [Har98b].

The HLA RTI according to [Dah99, CA02] acts as the distributed operating system providing the interface specification and management services to the federation. [BJ98] compared the HLA with other private sector distributing computing efforts including the Common Object Request Broker (CORBA) sponsored by the Object Management Group (OMG), and the Remote Method Invocation (RMI) from Sunsoft's Java Development Kit (JDK) [WWN97], and identified the strengths and weaknesses of each approach. [BCB02] discussed methods to manage, monitor, and manage federations. Viewing federation performance as a critical factor, [FMF02] explain RTI benchmark studies comparing three RTI implementation approaches, while [HW02] described the possible implementation of data distribution management services in the next-generation RTI.

Community researchers continually provide recommendations for HLA improvements and enhancements. [HHS+98] suggests specifying additional optional classes of information allowing the development of a more complete description of the federation structure or behavior. [HHS+98] also explains the capstone definition, class terminology, and classification rules of a reference FOM. [RD00] describe several additional FOMs, devel-

oped according to the format described in the HLA OMT, including a prototype C4I FOM. [GRH+01, GHL+01, and GHO+01] describe and specify the Base Object Model (BOM); a SISO-approved type of reference FOM designed to provide a component overlay capability to the HLA architecture. Other FOM initiatives include an intelligence and electronic warfare FOM developed by [WS02] to bridge constructive, virtual and live systems in a synthetic environment to improve interoperability; a federation object model for atmospheric dispersion proposed by [HCP+02]; and a real-time platform reference FOM suggested by [OF02].

In order for the Department to achieve M&S interoperability goals, [MCL+99] identified several issues including the need for “a better means of agreeing and translating alternative model data representations (e.g., model interoperability)” [MCL+99]. In support of improved simulation interoperability, [LLP+02] conducted and analyzed calibration exercises to baseline performance data, while [CCB+02] described the development of an HLA-compliant high-performance computing RTI to improve RTI interoperability performance issues identified by [MCL+99, FMF02]. The FOM and OMT (and their relationship with the RTI) also need extensions and better definitions according to [MCL+99].

After initial HLA federation experimentation and prototype efforts experienced a great deal of trial and error, the Department M&S community determined the need for additional guidance to improve cross-domain interoperability and future federation collaboration. The community achieved consensus for various “best practices” aspects of federation development, leading to the establishment of a common process view for HLA federations addressing the spectrum of interests within the HLA community, and introduced the Federation Development and Execution Process (FEDEP) [DMS99] in September 1996, followed by a release of the FEDEP concept of operations in 1997.

The FEDEP six-step process illustrated in Figure 5-2 continues to evolve as the community experience identifies improvements [Wai97, Wai02a, DMS99] and efforts are currently underway to propose FEDEP IEEE guidance for future HLA federation construction methods [LDG+02]. In a continuing trend to perform only inherently governmental functions, the Department transitioned the RTI development responsibilities from a DMSO-sponsored, government-funded project to the private sector for future development

and capitalization in October 2002. In December 2003, researchers at the Johns Hopkins Applied Physics Laboratory (APL) reported the first successful commercial application of the IEEE 1616 HLA specification in a medical federation between the Massachusetts Institute of Technology cardiovascular system simulation and the APL's detailed model of both the left and right ventricles [JHU03].

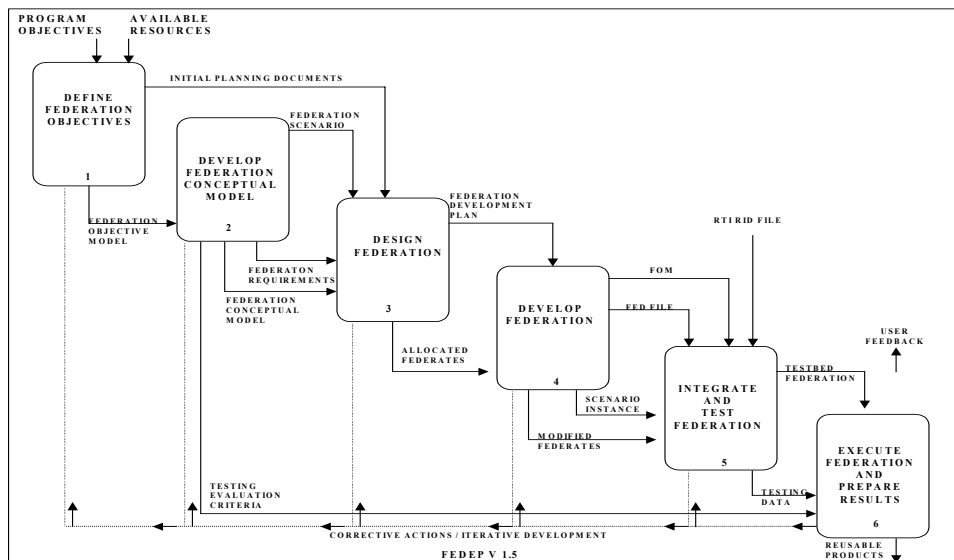


Figure 5-2. The FEDEP Process (From [DMS99])

2. Conceptual Model of the Mission Space (CMMS)

A conceptual model [DoD95, GMS+96] is the developer's method of translating the M&S requirements into a detailed design framework to develop the software, and describes the conceptual model components, interactions, and the M&S concept of operations. Several researchers, including [SPC+98, Pac00a, SH00, She00b, DDH01] validated the basic need for conceptual models, and developed supporting perspectives of conceptual modeling theory, identified systemic issues, and proposed solutions. However, the Department's management of conceptual models has two major challenges to overcome, an inconsistent past and a demanding future.

Today, the Department's M&S community currently lacks consensus and standards for conceptual models. Moreover, conceptual models are critical to the key Department's

M&S issues of authoritative representations, improved simulation credibility, enhanced confidence in simulation results, integrated simulation security, and improved simulation interoperability. Surveying the current M&S situation, [LRH+01] consider the term “conceptual model” extremely overloaded and overused within the M&S community. For example, the term “conceptual model” describes the first abstraction of representations in an M&S and describes a high-level design of how all the components in an M&S relate to another. The wide and varied body of literature on the subject of conceptual models is an indication of their importance, but also reveals a lack of consensus on conceptual model methods, verification and validation, format and development methodology.

A short survey on the breadth of conceptual modeling research, concepts, theories, and the different perspectives include the following widely-diverse conceptual model topics: verification and validation [Tho97b, Wai97, Pac98a, Pac98c, SPC+98, Pac99a, Pac00a, Pac00b, Pac00c, Met00, LRH+01, Pac01a, Bor02], conceptual model object-oriented schemas [Wai97], conceptual model quality [TB97], and conceptual model documentation [Tho97b, Pac99a]. The literature also includes significant information on: conceptual model environment [Bir98], conceptual model aggregation / disaggregation [Bid00], fidelity [Pac98c, Pac00a, Pac00c, Pac01a], conceptual models and the mission space [MM98], conceptual model validation [Pac98c], conceptual model/ functional description of the mission space transition [HM98, Jor01], multiple conceptual model approaches [Whi99], and conceptual model reuse [Pac00c, Pac02a]. In addition the literature search revealed additional citations for aircraft conceptual models [Cha00], assumptions document [LK00], joint conceptual models [Bir98, Met00, LRH+01], and conceptual models for HLA federations [Tho97b, LC97, Wai97, Pac98c, Bir98, Pac01a].

The conceptual model may also include the algorithms, equations as well as any assumptions and limitations. However, even the naming convention for the Department’s conceptual models is challenging with the CMMS [DoD95, GMS+96] term and the newer Functional Description of the Mission Space (FDMS) [RPG00] term used interchangeably. In this effort we will normally use the term conceptual model unless specifically addressing the CMMS- or FDMS-versions of the conceptual model methodology.

However, whichever term or format is used, whether its is the older CMMS term or the newer FDMS label, [GMS+96] cites a Department's key concern with M&S credibility as the lack of a detailed and documented conceptual model and design specification for existing legacy systems M&S. Conceptual model verification is the Department's process for ensuring the conceptual model meets specified requirements. However, the conceptual model documentation for many legacy M&S may be inadequate, if it exists at all, according to [Pac99b]. The conceptual model for simulations suggested by [Pac00a] addresses the simulation's requirements, context, entities and processes.

[Had98] proposed a specification of metrics for describing existing CMMS and for guiding subject matter experts in the acquisition and documentation of the Military Operations Mission Space. [FY97] decomposes his conceptual model design and M&S implementation according into a layered structure to account for model dependency, information flow, and model control; and envisions three different abstractions: the real world, the conceptual model of the mission space and the software implementation. Additional perspectives on the definition and purpose of conceptual models, where conceptual models fit into the M&S development lifecycle, and how conceptual models are developed include concepts forwarded by [GMS+96, Hom+97, SPC+98, DMS98a, DMS98b, DMS00a, Pac00a, RPG00, Bor02, Pac02d]. A properly developed conceptual model for the M&S is the best generic validity referent according to [Pac01b], since it captures both pertinent elements of system theory and intended M&S application.

Today, there still exist several systemic impediments to the development of credible explicit conceptual models. Two major issues addressed by [Pac02a] are the lack of formal requirements or standards for conceptual models and the lack of an explicit M&S conceptual model as a defined contract deliverable. There are also issues identified by [Pac00a] with abstracting representation from available information, or for describing and documenting the conceptual model. In addition, [Bor02] is concerned about the level of understanding and development of conceptual models of the mission.

Without a verified and validated conceptual model (e.g., CMMS or FDMS) upon which to base the verification of the software implementation, and ultimate validation of the simulation, it may be extremely difficult, if not technically impossible to credibly fol-

low the Department's recommended guidelines for verification and validation [GMS+96, RPG00]. This may result in an adverse opinion of the simulation's credibility, which undermines user confidence in the simulations results.

Definitions of conceptual model validity, model verification, operational validity, data validity, and recommended procedures forwarded by [Sar98] include the relationship of V&V to the development process. Models based on repeatable, measurable phenomena are usually empirical according to [Mey98] while simulations are normally probabilistic in nature since the supporting phenomena may be unknown, uncertain or questionable. [Mey98] also proposed the possible valid uses of a simulation if the real-world phenomena are unavailable, while acknowledging the possible invalid use of a model without knowing the supporting phenomena. [Pac01a] also provides ideas on how to perform validation assessments when there is inadequate information about one or more federate conceptual models.

[Had98] describes the problem of defining an appropriate level of resolution for the CMMS, and addresses ways that system purpose and desired capabilities can drive the development of prescribed requirements for CMMS. In a follow-on SISO-sponsored effort, [Gro99 and RGH00] proposed a new CMMS methodology to define the context for the development and support of simulation object models and federation object models, based on describing and developing several CMMS corresponding to the battlespace of real mission areas, including sufficient information to support the new CMMS role as a referent for fidelity measures. The overarching CMMS framework envisioned by [Gro99 and RGH00] would provide access to the necessary detailed data and support the development of consistent and interoperable authoritative representations within a standard fidelity framework. More importantly an expanded CMMS model provides a possible solution to the systemic issue of how to define a standard fidelity referent, which has been a major obstacle to the development of accepted definitions of fidelity, which [Gro99 and RGH00] contend is needed for supporting fidelity-based verification and validation process.

Conceptual models also support the knowledge acquisition process and provide a common starting point for constructing consistent and authoritative M&S representations. The knowledge acquisition alternatives addressed by [TS92, CPS+96, Ede97, Mai97,

DRC99, KBO99, DSB+01] support the development of a valid conceptual model. A conceptual model framework endorsed by [Pac00a] identifies four steps in developing a conceptual model, and proposes greater use of formal methods to develop M&S requirements and define the conceptual model. [HOM+97, RPG00] specify a CMMS technical framework providing:

- Technical standards,
- Administrative procedures,
- Operational infrastructure
- The common semantics and syntax¹³⁰ for describing the mission space,
- A common format database management system [RPG00],
- A closed-loop engineering process for creating and maintaining conceptual models,
- Data interchange format (DIF) standards for conceptual model integration and interoperability [HOM+97, RPG00].

Several DMSO sponsored reports [HSB98, She00b, HS00a, HS00b] address the state of current conceptual model development methods and tools developed by and for the Department's M&S community. A recent comprehensive assessment of the DMSO CMMS support effort by [Luq01] addressed the following areas: a) the effectiveness of software risk assessment models, b) enhancements to the FDMS resource center, c) XML and wrapper-based translators for Department databases, and d) metrics for selecting automated test tools; and suggested improvements based on the latest research.

Most experts agree on the need for conceptual models, although a wide variety of interpretations and a great deal of confusion for conceptual modeling exists within the Department's M&S community today; with consensus difficult to achieve on the exact definition of conceptual modes [LRH+01]. As a result, the current draft [DoD01a] cites a need for developing new and improved V&V technologies and methodologies, and improvements for the V&V of representations.

¹³⁰ Every language of communication possesses two identifiable properties, the form of the language and the meaning associated with the form. The syntax of the language is a set of rules specifying which forms of the language are grammatically acceptable, its grammar. The meaning derived from a syntactically correct instance of a language may be viewed from two points, the meaning intended by the originator, the semantics of the language, or the meaning interpreted by a receiver, the pragmatic meaning [RR83].

3. Functional Description of the Mission Space (FDMS)

The term Functional Description of the Mission Space (FDMS) [RPG00, Dou01] replaced the previous Conceptual Model of the Mission Space (CMMS) term, in the Department's M&S lexicon, as a required component of the M&S development and V&V processes. The FDMS provides the real-world description of entities, processes, and activities for the design phase, and detailed representations of the problem domain in the requirements development phase of the simulation life cycle. Most of the conceptual models [LRH+01] described support classification either as a domain-oriented classification, such as the battlespace [PRM97], or design-oriented classification. There are four basic formats to describing conceptual models identified by [RPG00]: the *ad hoc* method, design accommodation, development paradigm, and the scientific paper approach. The FDMS employs the development paradigm method, although the [RPG00] recommends the scientific paper. The *ad hoc* approach is the most common format employed today to develop the conceptual model [RPG00]. The M&S developer normally produces the FDMS.

4. Data Standards

Data standards are the third component of the common technical framework for M&S. In any M&S application, the associated data quality should be as creditable as the M&S itself if the user or sponsor is to attach confidence to the results. [DoD95] defines data quality¹³¹ for the Department's M&S programs. A proposed draft [DoD01a] places increased emphasis on data quality adding, "That a model implementation *and its associated data* accurately represents the developer's conceptual description and specification" [DoD01a]. [BCE+99] also suggests data quality attributes including source accuracy, fidelity, suitability, credibility, maturity, cost, availability, and similar characteristics. [Kil02] recommends data accuracy and quality information supported by a review at three levels: database, data element, and data value; employing descriptive, quality and usage metadata.

¹³¹ Data quality includes the correctness, timeliness, accuracy, completeness, relevance, and accessibility that makes data appropriate for use. Quality statements are required for source, accuracy (e.g., positional and attribute), up-to-dateness / currency, logical consistency, completeness (e.g., feature and attribute), clipping indicator, security classification, and releasability [DoD95].

A data verification and validation process assesses the quality of the data, employing data quality measures, and establishes the basis of confidence in the data [Kil02].

For each phase or aspect of the simulation's design, development and operation, possible sources of performance information include live tests, data centers, ground tests or analytical M&S. Theoretically, the data producer uses metadata to describe the quality of the data as a part of data V&V process to meet the specification requirements and provides data meeting the user's specification. The user assesses the applicability and quality of the data for a specific application based on the producer's verified metadata. As part of the users V&V process, the data verification process checks the model algorithms, and as the last step, validates the data and algorithms to establish overall M&S credibility. The DMSO-sponsored Authoritative Data Source (ADS) project [VED98] identified seven hundred and fifty-six authoritative data sources that met ADS data procedures, including metadata definitions, in an effort to decrease the redundancy of data development, and improve data quality, fidelity, and data sharing.

Environment data is critical to the M&S community. Sun Tzu realized the importance of knowing the environment when he wrote, "Know the other, know yourself and the victory will not be at risk; know the ground, know the natural conditions, and victory can be total" [Tzu71]. The second [DoD95] objective included the development of natural environment representations sub-objectives for terrain domains including: (2.1), ocean (2.2), atmosphere (2.3), and space (2.4) instantiated by the Ocean, Atmosphere and Space Environmental Services (OASES) System [RIO01].

However, quality environmental data is hard to acquire. [BPT96] studied the eight most-needed types of environmental data identified by earlier surveys and concluded that the capability to acquire these data, especially at high fidelity were insufficient to meet requirements, especially in the models of fog, aerosols, humidity, visibility, and databases. Responding to these concerns, the Department invested heavily in environmental data. In an early DMSO sponsored initiative to improve environmental representations, the Naval Research Laboratory [NRL95] developed the Environmental Effects for Distributed Interactive Simulation (E²DIS) program. The E²DIS program employed an object-oriented

methodology described by [HJG94, Hec94, WH94, AFW+94] to support distributed interactive M&S with environments and environmental effects.

The Synthetic Environment Data Representation and Interchange Specification (SEDRIS) program is a major Department follow-on initiative to the E²DIS program with objectives cited by [SED98a, SED98b, SED98c] to improve environment representations. The SEDRIS vision [SED98a] is to provide a common synthetic environment to reduce cost, improve reuse, support interoperability, and provide effective data transfer between heterogeneous systems, initially in the training functional area. SEDRIS employed a meta-model based transmittal methodology with a standardized data model and supporting application program interfaces. The SEDRIS Data Model, based on an Object Model Template (OMT), with minor extensions, employs the concept of a class of data [SED98a] supporting a common environmental architecture.

SEDRIS' major function is to serve as an interchange medium supporting diverse requirements for different customer domains [Bir97]. SEDRIS also includes a Geospatial Reference Model (ISO 18024) supporting the specification of coordinates, datum, projections, and several geo- and non-geo-referenced spatial reference systems, the SEDRIS transmittal format (ISO 18026) and the Environmental Data Coding Specification (ISO 180230) [RMJ+02, JMP02]. Complementing these initiatives are current or projected efforts to acquire science data from other the private sector, international scientific programs, or cooperative public-private partnerships.

NASA led the effort to collect data from space-based sensors, and continues to this day developing spacecraft for a complete Earth Observing System, with planned experimental Pathfinder missions under the auspices of the Earth Science Enterprise [PFL+00]. Another emerging international initiative collaborative effort promoting efficient geospatial data development, sharing and use is the Global Spatial Data Infrastructure [LWK+02].

However, remotely sensed data from space [JBS+93, NAB+92, FG99] may be difficult to share based on different sensor characteristics (e.g., sensitivity, spectral coverage, calibration), the lack of standards, proprietary data processing systems and algorithms, and subtle variations that may adversely influence data consistency [JSK02]. In addition, ensuring commercial data sources are available requires an understanding of negotiations for

intellectual property [IEE99b, ATL01] and understanding of how information technology standards¹³² emerge, survive and evolve [Lib01].

D. SUMMARY

Chapter V provided the Department's current architectural framework for resolving the underlying systemic causes generating the pre-conditions for heterogeneous system representation anomalies, especially in federation interoperability. The Department's initiatives to improve the credibility of large-scale, legacy Department M&S supporting distributed interoperability include the "as-is" architecture and the evolving "to-be" architecture: the Joint Technical Architecture, and the Common Technical Framework. The Common Technical Framework components include the High-Level Architecture, conceptual model requirements, and data standards supporting credible simulation development.

The Department established terms of reference for an M&S framework to improve the communication of concepts, interoperability, and development of authoritative representations including architectural implementations (e.g., SIMNET, ALSP, DIS, and HLA). The Federation Development and Execution Process (FEDEP) support the implementation of the HLA.

The Department's future "to-be" architecture development includes the evolving Joint Technical Architecture (JTA), common Operating Environment (COE) and C4ISR Framework supported by the Technical Reference Model (TRM) with a major focus to improve interoperability and information exchange. The COE, JTA, and C4ISR framework support the foundation of the software-intensive Department's To-Be Architecture.

The Department's Modeling and Simulation Domain prescribed by [TRM01a] includes the Common Technical Framework (CTF) [DoD95, GMS+96] for facilitating M&S interoperability and reuse. The CTF has three components: the High-Level Architecture (HLA) to which the Department's which M&S must conform; Conceptual Model of the Mission Space (CMMS) to provide a basis for the development of consistent and authoritative representations; and data standards to provide common representations of data across M&S and C4I systems.

¹³² Information technology standards are a means by which two or more products (or systems) can function together. [Lib01]

A conceptual model is the developer's method of translating the M&S requirements into a detailed design framework to develop the software, and describes the conceptual model components, interactions, and the M&S concept of operations. [SPC+98, Pac00a, SH00, She00b, DDH01] validated the basic need for conceptual models, and developed supporting perspectives of conceptual modeling theory, identified systemic issues, and proposed solutions. However, the Department's M&S community currently lacks consensus and standards for conceptual models, which are critical to the key Department's M&S issues of authoritative representations, improved simulation credibility, enhanced confidence in simulation results, integrated simulation security, and improved simulation interoperability.

Surveying the current M&S situation, [LRH+01] consider the term "conceptual model" extremely overloaded and overused within the M&S community. For example, the term "conceptual model" describes the first abstraction of representations in an M&S and describes a high-level design of how all the components in an M&S relate to another. The wide and varied body of literature on the subject of conceptual models is an indication of their importance, and also reveals a lack of consensus on conceptual model methods, verification and validation, format and development methodology.

The term Functional Description of the Mission Space (FDMS) replaced the previous Conceptual Model of the Mission Space (CMMS) term, in the Department's M&S lexicon, as a required component of the M&S development and V&V processes. The FDMS provides the real-world description of entities, processes, and activities for the design phase, and detailed representations of the problem domain in the requirements development phase of the simulation life cycle.

Data standards are the third component of the common technical framework for M&S. In any M&S application, the associated data quality should be as creditable as the M&S itself if the user or sponsor is to attach confidence to the results. [DoD95] defines data quality for the Department's M&S programs. [BCE+99] also suggests data quality attributes including source accuracy, fidelity, suitability, credibility, maturity, cost, availability, and similar characteristics. [Kil02] recommends data accuracy and quality informa-

tion supported by a review at three levels: database, data element, and data value; employing descriptive, quality and usage metadata.

Theoretically, the data producer uses metadata to describe the quality of the data as a part of data V&V process to meet the specification requirements and provides data meeting the user's specification. The user assesses the applicability and quality of the data for a specific application based on the producer's verified metadata. As part of the users V&V process, the data verification process checks the model algorithms, and as the last step, validates the data and algorithms to establish overall M&S credibility.

Environment data is critical to the M&S community. However, quality environmental data is hard to acquire. In an early DMSO sponsored initiative to improve environmental representations, the Naval Research Laboratory developed the Environmental Effects for Distributed Interactive Simulation (E²DIS) program. The Synthetic Environment Data Representation and Interchange Specification (SEDRIS) program is a major Department to improve environment representations. The SEDRIS vision is to provide a common synthetic environment to reduce cost, improve reuse, support interoperability, and provide effective data transfer between heterogeneous systems, initially in the training functional area. The SEDRIS program employs a meta-model based transmittal methodology with a standardized data model and supporting application program interfaces.

Complementing these initiatives are current or projected efforts to acquire science data from other the private sector, international scientific programs, or cooperative public-private partnerships. However, remotely sensed data from space may be difficult to share based on different sensor characteristics (e.g., sensitivity, spectral coverage, calibration), the lack of standards, proprietary data processing systems and algorithms, and subtle variations that may adversely influence data consistency. In addition, ensuring commercial data sources are available requires an understanding of negotiations for intellectual property.

VI. MODEL QUALITY ATTRIBUTES FOR IMPROVED CREDIBILITY

A. INTRODUCTION

Chapter VI addresses software and simulation model quality attributes affecting heterogeneous system representation anomalies and credibility in Department M&S, especially in federation interoperability. The chapter reviews the internal and external attributes of the ISO 9126-1 Quality model [ISO01], testing for quality attributes, and M&S quality attributes addressed in much of the current literature, including an overview of aggregation and disaggregation. The chapter concludes with a discussion on M&S quality approaches.

B. SYSTEMIC SIMULATION SOFTWARE QUALITY ISSUES

In the 1970s, the cost of software maintenance for the first time exceeded the cost of software development, creating a “software crisis” and identifying for the first time that the technology at the time was inadequate to develop large, complex software-based systems. By the mid-1980s “silver bullet” solutions such as computer-aided software engineering (CASE) tools were tried and found wanting to solve the “software crisis” as software evolved into a \$300 billion dollar a year industry [Rak01].

In 1988, the Department’s Director of Operational Test and Evaluation (OT&E) [Kri88] acknowledged that the evolving art of M&S needed guidance to improve quality and establish credibility of simulation results through verification, validation and credibility assessments. [Kri89] implemented a policy on M&S support of OT&E specified by [DOT89] that identified the following high-level objectives in the model development process supporting credibility, improved quality and confidence in M&S results:

- Acceptability of the M&S approach,
- Confidence in the model,
- Confidence in the M&S team,
- Confidence in methodology and use,
- M&S verification, validation and accreditation [DOT89].

Society’s growing dependency on large, complex software-intensive systems complicates the development of quality software. According to [Boo01], as “software continues to weave itself deeply into the fabric of society, the stakes have gotten higher” [Boo01].

This challenge has driven the development of many different methods, techniques and approaches to improve software quality supporting the Department's large-scale, software-intensive, simulation systems.

C. SOFTWARE QUALITY INITIATIVES

Today, Department testers still require sufficient confidence in the quality of M&S results [DOT99, COH00], to support a foundation of trust for using M&S results as the basis of management decisions or part of the formal test and evaluation process. The Department initiated several programs to improve the relationship between M&S and live testing, reduce costs and improve the overall testing process. In 1995, the Secretary of Defense approved five initiatives, including the more effective use of M&S, to mature the OT&E community culture from a pass/fail, event-driven paradigm to the view of testing as a learning process.

[Wai02b] discusses several of these initiatives, including the role of hardware-in-the-loop M&S in missile defense systems testing and the emerging collaborative testing environments developing in technical testing architectures such as the Test and Training Enabling Architecture (TENA) [PLL+02] and the Virtual Proving Ground (VPG). In an effort to improve M&S management in the Department's test and evaluation process [Obr99] proposed the Modeling and Simulation Test and Evaluation Reform (MASTER) initiative, based on specific categories of expertise (e.g., terrain, weather), etc called M&S vectors. The MASTER concept [Obr99] also supported verification and validation of assigned models performed by government R&D centers, and a consortium organizational structure to implement policy and maintain a repository of codes.

Quality is an essential attribute supporting the safe operation of M&S software systems, especially M&S involving possible life or death situations [Bow02]. [Lev00] described the root causes found in the software industry's flawed quality processes, which created the conditions for the Ariane 5 failure in 1996 resulting in an uninsured five billion dollar loss. Several major spacecraft losses including NASA's Mars Polar Lander, Mars Global Surveyor, and the Titan IV / MILSTAR spacecraft [Lev01a], revealed similar software engineering quality shortfalls. It is probable that similar software engineering quality

and safety issues exist within the Department's software engineering programs today according to [HNC+00]. Furthermore, [HNC+00] found the current level of the Department's software engineering practice lacks discipline and consistent enforcement mechanisms necessary to improve these conditions.

There have been several uses and revisions of the term software quality¹³³ including [MRW77, IEE89, IEE92, IEE98d] over the past twenty-five years. Today, software quality remains open to subjectivity, different views, and various interpretations of definitions. [MRW77] addressed software quality attributes in an early effort and proposed a software quality model that identified three categories of software quality attributes. The study of software quality and software quality metrics [IEE89, IEE98d] evolved continuously with the maturation of software engineering practices and the growth of the software industry.

The ISO developed a two-part software quality specification model [ISO01], provided at Figure 6-1, applicable to every type of software product. The current [ISO01] definition of quality, similar to the IEEE definition cites "the totality of characteristics of an entity that bare on its ability to satisfy stated and implied needs" [ISO01]. The ISO Quality Model identifies six characteristics for internal and external product quality and their associated sub characteristics. An additional four quality-in-use characteristics prescribed by [ISO01] describes the combined effects of internal and external product quality.

The study of software quality by [Dro95, Dro96] suggests that the emphasis on quality software development has been heavily process-oriented and proposes a product-based model composed of components and modules possessing three key quality properties: cohesion, coupling, and layering¹³⁴, linked to quality attributes. As object-oriented (OO) languages continue to evolve, [Ale01] also addressed the additional complexity of OO languages over previous procedural programming testing techniques in OO testing, including inheritance, polymorphism and complex data requiring adequate testing of all relationships. [Fow01] suggests the earliest indication of design quality is the level of coupling, high-level dependencies and cohesion.

¹³³ Software quality is the degree to which software possess a desired combination of quality attributes [IEE98d].

¹³⁴ Layering for objects, programs, processes and systems are governed by the constructive principle that components constructed at one level may only be used at the next higher level. [Dro96]

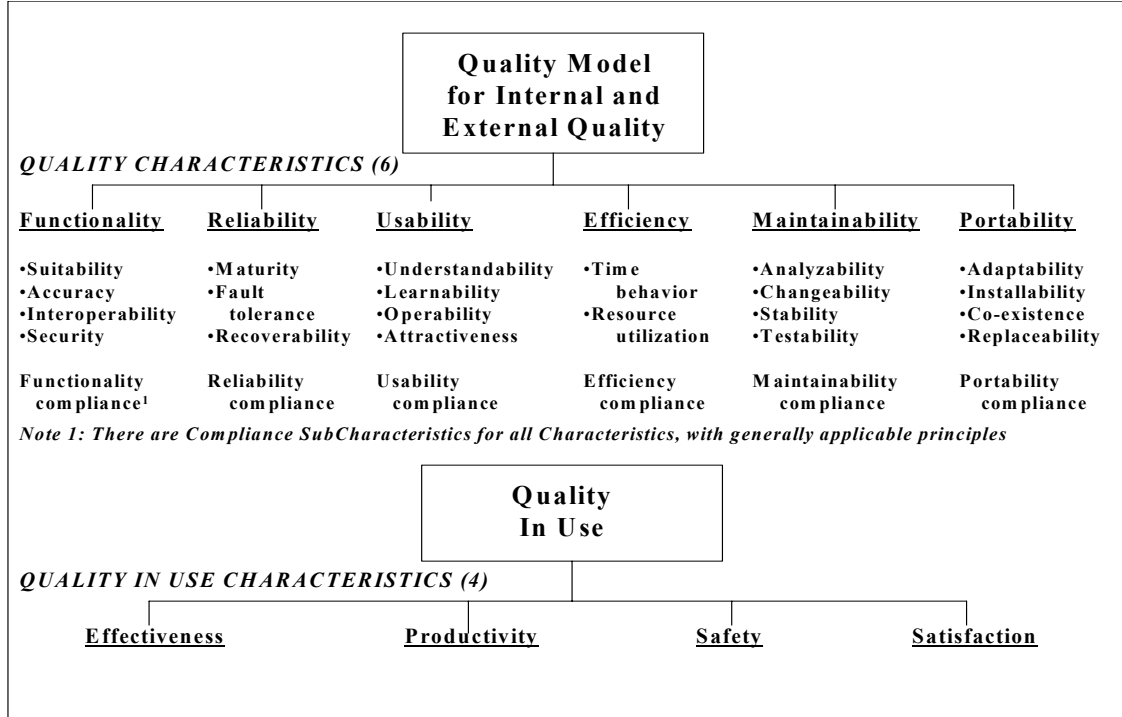


Figure 6-1. The ISO 9126-1 Quality Model (From [ISO01])

[IEE98d] provides a standard for software quality metrics supporting assessments for both the process and the product on the status of meeting quality requirements throughout the software life cycle. [Sch02a] incorporates quality metrics into software engineering practices, citing a growing quality measurement body of knowledge [IEE00a] augmenting the IEEE software quality standard. [BD02] added an extension to the [Dro96] hierarchical methodology quality framework work of developing the Quality Model for Object-oriented Design (QMOOD) to improve object-oriented design quality. [BD02] based the QMOOD model on the following broad quality attributes: functionality, effectiveness, understandability, extendibility, reusability, and flexibility, in lieu of the ISO 9126 quality attributes [ISO01] listed in Figure 6-1.

Methodologies for improved software quality continue to appear. [BHK01] suggests software documentation inspections in the early stages of a software project may allow for the early identification of defects, and experiments conducted by [BHK01] support the employment of a second inspection cycle to improve overall quality. [HJB+98] took a

different approach and developed a method for developing product/process dependency models (PPDMs), empirical software artifacts, reflecting past experiences to improve the planning, techniques, tools or methods for future projects. In addition, [FPC97, Kis01] identified measurement models and frameworks by supporting the software engineer's ability to establish product measures supporting improved overall software quality and reliability. [Boo01] suggests properly applied measures provide management with important insights into the health and welfare of a software project.

1. Internal / External Quality

[ISO01] divides software quality into six major characteristics: functionality, reliability, usability, efficiency, maintainability, and portability, with each characteristic composed of sub-characteristics¹³⁵. Functionality is the capability of the software product functions to meet stated and implied needs under certain specified conditions, and includes the sub-characteristics of suitability¹³⁶, accuracy, interoperability, and security.

The [DoD98] glossary defines accuracy as “the degree of exactness of a model or simulation, high accuracy implying low error”¹³⁷. In addition, “accuracy equates to the quality of the result, and is distinguished from precision” [DoD98]. A significantly different definition of accuracy by [RGH00] cites accuracy as, “the agreement between the performance of these models of each aspect and the real world performance” [RGH00]. An additional definition from the [RPG00] guidelines defines accuracy as the “degree parameters, parameter sets, or variables correspond exactly to the simulation reality, referent or some chosen standard” [RPG00].

Accuracy is also one of several model attributes and dimensions related to model real-world fidelity, and a quality sub-characteristic of software functionality identified by [ISO01] as the capability to achieve the agreed upon or correct solutions with the required

¹³⁵ There are *compliance* sub-characteristics for all six characteristics with similar standards and protocols [ISO01].

¹³⁶ Suitability is the software's fitness of purpose to provide an appropriate set of functions to complete certain tasks, meet user objectives, and affects operability [ISO01].

¹³⁷ Error: The difference between an observed, measured, or calculated value and a correct value [RPG00].

degree of precision¹³⁸. According to [Kil02] confidence in software quality and accuracy depends on the development environment, verification evidence, and software quality assurance [IEE89] and assessments, which includes accepted software development practices, personnel skill and experience attributes, sufficient resources and the existence of key life cycle development artifacts such as configuration management histories and logs. [Kil02] also notes, “dedicated software engineering experience is essential to most large-scale software development efforts” [Kil02].

M&S interoperability¹³⁹ is also an extremely complex systemic issue, and one of the central aspects of the Department’s current M&S efforts. Major M&S interoperability issues confronting joint model development and federations include data, algorithms [CLR97], representations, and joint C3I [You97, DPB+01]. [ISO01] defines interoperability as the ability of the software to interact with specified systems. However, interoperability is not just an M&S specific challenge, it is a universal Department issue, which impacts many sectors of the Department information domain [Ngu95, Wof95, HKL96, AH97, HMD99, Sut99, DSB+99, Ham00b, HDG00, YBG+01, HM00, GBL01, LBG+01, WHL+01, BKW+02, CW02]. Software developers and maintainers may introduce interoperability anomalies in autonomously developed software-intensive systems in all phases of a systems life cycle.

Interoperability is also a multi-dimensional challenge. The interoperability requirements listed by [DoD02d] are similar to the architecture concept for seamless M&S with a single integrated environment proposed by [Dow92]. The synthetic environment ideas advanced by [Dow92] identifies three classes of requirements: 1) simulation truth, 2) conceptual consistency, and 3) temporal consistency, which generate the supporting technical dimensions including the a) system architecture, b) data management, c) human machine interaction, and d) time management. [Ham96, HNP97] and [HM02] address the management of time, clocks, and synchronization in distributed simulation, the key role of

¹³⁸ Precision relates to the quality of the operation by which the result is obtained and can be repeated, and is distinguished from the term *accuracy* that is the quality of the result. [DoD98]

¹³⁹ M&S interoperability is the ability of a model or M&S to provide services to and accept services from other M&S, and to use the services so exchanged to operate effectively together [DoD98]

partitioning, and identifies five methods for partitioning a simulation for distributed execution, with synchronization noted as the most important implementation consideration.

The Internet is a major new interoperability dimension. With the explosive growth in the Internet in the 1990's, the Clinton Administration conceived a National Information Infrastructure (NII) program to deliver to all Americans the information they needed at an affordable price. A complementary Department initiative effort to the NII, the Defense Information Infrastructure (DII) [TM97], provides support for "dual-use" technology and added privacy, reliability, and security quality attributes to NII initiatives. The DII foundation elements¹⁴⁰ [TM97] included modeling and simulation, standards, architectures, software engineering, and test and evaluation.

However, [SFJ+98] cited continuing interoperability issues at almost every level. The Department determined a need to develop better policies and procedures to align and resource the technology base to support information assurance [DoD02e], information dissemination management, interoperability, network management, network operations, and enterprise computing and implemented the Global Information Grid¹⁴¹ (GIG) and the GIG Architecture¹⁴² (GIGA) [GIG00]. [Ham00b] provided Departmental GIG policy and guidance. [DPB+01, Cha01, Wol01, BBN+01] also identified the need for new analytical tools to support improved interoperability and facilitate the industrial age to information age transformation process.

The security quality attribute¹⁴³ addresses the ability of the software to prevent unauthorized access to programs or data, whether accidental or deliberate [ISO01]. The National and the Departmental strategies to counter threats and security challenges [WHM+79, LX99, KH02, SB02a] include natural environment, man-made physical hazards, and human actors, which are now considered more complex and more difficult than earlier threats [WHM+79]. The complex global interdependent information environment identified by the worldwide Year 2000 remediation effort [Gre97, WG99, Mus02] includes

¹⁴⁰ DII foundation elements included policy, requirements, modeling and M&S, standards, architectures, technology base, software engineering, test and evaluation, and joint spectrum management [TM97].

¹⁴¹ The GIG is "a globally interconnected end-to-end set of information capabilities, associated processes and personnel for collecting, storing, disseminating, and managing information on demand to warfighters, policy makers, and support personnel" [Ham00, GIG00].

¹⁴² The GIGA is composed of interrelated operational, systems, and technical views, which defines the characteristics of and relationships among current and planned Global Information Grid assets in support of National Security missions [GIG00].

¹⁴³ Security quality attributes are: availability, identification, authentication, confidentiality, integrity, and non-repudiation [Woo00b].

critical infrastructure¹⁴⁴ protection [EFL+97, ABB00, Bus01, GPH+01], information operations (IO)¹⁴⁵ doctrine [JP98], IO opportunities and vulnerabilities addressed by [SBJ+02], and information superiority (IS)¹⁴⁶ issues [JP98].

The Department is just beginning to understand the risks to an information superiority strategy and information operations (IO) as a basis for safeguarding the Nation's critical interests, and responding with information assurance (IA) measures. Information assurance¹⁴⁷ includes risks and protective measures [HNP97, Ham00b, ABB00, Woo00, Mon00b, WFP+01, DoD02d, DoD02e]. The Department selected Millennium Challenge 02 (MC02), a large-scale experiment to define the impact of IO measures employing the Joint Staff Analysis Model, as part of a limited-objective IO/IA experiment [All02]. Initial MC02 results suggest that modeling advanced IO, IS, and IA capabilities present significant challenges to the Departmental M&S community.

Reliability, the second of six major [ISO01] software quality characteristics, involves the capability of software used under certain conditions to meet a specified level of performance. Reliability is comprised of the following sub-characteristics: maturity, fault tolerance, and recoverability [ISO01]. Since software does not wear out or age, limitations in reliability are rooted in the requirements, design, implementation, and maintenance processes. One hundred percent software reliability may be an impossible goal to attain in the current software development environment when one considers that even 'six-sigma'¹⁴⁸ strategies seek only to reduce the number of errors in software systems.

Software reliability engineering (SRE)¹⁴⁹ practices are critical in high-risk, safety-critical areas, which affect national security or risk human life. NASA Shuttle mission engineers successfully implemented the Space Shuttle Flight Software Application methodol-

¹⁴⁴ Critical infrastructure was defined in the 1998 Presidential Decision Directive 63 as "those physical and cyber-based information systems essential to the minimum operations of the economy and the government" [SBJ+02].

¹⁴⁵ Information operations are operations conducted to defend our own information and information systems and affect adversary information and information systems [Woo00b].

¹⁴⁶ Information superiority is the capability to collect, process and disseminate an uninterrupted flow of information, while exploiting or denying an adversary's capability to do the same [Woo00b].

¹⁴⁷ Information assurance, a subset of information operations, is actions that protect and defend information and information systems by ensuring availability, integrity, authentication, confidentiality, and non-repudiation, and includes restoration of information systems by incorporating protection, detection, and reaction capabilities [Woo00b].

¹⁴⁸ Six Sigma objectives are to reduce the number of defects to 3.4 defects per million lines of code [STS00].

¹⁴⁹ SRE is the application of statistical techniques to data collected during system development and operation to specify, predict, estimate, and assess the reliability of software-based systems. [Sch99a]

ogy [KS97, Sch97a, Sch97b, Sch99a] to improve overall Shuttle software reliability. Software reliability affects all aspects of the software life cycle to include architecture [Lew02a], requirements [Sch01a, Sch01b], risk [Sch01d], software maintenance process [Sch99b], commercial-off-the-shelf software use [Sch00c], client-server systems [Sch96], testing [JDL01], and software quality control and prediction [Sch00c]. [Rus98, IC99, RC99, IC01, RC01] describe a decision support tool for selecting a reliability strategy based on project, product and resource decision factors. Other software reliability sub-characteristics include maturity, fault tolerance, and recoverability where:

- Maturity is the ability of the software to avoid failure stemming from faults in the software product, or the frequency of failures,
- Fault tolerance is the robustness of the system and its ability to maintain a specific level of performance in the event of faults or specified interface problems, and may include a fail-safe capability,
- Recoverability is the software's ability to regain a specified level of performance and recover affected data in the event of a failure [ISO01].

Usability is the third of six major [ISO01] software quality characteristics. Usability includes the ability of a user to understand, learn, and employ the software produce under specified conditions and is comprised of the following sub-characteristics: understanding, learnability, operability, and attractiveness [ISO01] where:

- Understandability is dependent on the documentation and the initial impression of the software, and allows the user to determine if the software is suitable, and how it may be used for specific tasks and under what conditions,
- Learnability is the ability of the user to learn to how to use the application,
- Operability is the ability of the user to operate and control the software product, and is affected by the attributes of suitability, changeability, adaptability, and installability,
- Attractiveness is normally associated with the graphical design, uses of color, and the ability of the software to be attractive to the user [ISO01].

The usability sub-characteristics are subjective, dependent, most difficult to measure, and are the least objective quality factors. [JWC01] acknowledge that usability is a difficult characteristic to integrate into any system, and requires specific knowledge of user requirements, preferences, and limitations. Research by [GRS02] borrowed from the psy-

chology field in studying the relationship between pain research and the implications for usability in software engineering, specifically the peak-end effect.¹⁵⁰

Research of simulation software usability by [Kil02] suggests that simulation software is credible only when employed within a well-defined context of use; that usability is more a case of reducing the probability of simulation misuse rather than the ease of use; and includes the M&S user support activities, which facilitate the credible use of the M&S. [DoD01a] found most Department simulations were too difficult to use and too resource intensive to employ as widely as needed to achieve the complete range and scope of required objectives. [Kil02] also identified configuration management as an indicator of M&S usability and the “glue that ties the version of the simulation to all the V&V results and M&S documentation” [Kil02].

Efficiency, the fourth of the six major [ISO01] software quality characteristics, involves the software’s resource dependent ability to provide appropriate performance, under stated conditions. Time-behavior and resource utilization comprise the [ISO01] sub-characteristics of efficiency where:

- Time behavior allows the software under stated conditions to provide appropriate responses, throughput rates, and processing times,
- Resource utilization is the ability of the software, under prescribed conditions to perform its function using appropriate amounts and types of software [ISO01].

The fifth of six major [ISO01] software quality characteristics is maintainability. Maintainability addresses the ability to modify the software product, including corrections, improvements, or adaptation of the software to changes in the requirements, environment or functional specifications, and includes the following [ISO01] sub-characteristics where:

- Analyzability is the ability to diagnose the software for deficiencies, failures, or for ease of maintenance and modification,.
- Changeability incorporates design, coding, and documentation and the ability to implement a specific modification,
- Stability allows the software to overcome unintended effects from software modifications,
- Testability provides the capability to validate the modified software [ISO01].

¹⁵⁰ Redelmeier and Kahneman discovered the peak-and-end effect in patient study research when patients were asked to rate their pain at regular intervals during the procedure [GRS02].

Portability, the sixth and last of the six major [ISO01] software quality characteristics deals with the sub-characteristics of adaptability, installability, co-existence, and replaceability to support standards and conventions of software portability from one environment to another.

2. Quality in Use

The [ISO01] standard specifies quality-in-use characteristics as enablers allowing users to satisfactorily, effectively, productively, and safely, achieve specified objectives. The [ISO01] quality-in-use characteristics include effectiveness and productivity where:

- Effectiveness is the software's ability to accurately and completely achieve specified goals,
- Productivity is the software's ability to achieve results effectively when compared to the expended resources [ISO01].

Technical constraints include the lack of tools, data security, data descriptions, variable resolutions [HOB95], and hardware/software limitations. Cultural challenges include the lack of trained personnel, immature processes and the lack of acceptance of M&S tools,

Safety is the software's ability to maintain acceptable levels of risk to people, environment, property, business, or software [ISO01]. [Lev00, Lev01a, Lev01b, Lev01c] identified the causal factors related to software safety issues in recent air and space accidents including the 1996 explosion of the Ariane 5 launcher, the 1999 loss of the Mars Climate Orbiter, the placement of a MILSTAR satellite in an incorrect and unusable orbit in 1999, and the destruction of the Mars Polar Lander in 2000. Several additional space accident reports [LLF+96, SMB+99, YAB+00] and aircraft accident reports [Lad93, SL94, Lad95] revealed very similar systemic safety issue factors including, but not limited to:

- Overconfidence and over-reliance on digital automation, where software complexity is underestimated and the effectiveness of testing is overestimated,
- Not understanding the risks associated with software,
- Confusing reliability with safety,
- Over-reliance on redundancy,
- Assuming risks decrease over time,
- Ignoring warning signals,
- Inadequate cognitive engineering,
- Inadequate specifications,

- Flawed review processes,
- Inadequate system safety engineering,
- Violation of basic safety engineering practices in the digital parts of the system,
- Software reuse without appropriate safety analysis,
- Unnecessary complexity and software functions,
- Operational personnel not understanding automation,
- Test and simulation environments that do not match the operational environment,
- Deficiencies in safety-related information collection and uses [Lev01b].

[Lev01a] expanded upon the cited accident-causal factors and noted their existence in three systemic organizational categories including flaws in the safety culture, ineffective organizational structure and communication, and inadequate or ineffective technical activities. In aerospace systems, a general testing heuristic is to fly what you test and test what you fly. However, [Lev01b] noted in the Ariane 5 case that developers experienced all three accident-causal factors and violated the general testing heuristic when they reused the trajectory data of the Ariane 4 in the simulation and specification of the Ariane 5 software, even though the trajectories were different.

[Lev95, HNP97] also cited similar software safety issues as the cause of the Therac-25 computer-controlled radiation therapy machine accidents, which caused massive radiation overdoses in six people between June 1985 and January 1987. [WLL+01] introduced a hierarchical accident model to help identify safety factors considerations with three levels of abstraction: 1) Level I factors include the chain of events; 2) Level II factors included the conditions or lack of conditions allowing the events at Level I to occur; and 3) Level III factors include the root cause or systemic factors of an accident.

Satisfaction, the final [ISO01] quality-in-use characteristic includes the software's ability to satisfy users. There are also a myriad of issues the Department must address to achieve customer satisfaction and confidence including:

- Credibility in Department M&S,
- Improved Department decision-makers confidence in simulation results,
- A quantifiable return-on-investment (ROI) methodology for simulations,
- Additional validated automated processes and tools for models and simulations,
- Simulation utility keeping pace with user requirements and decision-makers expectations [Sch01c],
- Interoperability of legacy M&S systems with war fighting systems [DoD01a].

3. Testing for Quality Attributes

[BM93] identified four distinct periods of software development identifiable by the following dominant themes and goals: 1) the functional era (1960s), 2) the schedule era (1970s), 3) the cost era (1980s), and most recently, 4) the quality era (1990s). Software testing [DPR+94, Roa98, Cla01, PLP01, Rak01] is a critical aspect of quality analysis and is most effective when based on quantitative measures integrated in the development environment, and supporting the test management process. Software testing is a complex undertaking, which addresses many dimensions including new test challenges for components, Java, active agents, object-oriented technology, software reliability engineering [IC99, IC01], website testing, test management, and the human element. Additional software test activities and challenges include the employment of Bayesian graphical models [WGC02], the context of software testing in the development process, the different roles of testers and developers [Whi00], statistical software engineering [JA97], and test case prioritization to increase the rate of fault detection [EMR02].

Software testing is an essential component of simulation software development, a critical technique for evaluating product quality, and essential for improving product quality by identifying defects and problems early in the development process [Som95, Pre97, Roa98, Rak01]. Software companies face major challenges testing products and [Whi00] predict these challenges will continue to grow with the increasing complexity levels of the software. [RTI02] cited improved software testing¹⁵¹ as a critical factor supporting software quality and reducing the software errors, estimated to cost the U.S economy \$59.5 billion annually.

After recent mission failures, NASA developed an independent verification and validation (IV&V) implementation approach [Ros01] for all agency software development with a focus on high-risk cutting edge projects. NASA based the new IV&V approach on nine factors, which NASA determined impacted software development including: software team complexity, contractor support, organization complexity, schedule pressure, process

¹⁵¹ Software testing is the process of applying metrics to determine product quality and is the dynamic execution of software and the comparison of the results against a set of pre-determined criteria. [RTI02]

maturity of software provider, degree of innovation, level of integration, requirements maturity, and software lines of code, evaluated against five risk categories [Ros01].

Software testing is also a best practice process and technique in the Department's developmental test and evaluation procedures according to [SNF+99]. Although they share similar test techniques and methods, [BCC+01b] differentiates software testing from software maintenance, noting that software testing is a component of the development process, whereas, software maintenance addresses system failures after the software is delivered. Software testing may also include static verification techniques. Software testers normally conduct testing at the unit, integration, and system level, applying a number of techniques including black-box and white-box testing, supporting different objectives cited by [Roa98, Rak01]:

- Acceptance / Qualification testing,
- Installation testing,
- Alpha and Beta testing,
- Conformance / Functional / Correctness testing,
- Reliability / Achievement / Evaluation testing,
- Regression testing,
- Performance testing,
- Stress testing,
- Back-to-back testing,
- Recovery testing,
- Configuration testing,
- Usability tests [Roa98, Rak01].

New verification techniques continue to evolve and now include the time warp technique for synchronizing parallel discrete event simulators [FRC+02]; model-based verification [GB99]; model extraction techniques for automated verification [HS02]; assurance-based testing [Pau99]; compositional verification methods [WH02]; and monitors employed as oracles or supervisors to analyze target systems [PP02]. [DMS00b] proposed that component-based M&S methodology articulated by [Hus00] "could allow more effective and affordable M&S verification" [Hus00].

4. Model and Simulation Quality Attributes

The M&S literature contains a wide range of M&S quality dimensions and attributes supporting key M&S credibility concepts, including the terms fidelity and accuracy. In practice today, the Department's M&S community often use the terms fidelity and accuracy synonymously. Unfortunately, the Department's M&S community lacks consensus on a fidelity definition, how to measure it, what it costs, or even its relative importance for achieving confidence and credibility in M&S results. The casual use of other terms closely related to fidelity such as resolution, detail, aggregation, and multi-resolution add to the general confusion limiting the common understanding of fidelity. This adversely affects simulation credibility and confidence in the simulation results.

a. Fidelity

Although fidelity is a critical component for credible M&S, it has proven elusive to implement and resisted multiple theoretical efforts [Bai92, Ham96, GF97, FY97, Pac97, SFL+97, HNP97, Fay98a, Fay98b, GFT98, GZ98, Had98, Har98b, McD98, Pac98a, Pac98b, RGJ98, SBL98, Pea99, Har99b, MBH99, RVJ+99, GJ00, LMR+00, Pac01b, HY01a, HY01b, BP02, RIO02] to describe in objective or quantitative terms. In addition, [HNP97] suggest bounding fidelity, defining perfect fidelity as a simulation indistinguishable from reality, which may be possible for some systems (e.g., virtual reality), although still out of the theoretical realm for non-trivial simulations.

The [DoD95] addressed the issues of fidelity in three of the six major objectives (e.g., objectives two, three, and four). The ability to describe and quantify M&S fidelity or "goodness" appropriately is essential for fully achieving other objectives identified in the master plan, such as HLA (Sub-Objective 1-1), CMMS (Sub-Objective 1-2), and support of endeavors such as Simulation Based Training, Analysis, and Acquisition (Sub-Objective 5-1). Systemic issues confronting improved fidelity include challenges with physical, visual, audio, motion, and temporal, environmental and behavioral fidelity [DoD95].

The [DoD95, DoD98] references contain the Department’s approved definition for fidelity¹⁵². According to the [RPG00], fidelity is a core concept involved in every issue in M&S, especially issues related to V&V. Fidelity is also one of the key areas directly affecting the credibility of the Department’s M&S programs and the confidence in the simulation or federation results provided to decision-makers. Fidelity is inherently an abstraction or a representation of some component of reality, a simuland¹⁵³, or referent developed in M&S. Table 6-2 provides an overview of the fidelity context from the [RPG00].

However, in practice, the M&S community describes fidelity by a number of dimensions and attributes: - accuracy, resolution, aggregation, de-aggregation, detail, extent, granularity, precision, repeatability, time, and spatial consistency. Many of these terms are ill defined and often cited interchangeably in the current literature. [SFL+97] define “fidelity” as a measure of how accurately and realistically a simulator or simulation represents the “real” world. [GFT98] provide a definition for fidelity noting “the extent to which the model reproduces the referent, along one or more aspects of interests” [GFT98]. [RGH00] provide two additional definitions of fidelity with the caveat that fidelity may describe model representations, a simulation, simulation data, or an exercise, with different implications for each use. [Ham96, HNP97] addressed the benefits of model simplification¹⁵⁴ and decomposition¹⁵⁵, without a loss of fidelity, especially of non-trivial systems.

There are many more definitions of fidelity in common use today, exacerbating the issue of defining fidelity. By 1992 there were at least twenty-two different definitions of the term fidelity cited by [LA92] for distributed interactive simulations, further subdivided into twenty different hardware and software components, requiring different levels of fidelity. The heuristic practice for describing fidelity includes short imprecise, subjective, qualitative descriptions for fidelity such as low, medium or high. In addition to these “short descriptions” for fidelity, [RGH00] identifies “shorthand descriptions” such as the Level D flight simulation classification consisting of over 100 attributes used by the

¹⁵² Fidelity is specified as the accuracy of the representation when compared to the real world [DoD95, DoD98]

¹⁵³ The simuland is the representation the simulation is intended to represent, not necessarily the real world, with the referent being the sum total of what is known, assumed, or projected about the simuland [RPG00].

¹⁵⁴ Model simplification includes three classes: brevity, clarity, and efficiency [HNP97].

¹⁵⁵ An alternative strategy to simplification, decomposition breaks the system down into component parts to model [HNP97].

Federal Aviation Administration, and “long descriptions” of multiple attributes involving enumeration, which requires quality attributes such as accuracy, resolution, or both.

[Gro99, RGH00] characterizes past approaches to specifying or measuring fidelity as *ad hoc*, artistic, or problem / domain specific.

[GFT98] propose that fidelity is the key to simulation validation, however, [Gro99] states, “fidelity has to be the least consistently used, yet most commonly used term in the M&S community” [Gro99]. [Mey98] concurs with [Gro99] and considers the term fidelity overloaded, with no context-free meaning, and almost useless in addressing the goodness of a simulation or federation. [Pac01b] adds that no widely accepted set of terms yet exists for the definitions of uncertainty, error, and variability as they apply to M&S fidelity, validity, and their referents.

[GF97] proposed that: 1) fidelity requires different measurements since there are fundamentally different aspects of fidelity; 2) fidelity is a property of simulation models; and 3) fidelity exhibits certain properties. Later, [GFT98, Pac98b] suggested the way to improve decision-makers credibility in the simulation and confidence in the simulation results is to describe fidelity and accuracy quantitatively. However, [Gro99, RGH00] contends there are two major obstacles blocking the development of a fidelity measurement standard: 1) there must exist an accepted definition of the real or imagined world with sufficient quantifiable characteristics to measure the difference between it and the simulation; and 2) the simulation must be similarly defined.

Fidelity according to [Mey98] addresses the M&S measure of agreement with the perceived reality within a specific context and suggests that fidelity is relative to simulation resolution, appropriate for determining differences between the resolution of the simulation and the details and accuracy of the models. A major cooperative government/industry/academia consortium collaborated on DIS interoperability standards for IEEE approval from 1989-1995. The objective of the collaborative initiative P1278.5 entitled *Standard for Distributed Interactive Simulation-Fidelity Description Requirements* [IEE95, STR96] was the interconnection of dissimilar components of DIS to provide real-time applications with appropriate fidelity. Shelved in 1995, the [IEE95] failed to span the

problem space, generated insufficient support competing against the emerging HLA standard, and lacked community support for its approach to fidelity.

The simulation literature has a deep, diverse body of theory on fidelity and its impact on M&S credibility. [Pac98] reviewed basic fidelity concepts, definitions and theory, noting the need to be concerned about M&S fidelity. [Pac98] also provides a synopsis of fidelity ideas and issues providing a framework for dealing with fidelity throughout the M&S development life cycle. In addition, [Pac98] suggests methods to measure, estimate or quantify fidelity including adopting a methodology for deriving and publishing M&S fidelity metrics; economic issues; and other challenges which have plagued analytical approaches, including classical and Bayesian statistics, game and decision theory. However, [Mey98] provides a counterpoint, arguing that quantifying M&S fidelity would be extremely questionable since M&S fidelity has almost no context-free definition. [Hug97b] further proposes that new technologies support 1) increased physical fidelity, 2) improved reasoning fidelity, 3) enhanced behavioral fidelity, 4) better flexibility, 5) improved information transfer and fusion.

The failure to describe M&S fidelity appropriately may adversely influence current and future Department interoperability, transformation, business, and warfighting initiatives. [Pac01b] proposes a referent¹⁵⁶ for fidelity and validity, where the fidelity referent provides the system theory, and conceptually is the most complete collection of information about the subject represented in the simulation, while the validity referent is more complex because it must involve intended use for the simulation. The referent is a standard from which the thing being represented in M&S is derived or the standard against which the correctness of M&S representation is measured [Pac01b]. [Pac00a] also discusses approaches for a conceptual model, supporting enhanced model completeness, consistency, and coherency.

[Had98] considers fidelity as an open issue, with respect to fidelity and resolution, scope, and completeness from the aspect of M&S developer. Moreover, [GF97] and [GFT98] note that M&S fidelity is a critical measure of M&S credibility, and one the largest cost drivers for M&S, although [GFT98] believes that fidelity is currently of little use

¹⁵⁶ A referent is a codified body of knowledge about a thing being simulated [Pac01b].

when engineering the M&S requirements. In addition, [Pac97, Pac98a] notes that there is no specific guidance for fidelity and that the M&S community has not yet agreed upon basic fidelity concepts and definitions. [Pac01b] also provides a broad overview of the issues confronting improved simulation and federation credibility and confidence in the results, including the important topic of quantifying the validity of HLA federates and federations.

In addition, [Had98] describes the problem of defining an appropriate level of fidelity for Conceptual Models of the Mission Space, and notes fidelity is the most difficult characteristic to define because it applies to many aspects of a problem. For example, a simulation may contain a “high fidelity” with respect to a sensor representation, but still labeled as “low fidelity” with respect to other representations such as human behavior or threat. [RGH00] added that fidelity requirements, specifications, and conceptual models have different research approaches although they share many shortcomings, including the lack of a formal and fundamental theory or framework. The [RPG00] guide provides a fidelity conceptual framework in Figure 6-2 with fidelity as a core concept integral to all aspects of M&S, especially VV&A; comprised of attributes such as resolution, error/accuracy, sensitivity¹⁵⁷, and precision; and concluding that qualitative terms inadequately express fidelity.

During the decade of the 1990’s, the Department and M&S community invested significant time, resources and attention to improve M&S verification, validation, and accreditation (VV&A). These VV&A efforts established processes and identified VV&A responsibilities, but have not provided methods for quantifying fidelity requirements or for determining complex M&S accuracy attributes quantitatively. Other groups, especially those involved in the Department of Energy’s Advanced Strategic Computing Initiative (ASCI) and others concerned with M&S of basic physical processes, have started significant efforts to quantify M&S validity [Pac01b]. These groups have begun to address questions of uncertainty and error for both M&S and experimental data used with the M&S as inputs and as standards for validity comparisons.

¹⁵⁷ Sensitivity: The ability of a component, model or simulation to respond to a low level stimulus [RPG00].

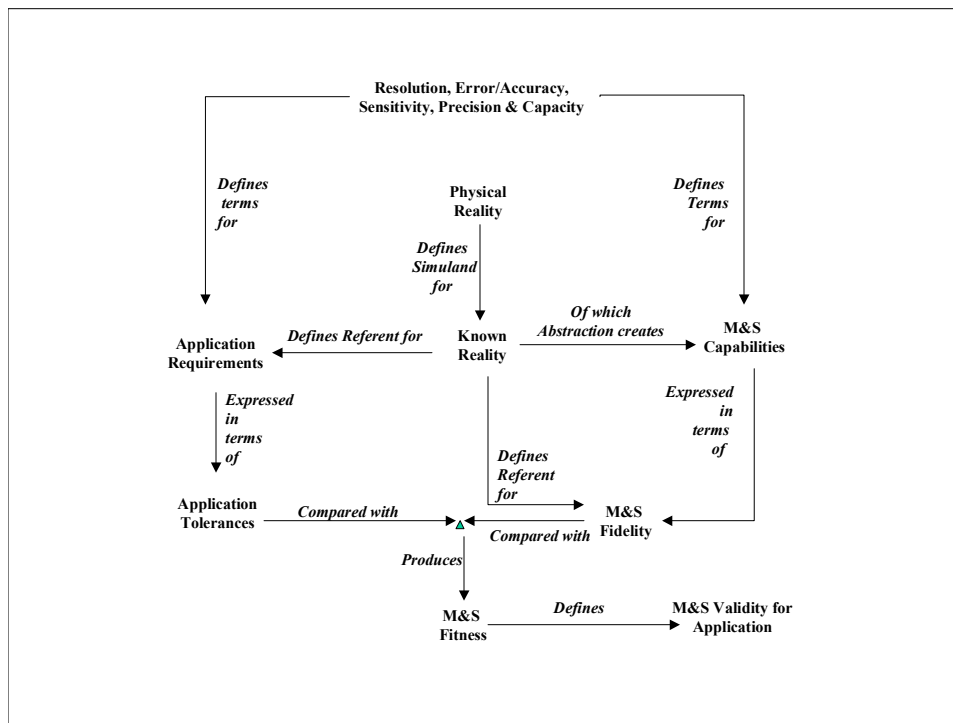


Figure 6-2. Overview of Fidelity Context (From [RPG00])

b. Accuracy

Accuracy, a sub-characteristic of the [ISO01] quality characteristic discussed earlier also plays a major role as an M&S quality attribute. Department M&S users also demand more accurate¹⁵⁸ elements of the mission space and developing authoritative representations remains a major objective of the draft *Department M&S Master Plan*, [DoD01a] currently under development. This has proved challenging to the Department since there are many views of accuracy. For instance, [Mey98] excludes the term simulation in his definition and proposes that accuracy “is a measure of the exactness of the model with respect to the characteristics and behaviors of the physical entity which the model represents” [Mey98].

¹⁵⁸ Accuracy is 1) the degree of exactness of a model or simulation, high accuracy implying low error. Accuracy equates to the quality of the result, and is distinguished from *precision*, which relates to the quality of the operation by which the result is obtained and may be repeated [DoD98], 2) the degree to which a parameter or variable or set of parameters or variables within a model or simulation conform exactly to reality to reality or to some chosen standard or referent (SISO) [RPG00].

Providing another viewpoint, [Bal98] addressed the significant accuracy quality characteristics inherent in verification, validation, testing, accreditation, certification, and credibility assessment activities. In yet another view for the perspective of the data, [Lev00, Kil02] propose:

- That data accuracy requires the appropriate data with the correct resolution for the intended use,
- The proper quality of data established by the data producer and reviewed by the user,
- The correct data transformation activities to make the data compatible with M&S [Lev00, Kil02].

There are also many other views of accuracy. [Pac00a] addresses conceptual model decomposition, and how the characteristics of the simulation elements are abstracted determine the accuracy and precision. [FY97] defines accuracy as being inversely proportional to an error measurement in which more accuracy implies less error. [FY97] also discusses how to determine and trace M&S accuracy through the models and processes associated with each layer, ensuring that the accuracy meets the design requirements, ensuring the M&S operates within the maximum error bounds and thus meet its intended requirements.

[Gro99, RGH00] suggest accuracy is determined by how well the simulation algorithm represents the subject that is simulated, and may be measured against reality, termed real accuracy¹⁵⁹; or against the articulation of the an application domain or mission space; abstraction accuracy¹⁶⁰; and should to relate to a single parameter or set of parameters. With the intent of achieving software engineering consistency, in this research, we employ the IEEE definition for accuracy¹⁶¹ provided in the *HLA Framework and Rules* [IEE00c].

¹⁵⁹ Real accuracy is a function of both the correctness of the abstraction and of the simulation's representation [Gro99, RGH00].

¹⁶⁰ Abstraction accuracy refers to the only the function of the simulation's representation of the abstraction [Gro99, RGH00].

¹⁶¹ Accuracy is the measure of the maximum deviation of an attribute or parameter value in simulation or federation from reality or some other chosen standard or referent [IEE00c].

c. Precision

Precision is also a close relative to accuracy and often used as a synonym in the body of research. [Gro99, RGH00] suggests that precision¹⁶² is a measure of the resolution or granularity with which a parameter may be determined, and always limits accuracy¹⁶³. Precision¹⁶⁴ receives additional clarification in the [RPG00]. Preferring quantitative attributes to qualitative identifiers, the selected IEEE standard glossary of software engineering terms [IEE90] definition for precision¹⁶⁵ suggests improved correctness in verification and validation processes, inferring higher credibility in the simulation and improved confidence in the results.

d. Timeliness

Timeliness¹⁶⁶, a factor in considering simulation fidelity, cautions [Gro99, RGH00] has the potential to create issues for parameter accuracy and precision in the parts of the simulation or federations, which may advance faster or slower than other parts. [Gro99, RGH00] indicates these manifestations may occur in distributed simulations, in unitary simulations employing distributed processing techniques, or in discrete event simulations, and is dependent on time management techniques in the simulation¹⁶⁷.

e. Error

Error¹⁶⁸ is a significant M&S quality attribute. [Ham96, HNP97] note an error occurs when a fault (e.g., a physical defect or flaw occurring in either the hardware or

¹⁶² Precision may normally be determined in a simulation by an understanding of the systems computational processes such as the number of significant digits, round off, interpolation intervals, update/refresh rates, and the quality of the real world data [Gro99, RGH00].

¹⁶³ Accuracy, precision, and timeliness characteristics describe how close the representation of an individual parameter is to reality [Gro99, RGH00].

¹⁶⁴ The SISO further defines *precision* as a 1. The quality or state of being clearly depicted, definite, measured or calculated, 2. A quality associated with the spread of data obtained in repetitions of an experiment as measured by variance, the higher the precision, 3) A measure of how meticulously or rigorously computational processes are described or performed by a model or M&S (SISO) [RPG00].

¹⁶⁵ Precision is the degree of exactness or discrimination with which quantity is stated (e.g., precision of 3 decimal places versus a 6 decimal place precision) [IEE90].

¹⁶⁶ Timeliness is impacted by the maximum magnitude of error between the parameter with and without the missed updates, caused by communications delays or an excessive computation time, supports quantifying the impact of timeliness on a specific parameter [Gro99, RGH00].

¹⁶⁷ Time in a simulation may be managed with various techniques such as continuous time, time step, discrete event, etc [Gro99, RGH00, RGH00].

¹⁶⁸ Error is the difference between a calculated, measured, or observed value and a correct value (SISO) [RPG00].

software) becomes apparent, adding that if the error causes the system to perform a function incorrectly, a failure occurs. According to [Gro99, RGH00], any review of simulation fidelity involves factoring in error. Possible M&S error sources include imperfect measurement of input data, deviation from correct data, imperfect algorithms, and the finite limitations of logical and computational processes.

A report on software error analysis [PW93] sponsored by the NIST found a wide variation of error techniques for high integrity software and proposed the development of an organizational error analysis database identifying the most effective techniques to support error detection. Moreover, the impact of error in software-intensive systems, including simulation software is staggering. Of the \$59.5 billion lost annually to the U.S. economy do to prevalent software errors; half of the cost is borne by the developers/vendors, with the other half of the costs paid by the users [RTI02]. Over half of these errors were uncovered late in the development process or while in use after the sale [RTI02].

f. Resolution

A significant M&S community issue includes the development of community standards defining resolution of model representations for use throughout the life cycle of the systems. [DoD98] defines resolution¹⁶⁹, while [RGH00] view resolution as “the extent to which the M&S models each aspect of the real world”. A definition in the web-based [RPG00] expanded the meaning of resolution¹⁷⁰. The [RPG00] definition also addressed granularity or the reduction of something into related parts or components. [HNP97] propose that the question of detail affects simulation resolution, with both factors related to fidelity. [Mey98] finds resolution erroneously used to describe the result of some measurement, rather than correctly identifying the resolution of the device taking the measurement (e.g., a sensor).

[Mey98] also proposes that M&S resolution like fidelity, derives from the context of a simulation, and is a measure of the minimum degree that the constituent mod-

¹⁶⁹ Resolution is the degree of detail and precision used in the real world aspects in a model or simulation [DoD98].

¹⁷⁰ The [RPG00] defines resolution as 1) the degree of detail used to represent aspects of the real world, or a specified standard or referent by a model or simulation, 2) separation or reduction of something into its constituent parts; granularity (SISO) [RPG00].

els' accuracy and detail correspond. The semantic relationship between the terms presented in Figure 6-2 delineates the delta between the fidelity required by the application or tolerances¹⁷¹ that define the range of values for dependent and independent variables, and the fidelity present in a model in a simulation, the knowable quantity. Figure 6-2 also illustrates that physical reality, either material or imagined, provides the basis for obtaining all knowledge of reality.

Known reality manifests this body of knowledge. Known reality also provides the source for referents supporting application requirements and model or simulation fidelity, and for abstractions of reality, in models and simulations. [HNP97] suggest that model resolution, a critical component when determining the usefulness of a simulation is conceptually closely related to fidelity, however, caution that one does not imply the other, and higher resolution does not necessarily increase fidelity. However, since resolution affects federation credibility supporting confidence in the results, the [IEE00c] definition for resolution¹⁷² is the most appropriate.

g. Detail

Simulation detail, like resolution, also includes many facets and views. [RR83] state that the model level of detail describes the number of functions, or level of structure included in the model, while the extent, a similar concept, relates to the range of system functions within the simulation. However, [Mey98] defines detail as a measure of model complexity and completeness when compared to the characteristics of the physical entity represented by the model.

[Fay98a, Fay98b] proposes a method for measuring the level of detail in a simulation based on identifying the important parameters and a sensitivity analysis approach, which provides a categorized hierarchy of phenomena, model, and value parameters. [Had98] further suggests that the sophistication of the developers in the problem do-

¹⁷¹ Tolerance: 1. The maximum permissible error or the difference between the maximum and minimum allowable values in the properties of any component, device, model, simulation, or system relative to a standard or referent. Tolerance may be expressed as a percent of nominal value, plus and minus so many units of a measurement, or parts per million. 2. The character, state or quality of not interfering with some thing or action [RPG00].

¹⁷² The IEEE HLA definition of resolution describes it as the smallest resolvable value separating attributes or parameter values that can be discriminated. Resolution may vary with magnitude for certain data types [IEE00c].

main factors significantly in the development of the appropriate level of detail in the application.

h. Aggregation / Disaggregation

Aggregating and disaggregating model representations present significant challenges to the development of credible simulations [HNP97]. The many complex issues involving aggregation¹⁷³ / disaggregation¹⁷⁴ complicate the development of improved M&S credibility. The [DoD02d] notes the lack of predominate type of M&S and identifies the need for varying levels of aggregation and flexible interoperability standards to meet diverse customer needs. Seeking more definitive answers, [HJ95] explored several different approaches to aggregation, and established requirements to develop consistency between aggregated and higher fidelity representations, concluding that common intuition about cause, effects and outcomes are often incorrect.

[Bid00] addressed additional areas for representing battlespace objects as elements of other aggregate element representations, including mechanisms for establishing aggregated representations on a permanent or temporary basis¹⁷⁵. Taking an object oriented approach, [HHP00] proposed a process transitioning aggregated / decomposed models into an object-oriented representation by defining modules in an object oriented domain. The object oriented methodology supports other possible structures proposed by [Tra93, CRB01, CM02].

The Department also requires credible, authoritative representations including aggregated and disaggregated system representations of U.S., allied, friendly, paramilitary, coalition, neutral, threat, systems, and for combat, combat support and combat service support systems and processes. This includes Department objectives for establishing standard taxonomies and common object classes for systems representations by FY2004 [DoD01a]. Two basic approaches have evolved for dealing with resolution and fidelity issues involved in aggregation / disaggregation: the top-down and the bottom-up approaches.

¹⁷³ Aggregation is defined as the ability to group entities while preserving the effects of entity behavior and interaction while grouped [DoD98].

¹⁷⁴ Disaggregation refers to the ability to represent the behavior of the aggregated unit in terms of its component entities [DoD98].

¹⁷⁵ This scheme deals with situations where an object needs to be represented both individually and as part of an aggregation at the same time. [Bid00]

Other resolution issues involve the development of acceptable algorithms for aggregating representations of a single system into groups of entities and disaggregation of grouped representations [You02a, You02b].

5. Model and Simulation Quality Approaches

Department- and National-level decision-maker and stakeholder knowledge and perceptions of M&S quality directly impacts credibility in the simulation or federation, and confidence in the results. [Wal94] described five approaches for viewing and defining quality based on the perspectives of different stakeholders — transcendental, user-related, buyer-related, production-related, and process-related. Today, the Department employs the “transcendental” approach to M&S software development, attempting to maintain high quality without precisely defining or measuring the product. In lieu of precise measurements in M&S software, the M&S community labels model representations with subjective, qualitative identifiers such as high, medium, or low fidelity.

The user-related approach for user satisfaction described by [Wal94] for the Department’s software intensive, simulation development best fits the current verification, validation, and accreditation processes. The Department’s acquisition process most appropriately instantiates the buyer-related viewpoints in the cost, schedule, and performance parameters of the acquisition process. Metrics, formal methods, and statistics represent the Department’s production-related approach, seeking to attain high quality by defining specific characteristics of a product with precise measurements, while the many process models (e.g. CMM, CMMI) provide the Department with process-related approach to prevents faults and minimizes rework.

The production-related approach has the potential to improve confidence in large scale Department M&S software systems. [Boo02] identifies the “use of certain reasonable and quantifiable measures of product and process” [Boo02] to address the maturity of software engineering team that are indicative of the product quality. This dissertation identifies a new architecture-centered, product line-based approach to replace the existing transcendental approach to the Department’s M&S software development and acquisition, and addresses the processes for a successful implementation.

D. SUMMARY

Chapter VI addressed software and simulation quality attributes affecting heterogeneous system representation anomalies and credibility in Department M&S, especially in federation interoperability. The chapter reviewed the internal and external attributes of the ISO 9126-1 Quality model [ISO01], testing for quality attributes, and M&S quality attributes addressed in much of the current literature, including an overview of aggregation and disaggregation. The chapter included a discussion on M&S quality approaches.

Quality is a major component of M&S credibility. The ISO Quality Model addresses the six major internal / external quality characteristics: functionality, reliability, usability, efficiency, maintainability, and portability; and the four qualities in use characteristics, allowing users to achieve specified M&S objectives. The term “software quality” remains widely used today, and open to subjectivity, different views and various interpretations of the definitions. Software quality continues to lag well behind the computer industry’s hardware engineering segment progress on quality, cost, and performance. While computer hardware engineering is well into the fourth computer era identified by [Pre97] and employs the *production* approach to quality according to criteria established by [Wal94], simulation software quality is best labeled as *transcendental*. The chapter also reviewed model and simulation quality attributes including fidelity, accuracy, resolution, error and detail, within a fidelity context [RPG00].

This research found little evidence that the M&S software development community will achieve any consistent future success: 1) developing quality metrics for M&S quality attributes, and 2) applying them early enough in the M&S software analysis and design phases to quantifiably improve M&S software quality employing current paradigms. In addition, many of the past recommendations to improve the Department’s M&S portfolio listed below have been reiterated and reissued on repeated occasions, although it is uncertain if more time and more resources will improve the results:

- Better verification and validation (V&V) techniques and accreditation processes,
- Better discipline in the software development process,
- Improved use of data,
- Improved quality,
- Improved documentation.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. SOFTWARE ENGINEERING EFFORTS SUPPORTING CREDIBILITY

A. INTRODUCTION

Chapter VII reviews three selected areas influencing credibility of large-scale, legacy simulations in the Department including software process improvement (SPI), the status of new major simulation software engineering initiatives (e.g., JWARS, JSIMS, JMASS) and the growing software engineering body of knowledge. The chapter also includes a discussion of the challenges generated by Department contract management oversight for achieving quality software and reducing heterogeneous system representation anomalies, Department institutional factors affecting M&S credibility, and recent Department software engineering education initiatives. In addition, research reveals a growing awareness that the Department needs qualified software engineers who understand the foundations of the software-intensive systems, including credible M&S, upon which we basing our future security, economic prosperity, and military preparedness.

B. PROCESS IMPROVEMENT SUPPORTING M&S DEVELOPMENT

Product-based activities evolved with the early software development industry, supported by artisans practicing software development as a craft, a dependence on heroes, and a largely anecdotal body of knowledge. Since the mid-1970s a view of building software as a process-based activity grew and began to challenge the traditional model of building software as product-based activity. This process-based activity concept drew heavily from the manufacturing sector with a theory that how the developer built the software affected the quality of the end product. Process-based activities now include a rapidly expanding field including process models, quality standards, and contractor evaluation criteria, including international process improvement, quality standards, and Department-based standards.

According to [SG99] American corporate project managers for information technology realized by 1996 the true cost, scope, and size of software project failures and finally acknowledged that no technological or silver bullet solution existed. The problem and solution identified by [SG99] included people and processes. [Bro96] cited the need

for fundamental process improvement in industry/government software development organizations driven by systemic failures to predict and meet cost and schedule targets, a need to reduce life cycle software costs, and an improved ability to identify and address technical risks.

Process-based activities including process improvement is multifaceted and incorporates continuous improvement process (e.g., Kaizen) championed by Deming [Sch91], business process reengineering concepts [HC93], statistical process control for software improvement [FC99, Wel01], and the process enterprise [Ham01b]. [BE99] addressed methods for improving the process and product by optimizing the software product throughout the life cycle, identifying follow-on efforts including product lines for further exploration. Process-based activities also entail event-driven learning approaches for training [Hil01], and project management processes [Fra94], which provide additional perspectives on the various dimensions of effective processes. [She97] noted the large set of process frameworks caused confusion and identified six specific compliance framework categories:

- Standards and Guidelines,
- Process Improvement Models and Internal Appraisal Methods,
- Contractor Selection Vehicles,
- Quality Awards,
- Software Engineering Life-Cycle Models,
- System Engineering Models [She97].

Standards and guidelines included U.S. military standards (e.g. MIL-STD 498), commercial standards (e.g., Electronic Industries Association (EIA) IS 632), and International Standards Organization (ISO) standards including the ISO 9000 series for quality systems and ISO Software Process Improvement Capability dEtermination (SPICE), an international standard for software process assessment. Process-based activities generally defined the characteristics of good processes, and did not prescribe the how to implement the processes. The 1991 introduction of the software Capability Maturity Model (CMM) for software (SW-CMM) [PWC+95] built on product quality principles promulgated since the industrial age production line practices of the 1930s. The objective of the CMM is to achieve defined, managed, and optimized processes, with the expectation that improved

process maturity [MCR+00, BBF+01] will also yield a higher quality software product [BE99].

Follow-on process improvement/maturity products including [Hum98a, Hum98b, Hum98c] based on the same underlying CMM appeared for other closely related disciplines including Software Acquisition [IEE98c] CMM [FCF+96, FDR+98, Gal99, SPM99, GF00], Systems Engineering [HO97, Hum98a, Hum98b, Hum98c, Hum99, MC01a], Integrated Product and Process Development [Zit97], People CMM [HB98c], the Personal (PSP) / Team Software Process (TSP) [HO97, Hum99, VFS+99, MC01a] and most recently, the Capability Maturity Model Integration (CMMI). In a similar initiative, [MCR+00] developed a five-stage maturity process similar to the CMM for U.S. Government investment management of information technology. [WPL02] addressed the possibility of improving M&S credibility, comparing and contrasting CMM key practice areas with several common V&V methods, while [Ric02] discussed the possibility of the a M&S CMMI to better identify mature companies to develop and maintain M&S products.

The Capability Maturity Model for Software (SW-CMM) [PWC+95] supports organizational development of improved, common, software processes based on a framework of best practices, which impact productivity, performance, cost and customer satisfaction. There are five CMM maturity levels 1-Initial, 2-Repeatable, 3-Defined, 4-Managed, and 5-Optimizing, supported by specific key processes areas (KPA) established at each maturity level. For instance, KPAs for level 2-Repeatable include requirements management, software project planning, software project tracking and oversight, software subcontract management, software configuration management, and software quality assurance.

Acquisition agencies may specify a SEI Software Capability Evaluation (SCE)¹⁷⁶ method or Software Development Capability Evaluation (SDCE) method employed by the Air Force, to have a third-party examiner review a competitors strengths and weaknesses in order to reduce risk to the acquiring agency. TG99] compared and contrasted both evaluation techniques noting both methods were time and resource-intensive for both the contractor and the government. In addition, [OS00] identified challenges and inconsistencies in the SCE process and suggested improvements to restore validity to the process. Confi-

¹⁷⁶ Software capability evaluations (SCE) are formal, systemic methods for assessing a contractor's software development process

dence in the evaluation process is critical since the most critical Department acquisition projects, termed ACAT 1 programs, must undergo an SCE or develop a risk mitigation plan [Gan99].

The debate over the value of software process improvement continues today. Although [Fal01] cites a draft document defining the requirements for process evaluation methods to improve process definition, [Mos02] contends that quality of software process improvement varies significantly within the Department today, and exhibits less disciplined processes when compared with the early 1990s. Concerned with the Department's software acquisition practices, [AS03] implements Section 804 of the Bob Stump National Defense Authorization Act for Fiscal Year 2003 requiring the establishment of software acquisition process improvement programs by Department agencies with a substantial software component, supporting a Department objective to improve the acquisition of software-intensive systems.

In an effort to improve quality, quality award initiatives evolved based on the premise that the quality of the processes used to develop and maintain a software system significantly influenced overall quality. Several organizations developed award programs to improve the quality situation, including the Malcolm Baldrige National Quality Award, established by the U.S. Government in 1987, and the European Quality Award [She97]. The apparent proliferation of these models had several consequences [Cla97], which resulted in the initiation of the CMM Integration (CMMI) project with staged representations [FAB+99a, FAB+99b], and continuous representations [FAB+99c, FAB+99d, and Ric02]. The development of the Capability Maturity Model-Integrated-Systems/Software Engineering (CMMI-SE/SW) built on three initial disciplines: software engineering, systems engineering, and integrated product and process development.

Organizations with software process improvement programs have reported benefits in product quality and productivity, corroborating increased product quality and decreased product costs results experienced in other sectors [BBF+01]. [Whi99] addressed the importance of VV&A for quality software engineering, and illustrated where VV&A processes integrated into the development process improved cost, performance and schedule. [Bal98,

PW93] also noted the importance of establishing an M&S software quality assurance program, such as the program detailed by [STS00].

Currently, the software acquisition management (SAM) model remains a separate model awaiting integration into the current CMMI framework. [San99] identified improved process and product discipline supporting improved software engineering practices [DL93, Pfl98]. Advocating continued professional development, [Whi99] endorses continuing training opportunities, and provides an outline for training M&S practitioners. Achieving these objectives requires a disciplined software engineering approach, mature software acquisition management processes [IEE98c, SPM99, SSJ+00], the capability to evolve and improve the current Department portfolio of legacy M&S, and maybe most importantly, management commitment supporting transformation and reengineering initiatives.

Process maturity closely correlates to improved system and software engineering practices and supports the management of large software projects with defined consistency, and process repeatability. [McG01] discusses the primary benefits and business case advantages for SPI from a properly run SPI program including improved cycle time, reduced software development costs, improved product quality, reduced maintenance costs, improved worker morale, and increased business competitiveness. [McG01] also identified possible impacts on secondary business case metrics from improved SPI practices including projected sales, average historical penalties (e.g., contract work), personnel turnover costs, employee development costs, repeat business, and a final category, the risk likelihood with and without software process improvement.

Although process improvement results are not yet conclusive, there are positive indications [McG01]. In dealing with projecting transitions to various maturity levels a linear correlation was established between the Quantitative Software Management¹⁷⁷ (QSM) quantitative Productivity Index and the SEI CMM [Put92]. The [Put92] analysis concluded that the number of companies in the two highest maturity levels will be quite small for the near term period, unless a dramatic change in the rate of improvement occurs. In a follow-

¹⁷⁷ QSM has been in business capturing management numbers for software development activities since 1978 and has a very large database populated with the complete spectrum of performers from poor to very good.

on paper [Put93] developed the expected economic benefits of moving up the SEI scale. In later research, [LFT95] studied the correlation between the CMM and software developer performance, citing improved cost and schedule performance with increasing process maturity and validating a correlation between project success and CMM ratings, suggesting that CMM ratings may be indicative of future success.

Continuing research by [Cla97] involving one hundred and twelve projects in the sample concluded in part that software process maturity was a significant factor affecting software development effort, and furthermore, that a one increment change in the process maturity rating, after normalizing, resulted in a “15 to 21% reduction in effort” [Cla97]. [Cla97] provides a significant assessment of the effects of process on software development efforts, and concluded the CMM is well defined and establishes criteria to evaluate processes, and proposes that process maturity should be a factor in all cost models.

The SEI compiled a comparison of process maturity profiles by a subset of the software community from 1996 [SEM97] through 1999 [SEM00] revealing:

- The number of organizations initiating software process improvement continues to increase [SEM97, SEM00],
- The proportion of commercial and in-house organizations initiating software process improvement are increasing [SEM97, SEM00],
- Manufacturing industries are conducting the most software process assessments [SEM97, SEM00],
- The service industry now conducts almost as many software process assessments as the manufacturing industries [SEM00],
- Nearly half of the reporting agencies have less than 100 software personnel [SEM97, SEM00],
- The overall community profile continues to shift towards higher maturity [SEM97, SEM00],
- The trend towards higher maturity profile for offshore organizations compared to U.S. organizations continues [SEM97, SEM00],
- Software Quality Assurance is the least frequently satisfied key process area (KPA) among organizations assessed at level 1 [SEM97, SEM00],
- Integrated software management, organization process definition, and training program are the least frequently satisfied level 3 KPAs among organizations assessed at level 2 [SEM97, SEM00],
- Higher process maturity has been reached among those organizations reporting re-assessments [SEM97, SEM00],

- All groupings exhibit a similar pattern for moving from maturity level 1 to level 2 and from level 2 to level 3. Furthermore, level 2 to 3 tends to be faster and have less variance [SEM97],
- For organizations that began their CMM-based SPI effort in 1992 or later, the median time to move from:
 - Maturity level 1 to level 2 is 25 months,
 - Maturity level 2 to level 3 is 23 months,
 - Maturity level 3 to level 4 is 36 months [SEM00].

[PGW00] notes the seventy-one level 4 or level 5 appraised organizations as of February 15, 2000, continued to grow since 1992 when there were no level 4 appraisals and only the IBM Onboard Shuttle team rated level 5 using the SCE method.

The Department's draft M&S strategic plan [DoD03a] supports accelerated organizational, operational, business, and process reforms. Current challenges to software process improvement implementation include the costs to implement a SPI program, ensuring development team follows processes, establishing process metrics, and maintaining process consistency. Common organization barriers to a SPI program include: a lack of resources, inadequate sizing of the SPI effort, lack of senior management sponsorship, middle management apathy, and staff tension [SEI95]. Building on the issues behind poor software quality identified by [Dem95a] and [You93], [Eva00] suggest that it is not only an issue of improving the software processes, but also changing the organization or project cultures into an atmosphere supportive of process improvement.

In the late 1990s, a countering effort, the Lightweight / Agile process movement (e.g., Extreme programming (XP), Adaptive software development) evolved in a countering effort to the plethora of process models which restricted creativity, according to the Agile process school of thought. The XP School reduces software development processes to the bare minimum [UHC02].

C. CONTRACT MANAGEMENT

Contract management oversight [MBG+02] is a critical process for an organization acquiring software. The challenges of managing the myriad of contracts supporting the Department's IT initiatives, including M&S software support continue to grow. Federal Government spending on IT services almost doubled from fiscal years 1997 to 2001, in-

creasing from \$9 billion to \$17 billion, with the Department of Defense remaining the largest single acquirer of IT services and increasing spending by about forty-one percent throughout the period [KSZ03]. During the same period spending on IT services from the General Service Administration (GSA) greatly increased with spending on the GSA federal supply service schedule growing from approximately \$405 million to \$4.3 billion [KSZ03].

[DoD03a] continues to support the objective of increased information sharing and collaboration within the Government (e.g., federal, state, local), academia, and industry. However, the inability of the government to establish M&S as contract deliverables, the absence of M&S requirements in many contract proposal plans, and the growing number of long-term contracts reduce the potential gains [HBD+01]. Another area of concern involves the possible ineffectiveness of the SCE process [FS94] for contractors identified by [Mos02]. In addition, [HBD+01] addressed several other systemic government-contractor issues impeding progress, also noted by other authors as indicated:

- Collaborative environments misunderstood,
- Contractual obligation constraints or restrictions [KM00],
- Only fifty percent of the reviewed programs identified M&S in the prime contractor's statement of work,
- Contractor ownership of reviewed M&S was nearly seventy-five percent,
- Issues with proprietary M&S [Sul02], and a corollary challenge—M&S products (e.g., conceptual models) as a contract deliverables [Pac02d],
- Lack of management visibility into programs,
- The lack of work breakdown structures [HBD+01].

D. SOFTWARE ENGINEERING OF MODERN DOD M&S

[Ham96, and HNP97] note that software engineering discipline overlaps and intertwines with simulation implementation and the system modeling process, which supports the simulation design. While reduced costs and improved performance have marked the evolution of hardware and network products in accordance with marketplace competition and Moore's Law [Moo65], a recent study commissioned by the Department of Commerce's National Institute of Standards and Technology [RTI02], concluded that software errors are prevalent and costs the U.S. economy an estimated \$59.5 billion dollars annually, a detrimental figure which equates to about 0.6 percent of the annual gross domestic product.

This is a systemic Department issue. The Department created several organizations and task forces over the past twenty-five years to evaluate methods for improving software quality and reliability, identify better software management processes, establish sound software engineering practices, reduce software development cost, and improve performance and schedule metrics [CHK90]:

- In 1979 a Joint Logistic Commanders Joint Policy Group on Computer Resources Management, concerned with adequate software support after the development period, evaluated post-deployment software support (PDSS) procedures supporting the transition and operational phases of the software system life cycle. The Department historically experiences the highest life cycle costs during the PDSS of a software system,
- The Very-High Speed Integrated Circuits (VHSIC) Program created by the Department in 1980 supported improved delivery time and performance of military integrated circuits with the development and insertion of military-qualified VHSIC chips into weapon systems. This early approach to a military hardware problem employed many of the product line principals explained later in the research and applied to the software challenge to fully use the hardware capabilities,
- The Software Technology for Adaptable Reliable Systems (STARS) established in 1983 by the Department investigated methods to reduce software development costs, increase software system reliability, improve software automation techniques, and identify the advantages of software reuse,
- The Department contracted with the SEI, located at Carnegie-Mellon University, in 1984 to investigate new software technology, analyze software development environments, and provide software and system engineering education [BBG+03],
- Finally, several Defense Science Boards task forces and summer studies since 1980 reviewed multiple defense programs and recommended changes to the Department's attitudes, policies, and practices regarding software development and acquisition [CHK90].

Today, many of the Department's software-intensive systems lack quality [HNC+00], while software costs identified by [Coh94, SG94, SG99, RJI02] continue to increase. The persistent systemic issues of late software deliveries; coupled with poor-quality continues to grow [Bow02, Mic02a, Mic02b]; and stymie the demand for reliable software systems. [KM01a] confirmed a causal link between simulation credibility and the quality of the software engineering processes and practices involved in simulation software development and maintenance practices identified in many of the earlier studies and reports. Achieving simulation credibility and confidence in M&S results also depends on the

quality and interoperability in the underlying software, hardware, networks and software development processes and products.

A preponderance of research indicates software engineering maturity lacks discipline and lags significantly behind other engineering disciplines. Our research suggests that simulation software development may be more complex than the development of other types of software, since a model is an “abstraction¹⁷⁸ or approximate representation of something” [GMS+96], where no model is ever completely representative [BCN96], and [Sha75] contends no model is absolutely correct. Disciplined development and the M&S verification and validation process is critical to the establishment of M&S software credibility and the establishment of confidence in the accreditation decision and the simulation results. To date, the Department has not achieved any long-lasting Department-wide quantifiable results indicating improved simulation credibility.

Research suggests that quality simulation software, based on a disciplined software engineering foundation [Sha93, VFS+99] supports simulation credibility and the maintenance of user confidence in the results. However, documented systemic simulation software engineering¹⁷⁹ issues [Arm64, PAD78, Che86a, HW97, GF00, Cra01a, Rie01] since the mid-1960s included the lack of adequate software requirements, software configuration management, verification and validation processes, software engineering processes, and software quality methods. Summarizing the simulation software portion of the Department’s software Tower of Babel, [BTE+93] stated,

Software will continue to be a problem because it is here that all of system complexity hides. The software problem will continue to be the problem of exactly what does the system do [BTE+93].

Simulation software development also shares all the same cost, performance, schedule, and quality software development challenges [Gib94, Coh94, SG94, SG96c, SG99, NJL00, BW01] found in other domains, and adds the complexity of developing and validating conceptual models with credible model representations of reality to achieve con-

¹⁷⁸ Abstraction is the selective emphasis on detail: specific details are suppressed and those pertinent to the problem are emphasized. Abstraction mechanisms focus on high-level aspects of an entity while concealing details. Three commonly used abstraction mechanisms are classification, aggregation, and generalization [Sow88].

¹⁷⁹ Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is the application of engineering to software [BCC+01b].

fidence in M&S results. In response to these issues, the Department initiated three large, joint software-intensive simulation development projects during the 1990's: Joint Modeling and Simulation System (JMASS), Joint Simulation System (JSIMS), and Joint Warfare System (JWARS). The Department approved, funded and initiated the three systems to replace Services and Agencies legacy M&S, with planned objectives of meeting software development cost, performance, and schedule milestones, improving system interoperability, reducing life-cycle maintenance costs, and conforming to the new HLA standards.

The Joint Modeling and Simulation System (JMASS) [Bro92, Mey01] started in 1990. JMASS is unlike other Department simulation software development programs, which plan to deliver a complete, integrated simulation. The JMASS program provides architectural components consisting mostly of the simulation engine and services, interface standards, and a set of GUI-based tools to build models to integrate the models with other architectural components and develop a simulation [Mey01]. The major deliverables expected from the JMASS effort include common digital simulation architecture; definition of standard interfaces; application of commercial standards; a CASE tool environment and M&S support for the entire acquisition lifecycle.

The Joint Simulation System (JSIMS) [Mar97], conceived as the primary Department simulation tool supporting future joint-and Service-based training, education, and mission rehearsal roles and functions, began with an approved July 1994 mission need statement. The JSIMS design envisioned a single, distributed, integrated simulation environment supported by a core infrastructure and mission space objects maintained in a common repository. However, by June 1999, [Ett99] declared that JSIMS had,

“Serious funding, technical, and management problems...JSIMS remains a troubled program¹⁸⁰”, with major architecture issues, and an inadequate management structure, requiring a major new baseline of the acquisition program, increased Department management oversight required, and the need for continued Department program support” [Ett99].

The Initial Operating Capability (IOC) for JSIMS, originally scheduled for December 1999, experienced several delays with the last JSIMS IOC scheduled for March 2003.

¹⁸⁰ The JSIMS PEO declared that (1) the current program was unexecutable; (2) the acquisition program baseline was breached and must be rebaselined; and, (3) the IOC for JSIMS would slip at least 11 months [Ett99]

The life-cycle program funding profile projects \$1.6 billion for the period of 1996 through 2007 [UMS+01a]. The JSIMS future is unclear according to [UMS+01b] citing a lack of assurance that the Department JSIMS Milestone Decision Authority would be able to make future informed investment decisions on JSIMS, based on reoccurring historical problems including multiple schedule delays, performance issues, and increased costs.

The Department approved a third major simulation development effort, the Joint Warfare System (JWARS) [Met00, Man01], in May 1995 to develop a state-of-the-art, multi-sided simulation of joint, campaign level warfare representing joint functions, processes, doctrine, and component warfare operations of future warfare to support analysis. The JWARS high-level design objectives included plans to represent future warfare capabilities supporting force structure decisions, course of action analysis for the Combatant Commanders, and the development of operational plans from a joint operation perspective.

The JWARS system objectives also included the need to provide a better assessment of C4ISR contributions, weapon systems, joint requirements and the analysis of mobility, logistic and contingency operations impacts on warfighting capability, and support for the analysis of investment alternatives. When software development began in April 1997, the JWARS program did not require a formal conceptual model product, and subsequently the V&V agent created a virtual conceptual model from available development artifacts [Met00]. Initial V&V efforts to include algorithm validation were unsuccessful according to [Met00]. The lack of a simulation conceptual model has been a recurring systemic issue with Department simulation software development, a major reason for the current lack of simulation and federation credibility and general skepticism of the Department's simulation results.

The three major Department simulation software engineering efforts: JMASS, JSIMS, and JWARS are all over cost, behind schedule, and have yet to demonstrate they can achieve performance objectives. Partly based on these experiences, the Department implemented a new approach for the Joint Distributed Engineering Plant (JDEP) initiative to improve System-of-System interoperability: large-scale reuse [Dah01a, Dah01b, DT01a, DT01b, Rad02]. [Wol01] established the Department policy for mission critical C2 interoperability for the Joint Task Force and below, including the JDEP. Rather than build the

JDEP from scratch, the Department is making use of large-grain reuse [CWS02], pulling the necessary assets from across the Department's enterprise to support integration, interoperability, and meet information assurance objectives.

Software engineering and the V&V processes support improved credibility and confidence. [KM01a] identified the link between simulation credibility and software engineering processes and good model management practices. The Department's verification process also "evaluates the extent to which the model or simulation has been developed using sound and established software engineering techniques" [DoD98, DoD95, AR97]. Standard software development processes and software engineering methodologies have been used to develop Department simulation software, including the incremental, prototype, evolutionary, spiral [Boe93, Boe00], and re-engineering methods [RPG00]. However, Department software development processes for the most part, lack the discipline of the more mature physical engineering fields [DoD01a], and also lack consistent enforcement mechanisms [HNC+00]. This statement may be significant since troubled or failed software projects, industry-wide, are still the norm, and the costs for failure are staggering.

Software, as a product and the technology for delivering a product, has made major contributions to the twenty-first century world, but major systemic challenges continue to persist according to [Rei93a, Rei93b], and many of these problems may cause unintended consequences. While there has been marked progress in many areas,

Many prior deficiencies continue to shape DoD M&S because they require long-term solutions and competing priorities that have slowed progress [DoD01a].

[Kil02] cites the lack of cooperation from M&S developers as the "single greatest obstacle to developing a meaningful assessment of the credibility and fitness¹⁸¹ of simulations" [Kil02]. As a result, [Pac01b] considers the state of current practices for simulation software development, associated M&S processes, and product management inadequate. Although [Pre97] suggests we have a chronic affliction as opposed to a software crisis¹⁸²,

¹⁸¹ Fitness: Providing the capabilities needed or being suitable for some purpose, function, situation or application [RPG00].

¹⁸² The term "software crisis" was identified by the DoD in the mid-1970s [SCC+98] and has been associated with the complex problems associated with producing software that was developed on time, within budget, operated properly, and was maintainable over its life cycle. More recently its was identified by [Gib94] as the need for a mature software engineering discipline supporting an information-age society, that he believes remains years, maybe decades away from achieving, even after fifty years of evolution [Gib94].

he cites an aging software plant and several unresolved systemic software-related problems including:

- Hardware advances continue to outpace software capabilities,
- Software has not kept up with the demand for new or replacement programs,
- Society is now dependent on the reliable operation of software as witnessed by the Year 2000 problem,
- High software reliability and quality have not been achieved,
- Poor design and inadequate resources [Pre97].

The software industry is also growing ever more important to the twenty-first century society. The demand for new software has grown so rapidly that the demand far exceeds the supply and the following conditions prevail according to [JBC+99]:

- Software is fragile, unreliable, difficult and labor-intensive to develop, test and evolve,
- The Nation's ability to produce software does not meet the demand,
- The Nation depends on fragile, inadequate, software products developed with immature processes,
- Current technologies to support reliable and secure software are inadequate,
- Software continues to grow more sophisticated and diverse,
- Software is replacing manual processes in many activities previously exercised manually by individuals,
- The Nation needs to invest more in software research [JBC+99].

Historically, the software engineering community viewed software engineering processes as discrete activities. However, while there are discrete activities including entrance and exit criteria, design, coding, and testing, most of the development process progresses in a continuous fashion [STS99]. The continuous software engineering activities [STS99] include configuration management, quality assurance, verification and validation, maintenance planning, risk analysis and management, and software engineering project management.

Software requirements for models, simulations, and federations include three major categories: the problem domain, the user domain, and the situation domain [RPG00]. The simulation software requirements process articulated by [BL91, Som95, SG96a, SG96b, BIL97, Pre97, BW97, CS99, HHP00, KG00, MWC00, RPG00, HL01] at the most basic level are properties, exhibited by a new or adapted M&S system to solve a particular prob-

lem. They may have different properties such as process, or system constraint, and product parameters, which include functional (capability) and non-functional requirements such as constraints and quality requirements. For instance, [Som95] suggests a reliability requirement places constraints on the system architecture.

Emerging requirements may be dependent upon the system interoperability, as opposed to a single feature or component, and critically dependent on system architecture decisions. Requirements elicitation is seldom easy and requires familiarity with many techniques and an understanding of the impact of social, political, or economic factors on the different stakeholders' view of the requirements. Many times requirements understanding continues to evolve during the design and development phases. The requirements then undergo analysis to understand the system components and how they interact with other components, establish baselines and prioritize the requirements, and finally develop the estimated cost. The root cause in the Ariane 5 disastrous failure was related to a breakdown in the requirements process [STS99].

Furthermore, software engineering processes, development and maintenance practices and procedures may provide significant insights into M&S credibility [KM01a]. It is also important to review M&S in the context of the software engineering environment. As the Department evolves from procedural-based software development techniques to an object-oriented¹⁸³ approach, [Mey98] suggests that a model may be defined as a “digital or software representation of a physical entity”, in an object-oriented paradigm comprised of entities with associated attributes and behaviors, while a simulation may be viewed as the “software framework or architecture within which physical entities are animated” [Mey98]. At Figure 7-1 is one version of the evolving software engineering body of knowledge (SWE-BOK) [Moo99, BCC+01b] supporting improved simulation credibility.

¹⁸³ Object-oriented is a method of implementation in which programs are organized as cooperative collections of objects, each of which represents an instance of some class, and whose classes are all members of a hierarchy of classes united by inheritance relationships [PMR94]. Includes object-oriented analysis, design and programming.

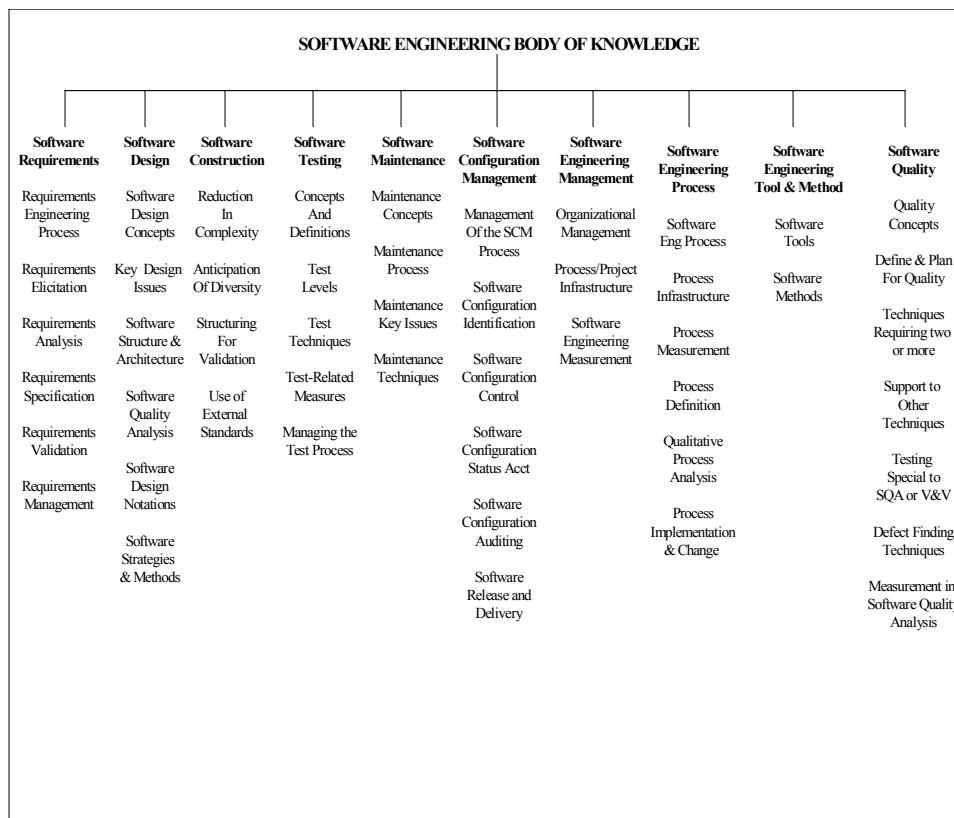


Figure 7-1. The SWEBOK (From [BCC+01b])

The software engineering process knowledge area is a dynamic, rapidly evolving field, with new paradigms and models constantly introduced. The Department's approach to software evolved from a software developer to a software acquirer, and today the Department acquires most software as commercial-off-the-shelf products or contracts with the private sector for the development effort. However, [Mos02] cautions that government program managers developing the contract request for proposal and the government contractor world winning the contract awards do not understand commercial best practices.

Conversely, [Mos02] contends the real commercial world contractors, bidding for private sector contracts know that best practices critical for success includes quality software, software architectures, and product lines, which provide a competitive advantage. [Mos02] concern echoes a recent Defense Science Board [HNC+00] report, which stated,

The major factor in successful software development is disciplined execution...too often; programs lacked well thought-out disciplined program management and / or software development processes. [HNC+00].

At a macro level, [Fer01] notes that software product quality and progress in the Department's weapon systems depend on a myriad of complex internal and external forces (e.g., Congress, DoD, Services), both within and outside the program manager's purview [Bro74]. In an article by [Jon00] on the best and worst software development practices, he characterized the military software domain with good development methods, supported by good quality control, but executing "marginal or deficient project management method" [Jon00]. However, there are some positive indicators. [SG99] cites improved project management, processes, and people as contributors to this positive trend, while interestingly enough, technology was not the problem, nor the silver bullet [Bro87, Bar02c] solution. [SG99] noted three factors contributing to this positive trend: smaller projects, better project management and greater use of standard infrastructures.

Future technical, organizational, cultural, and managerial factors in the missile domain simulation development domain cited in Table 7-1 will be significant challenges. Institutional change or transformation, led by early adapters requires reengineering of many institutions and processes, especially in the Department of Defense with its current and future strategy to defend the Nation built around information. Where once the prevailing school of thought in the Department's information technology world centered on processing power and bandwidth issues, these problems have technical solutions and are now more a matter of implementation. Replacing these older problem sets are many new, complex issues including information operations, information assurance, and the security of personal information from misuse.

Table 7-1 summarizes the Department institutional constraints adversely affecting M&S credibility today. Note the close correlation of the constraints cited in Table 7-1 to the major research areas identified in Figure 2-2. Significant persistent, organization, process, management, and technical constraints limit or adversely affect simulation software credibility in the Department [DoD95, San97b, Ett99, RPG00, HNC+00, Pac01b, DoD03a, Nor03]. Software engineering and software architecture skills are now emerging as critical

domain disciplines that must continue to mature the corresponding bodies of knowledge, if the Department and the Nation are to exploit the full potential of computer technology, and set the foundation for the next great era in computing, the Product Line Era. Product Line Practice [Nor03] areas address many of these constraints.

Department Model & Simulation Credibility Constraints			
SOURCE	CONSTRAINT I	CONSTRAINT II	CONSTRAINT III
M&S [DoD95, RPG00]	TECHNICAL	CULTURAL	MANAGEMENT
SBA [San97b]	PROCESS	CULTURAL	ENVIRONMENT
JSIMS [Ett99]	TECHNICAL	ARCHITECTURE	MANAGEMENT
DEF SCI BOARD [HNC+00]	PROCESS	DISCIPLINE	MANAGEMENT
VALIDATION [Pac01b]	PROCESS	DEVELOP SOFTWARE	MANAGEMENT
M&S STRATEGY [DoD03a]	PROCESS	ORGANIZATIONAL	BUSINESS
PRODUCT LINE [Nor03]	SOFTWARE ENGR	ORGANIZATIONAL	MANAGEMENT

Table 7-1. Department Institutional Constraints Affecting M&S Credibility

Future domain software engineers and domain software architects will need to understand the viewpoints of multiple, diverse stakeholders at all levels of the Department Enterprise and synchronize the software architecture and software engineering solution that meets cost, performance, schedule, and other constraints cited in Table 7-1. Developing and maintaining currency in the skill sets of domain software engineers and software architects grounded in the technical foundations of the field (e.g., computer science, software engineering, software architecture, information technology), with experience in multiple-domains and functional areas will also require continuous education opportunities and progressive developmental assignments.

E. SOFTWARE ENGINEERING EDUCATION INITIATIVES

The Department's draft strategic M&S plan [DoD03a] includes the a strategic goal for awareness, education, training, and collaboration, and proposes M&S education programs through the postgraduate level, providing Department employees with hands-on experience through on-the-job-training, internships, and rotational assignments. Systemic

information technology (IT) issues facing the Department today include the shortage of educated IT professionals [HP99a, PH99, Byn00, Mat00, RBB+00], with a special emphasis in the area of information assurance [Ste02b].

Improving the Department's IT competencies includes many disciplines and emerging bodies of knowledge, including software engineering. A growing body of literature addresses the complexity of the software engineering field including:

- Studies of the best software engineering training practices [MTC96, Wie96b],
- Software engineering education philosophy and alternative approaches [Par99, Bec99, Ber00b, CDC+00, His00, RBB+00, Sha00b, Rie01, CT02, BM02, HH02a, HH02b, SBM02, DT02, Jen02, BK02b],
- Teaching software engineering processes [Wie96b, BK01b, BCH+02, BKK02],
- Collaborative and distributed software engineering education to transform surplus engineers from traditional areas into software engineers [BPD02, HH02b, HMR02, UHC02],
- Applying the evolving software body of knowledge [Wie96b, TH02], and
- Adding a software engineering capability to existing military system engineering disciplines [Flo02b].

Emerging from the intellectual stimulus and software engineering studies are directories of industry and university software engineering collaboration programs [Bec99], guidelines for software engineering education [BHH+99], certification programs [OSA+01], computing curricula guidelines including software engineering [CDC+00], and new graduate- and postgraduate-level software engineering education opportunities [NPS02].

F. SUMMARY

In Chapter VII we reviewed three selected areas influencing credibility of large-scale, legacy simulations in the Department including process improvement, the status of new major simulation software engineering initiatives (e.g., JWARS, JSIMS, JMASS) and the growing software engineering body of knowledge. The chapter also included a discussion of the challenges generated by Department contract management oversight for achieving quality software and reducing heterogeneous system representation anomalies, Department institutional factors affecting M&S credibility, and recent Department software engineering education initiatives. In addition, research reveals a growing awareness that the Department needs qualified software engineers who understand the foundations of the

software-intensive systems upon which we basing our future security, economic prosperity, and military preparedness, including credible M&S.

The first component supporting M&S credibility is process improvement. By 1996, American corporate information technology project management realized no technological or silver bullet solution existed to reverse the tide of software project failures. The true scope of the problem and solution included processes and people. In an effort to improve quality, process improvement models including the Software Engineering Institute Capability Maturity Model Integration (CMMI), Capability Maturity Model (CMM) for software, Software Acquisition CMM, Systems Engineering CMM, ISO 9000, and the draft ISO 15504 standard for software process assessment (SPICE) initiatives evolved. These process models matured based on the premise that the quality of the processes used to develop and maintain the software significantly influenced the quality of a software system. Initial indications suggest that process improvement methods contribute to improve overall software product quality and productivity. However, the Department inconsistently supports software process improvement.

Contract management oversight is a critical process for an organization acquiring software. The challenges of managing the myriad of contracts supporting the Department's IT initiatives, including M&S software support, and improve collaboration, continue to grow. Federal Government spending on IT services almost doubled from fiscal years 1997 to 2001, increasing from \$9 billion to \$17 billion, with the Department of Defense remaining the largest single acquirer of IT services and increasing spending by about forty-one percent throughout the period. During the same period spending on IT services from the General Service Administration (GSA) greatly increased with spending on the GSA federal supply service schedule growing from approximately \$405 million to \$4.3 billion.

Research indicates software engineering maturity lacks discipline and lags significantly behind other engineering disciplines. Our research also suggests that simulation software development may be more complex than the development of other types of software, since a model is an “abstraction or approximate representation of something” [GMS+96], where no model is ever completely representative [BCN96], and [Sha75] contends no model is absolutely correct. Disciplined development and the M&S verification

and validation process is critical to the establishment of M&S software credibility and the establishment of confidence in the accreditation decision and the M&S results. However, the Department has not achieved any long-lasting Department-wide results improving simulation credibility. Quality simulation software, based on a disciplined software engineering foundation [Sha93, VFS+99] supports simulation credibility and the maintenance of user confidence in the results. However, documented systemic simulation software engineering issues [Arm64, PAD78, Che86a, HW97, GF00, Cra01a, Rie01] since the mid-1960s included the lack of adequate software requirements, software, configuration management, verification and validation, software engineering, processes, and software quality.

The next component of this research supporting M&S credibility is a review of the case studies of three modern software engineering projects involving new Department M&S initiatives: JWARS, JSIMS, and JMASS. This review was significant for several reasons. First, these replacement systems are overdue. The new Joint M&S programs JWARS, JSIMS, and JMASS, scheduled to replace many legacy systems are also over cost, and have reduced the original number of requirements and requirements to meet a future operational status. It is still to be determined if they will be useful for the emerging and rapidly evolving requirements of the 21st century.

Improving the Department's IT competencies includes many disciplines and emerging bodies of knowledge, including software engineering. A growing body of literature addresses the complexity of the software engineering field. The Department's draft strategic M&S plan [DoD03a] includes the a strategic goal for awareness, education, training, and collaboration, and proposes M&S education programs through the postgraduate level, providing Department employees with hands-on experience through on-the-job-training, internships, and rotational assignments. However, systemic information technology (IT) issues facing the Department today include the shortage of educated IT professionals [HP99a, PH99, Byn00, Mat00, RBB+00], with a special emphasis in the area of information assurance [Ste02b].

For over twenty years the Department hoped for a "silver bullet" solution to resolve the "software crisis". However, as hope for a "silver bullet" technical breakthrough to improve the Department's software, including simulation software, waned in the 1990s, it be-

came apparent that other non-technical solutions were required. A second underlying cause for the current situation is the apparent dichotomy presented by the similar M&S factors cited in Table 7-1 concurrently identified as the problem and the solution. In 1996 a major Department M&S study [PHP+96] identified three major types of systemic Department M&S challenges, technical, cultural, and managerial; while concurrently the Simulation-Based Acquisition (SBA) initiative identified three similar factors, process, culture, and environment as SBA enablers. More recently the SEI endorsed three related practice areas, technical management, organizational, and software engineering to support the successful implementation of software product lines.

These factors are significant since this methodology represents an institutional realization that the resolution to the twenty year old “software crisis” is not a “silver bullet” solution. Instead, the problem and the solution are multi-dimensioned, complex issues that require changes in all sectors of the Department. This change or transformation, led by early adapters requires a reengineering of many institutions and processes, especially in the Department with its current and future strategy to defend the Nation built around information. Where once the prevailing school of thought in the information technology world centered on processing power and bandwidth issues, these problems are technically solved and more a matter of implementation.

This realization will take time for society and the Department to truly assimilate and adapt to during the ongoing transition phase into the Information Age. We must however, also be mindful of the many Industrial Age vignettes where nations, armies, navies, or air forces were slow to adapt and the severe consequences of that failure in terms of defeat or loss of life.

VIII. SOFTWARE ENGINEERING OPTIONS TO IMPROVE CREDIBILITY

A. INTRODUCTION

Chapter VIII discusses traditional product lines, and introduces software product lines and software architectures. The chapter also reviews component development supporting a product line methodology, product line practice areas, and an overview of evolving software architecture theory potentially applicable to improved M&S credibility and reducing heterogeneous system representation anomalies.

B. PRODUCT LINES

Historically, Industrial Age companies found that using common assets to build related systems yielded significant market opportunities and improvements in time to market, customer satisfaction, product quality, meaningful metrics, and supported a competitive advantage for additional market-share. Henry Ford adopted the same concept when he introduced the assembly line, or product line method to the automobile industry, establishing the private industry product line model [MNJ+02].

In possibly the largest historical application of the product line concept, the United States private industry in World War II manufactured the majority of the ships, planes, tanks, and other major end items for the Allied war effort in numbers that the Axis powers could never match, establishing the same competitive advantage for wartime victory. Product lines continued to evolve in many Fortune 500 companies including Ford, McDonald's and Boeing, albeit differently. In the Boeing product line example, the parts list of two entirely different aircraft, the 757 and 767 models, overlapped by approximately sixty percent [Cle99].

C. SOFTWARE PRODUCT LINES

As the Information Age evolved and the corpus of software core assets accumulated, [Par76] noted,

We consider a set of programs to constitute a family whenever it is worthwhile to form the set by first studying the common properties of the set and

then determining the common properties of the individual family members [Par76].

The software product line concept builds on the experiences of component reuse including subroutines in the 1960s, modules in the 1970s, objects in the 1980s and component-based systems in the 1990's. The SEI developed a product line definition from the successful private industry model and early analysis by [Par76]. A product line involves:

A set of products sharing a common, managed set of features that satisfy specific needs of a selected market segment or mission [BFG+00, Nor03].

Software product lines building on product commonality however are a relatively new concept. The SEI Product Line Practices Initiative built on the concept of a product line and introduced the concept of software product lines as “a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets¹⁸⁴ in a prescribed way” [CN02]. The establishment of the software product line fielding approach requires core asset¹⁸⁵ development, or domain engineering¹⁸⁶, and product development, also called application engineering¹⁸⁷ using these core assets [BC96, Jon99, Nor03].

As an organization's potential strategic information resource, the evolution of a software product line requires an understanding of the organization's strategic plans, goals, business objectives, culture, technical acumen, risk threshold, and life cycle management plans [ISO00]. This analysis supports the multiple related software product line families and products, each with concurrent versions and releases. A product family¹⁸⁸ is also a set of products built from a common set of core assets [Nor03, CN02]. Although a product line does not require a product family, and a product family does not necessarily constitute

¹⁸⁴ Core assets are those assets that form the basis for the software product line. Core assets often include but are not limited to, the architecture, reusable software components, domain models, requirements statements, documentation and specifications, performance models, schedules, budgets, test plans, test cases, work plans, and process descriptions. The architecture is key among the collection of core assets [Nor03].

¹⁸⁵ A core asset is a software artifact used in the production of more than one product in a software product line. A core asset may be architecture, a software component, a process model, a plan, a document, or any other result of building a system [Nor03].

¹⁸⁶ The domain engineering process develops software assets for one or more domains [Nor03].

¹⁸⁷ The application engineering process develops software products from partial solutions or knowledge embodied in software assets [Nor03].

¹⁸⁸ A product family method may yield the greatest efficiencies for developing product lines, depending upon market targets or feature relationships, although it is not a requirement [Nor03].

a product line, the SEI definition of a software product line according to [Nor03, CN02] implies a product line of software products developed as a product family in a prescribed way, since this method leverages and amortizes prior investment [BC96, BBW+00], and potentially produces greater efficiencies, based on economies of scope and economies of scale¹⁸⁹.

Software product lines experience and growth in the commercial software sector and the Department provides a growing body of software product line case studies in the following areas: product line investment analysis [Wit96, BBW+00], product line experience and practice [BC96, WL99, Don00, CN02], product line organization and management [BCC+97, BCC+98, BCN+98, Cle99, BCD+00, CGF+00, Wap00, VW00, TCO00, MNJ+02], product line methods [CJB00, KLD02, KN00], product line process [Woo00a, FHR00], product line components (reuse) [Cle97, JNR98, ABM00, BCS00, PP00, Gri00, CN02], product line architectures [DS00, Pro00, Sha00a], product line tools and techniques [CJB00, AOV+00, YMK+00, SSP+00, ST00, HOF+00], product line domain engineering [ADD+00, LKK+00, Dag00, HSV00, TP00, MD02], and product lines in the Department [Jon99, BFJ99, BFG+00, BOS02, CDS02, Cam02].

[BFG+00, DDW01, CN02, KLD02] suggest employing a software product line at one or more of the following levels: system¹⁹⁰, subsystem, or component, depending on the application domain, degree of commonality, and feature variability. A software product line notes [Nor03] includes multiple, related products with product-specific cycles of releases and versions, which evolve in consonance with the product line as a whole. The upfront cost of developing the first software product line member(s) often requires a significant investment, supported by a business case and market analysis. Fielding a software product line includes the development of core assets and products coordinated by management activities. Software product line development includes mining existing products for generic assets or core assets for later development use in an iterative manner, based on current needed capabilities, anticipated future requirements, and likely future product variants [BOS00, CDK+01, CN02].

¹⁸⁹ The condition where fewer inputs such as effort and time are needed to produce greater quantities of a single output (e.g., economy of scale), or a greater variety of outputs (e.g., economy of scope) [Nor03].

¹⁹⁰ A product line system is a member of a software product line [Nor03].

1. Potential Benefit of a Software Product Line

A product line approach to developing and deploying software-intensive systems offers great promise according to [WL99, BFG+00, CN02] for delivering higher quality systems in a shorter time and at reduced cost. According to analysis of the industrial sector experience by [BFG+00, LKK+00, Dag00, HSV00, and CN02] a product line approach to software-intensive systems can save money and result in a faster time to field quality systems. [Jon99, CN02] cite a number of organizations gaining order-of-magnitude improvements in efficiency, productivity, and quality through a product line approach. [BC96, Jon99, WL99, BBW+00, CN02] also note that even more important than cost savings is the fact that product line practice enables an organization to get its product to the market more rapidly. As an example, [BC96] provides a case study of a successful product line implementation by the CelsiusTech Systems AB of Sweden in the area of large, embedded, real-time shipboard command and control systems.

The SEI identified organizations benefiting from software product line practice in workshops cited by [BCC+97, Cle97, BCC+98, BCN+98, CW98, BCD+00, CGF+00] and case studies [BC96] including:

- The Swedish naval defense contractor, CelsiusTech experienced a favorable reversal in the hardware-to-software cost ratio from 35:65 to 60:20, now favoring software,
- Hewlett-Packard has metrics reflecting two to seven times cycle time improvements,
- Motorola experienced a four times cycle improvement with 80% reuse on a pager product line,
- Cummings Engine recorded a dramatic reduction in one case for a system build and integration from about one year to three days,
- Thompson-CSF with air traffic control systems,
- Alltel supporting commercial bank systems,
- Ericsson, Noki, Lucent, and AT&T in communication systems,
- Boeing in air flight software, and
- The National Reconnaissance Office command and control systems for satellites [BC96, BCC+97, Cle97, BCC+98, BCN+98, WL99, BCD+00, BFG+00, CGF+00, CN02].

An organization's business analysis supports the decision to establish a product line approach with product line goals, objectives, strategies and the development of a Concept

of Operations (CONOPS). The product line CONOPS [AIA93, CFM+96, Coh99, CN02] establishes an approach for achieving the organization's product line goals of doing a job better, faster, and cheaper by focusing on efforts that reduce the costs and risks, associated with system development. The CONOPS is the system-users operational view of the system under development, and [Coh99] notes the CONOPS identifies in-house product line responsibilities for the government and commissioning organizations, and establishes acquisition/supplier relationships, determines the ownership of product line assets and defines access policies.

However, there are also significant implementation issues with product lines identified by [WL99, and Ebe01]. As a result, the SEI explored the range of issues and practices necessary for the successful implementation of software product lines [KZ96, Cle97, CN02], and developed several products and including an investment analysis methodology for software product lines [Wit96]. The SEI also sponsored a series of Product Line Practice (PLP) Workshops [BCC+97, BCC+98, BCN+98, BCD+00, CGF+00] to share industry practices in software product lines and to explore the technical and non-technical issues involved with software product line ventures, including software engineering, technical management, and enterprise management functions, responsibilities, and issues.

2. Challenges for Software Product Lines

a. Developing Software Product Lines in the Department

Any software product line implementation strategy in the Department must take into account the Department is an acquirer of software systems and not the software developer, and factor in product line practice areas supporting the Department's acquisition process. Initial efforts to employ product lines in the Department's M&S domain have been limited. The Air Force employed a structural modeling¹⁹¹ [ASC94, CB96] initiative to support the development of aircrew trainer simulator¹⁹² software for the B-2 Weapons

¹⁹¹ Structural modeling has been in use since the mid-1980s during which it was expanded from a method for constructing a software architecture into an architecture-based development method. It includes the general engineering principles and technologies including prescriptive architecture, incremental development, and prototyping [CB96].

¹⁹² A simulator is a device, computer program, or system that performs simulation or for training, a device which duplicates the essential features of a task situation and provides for direct human operation [DoD98].

System Trainer (1986), C-17 Aircrew Training System (ATS) (1990), Special Operation Forces (SOF) ATS (1991), and the Simulator Electric Combat Training (1992) as a method to improve changeability¹⁹³ and resolve the limitations¹⁹⁴ of the previously employed data-driven architecture¹⁹⁵. [CB96] noted that the simulation software was only a part of the B2 trainer supporting the physical cockpit, the cueing system, and the display software, and not a complete architectural solution, as presented by this dissertation. A major point addressed by [BC96] is that non-technical issues including business, personnel, management practices, cultural and process issues are equally as important to the success of a product line initiative as the technical issues.

Several Department research and acquisition organizations recently explored product lines applicability within the Department's M&S domain. The Naval Postgraduate School (NPS) developed a body of knowledge germane to this research initiative with research on the Janus model [JBL97, SLB+98, SLB+99, BLS+99], but it was limited in scope to a single model. The U.S. Army will use product lines as a method to replace outdated legacy instrumentation and simulation systems used in support of the Maneuver Combat Training Centers with the next generation Objective Instrumentation Systems [DDW01, MD02]. However, the [DDW01, MD02] initiative appears more focused on the hardware components of a product line, and it is much too soon to determine if it will be successful. [Bru98] also evaluated the product line methodology to determine the feasibility of the product line approach for software development in a single model.

The Department recognizes the potential benefits of product lines along with an acknowledgment of the significant challenges to adopting this approach [Wit96, JNR98, BFG+00, Jon00, BOS02, Cam02, CDS02]. Many of the challenges stem from the fact that the Department is in the business of acquiring systems rather than developing them. Although product line practice works in industry, many attempts to emulate this success within the Department encountered problems [BNS97, BFG+00]. Product lines face im-

¹⁹³ Changeability is defined as the ease with which the software standard can be maintained throughout its life cycle, and extends the concepts of modifiability and maintainability to include changes in requirements and specifications [CN96].

¹⁹⁴ The data-driven software architecture, a *de facto* standard since the 1960s increased the complexities and risk during integration and maintenance phases of the life cycle, most significantly data coherence problems stemming from spreading out the state information and communication of the state information across subsystems [CB96].

¹⁹⁵ The data-driven architecture consists of the executive code/scheduler, system routines, scheduling table, and the data pool, which is the data area shared by the system routines for storage and communication of state information [CB96]

plementations in the Department face technical challenges and significant non-technical barriers such as culture and acquisition-related issues. The SEI identified several key issues an organization should address before moving to a product line approach for acquiring or developing software:

- What constitutes the product line?
- How will it be introduced?
- What key organizational elements will be involved in defining, developing, and fielding the product line?
- What is the relationship between product line assets and systems within the product line?
- How will the architecture be developed and maintained?
- What are the sources of software components [BNS97, BFJ99, BFG+00]?

[BFJ99] recommends three product line acquisition activities for acquiring a product line in the Department's acquisition environment. These activities include: 1) acquire architecture and other elements of an asset base to enable a product line approach; 2) acquire software products utilizing the asset base; and 3) acquire the services to maintain the asset base and support the development and enhancement of derivative products [BFJ99]. [BFJ99] also suggests that product line start-ups may be more successful in a system's operational support phase, vice system start up phase. [BFJ99] also believes the strategically positioned system sustainers may be motivated to adopt a product line approach, since they may have responsibility for sustaining and enhancing similar operational systems.

b. Component Development for Software Product Lines

Components¹⁹⁶ play a major role in many software-intensive systems [LG96, HBL97, BBB+00, CN02, FEA02b, SAG02]. There are several motivations for a components-based software approach to software: providing a basis for commerce in reusable software to meet demand, facilitating the development of flexible systems, and reduce the time to design, implement, and deploy systems. As an indication of the need for soft-

¹⁹⁶ A software component is an opaque implementation of functionality, subject to third-party composition, conformant with a component model [BBB+00].

ware components¹⁹⁷ at the United States Federal Government level, the Office of Management and Budget recently established the Federal Enterprise Architecture Program Management Office to remedy the lack of a Federal Enterprise Architecture (FEA) [FEA02b, Hay03b], a major barrier to the success of the E-Government initiative¹⁹⁸. The FEA program includes a Component-Based Architecture (CBA) [FEA02b] supported by tools, technologies, and standards facilitating component reuse, distribution, and cross-agency collaboration.

The CBA framework [FEA02b, Gar02] builds on a tier-based architecture, employing layers to support the creation of components facilitating reuse and interoperability. The integrated CBA model [FEA02b] includes five architecture layers: Information, Technical, Security, Application, and Business employing an “Import and Export” methodology supported by Extensible Markup Language (XML)¹⁹⁹ schemas to support inter-agency interoperability and data sharing. The CBA [FEA02b] involves relevant industry standards (e.g., Hypertext Markup Language (HTML), XML, XML Web Services, Simple Mail Transfer Protocol (SMTP), and Simple Network Management Protocol (SNMP)) providing a foundation for interoperability, growth, integration, and expansion.

The Department supports the CBA initiative in many areas [Lat97], including M&S. In the Department’s M&S arena, the DMSO initiated the Composable Mission Space Environment (CMSE)²⁰⁰ project and sponsored a Composable²⁰¹ M&S Workshop [ES02, Gar02, Lor02, Mat02, Pet02, Ves02, ZHS02] in July 2002 to explore ways component-based technology may reduce the manpower, time, and resources currently needed to execute Department simulations in a transformation environment with joint, interoperable, re-useable models [CA02, PLN02, RPK+02, ZHS02]. The Department’s on-going transformation process drives the transition requirement from a threat-based force applying system-focused, platform-centric approaches to a capabilities-based force employing mission-

¹⁹⁷ At the Government Enterprise level a component is an application, capability, or service that leverages technology to perform a specific business function [FEA02b].

¹⁹⁸ The President approved twenty-four priority E-Gov initiatives to meet the business needs of the Federal Government [FEA02b].

¹⁹⁹ XML is a platform independent, universal language used to support the structuring and integration of documents and data on the web with a flexible set of standards for tagging and classifying information readable by humans and data exchange systems [FEA02b].

²⁰⁰ The CMSE is an interrelated collection of enabling M&S technologies, tools, and procedures.

²⁰¹ Composability is the capability to select and assemble simulation components in various combinations into simulation systems to satisfy specific user requirements [Pet02].

focused network-centric approaches in how the Department trains, procures, equips, operates and fights. The Workshop identified several specific recommendations for composable M&S standard development:

- Develop and interchange standard for data and models,
- Develop a Composable M&S Ontology,
- Conduct a business case analysis,
- Conduct further research in many less-understood areas including the impact of culture and organizations on adoption of a composable M&S approach,
- Engage basic research in component technology needs,
- Identify VV&A issues [ZHS02].

The SEI recently initiated a component-based software engineering (CBSE) study to determine if they could extract predicted properties of a CBSE system of components made from the components themselves [BBB+00]. During the process [BBB+00] adopted the following narrow vision of CBSE:

Component-based software engineering is concerned with the *rapid assembly* of systems from components where components and frameworks have *certified properties*; and these certified properties provide the basis for *predicting the properties of systems* built from components [BBB+00].

Component-based systems rely on defined standards and conventions, or *component model*, and a support infrastructure, or *component framework* [BBB+00]. A component model specifies the component standards and conventions (e.g., component typing, interaction schemes, resource binding²⁰²) whereas a component framework provides the developer with an implementation of services supporting or enforcing component model standards and conventions [BBB+00]. Although there still remains a lack of consensus on the contents of a component model, [BBB+00] suggest:

- A uniform composition constraining how components interact if and only if they share consistent assumptions about what each component provides and requires of another component. Although some assumptions are unique to the component, a standardization of assumptions (e.g., component location, control flow, data encoding, communication protocol) reduces chances for accidental mismatch, which adversely affect composition.

²⁰² A component type is defined by the interfaces it implements. An interaction scheme specifies how components are located, communication protocols, and quality of service attributes including security and standard transaction assumptions. Resource binding identifies the process of composing components in terms of binding a component to one or more resources [BBB+00].

- Appropriate quality attributes, which depend on the software architecture and architectural style²⁰³ supports the desired quality attributes for a system composed from third-party components and the quality of service²⁰⁴.
- Deployment of components, application development, and the emergence of a viable market in third-party components are critical to the success of a CBSE environment. Historical examples abound where technologies survived and economically prospered in the face of arguably superior technology (e.g., Apple versus Microsoft, the Beta tape format versus the VHS tape format). This precondition for component composition suggests that components successfully transition from the component developer to an application developer's composition environment, and finally, operate in the customer's environment [BBB+00].

The second component concept advanced by [BBB+00] is the component framework. In an analogous sense, a component framework is comparable to an operating system, in which components are to frameworks what processes are to operating systems, and the framework manages resources shared by the components, and provide the underlying services enabling communication among components [BBB+00].

In practice, component frameworks include the Enterprise JavaBeansTM specification of servers and containers, the WaterBeans framework for real-time visualization of high-performance data streams, and Microsoft's VisualBasic framework for visual composition of components [BBB+00]. A major challenge facing the successful implementation of component frameworks is the issue of standard versus custom component models and frameworks. These requirements create competition between the maximum flexibility school of thought supporting different architectures, different styles, and different allocation of quality attributes based on the application; and the business case school of thought supporting the establishment of a viable market in software components [BBB+00].

Component interfaces are the foundation for component substitutability and significantly more complex than traditional system interfaces. This generated the idea of an *interface contract* [BBB+00] where,

- A contract binds two or more parties,
- The parties negotiate details of the contract before they sign the contract,

²⁰³ The architectural style includes standardization of the types of components used in a system and their patterns of interaction [BBB+00].

²⁰⁴ The quality of service specifies the patterns of interaction (e.g., Type of transactions, encryption methods) [BBB+00].

- The contract prescribes normative and measurable behavior by all parties,
- A contract cannot be changed unless signatories approve all changes [BBB+00].

[LG96, OHR99, BCS00, PP00, Gri00, ABM00, and AB02] explain the design of generic systems based on the reuse of software designs and components, supporting designs of families of systems or product lines based on commonalities, including component metadata [OHR99], within the product lines. In object-oriented design, a similar concept for extensible software system provides for a framework and specific plug-ins. Design quality is critical and some quality attributes may or may not be discernable at run-time, while other attributes relate to the architectural qualities such as conceptual integrity, correctness, completeness, and build ability. Four general principles identified by [BL91, Som95, BW97, Pre97] impact software component construction:

- Reduction of complexity,
- Anticipation of diversity,
- Structuring for validation,
- Use of external standards [BL91, Som95, BW97, Pre97].

Composition²⁰⁵ is a key process within the software component development life cycle. The software composition design²⁰⁶ activity includes analysis of the software requirements [HBL97, Boc00, FHR00] and results in the development of the software product [BL91, Som95, LG96, LZB+96, Lat97, Pre97, Jac98, BBB+00, RPG00, Co01b, NS01, AKN02]. Key enabling design techniques and principles include abstraction, coupling, cohesion, decomposition, modularization, encapsulation, information hiding, separation of interface and implementation, sufficiency, completeness, and primitiveness. [Wil96, VT01] discuss a number of key issues for consideration during the design phase including quality and the decomposition, organization, and packaging of the software components. [BW97, CA02, PLN02, RPK02] review other aspects of the systems behavior, which horizontally crosscut the system functionality, including concurrency, event handling and control, distribution, exception handling, interactive systems, and data persistence issues.

²⁰⁵ Composition is the term used in component-based development to explain how to develop or assemble systems [BBB+00]

²⁰⁶ [IEE90] defines software design as both a process and a product.

Several critical aspect of component-based software still require attention according to [BBB+00, VT01, and Voa02], including the need for systems that will predictably exhibit the quality attributes required of them (e.g., predictive composition) and address,

- Changeable environment and dependencies,
- Variable usage,
- Large number of small parts to manage,
- Component flux (e.g., version creep),
- Developer inconsistencies [VT01].

Key technical areas of component-based development addressed earlier by [LG96, HBL97] and recently by [BBB+00] include components, interfaces, component models, component framework [Lat97, Wil01], composition [LG96, Wil01, AKN02, Voa02, KH02, PLN02], component-based product line engineering [ABB+02], the component certification process [WR94, POP00, CSS+01, GM01, GS01, JDL01, Mas01, MC01b, PSR+01, SW01], and component wrapping [PSR+01]. In essence at this point on the maturity curve, the current body of knowledge on components, component trust and certification, component technology, and software architecture,

- Raises concerns about properties of assemblies of components,
- Lacks information about component behavior,
- Lacks an understanding of the functional and extra-functional properties of systems,
- Lacks the ability to determine properties of “black-box” component assemblies [CSS+01].

Currently there is not an established basis for how well component models and frameworks contribute to achieving the desired quality [WR94, LG96, JDL01, VT01, KH02, AKN02, Voa02]; nor is there any basis for assessing the quality of software component interfaces, composition, and component certification. [WR94, BBB+00, KH02, and Wr02] address components and component certification issues and analyzed component complexity and interfaces. [CTW98] evaluated software component licensing issues, while [Voa02] reviews several significant issues with composition practices. Consensus on the many solutions required for contentious component / composition issues do not appear on the near-term horizon.

3. Product Line Practice Areas

[CN02, Nor03] defines a software product line practice area as a body of work or a collection of activities, which an organization must successfully master to carry out the essential work of a software product line. Many, if not the majority of the practice areas describe activities, which are critical to the success of any software project, and provide starting points for organizations to initiate and master activities and measure progress in adopting a product line approach. The practice areas in the framework divide into three categories: 1) software engineering practices, 2) technical management practices, and 3) organizational management practices illustrated in Table 8-1.

Implementing software product lines offers potential gains, and entails significant risks, including technical, organizational, and management issues. [Nor03] explains that building and acquiring a software product line requires disciplined engineering supported by mature technical and organization management processes of universal essential activities and practices, which the SEI developed into an online framework²⁰⁷, a web-based document describing the competencies needed to develop and field a successful a software product line or any software-intensive system.

A Framework for Software Product Line Practice, (Version 3.0) [Nor03] is a web-based tool introduced and designed to support the software community in software product line endeavors. Each version represents an incremental attempt to capture information about successful software product line practices. This information builds on the work of the SEI Product Line Practice Initiative, including studies of organizations, which have built product lines, from direct collaborations on software product lines with customer organizations, and from leading practitioners in software product lines. The SEI website defines all of the practice based upon ongoing software product line collaborations and the feedback from the community. Future versions will build upon the current foundation and the growing body of knowledge, refining current knowledge, and describing a small num-

²⁰⁷ Framework in this context suggests a conceptual index, a frame of reference for the information essential for success with product lines [Nor03].

ber of product line scenarios involving the development, acquisition, and/or evolution of a software product line. The Product Line Practice envisions:

- Product line development as low risk / high return
- Techniques for exploiting system commonalities and controlling variability with standard Department, government, and industry software engineering practices [Nor03].

There are several overlapping terms from previous information technology areas synonymous with product line terms. [ADD+00, TP00, and Sch00a] address core asset development activities, (e.g., domain engineering), while [BFG+00] addresses product development from core assets, often cited as application engineering. The architecture and components are central to the set of core assets (e.g., platform), used to construct and evolve the products in the product line. Development is a generic term to describe acquiring software. There are several viable options for acquiring software [CN02, Nor03, MNJ02]. An organization may build the software in-house from scratch, mine existing legacy software assets (e.g., core assets), purchase commercial-off-the-shelf-software (COTS), or commission the development, normally through a contract with someone to build the software [Nor03, MNJ02], or even combine several options. The objective of the core asset activity is to support development of a production capability for products and requires three component activities:

- Defining the product line scope,
- Developing the production plan
- Producing the core assets, [Nor03].

Table 8-1 identifies practice areas as a body of work or a collection of activities, which an organization must successfully master to carry out the essential work of a software product line. The following sections describe the three categories and the associated practice areas. The number of products developed from core assets will form the foundation for metrics indicating the real potential for mining core assets [BFG+00, BOS00]. A short summary of each product line practice area follows.

PRODUCT LINE PRACTICE AREAS v 4.1

<u>Software Engineering</u>	<u>Technical Management</u>	<u>Organization Management</u>
•Architecture Definition	•Configuration Management	•Business Case
•Architecture Evaluation	•Process Definition	•Customer Interface Mgt
•Component Development	•Product Line Scoping	•Acquisition Strategy
•COTS Utilization	•Technical Planning	•Funding
•Mining Existing Assets	•Technical Risk	•Launching/Institutionalize
•Requirements Engineering	Management	•Market Analysis
•Software System	•Tool Support	•Operations
Integration	•Data Collection, Metrics,	•Org Planning
•Testing	Track	•Org Risk Mgt
•Understanding Relevant	•Make/Buy/Mine/	•Structure Org
Domains	Commission	•Org Technology Forecast
		•Training

Table 8-1. Product Line Practice Areas (From [Nor03])

a. Software Engineering Practice Area

Based on research to date, the SEI [CN02, Nor03] proposes the following software engineering practices listed in Table 8-1 as necessary practices supporting an organization's capability and technology to create, evolve, and maintain core assets and products:

- **Architecture Definition** – This practice area defines the activities supporting the definition of software architecture for a product line and addresses quality attributes, system interaction requirements, and organization business goals. The focus may be infrastructure focused (e.g., operating system, protocols, middleware), or focus on the application functionality. The software architecture²⁰⁸ is a major component supporting the development of the core assets, affecting how well an organization fields products form a shared asset repository, supporting explicitly allowed variations [CN02, Nor03].
- **Architecture Evaluation** –The architecture of a system, an early design artifact, represents one of the earliest design decisions, and perhaps the most difficult to

²⁰⁸ No completely accepted definition for software architecture has emerged, although over ninety have been discovered by [BCC+03]. [CBB+03]: suggests the following definition:- Software Architecture is the structure or structures of a system, which comprise elements, their externally visible properties, and the relationships among them [CBB+03].

change over time. It is at this point that the software architects establish the system's quality foundations (e.g., security, reliability, usability). The abstraction of the architecture at this point supports communication among the system's key stakeholders. Several different architecture evaluation techniques exist including: the Architecture Tradeoff Analysis Method, the Software Architecture Analysis Method, active design reviews, Software Performance Engineering, and active reviews for intermediate designs [CN02, Nor03].

- **Component Development** – The term component²⁰⁹ like software architecture and object has collected many meanings, and not yet standardized. The component functionality is normally encapsulated and packaged with an interface provided to other components employing an agreed upon interconnection method. Components form the building blocks for applications, linked by communication and connection technologies including the Common Object Request Broker Architecture (CORBA), from the Object Management Group; Distributed Common Object Model (DCOM), originally from Microsoft; and JavaBean [DRS99], developed by Sun [CN02, Nor03].
- **COTS Utilization** – Commercial-off-the-shelf and non-developmental items (NDI) may also be used as core assets in a product line. In the past decade, the growth of middleware technologies supported the employment of COTS / NDI components in large-scale systems, introducing a new set of risks, constraints, and tradeoffs [Nor03]. Key issues for consideration include COTS evaluation process, adaptability, and vendor support for COTS products [BB99, Car99, Voa99, HP99b, SWR+99, PHW99, MG00, ABC00, CN02].
- **Mining Existing Assets** – Much of today's software systems evolve as extensions of legacy systems. However, legacy software involves *resurrected* and *rehabilitated* software or service in a new system beyond the original design. Mining expands significantly upon the current practice of small-grained reuse of code, sub-routines, or small programs and may include a higher-level focus on the organization's business processes and software architecture, in addition to consideration of the normal constraints of cost, schedule and functionality. Mining assets are resource intensive activities and focus on a wide range of assets besides code, including business models, rule bases, and budgets. Prime candidates include algorithms, interface specifications, performance models, test plans, and available architecture documentation. If quality documentation does not exist, reconstruction of existing documentation with enhancements such as tradeoff options, presents an alternative approach supporting improved use of the component [BOS00, CN02, Nor03].
- **Requirements Engineering** – The IEEE defines a requirement²¹⁰, and [Bro87] noted the "hardest single part of building a software system is deciding precisely what to build...the detailed technical requirements" [Bro87]. [BL91, GGR+94, GGR+95, Som95, Pre97, LK00] provide in-depth coverage of requirements engi-

²⁰⁹ Components are the principal computational elements and data sources that execute in a system [CBB+03].

²¹⁰ A requirement is: (1) A condition or capability needed by the user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a system or a system component to satisfy a contract, standard, specification, or other formally imposed documents. (3) A documented representation of a condition or capability as in (1) or (2) [IEE90].

neering techniques supporting systematic and repeatable processes for completely, consistently, and relevantly eliciting, analyzing, specifying, verifying, and managing requirements. Requirements engineering for a product line differs from the requirement process of a single system and includes: capturing the anticipated variations over the projected future of the product line, finding commonalities and identifying variations, preparing a product-line-wide set of requirements and product – specific requirements. It may involve a broader verification process occurring most likely in stages, and adaptation to support the dual nature and staged (e.g., common, specific) nature of product line requirements engineering [CN02, Nor03].

Product line analysis is a requirements process for engineering a product line of software-intensive systems [CDK+01]. The [CDK+01] suggested methodology encompasses the elicitation, analysis, specification, and verification and validation of the requirements for a product line. Four interrelated products support the product line requirements model for elicitation, analysis, specification, and verification: object modeling [RF01], use-case modeling, feature-modeling, and the dictionary, with the goal to achieve high cohesion / low coupling [CDK+01]. [CDK+01] also propose product line analysis methods for establishing the requirements for a product line of software-intensive systems supported in the context of product line development, assisted by modeling techniques and tools [YMK+00, SSP+00, ST00, HOF+00, CN02].

[CDK+01] also show how to build a requirements model from work products, based on object modeling, use-case modeling, and feature-modeling techniques. Requirements are statements of what a system must do, how the system must behave, the properties it must exhibit, the qualities it must possess, and the constraints that the system and its development must satisfy. A feature is a distinct aspect, quality, or characteristic of a software system or systems visible to the user. Strategies and methods for product line requirements modeling [KN00, CJB00, AOV+00] include the feature-driven strategy and a use-case-driven strategy. In a feature-driven approach, requirement modeling focuses on the features, while developers use the use-case strategy to discover requirements [CDK+01].

- **Software System Integration** – This practice involves combining individual software components into an integrated whole in a process where components are integrated into subsystems or when subsystems are combined into products, either discretely supporting a waterfall approach, or continuously supporting an incremental methodology. A key point about product line integration is that cost of integration identified in the product line scope, core assets, and production plan for the architecture’s planned life cycle is amortized over many products, versus a single system integration. Once the components and interfaces have been tested and verified there should be very little effort needed for new variations and adaptations. Specific practices supporting component integration includes patterns [Fow97], object technology, wrapping for recovery or discovery solutions, and middleware [CN02, Nor03].
- **Testing** – Testing performs two primary functions described in detailed by [BL91, Som95, Pre97, Roa98, and LK00]. First, testing continually supports developers

ability to identify faults leading to failure in the development phase, and secondly, at a later time when testers determine whether a system can perform to meet its requirements [CN02, Nor03]. Testing is also a critical part of the V&V and quality processes. Metrics and measurements for testing and reliability supporting credibility in simulation software include [Ber94, CFG+94, Jon95, FV96, PGF96, Sta96, MV96, JA97, Wie97b, AVL+97, Sch99b].

- ***Understanding Relevant Domains*** – Domain understanding evolved as a major factor in this research. Domains are areas of expertise and domain knowledge available for creating future systems or set of systems, with an understanding that knowledge from several domains is normally required to build a single product. Understanding a specific domain normally entails the identification of areas of expertise, identification of recurring domain problems and known solutions, capturing and representing this information to stakeholders, for the duration of the effort. Domain comprehension supports understanding the *commonality* and *variability* of potential future product within the scope of the product line [CN02, Nor03].

b. Technical Management Practice Area

Technical management practices are those management practices necessary to engineer, develop, evolve, and maintain to core assets and the products and encompass:

- ***Configuration Management*** - Software configuration management (CM) [Bro98, CN02, Nor03, BCC+01b] includes the following activities: software configuration identification, software configuration control, software configuration status accounting, software configuration auditing, and software release. Configuration management processes matured steadily since the early 1970s, and include new or reengineered methods, management techniques [Bro96, BL91, Som95, Pre97, LK00], technology [STS94a], and products [BH99] continuously evolve [Bro87].

However, CM for a product line is complex. The core assets and each of the products in the product line constitute a configuration to manage, and the management of all configurations needs coordination under one process. We will compare and contrast the CM requirements for a single system with the CM requirements for a product line. First, a single system manages each version's configuration; a product line requires CM for each version of each product. A single-system CM process may separately manage each product and all its versions; however, in a product line system the core assets integrated across all products require a single, unified CM process. Third, in a single system, the component developers and product developers are often the same, whereas in a product line must support the CM of the core assets, normally developed by one team and supporting parallel production by several other teams [CN02, Nor03]. The additional requirements levied by a product line CM suggest the need for disciplined CM techniques and processes, supported by robust CM tools.

Additional new CM techniques include the uniform version management framework, explained by [WMC02], which supports the definition of version models and addresses the orthogonal differences between the version and data models; a test bed model that separates CM repositories from CM policies detailed by [HCH+02]; visualization techniques [EGF02], and software merging version control [Men02]. [AB02] describes open source software projects CM methods, while [SJW+02] discusses open source software maintainability. [Kil02] cites the following benefits to software accuracy attributable to good software configuration management: longer shelf life of V&V work, confidence that model results are consistent, lower likelihood of undetected errors in the code, less experimentation and training needed to operate the model, and CM provides a venue to pool resources for model changes, including legacy models with multiple users [Kil02].

Configuration management has proven inconsistent in the Department's M&S domain. Citing the restrictions of the current VV&A directives, [Cau95] is concerned that the process is so complex, time-consuming and expensive that changes are often made to the M&S before the V&V process is completed, potentially invalidating the process. [Mue97a] described the Susceptibility Model Assessment and Range Test (SMART) project, which integrated configuration management with V&V processes to improve M&S credibility. Decision-makers trust in an M&S tool requires strict configuration management, yet the M&S analytical tool set must have the flexibility and depth to resolve complex problems.

- **Process Definition** – The process definition practice area involves an organization's capability to define and follow documented processes. Product line management requires the disciplined interaction of separate organizational entities adhering to mature processes. Each core asset has an *attached* process explaining how to employ the core asset to build products in a product line [CN02, Nor03]. Software process modeling supports process definition, describing the abstract description and models of important defined software development processes executed by a human or a machine to achieve the following goals: 1) facilitate human understanding, 2) support process management, and 3) support process improvement..
- **Product Line Scoping** – Scoping bounds a system or set of systems defining behaviors, characteristics or aspects included or excluded from the product line formalized in a scope definition document supporting the requirements engineering process, or influencing the market analysis for a product line variant [CN02, Nor03].
- **Technical Planning** – The product line requires no new planning processes, however, core asset development, core asset maintenance, core asset production, and core asset reuse plans are unique product line endeavors [CN02, Nor03]..
- **Technical Risk Management** – Risk management is the practice of managing risks within a project, organization or a group of organizations. Risk management is a critical process for a product line since the risks involve more than one product, with potentially far-reaching consequences [CN02, Nor03].
- **Tool Support** – A product line requires tools to support concurrent development, employment and maintenance of multiple core asset artifacts. Most likely several

interoperable computer-aided software engineering tools will manage the product line [CN02, Nor03].

- **Data Collection, Metrics, and Tracking** – Measurements and metrics support and guide management decisions about whether organization goals and objectives are met over time [CN02, Nor03]. Product line systems are managed in many ways just like a single product system, except for the additional requirements to track the core asset development, product development, and overall management of the product line. [Ber94, CFG+94, Jon95, FV96, PGF96, Sta96, MV96, JA97, Wie97b, AVL+97, Sch99b, and IEE001] provide guidance on measurement and metrics.
- **Make / Buy / Mine /Commission Analysis** – Normally a product may be built in-house, purchased from a commercial company, commissioned for development, or mined from in-house assets based on a core asset development decision process including quality, cost, product line requirements, architecture, variation flexibility, maintainability, and schedule [BSW+99, CN02, Nor03]. The product development activity depends upon the product line scope, core assets, production plan, and the requirements for the individual products. Product line organizations are flexible and may include a product group for several products, or one product group per product. Management according to [Wap00, VW00, TCO00] also makes assets available for reuse, retains responsibility for success or failure, manages external interfaces, creates an adoption plan, and acts as or empowers the product line champion. Management support required changes to cultural perspectives [Wie96b] and allows new products to align with existing assets, or update the assets to meet the new requirements.

-

c. Organizational Management Practice Area

Organizational management practices are those management practices necessary to orchestrate the entire core assets and products line effort and include:

- **Building and Communicating a Business Case** - There are several generally accepted costing approaches for building and communicating a business case, including top-down, bottoms-up, algorithmic, analogy, and expert judgment [BBW+00, CN02]. A significant body of work exists on cost estimation approaches based on project characteristics. Software estimation tools include the non-proprietary Constructive Cost Model models first introduced by Barry Boehm COCOMO model in 1981, the Revised Intermediate COCOMO (REVIC) model, Putnam's Software Lifecycle Model (SLIM) and Boehm's COCOMO II; in addition to software size estimation tools such as SEER-SSM discussed and compared by [BNT93, STS93a, STS93b, STS94b, McG97, BA99, and Nog00]. [SK02] advocates the development of newer empirical cost and schedule estimation approaches based on the increased use of COTS, application generators, simulations, and fourth generation languages

today, while [Wit96] proposes an investment analysis tool for software product line business case development.

- **Customer Interface Management** – Managing the customer interface for a product line differs significantly from a single-product organization, requiring new methods for managing customer expectations, negotiating requirements, developing, evolving, and maintaining customer products and providing customer support [CN002, Nor03].
- **Developing and Implementing an Acquisition Strategy**– This process supports the acquisition of products and services by contract, such as those organizations purchasing or commissioning products, and especially important for government agencies, such as the Department. Acquisition of product lines are typically structured differently with the acquisition of the core assets versus a product as a contract deliverable, although the role of the architecture in a product line may provide opportunities for contracting flexibility [CN02, Nor03]. Product lines present several challenges to the Department [Jon99], requiring the development of Department product line acquisition processes [BFJ99, BFG+00, Cam02, BOS02], and the sharing of lessons-learned from successful Department product line initiatives [CDS02].
- **Funding** – A product line requires significant up-front investment to build or acquire the core asset base, complete initial analysis efforts, and establish the production infrastructure; then evolving and sustaining the effort [CN02, Nor03]. This indicates a need for a strategic plan and stable, sustained funding, a significant challenge for product lines in the Department.
- **Launching and Institutionalizing a Product Line** - Launching and institutionalizing a product line involves the initiation and improvement of the product line practices appropriate for a given organization, and viewed as a practice area for applying the other practice areas [CN02, Nor03].
- **Market Analysis** – A market analysis is an early step for a product line, establishes the product line scope, providing analysis on the possible product line commonality and variability [CN02, Nor03].
- **Operations** – Operations define which part of the organization develops the core assets and products, and directs planning, processes, strategies, policies, and constraints [Nor03]. Management provides the resources, coordination, and supervision, critical to success, ensuring coordination of the operations and communications activities of the product line effort with a concept of operations (CONOPS).
- **Organization Planning** – The product line planning process, as mentioned earlier, is not unique, but does require product line adoption plans, core asset funding plans, and due to its importance, organization-level configuration management plans [CN02, Nor03].
- **Organizational Risk Management** - Organizational risk management involves risk management at the strategic level, and requires a great deal of coordination across project boundaries supported by open communication, integrated management, teamwork, a forward-looking view, global perspective, and shared product vision [CN02, Nor03].

- ***Structuring the Organization*** – A product line approach requires new roles and responsibilities supporting core asset development and product development from the core asset base. An early decision identifies where to locate the developers and maintainers of the product lines core assets and the organizations that build the product lines. Typically, the developers and maintainers may be located separately from the product line builders or they may be located together [CN02, Nor03].
- ***Technology Forecasting*** – Technology forecasts for product lines include two focus areas, internal development for tools, processes, or methods; or customer solutions such as technology or solutions that may possibly affect product line features or capabilities [CN02, Nor03].
- ***Training*** – Training is a core activity of organizations developing software and supports the initial product line adoption and the long-term product evolution; and requires management’s commitment, an effective training plan, and support of product line adoption process or process improvement [CN02, Nor03].

D. EVOLVING SOFTWARE ARCHITECTURE THEORY

1. The Federal Enterprise Architecture (FEA)

The Office of Management and Budget recently initiated a Federal Enterprise Architecture (FEA) program [OMB02, FEA02a, FEA02b, SAG02, Hay03b, BBT+03] supporting the President’s e-Gov guidance, focuses on information technology investments, and designed to facilitate cross-agency analysis and improvement. The lack of architecture in the Federal Government is a systemic management weakness repeatedly cited since the early 1990s [BBT+03]. The Federal Enterprise Architecture’s four-layer²¹¹ segmented structure [FEA02a] include systematically derived and captured structural descriptions in useful documentation (e.g., models, diagrams, narrative) for a given enterprise, including a single organization, or functional area, or a mission area that transcends more than one organizational boundary [BBT+03]. The existing agency-specific architectures will serve as the foundation for the FEA, with five reference models or views under development to assist with aligning existing data with the FEA,

- The *Business Reference Model* describes Federal Government business operations, independent of individual agency implementation,
- The *Performance Reference Model* provides a common set of general performance outputs and measures for agencies,

²¹¹ The four layers of the FEA are the Technology, Application, Data, and Business Architectures [FEA02a].

- The *Data and Information Reference Model* describes at an aggregate level, the data types supporting agency operations, and the relationships among the data,
- The *Service Component Reference Model* identifies and classifies information technology service components, and promotes reuse,
- The *Technical Reference Model* describes how technology supports delivery of service components and includes relevant standards [BBT+03].

The Department, as an agency of the Federal Government develops software-intensive systems with the intent of achieving interoperability with other stakeholders, including other Government agencies (OGA), will support the FEA program.

The FEA suggests the use of Unified Modeling Language (UML) [UML99] for defining and applying data models and standards [FEA02a]. In addition, XML, emerging as a government and industry standard, provides a recommended foundation as the default format for moving and sharing highly structured information, as well as less highly structured information between E-Gov data architectures [KJ01, LE01, FEA02a]. Data interoperability principles for the FEA Data Architecture support improved interoperability by,

- Avoiding non-standard data syntaxes,
- Seeking industry vocabularies before developing custom schemas,
- Avoiding the “one size fits all” schema concept,
- Registering the semantics of shared data elements,
- Documenting service interfaces in a standard way [FEA02a].

2. Emerging Software Architecture Concepts

The software architecture discipline is relatively new, and has not been completely defined and applied consistently to the life-cycle management of software-intensive systems. Much of the current software architecture research stems from the earlier works of [Par72, Par76, Par79, PW92, SG93, and SC96] and their observations that software consists of many structures, and that a system is a collection of related parts. Although there is not currently agreement on a precise definition of a system’s architecture²¹², the *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems* [IEE00b], cites a consensus on the use of multiple views²¹³, reusable specifications [BS95, Gac95,

²¹² Architecture is the fundamental organization of a system, embodied in its components, their relationship to each other and the environment, and the principles governing its design and evolution [IEE00b]

²¹³ a) A view is a representation of a whole system from the perspective of a related set of concerns [IEE00b]. b) A view is a representation of a set of system elements and the relationships associated with them [CBB+03].

HBL97, MR02] for models within views, and the relationship of architecture to the system context. The foundation of the approved [IEE00b] builds on the following concepts and relationships,

- Every software-intensive system has an architecture, however, an architecture is not a system,
- An architecture and an architectural description are not the same,
- Architecture standards, descriptions, and development processes can differ and be separately developed,
- Architecture descriptions are inherently multi-viewed,
- Effective architecture description standards support the separation of an object's view from its specification [MEH01].

The architecture of a system²¹⁴ is a critical component supporting the engineering process and the life cycle model²¹⁵ of the system [LB95, SNH95, MTW96, CN96, Fow97, HBL97, LB98, CWK99, BBG+00, Bos00, IEE00b, HHP00, MEH01, MM01, FG02, LLC+02, MR02, SB02b, Fra03, CBB+03, GA03, SPG03, Tol03]. As large-scale, legacy software-intensive systems evolved, the initial architectural emphasis was on the hardware and network architecture components of the information systems, until the complexity of the software technology and the cost of software development necessitated a change in the emphasis to include the software-related architecture issues [Sha01] of today's software-intensive system. Software architecture especially for large, software-intensive systems,

- Serves as the system blueprint, a focal point for the project development team and mutual communication,
- Serves as the foundation for the system's quality attributes,
- Provides the first artifact for early design decisions indicating the system meets requirements,
- Supports a transferable abstraction of the system for activities including post-deployment maintenance or mining [CN96, BBG+00, BOS00, MR02].

The software architecture is key to the success of any software project [LB98, BCK98] and critical to the success of a product line initiatives [Jon99, Bos00, DS00, Pro00, Sha00a, Mor00a, Don00, CN02]. Architectural drivers, influencing the entire life

²¹⁴ System is defined as a collection of components organized to accomplish a specific function or set of functions [IEE00b].

²¹⁵ The life cycle model is a framework containing the processes, activities, and tasks involved in the development, operation, and maintenance of a software product, which spans the life of the system from the definition of its requirements to the termination of its use [IEE00b].

cycle of a system, include quality attributes, business goals, and system interactions [MR02]. Product line architectures also include the allowable product variability and the different instantiations of the product, since these too are products.

There may also be many views in a software architecture domain, which show specific properties of the software system [IEE00b]. These different architecture views address diverse issues encountered with the design process, including the logical view, process view, physical view, and the development view. Other terms of references for software architecture design views include behavioral, functional, structural, and data modeling views, however, the key point made by [BB00] is that software architecture design is the product of a multi-dimensioned process composed of independent and orthogonal views, impacted by variability, either planned or unintended.

[CBB+03] defines an architectural style²¹⁶ (e.g., pattern) as a specialization of element and relation types, together with a set of constraints on their use. In this research effort, an architectural style consists of a set of constraints on the architecture, which define the set or family of architectures, and satisfy them with a number of major styles. These styles may include general structures of pipes and filters [SC96], layers [BBG+00], blackboards, object-orientation, and implicit invocation²¹⁷ [Bos00]; distributed systems; interactive systems; adaptable systems; and other styles including batch, interpreters, process control, specification-based, and rule-based.

Software architecture theory is a rapidly evolving component of software engineering with a wide range of diverse concepts, styles, and views including: the development of domain specific repository for components [Gac95, HBL97], structural modeling [CB96], software architecture overview [CN96], transitioning to a model-based engineering architectural style [GP96], and includes industrial best-practices for evaluating software architectures [ABC+97]. The evolving software architecture theory body of knowledge includes a wide spectrum of topics: evaluating the quality attributes of a software architecture [BKW97], reconstructing a software architecture with automated support [KC97], reusable

²¹⁶ Shaw and Clements define architectural styles as a set of design rules that identify the kinds of components and connectors that may be used to compose a system or subsystem, together with local or global constraints on the way the composition is done [SC96].

²¹⁷ Mary Shaw and David Garlan introduced implicit invocation in (1996) as a style that organizes the system in terms of components that generate events, possibly containing data, and that consume events. An example is the JavaBeans standard [Bos00].

components implemented via a relational hydrograph model [Luq90, HBL97, LG97], software architectural transformation via automated code transformation [CWK99], interoperability [Sut99], attribute-based architectural styles [KK99], and methods for documenting architectural layers [BBG+00].

3. Software Architecture Views

A system²¹⁸ is a collection of components organized to accomplish a specific function or set of functions [IEE00b]. In addition to defining architecture, [PW92] emphasized certain architectural considerations for different stakeholders or for different uses with a variety of system views. A sample of other recent advances in software architectures noted in Table 8-2 include [Kru95] describing four views of software architecture for system building; the collaborative work of [SNH95] who identified four additional industrial use views; four business views [HS00c], the development of viewpoints²¹⁹ [IEE00b] to designate the means used to construct individual views; and the three categories of views identified as viewtypes²²⁰ [CBB+03].

[PW92]	[SG93]	[Kru95]	[SNH95]	[HS00c]	[IEE00b]	[CBB+03]
Building	Components	“4+1” RUP	Siemens 4	View Sets	Viewpoints	Viewtypes
Elements (3)	Components	Logical	Conceptual	Technical	Structural	Module
Form	Structural Organization	Process	Module-Interconnection	Project Management	Physical Interconnect	Component & Connector
Rationale	Connectors	Development	Execution	Application	Behavioral	Allocation
	Constraints	Physical	Code	Functional	Link Bit	
		Scenarios +1				

Table 8-2. Software Architectural Views / Viewpoints / Viewtypes (From [CBB+03])

²¹⁸ A system also includes individual applications, traditional systems, subsystems, systems of systems, product lines, product families, whole enterprises, and other aggregated types [IEE00b].

²¹⁹ A viewpoint is specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis [IEE00b].

²²⁰ A viewtype defines the element types and relationship types used to describe the architecture of a software system from a particular perspective [CBB+03].

There have been many attempts to overcome the formidable risks and difficulties experienced in the design, development, deployment, and evolution of software-intensive systems, with improved software engineering practices, procedures, and techniques [MR02]. In 1996, the IEEE chartered the Architecture Working Group (AWG) to implement the approved recommendations from the 1995 IEEE Architecture Planning Group for software-intensive systems. The [IEE00b] charter tasked the AWG to define terms, principles and guidelines for the consistent application of architectural precepts for systems throughout the entire life cycle.

The AWG elaborated on architectural precepts and potential benefits for software products, systems, and aggregated systems (e.g., systems of systems); provided a framework for architectural attributes; and developed a roadmap for architectural precepts in the generation, revision and application of IEEE standards. In developing [IEE00b], the AWG intended to capture the architectural information contained in the various products of the system development process illustrated in Figure 8-1 and devise an architectural description²²¹:

- Expressing the system and its evolution,
- Supporting communication among the system stakeholders,
- Allowing a consistent comparison of architectures,
- Supporting system development,
- Identifying the system's persistent characteristics and supporting principles for future changes,
- Verifying the system's implementation as compliant with an architectural description,
- Recording contributions to the body of knowledge of software-intensive system architectures [IEE00b].

²²¹ An architectural description is a collection of products to document an architecture [IEE00b].

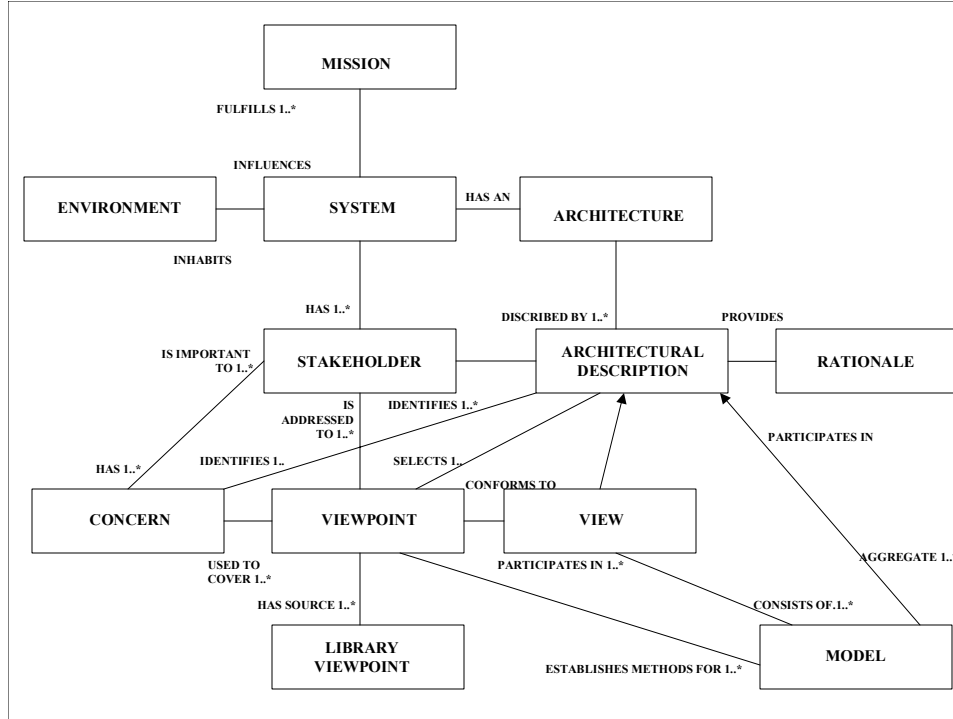


Figure 8-1. Model of an Architectural Description (From [IEE00b])

4. Software Architecture Quality Analysis Methods

[DN02] suggest that over the past several years, software architecture emerged as the appropriate level for addressing software quality, and recent efforts to understand the design patterns and architectural styles contribute to that quality analysis effort. In their research [DN02] provide a concise and recommended survey on eight representative software analysis methods in different domains, compare and contrast similarities and differences through study, comparison, and classification, and offer guidelines supporting the use of the most suitable method for an architecture assessment process.

[DN02] fit these eight methods into three categories or communities: software metrics, scenario-based, and attribute model-based analysis. The software metrics analysis technique uses module coupling and cohesion theories or more abstract evaluations to define predictive measures, of quality; while scenario-based methods applied over the past several years have a maturity and a certain level of face validation based on use; while the attribute model-based is too new to evaluate [DN02]. These eight methods include:

- Scenario-Based Architecture Analysis Method (SAAM) supports a general understanding of the general architectural concepts supporting proof that the software system meets more than the functional requirements,
- SAAM Founded on Complex Scenarios (SAAMCS) considers that the complexity of scenarios is the most important factor for risk assessment,
- Extending SAAM by Integration in the Domain (ESAAMI) a combination of analytical and reuse concepts, integrates SAAM in the domain-specific and reuse-based development process, considering only the problem description, requirements statement, and architecture description,
- Software Architecture Analysis Method for Evolution and Reuse (SAAMER), an extension of SAAM assessed quality objectives based on two attributes, evolution and reusability,
- The Architecture Trade-off Method (ATAM) an attribute-based model, provides a way of understanding a software architecture's capability with multiple, competing quality attributes,
- Scenario-Based Architecture Reengineering (SBAR) evaluates multiple quality attributes of the architecture design in a scenario-based review of the software qualities of a specific software architecture or system,
- Architecture Level Prediction of Software Maintenance (ALPSM) analyzes maintainability of a software system employing scenarios at the software architecture level and using the size of the change as a predictor,
- Software Architecture Evaluation Model (SAEM), a quality attribute model, establishes the basis for the software architecture internal and external quality evaluation and prediction of final system quality [DN02].

Software quality is one of three dependent, user-oriented product characteristics, in addition to cost and schedule [BKW97]. A process mature organization may predict and control cost, however, process maturity does not automatically translate into product quality, which requires mature technology to predict and control quality attributes [BKW97]. [BKW97, BKB00, BKB01, and BBK02] describe the quality attribute requirements for performance, security, modifiability, reliability, and usability and the significant influence of these attributes for evaluating software architectures.

Other representative software architecture quality assessment techniques include the Model-Driven Architecture Theory [Sie01, Fra03], supported by the Object Management Group, simulation, mathematical modeling, and experienced-based assessment techniques [Bos00]. [CBB+03] identified Department software architecture theory as a fledgling practice and considers the JTA and C4ISR architecture framework focus on the system architecture, and HLA publish-subscribe mechanism lacking in software architecture substance.

However, an understanding of software architecture in the M&S domain continues to grow, and [Tol02, Tol03] discusses future architecture requirements for HLA development. The SEI also contributed the Attribute-Based Architecture Style (ABAS) [KK99], Architecture Based Design (ABD) [BK99] and Attribute Driven Design (ADD) [BKB00] methods.

The Architecture Based Design (ABD) method [BK99, BBC+00, BBK00] provides a recursive framework with three foundations of attributes (e.g., functional, quality, and business requirements) at a level of abstraction supporting the necessary variation for producing products, and includes function decomposition, architectural styles, and software templates for designing high-level software architectures. [BKB01] develops functional, quality and business requirements, or architectural drivers, at a level of abstraction that allows for the variation needed to produce specific products for a product line or any type of system with a long lifeline. The ABD method [BK99, BBC+00, BBK00] provides a process for designing the conceptual software architecture, support for organization functions, identification of synchronization points, and allocation of functions to processes, concluding with allocation commitments to classes, processes or operating systems. A product of the ABD process [BBC+00] is a collection of software templates²²², which constrain the implementation of different types of components with a description of component interactions and responsibilities.

Another SEI-sponsored approach for defining a software architecture based on a design process driven with specified functional and quality attribute requirements possessed by the software is the Attribute Driven Design (ADD) [BKB00, BKB01, BBK02] method, a recursive, decomposition process where at each stage of the decomposition process, chosen attribute primitives²²³ [BKB00] attempt to satisfy a set of quality scenarios [BKB01]. In addition the development of quality architectures necessitates early consideration of factors affecting the various quality attributes and the impact on software design [BKB00, BBK02]. [BKB00, BKB01, and BBK02] further suggest the use of general scenarios to support development of quality attributes, with each general scenario consisting of:

²²² A software template defines the software element of a particular type, including patterns describing interactions with shared services and infrastructure, and citizenship responsibilities [BBC+00].

²²³ An attribute primitive is a collection of components and connectors that 1) collaborate to achieve some quality attribute goal, normally expressed in a general scenario; and 2) is minimal with respect to the achievement of these goals [BKB01].

- The stimuli requiring an architecture's response,
- The source of the stimuli,
- The context in which the stimuli occurs,
- The type of system elements involved in the response,
- Possible responses,
- The measure's employed to characterize the architecture's response [BKB00, BKB01, BBK02].

An example of a reliability general scenario is the case where an internal component fails. The system recognizes a failure of an internal component and possesses capabilities to compensate for the fault. The SEI developed quality attribute workshops as a method to analyze a system against a number of critical quality attributes [BW02, BEL+02].

Quality attribute design primitives are basically templates and provide building blocks for developing architecture designs supporting the achievement of specific quality attribute goals [BKB01]. An attribute primitives addresses one or more quality attributes characterized by one or more general scenarios [BKB01]. In a product line different products may have different quality attribute requirements or the products may exist simultaneously and only vary in terms of different attributes, characteristics, or scale factors.

5. The eXtensible Markup Language (XML)

The World Wide Web Consortium (W3C) developed the key technical standards for XML, including Namespaces the XML Information Set, and XML Inclusions [Sal02]. XML, as an application profile or restricted form of the Standard Generalized Markup Language (SGML) describes a class of data objects called XML documents and partially describes the behavior of programs which process them [Sal02]. Storage units called entities, which contain parsed or unparsed data, comprise XML documents. Parsed data includes characters, which form either character data or markup. Markup encodes a description of the document's storage layout and logical structure [Sal02].

Expanding areas of XML research include W3C working group efforts for an XML Query (XQuery) data model [CFM+03], and query language [FMM+03]. XML application areas include: database interoperability [Hin01], heterogeneous software systems [You01, TA01], XML schema integration [Hal01b], real-time system data interchange [Pra01],

common data attributes [Zob01], an XML technology assessment [Ber01], and workflow and document management [ACD+02].

In the M&S area, XML uses include: data interchanges for equipment and performance information [LKB01, LPA+02], support of command and control communications requirements [SB00], data management and architecture description [SB02b], the development of integration and collaborative toolsets [GRG+02], support to major simulation development efforts [And02], and scenario generation [RK02]. Web-based M&S techniques employing XML for development and interoperability have also emerged [BZP+02, Hob03, Tol03].

However, [MCF+02] identify challenges in realizing XML's full potential including risks that data will lack definition, incompatible data definitions, and the proliferation of vocabularies and structures. [Ber01] suggest that the lack of agreement on data representations and conceptual data models represent a major obstacle to data interchange among legacy systems, but concludes that:

Data interoperability is feasible without requiring a comprehensive data standard, and that methods for incrementally growing localized standards and bridging the gaps among them without requiring global agreement appear possible. Further assessments are proposed to determine the practical feasibility of applying this approach to DoD operations [Ber01].

6. Software Architecture Description Languages (ADL)

Various Architectural Description Languages (ADL) support current software architecture research and development efforts with the potential for employing common coarser-grained architectural elements (e.g., components and connectors) and interconnectivity schemes for architecture-based development, and featuring formal modeling notations, analysis, and development tools operating on architectural specifications [MT00]. However, the research community currently lacks consensus on ADLs, the capabilities expected from an ADL toolset, and what aspects of a software architecture to model. [MT00, CBB+03] suggest that no existing ADL tool provides the complete capability to document software architectures, although many ADLs perform well in specific areas such as conceptual frameworks, concrete syntax, parsing, displaying, compiling, analyzing, or simulating

architectural descriptions in specific languages. While there is no generally accepted definition²²⁴ of an ADL, [MT00] proposes a definition and classification for ADLs:

An ADL must explicitly model components, connectors, and their configurations; furthermore, to be truly usable and useful, it must provide tool support for architecture-based development and evolution. These four elements of an ADL are further broken down into constituent parts [MT00].

Although lacking a consistent definition, [MT00, CBB+00] cite several domain-specific and general purpose ADLs including ACME, Aesop, Darwin, MetaH, Rapide, SADL, UniCon, and Wright. Emphasizing the lack of a standard definition and scope, [MT00] describe ACME as an architectural interchange language enabling integration of support tools across ADLs, while [CBB+00] categorize ACME as an ADL. [MT00] provide a succinct ADL classification and comparison framework of key ADL properties, identifying capabilities and deficiencies.

E. SUMMARY

Chapter VIII discussed traditional product lines, and introduces software product lines and software architectures. The chapter also reviews core asset development, reverse engineering to develop core assets, reengineering core assets, component development supporting a product line methodology, product line practice areas, and an overview of evolving software architecture theory potentially applicable to improved M&S credibility and reducing heterogeneous system representation anomalies.

Product lines and product line practices are new to software engineering, the Department, and the Department M&S domain. Product line practice areas include software engineering, technical management and organizational management, which may be applicable to meet some or all of the technical, cultural, and managerial challenges collectively hindering the development of improved Department M&S credibility.

A product line approach to developing and deploying software-intensive systems offers great promise for delivering higher quality systems in a shorter time and at reduced cost. A product line approach to software-intensive systems can save money and result in a

²²⁴ Another definition of an ADL is a language (graphical, textual, or both) for describing a software system in terms of its architectural element and the relationships among them [CBB+00].

faster time to field quality systems. A number of organizations gained order-of-magnitude improvements in efficiency, productivity, and quality through a product line approach. Product line practice also enables an organization to get its product to the market more rapidly. As an example, [BC96] provides a case study of a successful product line implementation by the CelsiusTech Systems AB of Sweden in the area of large, embedded real-time shipboard command and control systems.

Any software product line implementation strategy in the Department must take into account the Department is an acquirer of software systems and not the software developer, and factor in product line practice areas supporting the Department's acquisition process. Initial efforts to employ product lines in the Department's M&S domain have been limited.

Components play a major role in many software-intensive systems. There are several motivations for a components-based software approach to software: providing a basis for commerce in reusable software to meet demand, facilitating the development of flexible systems, and reducing the time to design, implement, and deploy systems. As an indication of the need for software components at the United States Federal Government level, the Office of Management and Budget recently established the Federal Enterprise Architecture Program Management Office to remedy the lack of a Federal Enterprise Architecture (FEA), a major barrier to the success of the E-Government initiative. Currently there is not an established basis for how well component models and frameworks contribute to achieving the desired quality; nor is there any basis for assessing the quality of software component interfaces, composition, and component certification. Consensus on the many solutions required for contentious component / composition issues do not appear on the near-term horizon.

A software product line practice area is a body of work or a collection of activities, which an organization must successfully master to carry out the essential work of a software product line. Many, if not the majority of the practice areas describe activities, which are critical to the success of any software project, and provide starting points for organizations to initiate and master activities and measure progress in adopting a product line approach. The practice areas in the framework divide into three categories: 1) software engi-

neering practices, 2) technical management practices, and 3) organizational management practices illustrated in Table 8-1.

Implementing software product lines offers potential gains, and entails significant risks, including technical, organizational, and management issues. [Nor03] explains that building and acquiring a software product line requires disciplined engineering supported by mature technical and organization management processes of universal essential activities and practices, which the SEI developed into an online framework²²⁵, a web-based document describing the competencies needed to develop and field a successful a software product line or any software-intensive system. Based on research to date, the SEI proposes the software engineering practices listed in Table 8-1 as necessary practices supporting an organization's capability and technology to create, evolve, and maintain core assets and products.

Software architecture theory is a rapidly evolving component of software engineering with a wide range of diverse concepts, styles, and views. The evolving software architecture theory body of knowledge includes a wide spectrum of topics: evaluating the quality attributes of a software architecture, reconstructing a software architecture with automated support, reusable components, software architectural transformation, interoperability, attribute-based architectural styles, new XML architecture-focused techniques and applications, emerging architecture description languages, and methods for documenting architectural layers.

The software architecture is key to the success of any software project and critical to the success of a product line initiatives. Architectural drivers, influencing the entire life cycle of a system, include quality attributes, business goals, and system interactions. Product line architectures also include the allowable product variability and the different instantiations of the product, since these too are products. There may also be many views in a software architecture domain, which show specific properties of the software system. These different architecture views address diverse issues encountered with the design process, including the logical view, process view, physical view, and the development view.

²²⁵ Framework in this context suggests a conceptual index, a frame of reference for the information essential for success with product lines [Nor03].

Other terms of references for software architecture design views include behavioral, functional, structural, and data modeling views, however, the key point is that software architecture design is the product of a multi-dimensioned process composed of independent and orthogonal views, impacted by variability, either planned or unintended.

An architectural style (e.g., pattern) is a specialization of element and relation types, together with a set of constraints on their use. In this research effort, an architectural style consists of a set of constraints on the architecture, which define the set or family of architectures, and satisfy them with a number of major styles. These styles may include general structures of pipes and filters, layers, blackboards, object-orientation, and implicit invocation, distributed systems; interactive systems; adaptable systems; and other styles including batch, interpreters, process control, specification-based, and rule-based.

A sample of other recent advances in software architectures noted in Table 8-2 include [Kru95] describing four views of software architecture for system building; the collaborative work of [SNH95] who identified four additional industrial use views; four business views [HS00c], the development of viewpoints [IEE00b] to designate the means used to construct individual views; and the three categories of views identified as viewtypes [CBB+03]. [DN02] suggest that over the past several years, software architecture emerged as the appropriate level for addressing software quality, and that recent systemic efforts of understanding how the design patterns and architectural styles contribute to that quality analysis effort. In their research [DN02] provide a concise and recommended survey on eight representative software analysis methods in different domains, compare and contrast similarities and differences through study, comparison, and classification, and offer guidelines supporting the use of the most suitable method for an architecture assessment process.

Emerging software tools including XML and ADLs support current software architecture research and development efforts with the potential for employing common coarser-grained architectural elements (e.g., components and connectors) and interconnectivity schemes for architecture-based development, and featuring formal modeling notations, analysis, and development tools operating on architectural specifications. However, the research community currently lacks consensus on ADLs, the capabilities expected from an ADL toolset, and what aspects of a software architecture to model.

IX. ANALYSIS OF MODEL CREDIBILITY IN FIVE BMDS M&S

A. INTRODUCTION

Chapter IX provides an overview of the Ballistic Missile Defense System (BMDS), domain M&S, a concise description of the BMDS domain background²²⁶, the BMDS domain M&S hierarchy, a BMDS System-Level M&S synopsis, BMDS M&S demographics, and a review of the BMDS model representations populating the five BMD System-Level simulations under study in this dissertation. The BMD domain overview section provides the Agency background and highlights the significant role Department organizational changes played in the current status of BMD System-Level M&S. Building on the organizational outline and M&S domain overview, the chapter continues with a review on the Agency's implementation of the Department's policies for establishing simulation credibility supporting confidence in BMDS simulation results, and VV&A background information for each of the BMD System-Level M&S.

Summary level information of the other M&S in the domain M&S hierarchy provide additional context for the analysis. A top-level review of the BMD System-Level M&S fidelity, and the foundations for radar sensor [Mac92, Cla93, EB01] fidelity follow. The analysis identifies additional root causes for heterogeneous anomalies in the BMD System-Level M&S. The research methodology supported by the NPS software engineering distance-learning model facilitates the timely study of Department primary source material for software-intensive legacy simulation systems. This case study also employs selected product line practice areas (e.g., Organization Management, Technical Management, Software Engineering) [Coh02] as a tailored framework for the missile defense domain analysis. The scope of this research includes five large-scale, Missile Defense Agency legacy simulations, identified later in the chapter, and supporting appendices.

²²⁶ The JTA includes specific functional domains, separate from the JTA Core standards and guidelines and which may be inappropriate for systems in other domains, only to ensure interoperability within the domain. These domains include subdomains containing JTA elements applicable to systems within that subdomain. The intention of the JTA is for the systems in the subdomains and domains to eventually adopt the JTA elements in the JTA Core standards [JTA02a].

B. SELECTION CRITERIA OF A RELEVANT DOD DOMAIN

The selected Department M&S domain²²⁷, the Ballistic Missile Defense System (BMDS), met the following research criteria: 1) the BMDS domain is instantiated in the Department's Joint Technical Architecture [JTA02a] specifically in the weapon system (e.g., missile defense sub-domain), M&S, and C4ISR²²⁸ (e.g., space reconnaissance sub-domain) domains identified by the bold areas in Figure 9-1; 3) the selected BMDS domain is also currently engaged with multiple international cooperative M&S programs; 4) the BMDS domain is part of the Department's Transformation process [Rum01] with transformation criteria established by [Rum02c, Ald02e]; 5) the BMDS is a high-risk, mission-critical defense capability (Figure 9-2) for the Nation, allies, and friends; 2) the Missile Defense Agency (MDA), responsible for the BMDS is a jointly organized Department domain including Army, Navy, and Air Force systems; 6) the BMDS domain will implement the Department's evolutionary acquisition strategy [Ald02d]; with the acquisition strategy criteria established by [Rum02c, Ald02e]; and 7) the BMDS domain possessed an established hierarchy of M&S, introduced in Figure 9-3, capable of supporting an analysis of the Department's legacy architecture for reverse engineering, reengineering, and reuse (R3).

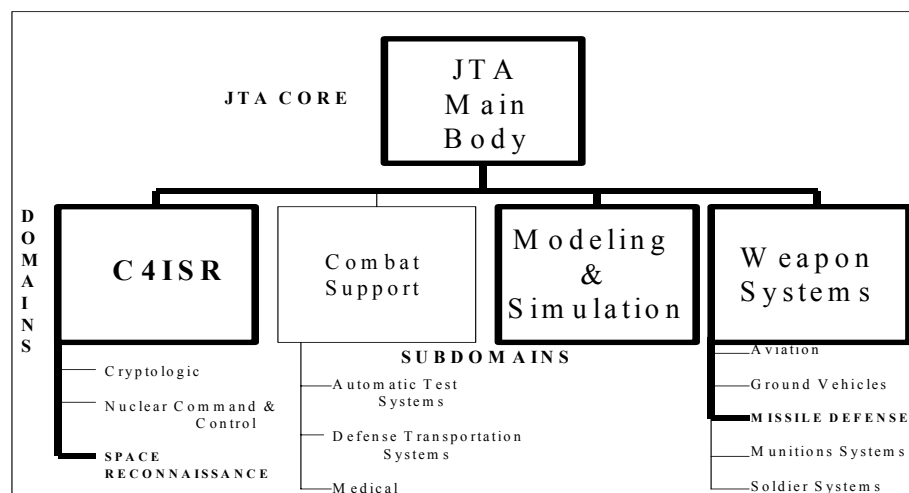


Figure 9-1. JTA Domains Included in this Study (After [JTA02a])

²²⁷ In this context a domain is an area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area.

²²⁸ The acronym C2/BM represents the BMDS C4ISR function in this work.

C. ORGANIZATION MANAGEMENT

1. Domain Overview of the Organization Structure

In order to meet the changing ballistic missile threat to the Nation and adapt from a Cold War monolithic threat environment to the current Post-Cold War asymmetrical threats, the Nation has continued to evolve its missile defense capabilities. This evolution began with President Reagan's Strategic Defense Initiative in 1983, which established the Strategic Defense Initiative Organization (SDIO), continued with the establishment of its successor, the Ballistic Missile Defense Organization (BMDO) [BMD00a], and most recently, the current Missile Defense Agency (MDA) [Rum02c, Ald02e]. As the world geopolitical situation changed in the 1980s and 1990s, national-level policy and defense strategy reordered the priorities and focus of the Agency several times.

In addition, the complicated roles and responsibilities of a joint program integrating service-led major defense acquisition programs (MDAPs), a cumbersome Department requirements generation and acquisition process, international treaty constraints, and an inconsistent budget process affected the BMD system development and the supporting M&S effort. The current state of missile defense system representations in the BMD System-Level M&S, discussed in this chapter, occurred partly as the result of the many organizational impacts and Department process constraints.

Since the 1980's the Agency's models and simulation program reflected changing National missile defense priorities evolving from the bi-polar Cold War environment to counter the myriad of emerging and uncertain threats of the post-Cold War world:

- 1984-1987 – Explore technologies for national ballistic missile defense,
- 1987-1991 – Start acquisition of a phased national ballistic missile defense (NMD),
- 1991-1993 – Acquire a limited global ballistic missile defense (GPALS),
- 1993-1996 – Develop and field a Theater Missile Defense (TMD). Continue NMD as a technology readiness program,
- 1996-2001 – Continue to acquire TMD. Develop a NMD system for possible limited deployment [BMD00a],
- 2002-Present -Develop and field an integrated Ballistic Missile Defense System (BMDS) [Rum02c, Ald02e].

At the turn of the century the Agency managed separate TMD and NMD system architectures²²⁹, and lacked a single unified missile defense system architecture, while the Agency executed three concurrent, and often-competing acquisition strategies:

- Field TMD systems quickly,
- Determine the deployment strategy for the NMD systems,
- Continue to advance the technology [BMD00a].

The Service Components TMD programs, and Joint Program office-managed NMD program developed their respective missile defense system efforts as MDAPs, with the BMDO responsible for integrating the diverse, complex weapon and sensor systems into single, seamless interoperable operational system architecture [BMD00d]. The Department designed the Army's PATRIOT program (e.g., PAC-2, PAC-2 GEM, and PAC-3) and Navy Area Defense (NAD) lower-tier²³⁰ missile defense systems to defend against terminal endoatmospheric²³¹ threats. At higher altitudes, the Department devised the Army's Theater Area Air Defense System (THAAD) and Navy Theater Wide (NTW) missile defense systems²³² to counter upper-tier exoatmospheric²³³ missile threats [BMD00d].

The NAD and NTW missile defense systems evolved from the Aegis Weapon System's primary mission as a fleet air defense system [BMD99a]. Other missile defense efforts included the Airborne Laser (ABL)²³⁴ program, an Air Force managed system. The PAC-3, THAAD, and NTW systems employed hit-to-kill (HTK)²³⁵ technology. Appendix A summarizes the mission, organization, technical, acquisition, management, procedures processes supporting the Agency's evolution and organizational responsibilities in the 1996-2001 timeframe.

The Agency also supported three major international collaborative programs including an international cooperative program with Germany and Italy, the Medium Extended Air Defense System (MEADS)²³⁶. The Israeli ARROW program [BMD00j], now deployed

²²⁹ System architecture in this context is the software architecture plus execution and development environments.

²³⁰ The PAC-3 and NAD missile defense weapons employed against endoatmospheric threats are referred to as lower-tier systems.

²³¹ Endoatmospheric refers to being within the Earth's atmosphere; generally considered to be altitudes below 100 km [MDA02b].

²³² THAAD and NTW are also called mid-course systems

²³³ Exoatmospheric refers to being outside the Earth's atmosphere; generally considered to be altitudes above 100 km [MDA02b].

²³⁴ The ABL concept employed a multimegawatt chemical laser to destroy missiles in the boost phase [BMD00a].

²³⁵ Hit-to-kill technology (HTK) employs kinetic energy to destroy the target [BMD00a].

²³⁶ MEADS was conceived to provide deployed NATO maneuver forces with 360 degree protection [BMD00a].

and operational is another international missile defense effort developed with U.S. support, and employs a high-explosive warhead rather than the HTK technology designed into U.S. missile systems. The Russian-American Observable Satellite (RAMOS) is a cooperative effort to observe the earth's atmosphere and ballistic missile launches.

However, as the United States enters the 21st century, the dynamic geo-strategic environment no longer checked by bi-polar world power control presents new challenges. New and uncertain 21st century threats such as missile attacks from rogue nations, terrorism, and weapons of mass destruction (WMD), brutally driven home to the American Nation on September 11, 2001, eclipse the Cold War focus on a massive Soviet missile attack.

2. A MISSION-CRITICAL SYSTEM IN THE NATIONAL DEFENSE

On January 2, 2002, the Secretary of Defense [Rum02c] established the Missile Defense Agency (MDA), revised the agency concept of operations²³⁷, and directed the initiation of a single joint program to develop an integrated ballistic missile defense system (BMDS). [Rum02c] also directed a capability-based requirements process, supported by the full and cooperative efforts of the Services, Joint Staff, and defense agencies to achieve the objectives of this National priority (Appendix B).

The Department's organizational change of the BMDO to the MDA was part of a transformation process to replace an overly restrictive, non-responsive, and overly prescriptive requirements and acquisition process [Ald02e]. These significant policy changes removed the communication barriers with the former MDAPs, and permitted a two-way dialogue establishing a foundation for improved system representations in the BMDS System-Level M&S. However, the creation of the BMDS also added: an expanded mission, a wider scope of ballistic missile defense responsibility, many new complex program requirements, and an accelerated milestone schedule placing renewed emphasis on the development of a credible BMDS M&S program.

Nearly one year later on December 17, 2002, President Bush identified the important role of missile defense to the Nation, friends, and allies, and directed the Secretary of

²³⁷ [Rum02c] described the new MDA organizational structure, roles, responsibilities, processes, and policies that detailed the way the MDA operates.

Defense to “proceed with the fielding of an initial set of missile defense capabilities” [Bus02f]. In addition the President made the following statement underscoring the mission-critical nature of missile defense:

The deployment of missile defenses is an essential element in our broader efforts to transform our defense and deterrence policies and capabilities to meet the new threats we face. Defending the American people against these new threats is my highest priority as Commander-in-Chief, and the highest priority of my Administration [Bus02f].

[Rum02c] requires a single missile defense program to design, develop, and test the BMD system elements (e.g., the former MDAPS) of an integrated BMDS to defend the U.S., deployed forces, allies, and friends, with a BMDS that layers defenses to intercept missiles in all phases of their flight phases (boost, midcourse, and terminal) against all ranges of threats [MDA02a, MDA02c, MDA02b, MDA02e, MDA02f, MDA02h, MDA02j, MDA02m, MDA02x, MDA02y, MDA03b]; support the fielding of element capabilities, such as PATRIOT Advanced Capability-3 (PAC-3) system, as soon as practicable; develop, test technologies and improve missile defense test ranges [Pat99, TRG01]; and provide early capability, if necessary, inserting new technologies as they become available, or when the threat warrants [Bus02e, Rum02c, Kad02b, Kad02c, MDA02i, MDA03e]. The Under Secretary of Defense (AT&L) [Ald03] noted, “space and missile defense is central to the future of our national security” [Ald03], reconfirming missile defense as a mission-critical requirement for the Nation.

In addition to the HTK interceptor technology and directed energy weapons, the Agency is also developing sensor systems that will improve the ability to detect, track, and identify ballistic missile warheads from countermeasures, which will be integrated in the BMDS through a Battle Management / Command and Control (BM/C2) system. The Agency is also exploring advanced weapon capabilities including space-based lasers, and sea- and space-based kinetic energy systems [MDA02p].

Figure 9-2 illustrates the major BMDS functions required of an integrated collection of defense-in depth capabilities (e.g., detection, identification, classification, battle management, sustainment, engagement, kill assessment) supporting the Boost Phase Defense Segment (BDS), Midcourse Defense Segment (MDS), and a Terminal Defense Segment

(TDS) (e.g., the missile defense kill-chain). A planned Missile Defense Test Bed in the Pacific will support these three segments with additional test realism, the ability to support multiple engagements, and provide a limited contingency capability. Appendix B provides a synopsis of U.S. missile defense acquisition programs, international collaborative efforts, technology programs, and the BM/C2, weapon, sensor, and communication components of these programs in 2002.

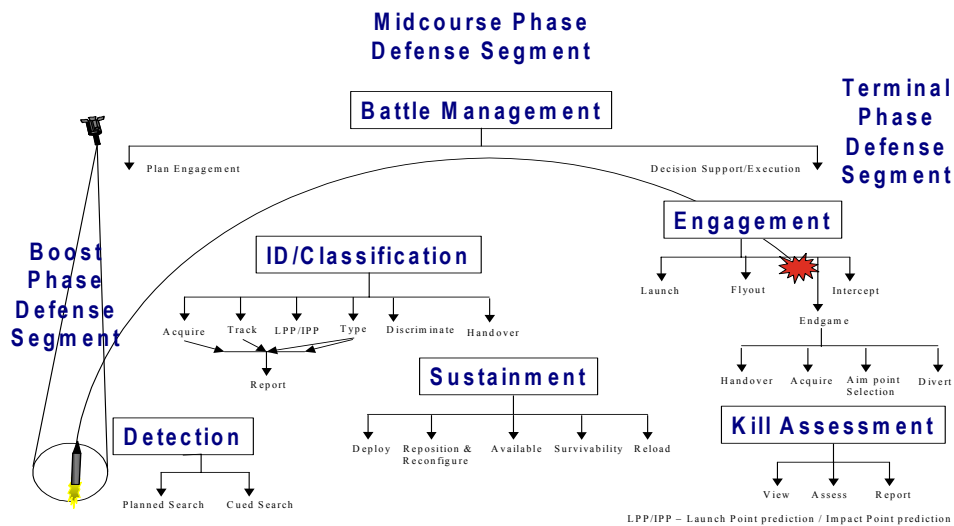


Figure 9-2. BMDS Defense Phase Segments and Required Capabilities (After [Ray01c])

3. A HIGH-RISK, SOFTWARE-INTENSIVE SYSTEM

Missile defense has other hurdles including the software challenge identified by Lieutenant General George Monahan for the SDI program in his 1990 report to the Secretary of Defense when he noted,

The greatest engineering, vice technical, challenge in the SDI program is software...[BMD00a].

BMDS software-intensive systems will be complex, expected to support system evolution, perform the most difficult tasks, including battle management, recover from software and hardware failures, and respond correctly to anticipated and unanticipated threats to the system. Critics and supporters agree that software errors will occur in a system as complex as

the BMD, but argue over whether the failures would be catastrophic²³⁸ [Bow02]. [Par01] believes the development of dependable, trustworthy software-intensive systems for the BMDS is a very high-risk undertaking and may very well be impossible.

Correctly assessing and meeting the many challenges for developing quality software-intensive systems supporting systems engineering/integration, BMD System testing, operations, and simulation development with credible verified and validated software products is on the critical path to fielding an integrated missile defense system. In addition to supporting the evolutionary acquisition of the BMDS program and fielding of block capabilities every two years, the evolving BMD M&S program will also support all missile defense life cycle support requirements.

D. BMD SYSTEM M&S DOMAIN OVERVIEW

The Agency M&S program [MDA03M] supporting the development of the BMDS evolved from the large-scale, legacy M&S development efforts of previous missile defense programs [SCC+88, SW96]. The current M&S programs has three categories mapped to the Department's M&S Hierarchy in Figure 2-2. The three categories of the Agency's four-level M&S Hierarchy illustrated in Figure 9-3 are the: BMD System Threat, Signature, Environment, and Lethality (TSEL). The BMD System TSEL M&S, at the bottom layer of the Agency's four-level hierarchy support the development of the BMD Element-Level components/sub-systems/systems, at the second layer of the hierarchy. The BMD Element-level M&S directly support the Element programs. The BMD System-Level M&S at the third level of the hierarchy M&S supports the system-wide integration and interoperability of the element representations into the BMD system.

The BMD System-Level M&S and the BMD System TSEL comprise the BMDS Core Model set. The BMD System-Level M&S is the only category addressed in detail within this dissertation, while the Element-Level M&S, and BMD System Threat, Signature, Environment, and Lethality M&S are summarized at a very high level. See Chapter

²³⁸ Several meanings of catastrophic have been developed, but [SCC+88] defined a catastrophic failure as a decline in system performance to a 10 percent or less of expected performance.

VII for a summary of the Department's Joint simulation development program, (e.g., JWARS JSIM, and JMASS), the Agency's capstone M&S category shown in Figure 9-3.

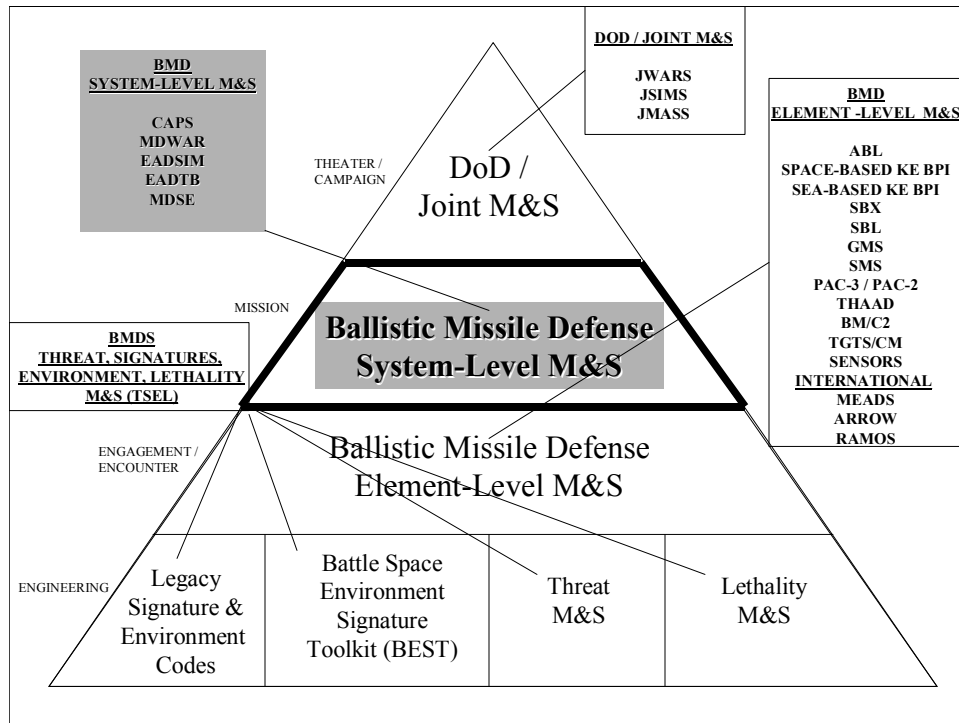


Figure 9-3. BMD M&S Addressed in this Study (After [PMR97])

1. BMDS Legacy Model and Simulation Systems

A major MDA objective is to provide credible M&S and improve the Department's confidence in BMDS simulation results [Kad02a, Kad02b, Kad02c]. However, the BMDS M&S program has four major challenges to overcome in support of the vision to make missile defense a reality: 1) developing credibility in the M&S and building user confidence in the results, 2) accurately modeling the significant technological hurdles inherent in missile defense, 3) dealing with major acquisition program complexity, and 4) the software engineering challenge itself.

The Agency and its predecessor Service / Agency / Component organizations developed an expensive portfolio of M&S including large-scale, legacy simulation systems: Adage, Carmonette, and COMO III [RBB+82, Che87] to support the air and missile de-

fense mission. Many of these M&S evolved in an *ad hoc* manner, evolving into large systems with multiple stakeholders and significant support infrastructures. A significant number of the existing systems lacked formal documented conceptual models and reflected an inconsistent V&V history [RRB+82, Che87].

In addition, the number of M&S tools in the missile defense domain proliferated in the 1990's until there were nearly three hundred M&S systems competing for BMDO M&S life cycle funding and support [BMD99b]. The Agency's legacy M&S systems supported experiments, analysis, studies, and tests with little documented V&V, resulting in reduced credibility of the accreditation process for these simulations, and limiting confidence in experiment, analysis, study, or test results [RRB+82, Che87].

In a series of agency-level M&S reviews conducted during 1998 and 1999, redundant, low-use M&S, and simulation systems with a low potential for future integration or HLA interoperability were discontinued, while approximately eighty-eight M&S tools were determined to be MDAP-unique²³⁹ and management responsibilities were assigned to the former major defense acquisition programs (e.g., PATRIOT, THAAD) [BMD99b]. The MDAP-unique tools currently constitute the MDA Element-Level M&S category of the MDA M&S Hierarchy in Figure 9-3. The remaining common-, and general-use legacy tools, supporting multiple internal and external MDA stakeholders, employed in all phases of the BMD system evolutionary program development (e.g., RDT&E, Transition, Operations & Maintenance phases), and integrated into multiple functional areas (e.g., analysis, training, experimentation, acquisition, and operations), became the BMDS Core M&S. Figure 9-3 illustrates the two categories of the BMDS Core M&S: the BMDS Threat, Signatures, Environment, Lethality and Threat (TSEL) category and the BMDS System-Level M&S.

2. BMDS System-Level Legacy M&S Systems

The five simulations comprising the MDA System-Level M&S layer of the MDA M&S Hierarchy [SW96] shown in Figure 9-3 comprise the scope of this research. The systems under study are: the Commanders Analysis and Planning Simulation (CAPS), the

²³⁹ MDAP-unique M&S were determined to be valid requirements for an individual MDAP, but did not have system-wide applicability.

Missile Defense Wargame and Analysis Repository (MDWAR), the Extended Air Defense Simulation (EADSIM), the Extended Air Defense Testbed (EADTB), and the Missile Defense System Exerciser (MDSE). Theoretically, the BMD System-Level M&S in this layer include representations of the Element-Level systems at varying degrees of fidelity, accuracy, precision, and resolution. In the future, the BMD System-Level M&S will in turn support development of valid BMDS representations in the Department's capstone-level joint M&S.

The BMD System-Level M&S support environments²⁴⁰ are repositories of missile defense domain expertise, critical intellectual property, and the foundation for the development of future components in the missile defense domain core asset portfolio. This constituency of knowledgeable stakeholder supporters is almost always lacking in the development of new systems, where requirement-stakeholders often compete for priority, funding, and allocation of development resources. The MDA also supports external stakeholders including the Department's operational test community, National-level decision-makers, and international cooperative partners.

a. Commanders Analysis and Planning Simulation (CAPS)

The CAPS model is a theater level, force-on-force, missile defense system analysis tool, and qualitatively considered low-fidelity. CAPS simulates active theater defense against ballistic missiles and air breathing threats and is employed as a planning, training and analysis tool with four views: footprint view, operating area view, defended area view, and scenario view. These capabilities illustrate defensive footprints, or areas on the ground that may be protected by missile defense units, possible operating regions for defensive systems, areas defended by various positioning of defensive forces, and expected performance of missile defense systems in a campaign. See Appendix C for additional information.

²⁴⁰ The support environment includes the multiple stakeholders, both government and private sector, who manage, develop, test, V&V, use, or support the M&S development process.

b. Missile Defense Wargame and Analysis Resource (MDWAR)

The MDWAR simulation, formerly called WARGAME 2000, is a considered a low-medium fidelity simulation designed to provide the missile defense community with a Human-In-Control (HIC)²⁴¹ wargaming capability. MDWAR provides a constructive M&S tool to develop missile defense concepts of operation (CONOPS), doctrine, tactics, techniques and procedures²⁴² (TTPs) through the use of virtual experimentation in a synthetic environment. MDWAR is the successor system to ARGUS simulation, developed to provide a real-time, discrete-event simulation capability supporting the development of missile defense [TOM99]. MDWAR also supports missile defense architecture evaluation, joint missile defense exercises, and the development of the BMDS BM/C2 element.

MDWAR HIC experiments interactively simulate the complex TMD and NMD threat environment with scenarios designed to induce the effects the real world confusion to the operators manning realistic operator consoles and displays. The objectives of the MDWAR war games are to exercise command and control (C2) processes in a HIC environment with realistic battle management defense system features that enable the operators to perceive and interpret the battlefield situation, identify problems, develop courses of action, and allows operators to select, implement and monitor the corrective action and its impact on the situation. See Appendix D for additional information.

c. Extended Air Defense Simulation (EADSIM)

EADSIM is a constructive simulation first released in 1989 [JAS97c] as a system-level simulation providing a many-on-many theater-level simulation of air, space, and missile warfare. EADSIM is generally considered a medium fidelity simulation. EADSIM models joint and combined force air and missile defense warfare, ranging from few-on-few to many-on-many theater scenarios, to determine the effectiveness of air and missile defenses against the full spectrum of threats as an analysis tool and to augment

²⁴¹ Also referred to as human-in-the-loop (HIL)

²⁴² Computer-based TTP, CONOPS, and doctrine development supports the force and combat development functions of the services and joint forces and provides an early opportunity for the war fighter to influence the system development and employment. TTPs are also reviewed early in the material development process to see if a change to TTPs CONOPS, or doctrine negated the need for a materiel solution.

training. EADSIM supports the four pillars of theater missile defense: active defense, passive, defense, attack operations, and BM/C2. EADSIM also models the command and control (C2) decision processes, intelligence gathering, and the inter-platform communication process. See Appendix E for additional information.

d. Extended Air Defense Testbed (EADTB)

The EADTB simulation, an event-driven, Monte Carlo constructive simulation framework, supports missile defense modeling from the fire-unit to the theater level [BMD01]. EADTB evaluates the effectiveness and efficiency of weapon systems against targets and the value of different force structures and resources. The EADTB system design offers object-based²⁴³ simulation architecture with the flexibility to use high- or low-detail, user-developed, specific system representations (SSRs)²⁴⁴, for play on a simulated game board. [SMP98, Ray02a]. EADTB, considered medium-to-high fidelity is capable of supporting a wide range in the level of SSR detail in a single simulation exercise, running interactively or in batch mode, and provides analytical flexibility, with the objective of reducing the need for multiple simulations [BBB+96]. See Appendix F for additional information.

e. Missile Defense System Exerciser (MDSE)

MDSE, originally called the Theater Missile Defense System Exerciser (TMDSE) [Too94] addressed the objectives, architecture and engineering considerations of the theater level missile defense System-Level hardware-in-the-loop capability [BB98a, BB98b]. MDSE is the highest fidelity BMD System-level simulation. MDSE was originally designed to demonstrate the interoperability of theater ballistic missile defense (TBMD) weapon systems and national sensor elements in a real-time, GPS-synchronized, centrally controlled, geographically distributed hardware-in-the-loop (HWIL) tool to test the TBMD family of systems (FoS) in a dynamic environment [Too94]. The original MDSE development concept specified by [Too94] identified the requirements for the first

²⁴³ EADTB is object-based, versus object oriented, since it does not support inheritance.

²⁴⁴ SSRs are data and code rule sets developed to represent objects (THAAD, PATRIOT) and control their behavior [BBB+96].

three TMDSE builds and called for an incremental build schedule to develop a test resource supporting interoperability and integration tests of the theater missile defense architecture. The GMD (formerly NMD) architecture was not a part of the original TMDSE requirements, and will be addressed in future MDSE build requirements. See Appendix G for additional information.

f. BMD System-Level M&S Status

These simulations experienced different growth patterns. In the case of CAPS and EADSIM the documented start dates for simulation development do not reflect earlier development efforts. For instance, CAPS evolved from the Operation Planning Simulation (OPS) [BBG+99f], while EADSIM built on the foundation of the earlier C3ISIM simulation [BBG+99e]. In comparison to the CAPS and EADSIM evolutionary development, MDWAR and MDSE were more deliberate designs. The five BMD System-Level simulations under study are currently in the maintenance phase of the M&S software life cycle and executing to a stable program plan²⁴⁵.

E. BMD SYSTEM-LEVEL SIMULATION MODEL REPRESENTATIONS

This section provides the analysis results illustrating the status of acceptable BMD System representations within the five BMDS System-Level M&S (e.g., CAPS, MDWAR, EADSIM, EADTB, and MDSE). The analysis includes the following missile defense representations:

- Missile Defense Weapons – Terminal Defense Systems (TDS), Midcourse Defense Systems (MDS), and International,
- Directed Energy Weapons – Airborne- and Space-based Boost Defense Systems (BDS),
- Kinetic Energy Weapons – Sea- and Space-based Boost Defense Systems (BDS),
- Radar Sensors - Terminal, Midcourse, and International systems,
- Satellite / Infrared Sensors,
- Airborne Laser Sensors,
- Platform Systems
- Geo-Spatial Environment representations.

²⁴⁵ The current program plan for the M&S under study shows a steady state program of resources, factoring in inflation, for most of the decade. Any significant decrements to the projected program will extend the time needed to improve the credibility of the BMDS M&S.

For the purpose of this research an acceptable authoritative representation met the following criteria: the appropriate program office previously accredited the representation for a specific use, or the program office developed and documented the representation with an understanding of the representations' planned intended use, or the program office reviewed and accepted the system representation, with specific caveats for some documented intended use. Documentation supporting the PAC-3 Initial Operational Test and Evaluation (IOT&E) Interoperability Demonstration (PIID) [CRC02a, Bar02a, CRC02b, Bar02b] conducted in April 2002 proved invaluable for the purpose of assessing representation acceptability. Based on this criteria, a "Y" in the following tables indicates evidence of at least one Acceptable System Representation, and an "N" indicates an Unacceptable / Missing System Representation. These statistics do not state, imply or suggest a blanket approval of system representation certification, accreditation, or validation by the respective MDA element program office for any future uses.

1. BMD System-Level M&S TDS Weapon System Representations

In the current TDS population of six missile systems identified in Table 9-1, five systems (83%) have acceptable representations of the specific missile in the BMD System-Level core simulations. One specific representation configuration of the THAAD EMD missile system is currently available at a very high level of aggregation in CAPS. Overall, eighty-seven percent (26/30) of the TDS missile representations are included in the BMD System-Level M&S. This is the highest percentage of weapon system representations found in the BMD System-Level M&S. Only the TDS sensor system representations, discussed later in the chapter, exceed this metric in the System-Level models. The relatively high percentage of missile representations within the TDS may be attributed to the following:

- A focus on terminal-phase systems (PATRIOT, THAAD, Aegis) during the development phase of the simulations,
- An emphasis on interoperability among the different terminal-phase systems,
- The requirement for a Single Integrated Air Picture (SIAP) including theater ballistic missiles, cruise missiles, and aircraft, both enemy and friendly,

- A missile defense program priority during most of the 1990s to defend U.S. forces and allies in a theater-wide operational scenario,
- The PATRIOT and the Aegis terminal defense weapon systems were the most mature missile defense systems,
- A higher probability of M&S reuse within a mature program office with a family of related systems, such as the PATRIOT program office,
- Support contractor longevity and domain expertise for different generations of similar missile system (e.g., PAC-2, PAC-2 GEM, PAC-3) [BRC+01].

Of significance, a specific system representation based on correct specifications for the system at certain stage of the development cycle, even with an accurate verified and validated conceptual model may be completely invalid for another intended use. Table 9-1, illustrates this apparent inconsistency of specific system representations with the availability of the THAAD Program Definition and Risk Reduction (PDRR)²⁴⁶ phase missile system version and the absence of the THAAD Engineering and Manufacturing (EMD²⁴⁷) phase missile system within the BMD System-Level simulation systems.

Missing or inaccurate system characteristics or critical attributes [MDW+01] may adversely impact interoperability among the federation entities resulting in substantive interoperability issues including representational accuracy and internal sensitivity problems. There are several plausible reasons for the situation including a lack of coordination for a federation conceptual model, inadequate requirement management, poor configuration management, lack of system information, non-responsive contract support, funding shortfalls, and scheduling. Whatever the root cause, the lack of specific valid system representations may adversely impact the simulation or federations intended use and / or provide decision-makers with inaccurate simulation results.

BMDS TDS MISSILE SYSTEM WEAPON REPRESENTATIONS					
TDS MISSILES	CAPS	MDWAR	EADSIM	EADTB	MDSE
PAC-2	Y	Y	Y	Y	Y
PAC-2 GEM	Y	Y	Y	Y	Y
PAC-3	Y	Y	Y	Y	Y
THAAD PDRR	Y	Y	Y	Y	Y

²⁴⁶ PDRR is the early Program Definition and Risk-Reduction Phase of a system's development.

²⁴⁷ EMD systems have progressed beyond PDRR, with more mature engineering specifications supporting manufacturing of the system.

THAAD EMD	Y	N	N	N	N
AEGIS SM-2 BLK IVA ²⁴⁸	Y	Y	Y	Y	Y
# Avail /Total/ % Avail	6 / 6 / 100%	5 / 6 / 83%	5 / 6 / 83%	5 / 6 / 83%	5 / 6 / 83%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-1. Terminal Defense Segment Missile Representations

The BMDS System-level M&S program also supports two significant international simulation development activities cited in Table 9-2, and an international collaborative M&S program detailed in Tale 9-17. Systems such as the Israeli ARROW Weapon System are deployed operational systems with real-world interoperability requirements, while MEADS is a joint cooperative German, Italian, and U.S. missile defense development program.

BMDS INTERNATIONAL TDS MISSILE WEAPON SYSTEM REPRESENTATIONS					
INTL MISSILES	CAPS	MDWAR	EADSIM	EADTB	MDSE
ARROW (ISRAEL)	Y	Y	Y	Y	Y
MEADS (GE, IT, US)	N	Y	N	N	N
1 / 2 / 50%	2 / 2 / 100%	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%	
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-2. International Terminal Defense Segment Missile Representations

2. BMDS System-Level M&S MDS Weapon System Representations

The MDS missile system weapon representations listed in Table 9-3 are under-represented in the BMD System-Level M&S. There are only four weapon system representations of the GMD ground-based interceptor (GBI) and one Navy Leap ALI missile system, based on parameter files, resident in the BMD System-Level M&S. CAPS,

²⁴⁸ All AEGIS SM-2, weapon systems in the Navy Area Defense (NAD) program in 2001 were counted as TDS representations. With the evolution of the Navy Theater Wide (NTW) program into the midcourse defense segment (MDS) the AEGIS SM-3 and LEAP ALI were counted in the MDS category for this study.

MDWAR, and EADSIM, three of the relatively aggregated M&S include the Aegis SM-3 missile. However, acceptable SM-3 missile representations are not yet available in the EADTB constructive simulations or in the MDSE hardware in-the-loop tool. Reasons for the lack of MDS missile representations in the BMD System-Level M&S include the following:

- National-level mission defense priorities shifted several times during the 1990s perturbing the acquisition strategies, funding levels, and priority of the different systems and their associated M&S programs,
- The GMD (former NMD) program was developed as a single joint program with a specific mission versus the family of systems (FoS) approach employed for theater level missile defense systems requiring interoperability,
- The GMD/NMD mission was to defend the U.S. against ICBM-class missile threats, as opposed to the FoS TAMD / TMD / TBMD mission against shorter range missile, aircraft, and cruise missile threats in a theater of operations,
- The GMD/NMD architecture was not required or designed to be interoperable with the FoS TAMD / TMD / TBMD architecture,
- With the exception of MDWAR, the GMD/NMD missile representations were not included in the development of the legacy BMD System-Level M&S,
- The GMD/NMD program developed the BMD System-Level M&S equivalent incorporating model representations of the former NMD component systems (e.g., UEWRS, DSP, BM/C2) in the GMD/NMD system architecture
- Overall, Table 9-3 shows only forty percent (8/20) of the MDS missile system weapon representations are included in the BMD System-Level M&S, with no acceptable MDS missile system representations available in EADTB or MDSE.

BMDS MDS MISSILE SYSTEM WEAPON REPRESENTATIONS					
MDS MISSILES	CAPS	MDWAR	EADSIM	EADTB	MDSE
GMD GBI I W/EKV	Y	Y ²⁴⁹	N	N	N
GMD GBI II W/EKV	Y	Y	N	N	N
AEGIS SM-3 BLK 1A	Y	Y	Y	N	N
NAVY LEAP ALI	Y	N	N	N	N
# Avail /Total/ % Avail	4 / 4 / 100%	3 / 4 / 75%	1 / 4 / 25%	0 / 4 / 0%	0 / 4 / 0%

²⁴⁹ The GBI model is a combination of fly out tables and Keplerian propagation with the boost phase GBI modeled as a unitary or single stage interceptor defined by a two-dimension fly out table. Interceptor burnout conditions are extracted from the flyout tables and modified to compensate for earth's influence. A point-in-space navigation method is employed because the two-dimensional fly out tables cannot represent trajectory shaping. Constraints include endgame maneuver and the interceptor seeker is not explicitly modeled, although a field-of-view is calculated. Intercept occurs at the time of closest proximity to the target with intercept outcomes modeled with two probabilistic draws: probability of hit (Ph) and probability of kill (Pk), and do not represent complex functions of intercept geometry, velocity, or closing angle [TOM99].

Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-3. BMDS Midcourse Defense Segment Missile Representations

Major missile defense weapon system platforms (e.g., Aegis) must also accommodate multiple configurations of weapons, sensors, battle management, or communication systems during the same timeframe. This is an example of real world interoperability challenges. It is also potential root cause for substantive interoperability issues in federations involving the real-world AEGIS SM-3 BLK 1A and NAVY LEAP ALI systems and their respective system representations in CAPS, MDWAR, and EADSIM illustrated in Table 9-3. In the example highlighted in Table 9-3, there is the potential for inadvertently introducing the incorrect or “almost good” enough missile weapon system representation into a federation, with a potential to produce results similar to the Ariane 5 mishap.

3. BMD System-Level M&S BDS Weapon System Representations

Supporting the Department’s transformation process, evolutionary acquisition strategy, and an overall reduction in product cycle time, the Department’s M&S programs must also transform and reduce simulation development cycle time, especially for new and proposed systems such as the BDS. Under the Block Development concept supporting the fielding of missile defense capability by blocks in two-year intervals (e.g., Block 2004, Block 2006...Block 2016) there is a requirement to support the system engineering process with an M&S capability providing credible results across the temporal spectrum. This would suggest that as the Department system engineering process and M&S processes mature together, the system engineer would allocate future M&S requirements, including fidelity, resolution, precision, accuracy attributes [MDW+01] to future blocks and the simulation community would respond with quality products supporting credibility and confidence.

The next category of BMDS model representations is the BDS Directed Energy Weapon system category. In the BMD system boost defense segment today there is a shortfall of BDS directed energy (Table 9-4) and kinetic energy (Table 9-5) weapon system representations, with the exception of the Airborne Laser (ABL) representations. The PLASTR and ISAAC models represent the ABL system in CAPS [BBG+99e], while the ABL representation in MDWAR is instantiated through a DIS gateway with the ABL operator-in-the loop simulation residing at the Theater Air Command and Control Simulation Facility. Overall, only thirty-three percent of BDS directed energy system representations are currently available in the BMD System-Level M&S.

BMDS BDS DIRECTED ENERGY WEAPON SYSTEM REPRESENTATIONS					
<u>DIRECTED ENERGY</u>	CAPS	MDWAR	EADSIM	EADTB	MDSE
AIRBORNE LASER	Y	Y	Y	N	N
SPACE-BASE LASER	N	N	N	N	N
# Avail /Total/ % Avail	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%	0 / 2 / 0%	0 / 2 / 0%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-4. Boost Defense Segment Laser Representations

The kinetic energy systems envisioned for the future of missile defense and listed in Table 9-5 have no acceptable representations available, and are maybe the most challenging for the unprecedented missile defense system development effort. Unlike the earlier TDS and MDS systems, which built on previous legacy missile defense systems, sub-systems, and domain bodies of knowledge, future system capability development faces many unknowns. Conversely, since the future may hold fewer constraints from current stakeholders and provide more trade space flexibility, the system engineer has the potential to evolve the system into areas with the most promise, assuming that the BMDS M&S development process is responsive, timely, and credible.

BMDS BDS KINETIC ENERGY WEAPON SYSTEM REPRESENTATIONS					
KINECTIC ENERGY	CAPS	MDWAR	EADSIM	EADTB	MDSE
SEA-BASE KINETIC	N	N	N	N	N
SPACE-BASE KINETIC	N	N	N	N	N
# Avail /Total/ % Avail	0 / 2 / 0%	0 / 2 / 0%	0 / 2 / 0%	0 / 2 / 0%	0 / 2 / 0%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-5. BMDS Boost Defense Segment Kinetic Energy Representations

4. BMD System-Level M&S Weapon System Representation Summary

Currently, there are fourteen U.S. BMD weapon systems in four separate categories (e.g., six TDS missile systems, four MDS missile systems, two directed energy systems, and two kinetic energy systems) and two international weapon systems modeled across the five BMD System-Level M&S. Projecting future requirements for US-only weapon system representations in the BMDS weapon system category, and assuming the current set of weapon systems, there is a potential requirement for seventy weapon representations in every block build (e.g., Block 2004, Block 2006...Block 2016). Assuming a requirement by the BMDS system engineering stakeholders for future block system representations, and the need to retain configuration-managed copies of past and present system representation configurations, the potential population for credible BMDS weapon system representations may soon number in the hundreds. This suggests the need for a methodology to manage model representation variability with a disciplined configuration management process.

Possible future requirements for additional acceptable BMDS representations are daunting when a review of the Block 2004 representation inventory reflects only thirty-six acceptable TDS, MDS, and BDS weapon system representations, or fifty-one percent of acceptable system representations available. The international weapon system programs, Arrow and MEADS, have sixty percent (6 for 9) acceptable system representations. These statistics do not indicate any level of weapon representation certification or validation by the respective program office.

5. BMD System-Level M&S Sensor System Representations

The BMDS terminal defense segment radar sensors (Table 9-6) have the highest percentage of system representations in the System-Level M&S with a one hundred percent fill rate. The rationale for this high rate is the same reason provided earlier in this section for the TDS weapon systems. However, as with the missile systems weapon category there is a significant decrease in system representations in the MDS (Table 9-8) and BDS (Table 9-9) segments, with the largest void in three simulations, EADSIM, EADTB, and MDSE.

BMDS TDS RADAR SENSOR SYSTEM REPRESENTATIONS					
<u>TDS RADARS</u>	CAPS	MDWAR	EADSIM	EADTB	MDSE
PAC-2 AN/MPQ-53	Y	Y	Y	Y	Y
PAC-3 AN/MPQ-65	Y	Y	Y	Y	Y
TPS-59 (USMC)	Y	Y	Y	Y	Y
TPS-75 (USAF)	Y	Y	Y	Y	Y
AEGIS SPY-1B/D	Y	Y	Y	Y	Y
# Avail/Total/ % Avail	6 / 6 / 100%	6 / 6 / 100%	6 / 6 / 100%	6 / 6 / 100%	6 / 6 / 100%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-6. BMDS TDS Radar Sensor System Representations

The availability of CAPS and MDWAR sensor representations for the International TDS radars in Table 9-7 reflect the relatively low fidelity of radar system representations in CAPS and MDWAR. The ARROW sensor representations in EADTB and MDSE support international interoperability exercises and testing. Although the current fill rate for International TDS radar representations is sixty percent overall, the international M&S program is expected to grow significantly requiring the rapid addition of system representations of all types.

BMDS INTERNATIONAL TDS RADAR SENSOR SYSTEM REPRESENTATIONS					
<u>INTL TDS RADARS</u>	CAPS	MDWAR	EADSIM	EADTB	MDSE
ARROW RADAR	Y	Y	Y	Y	Y
MEADS FIRE CNTRL	Y	Y	N	N	N

MEADS SURVLNCE	Y	Y	N	N	N
# Avail /Total/ % Avail	3 / 3 / 100%	3 / 3 / 100%	1 / 3 / 33%	1 / 3 / 33%	1 / 3 / 33%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-7. BMDS International TDS Sensor System Representations

The next three BMDS sensor system representation tables represent radar, satellite / infrared, and ABL-specific sensor systems. Table 9-8 through Table 9-9 develop a similar pattern, high availability of representations at the lower-fidelity spectrum and few representations available for the higher-fidelity simulations, EADTB and MDSE. Table 9-8, MDS radar sensor representations achieve only a fifty percent fill rate (15/30).

BMDS MDS RADAR SENSOR SYSTEM REPRESENTATIONS					
<u>MDS RADARS</u>	CAPS	MDWAR	EADSIM	EADTB	MDSE
THAAD PDRR	Y	Y	Y	Y	Y
THAAD EMD	Y	Y	Y	N	N
COBRA DANE	Y	Y	N	N	N
UEWR	Y	Y	N	N	N
SEA-BASE XBR	Y	N	N	N	N
GBR/XBR	Y	Y	N	N	N
# Avail /Total/ % Avail	6 / 6 / 100%	5 / 6 / 83%	2 / 6 / 33%	1 / 6 / 17%	1 / 6 / 17%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-8. BMDS MDS Radar Sensor System Representations

Table 9-9, BMDS satellite / infrared sensors, account for only forty five percent (9/20) of the possible sensor representations.

BMDS BDS SATELLITE / IR SENSOR SYSTEM REPRESENTATIONS					
SATELLITES / IR	CAPS	MDWAR	EADSIM	EADTB	MDSE
DSP SATELLITES	Y	Y	Y	Y	Y
SBIRS-HIGH	Y	Y ²⁵⁰	N	N	N
SSTS (SBIRS-LOW)	Y	Y	N	N	N
RAMOS (US/RF Program)	N	N	N	N	N
# Avail /Total/ % Avail	3 / 4 / 75%	3 / 4 / 75%	1 / 4 / 25%	1 / 4 / 25%	1 / 4 / 25%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-9. BMDS BDS Satellite / Infrared (IR) Sensor System Representations

Table 9-10 lists the Airborne Laser (ABL) sensor representations, which attain a sixty per cent availability rate (9/15) of ABL representations, assisted by embedded model and federation techniques.

BMDS ABL SENSOR SYSTEM REPRESENTATIONS					
ABL SENSOR	CAPS	MDWAR	EADSIM	EADTB	MDSE
BEACON ILLUM.	Y	Y	Y	N	N
TRACKING ILLUM.	Y	Y	Y	N	N
LASER RANGER	Y	Y	Y	N	N
# Avail /Total/ % Avail	3 / 3 / 100%	3 / 3 / 100%	3 / 3 / 100%	0 / 3 / 0%	0 / 3 / 0%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-10. BMDS Airborne Laser Sensor System Representations

6. BMD System-Level M&S Sensor System Representations Summary

There are eighteen BMDS sensor systems modeled across the five BMD System-Level simulations. Out of a possible ninety sensor system representations, fifty-eight are available, a sixty four percent availability rate. Individual sensor category statistics range from a high of one hundred percent for the five sensor system representations in the TDS

²⁵⁰ SBIRS-High and SSTs/SBIRS-Low are represented in MDWAR by a federation with the Missile Defense Space Tool (MDST)

radar sensor category to a low of forty-five percent in the satellite / infrared sensor category. The international sensor system category had representations for sixty percent of the simulations under study.

7. BMD System-Level M&S Battle Management System Representations

The BMDS battle management (BM) software is critical to the success of missile defense, a fact recognized from the earliest SDI studies including the *Fletcher Report* [FMA+84] and *Eastport Study Group* [CCL+85], and its success depends largely on M&S support. Command and control (C2) is the human element of the BM/C2, supported by the automated BM functions and communications network. The BMDS Battle Management system representation will address the system wide BMDS BM capabilities including the engineering needed to resolve protocol differences between TADIL-J and TADIL-K message formats employed by the TDS and MDS segments respectively, integrate the separate battle managers developed by existing MDA programs, and support BMDS system-level future battle manager development.

BMDS BM systems cited in Table 9-11 represent high risks for the BMDS. The *Fletcher Report* [FMA+84] provided strong justification for reliable, safe, effective missile defense software and concluded:

Further emphasis is needed on simulation as a means to assist the design of battle management systems and software. Specific work is needed on algorithms related to critical battle management functions [FMA+84].

Currently, the development of the BMDS System-Level BM is in the embryonic stage and not yet adequately represented in any of the BMDS System-Level M&S. However, BMD element-level legacy BM systems exist and many of their representations exist with the BMD System-levels simulations. The five TDS battle managers allocated to the five BMD System-level simulations noted in Table 9-11 have twenty-five acceptable model representations out of a possible twenty-five instantiations for a one hundred percent availability rate. Acceptable GMD BM representations reside in both CAPS and MDWARS,

although missing BM representations in EADSIM, EADTB, and CAPS, create an overall forty percent BM system availability rate.

The two BDS battle managers, ABL and SBL, available in only two of ten possible BM representations, reflect a twenty percent availability rate. Table 9-11 lists the available representations and percent availability for each simulation. The entire BM representation population in Table 9-11 has an overall sixty-four percent (29/45) availability.

BMDS BATTLEMANAGER SYSTEM REPRESENTATIONS					
C2BM	CAPS	MDWAR	EADSIM	EADTB	MDSE
BMDS C2BM – System-Level	N	N	N	N	N
PAC-2 BMC2 - TDS	Y	Y	Y	Y	Y
PAC-3 BMC2-PDB 5+ - TDS	Y	Y	Y	Y	Y
AEGIS BASELINE 5 - TDS	Y	Y	Y	Y	Y
JTAGS/ALERT - TDS	Y	Y	Y	Y	Y
THAAD BMC3 - TDS	Y	Y	Y	Y	Y
GMD BMC2 - MDS	Y	Y	N	N	N
ABL BMC41 - BDS	Y	Y	N	N	N
SBL BMC2 - BDS	N	N	N	N	N
# Avail /Total/ % Avail	7 / 9 / 78%	7 / 9 / 78%	5 / 9 / 56%	5 / 9 / 56%	5 / 9 / 56%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Ray02a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-11. BMDS Battle Manager System Representations

The International BM system representations for the ARROW and MEADS representations in Table 9-12, achieved a slightly lower fifty percent fill rate, with ARROW BM represented at a one hundred percent fill.

BMDS INTERNATIONAL BM SYSTEM REPRESENTATIONS					
INTL C2BM	CAPS	MDWAR	EADSIM	EADTB	MDSE
ARROW BMC2	Y	Y	Y	Y	Y ²⁵¹
MEADS BMC2	N	N	N	N	N
# Avail /Total/ % Avail	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%

²⁵¹ The ARROW-MDSE interoperability program achieves this capability.

Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-12. BMDS International BM System Representations

8. BMD System-Level M&S Communication System Representations

The BMD system-Level M&S communication network representations in Table 9-13 include the Joint Data Network (JDN), Joint Planning Network (JPN), and the Joint Composite Tracking Network (JCTN). The BMD System-Level M&S, except CAPS, explicitly model the JDN, which forms the communication network for the TDS. The absence of explicit communication representations in CAPS does not degrade the simulation, since it is not a communications planner. EADTB has the highest percent fill of communication representations, confirming its primary role as a communications interoperability tool. Overall the three communication networks have five of fifteen possible representations in the BMD System-Level M&S, an availability rate of thirty-three percent.

BMDS COMMUNICATION SYSTEM REPRESENTATIONS					
COMMUNICATIONS	CAPS ²⁵²	MDWAR	EADSIM	EADTB	MDSE
JDN ²⁵³	N	Y	Y	Y	Y
JPN ²⁵⁴	N	N	N	N	N
JCTN ²⁵⁵	N	N	N	Y	N
# Avail /Total/ % Avail	0 / 3 / 0%	1 / 3 / 33%	1 / 3 / 33%	2 / 3 / 66%	1 / 3 / 33%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-13. BMDS Communication System Representations

²⁵² CAPS network are generic with no explicitly documented network models, although the JDN is the implicit model [BBG+99f].

²⁵³ The Joint Data Network (JDN) is comprised of DoD Link 11, 16, and 22 message protocols sharing sensor data between all subscribers in near-real-time.

²⁵⁴ The Joint Planning Network (JPN) supports the broadcast of TRAP (TDS) and TIBS traffic and non-real-time status and planning.

²⁵⁵ The Joint Composite Tracking Network (JCTN) provides for the distribution of best quality engagement data to all subscribers, in as close to real-time as possible, and supports the extension of the Navy's Cooperative Engagement Capability (CEC).

9. BMD System-Level M&S Platform System Representations

The BMDS System-Level Model and Simulation Platform Representations data in Table 9-14 shows a sixty-four percent availability rate of launcher platforms, with the highest density of available representations in the TDS systems designed for theater mobility (e.g., PAC-2, PAC-3, THAAD) and the Aegis Weapon System. The CAPS model, as noted with the communication systems, operates at a high level of aggregation and does not require detailed platform information. The ABL aircraft, a YAL-IA 747-400, requires the development of a specific track configuration in all the System-Level M&S in support of its current doctrinal employment strategy.

BMDS PLATFORM SYSTEM REPRESENTATIONS					
PLATFORMS	CAPS	MDWAR	EADSIM	EADTB	MDSE
ABL YAL-IA 747-400	N	N	N	N	N
THAAD LAUNCHER	N	Y	Y	Y	Y
PAC-2 LAUNCHER	N	Y	Y	Y	Y
PAC-3 LAUNCHER	N	Y	Y	Y	Y
AEGIS WPN SYS	N	Y	Y	Y	Y
# Avail /Total/ % Avail	0 / 5 / 0%	4 / 5 / 80%	4 / 5 / 80%	4 / 5 / 80%	4 / 5 / 80%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Ray02a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-14. BMDS Platform System Representations

10. BMD System-Level M&S Geodetic Coordinate Representations

Geospatial information provides the capability to describe a location or position, normally exchanged through the use of coordinate locations. Common, well-known geospatial reference systems support interoperability with accurate coordinate locations and support the transformation of a coordinate system into multiple definitions of geo- and non-geo-referenced space. The coordinate systems include:

- Inertial coordinates,
- Earth-centered, Earth-Fixed coordinates,
- Local coordinates [Wil00].

The inertial coordinate systems (e.g., non-rotating coordinates) have an orientation fixed in space, with the Earth-Centered Inertial Model (ECI) coordinate frame commonly employed to define the motions of ballistic missiles or spacecraft [Wil00]. Spatial information consists of wide population of measurements and data types. Earth-centric spatial locations include an Earth-centric view, indicating that spatial locations are geospatial. Spatial locations also support the identification of a fixed reference frame rotating with the earth, the geographic or geodetic reference. Sensors may use local coordinate systems to measure the positions of other bodies relative to the position of the sensor(s) [Wil00]. Conversely there is also a significant amount of spatial data (e.g., astronomical, orbital, geomagnetic and local observations) with reference frameworks fixed on the observer, solar, celestial or other positional standards. Geospatial information interoperability requires that:

- Coordinate systems be defined such that coordinates describe location uniquely,
- A data transfer mechanism exists to transfer data to other defined coordinate systems [Bir97].

The approximate shape of the Earth, originally defined by early navigators as a spherical body supported the development of geodetic coordinate (GDC) systems. However, more accurate geodetic coordinates requires the modeling of Earth as an oblate spheroid or ellipsoid-of-rotation [Bir97]. GDCs, composed of the ellipsoid and the datum, relate Earth-centered, angular, geodetic latitude²⁵⁶ and longitude²⁵⁷ to an actual point, near or on the Earth's surface. The ellipsoid is a good approximation of the geoid²⁵⁸ and used for many mapping applications. Many ellipsoids model the earth, and each ellipsoid may be defined in many ways in reference to the Earth. The datum defines the location of the ellipsoid with respect to the Earth.

Modern GDCs, derived from satellite data, provide measurements defining Earth-based ellipsoids with accurate global data. The World Geodetic System 1984 [WGS84]

²⁵⁶ Latitude is the angle subtended with the ellipsoid's equatorial plane by a perpendicular through the surface of the ellipsoid from a point, with positive latitude north of the equator and negative latitude south of the equator [Bir97]

²⁵⁷ Longitude is defined as the angle measured about the minor, or polar axis of the ellipsoid from a prime meridian to the meridian through a point, positive to the east of the prime meridian (e.g., Greenwich, England) or negative if west of the prime meridian [Bir97].

²⁵⁸ The geoid is a physical surface of equi-gravitational potential corresponding to mean sea-level, extending into landforms as an approximation of mean sea-level, providing surface from which elevations may be directly measured [Bir97]

developed by the National Imagery and Mapping Agency (formerly the Defense Mapping Agency), defines the current Department standard datum and geoid, and employs a consistent global set of 3-dimensional station coordinates inferring the location of an origin, the orientation of an orthogonal set of Cartesian axes, and a scale. The [WGS84] standard, current as of 1996, uses the refined coordinates of the permanent Department's Global Positioning System (GPS) monitor stations as the operational WGS-84 reference frame.

BMDS GEODETIC COORDINATE REPRESENTATIONS					
<u>GEODETIC COORDINATE</u>	CAPS	MDWAR	EADSIM	EADTB	MDSE
WGS-84	N	Y	Y	N	Y
Round, Rotating Earth	Y	N	N	Y	N
Non-rotating Earth	Y	N	N	N	N
Earth Centered Inertial (ECI)	N	Y	N	N	N
Locally Spherical Earth	N	N	N	N	N
# Avail/Total/ % Avail	2 / 5 / 40%	2 / 5 / 40%	1 / 5 / 20%	1 / 5 / 20%	1 / 5 / 20%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Ray02a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-15. BMDS Geodetic Coordinate Representations

In addition, a new global model of the Earth's gravitational field, the Earth Gravitational Model 1996 (EGM-96), replaced the outdated WGS-84 gravitational model [WGS84]. WGS-84 is the current JTA geospatial data interchange standard and prescribed with the EGM-96 for the BMDS positioning, navigation, and timing standards [WBF+00]. The EGM-96 supports the accurate assessment of the Earth's gravitational function supporting the prediction of an object's state vector in time, a critical component and a possible source of heterogeneous system representation anomalies in threat and BMDS component trajectories, sensor target tracking, geodetic information exchange or target track data (e.g., substantive interoperability anomaly) if missing or modeled incorrectly [WBF+00]. Table 9-15 lists the status of geodetic coordinate representations in the BMD System-level M&S. Other environmental models currently under assessment for standardization and inclusion in the set of BMD System-level M&S include local weather models, earth atmosphere models, sun position, lunar position, and star catalog [Wil02, Wil03].

11. BMD System-Level M&S Summary of Representations

The summary of acceptable authoritative representations in Table 9-16 of the five BMDS simulations under study provides the following statistics:

- All five M&S systems included representations in the weapon, sensor, communication, platform, and BM/C2 categories or product lines.
- CAPS accounted for the highest percentage of authoritative representations in the missile and sensor product lines with eighty-two percent and ninety-five percent, respectively; and tied with MDWAR in the BM/C2 product line with seventy-eight percent. The ninety-five percent level for the CAPS sensor product line was the highest percentage of authoritative representations in a product line found in the study. Conversely, CAPS also had the only two product line categories, the communication and platform with no authoritative representations. The level of aggregation and abstraction in authoritative representations common to both CAPS and MDWAR contributed to the CAPS and MDWAR statistics.
- The weapon product line authoritative representations ranged from a high of eighty-two percent to thirty-eight percent. The mode for weapon authoritative representations was thirty-eight percent.
- The sensor product line authoritative representations ranged from a high of ninety-five percent to forty-one percent. The mode for sensor authoritative representations was forty-one percent.
- The BM/C2 product line authoritative representations ranged from a high of seventy-eight percent to fifty-six percent. The mode for BM/C2 authoritative representations was fifty-six percent.
- The communications product line authoritative representations ranged from a high of sixty-six percent to zero. The mode for communications authoritative representations was thirty-three percent.
- The platform product line authoritative representations ranged from a high of eighty percent to zero. The mode for platform authoritative representations was eighty percent, the highest mode statistic in the study.
- The five-simulation total product line authoritative representations ranged from a high of seventy-five percent to forty-seven percent. The mode for the five simulation total authoritative representations was forty-seven percent.
- The five-simulation population including all product line authoritative representations equaled sixty percent.

BMDS SYSTEM-LEVEL PRODUCT LINE REPRESENTATION SUMMARY					
SYSTEM REPRESENTATIONS	CAPS	MDWAR	EADSIM	EADTB	MDSE
TDS Missiles	6 / 6 / 100%	5 / 6 / 83%	5 / 6 / 83%	5 / 6 / 83%	5 / 6 / 83%
International TDS Missiles	2 / 2 / 100%	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%
MDS Missiles	4 / 4 / 100%	3 / 4 / 75%	1 / 4 / 25%	0 / 4 / 0%	0 / 4 / 0%
Directed Energy Weapons	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%	0 / 2 / 0%	0 / 2 / 0%

Kinetic Energy Weapons	0 / 2 / 0%	0 / 2 / 0%	0 / 2 / 0%	0 / 2 / 0%	0 / 2 / 0%
WEAPON SUB-TOTAL	13 / 16 / 82%	10 / 16 / 63%	8 / 16 / 50%	6 / 16 / 38%	6 / 16 / 38%
TDS Radars	6 / 6 / 100%	6 / 6 / 100%	6 / 6 / 100%	6 / 6 / 100%	6 / 6 / 100%
International TDS Radars	3 / 3 / 100%	3 / 3 / 100%	1 / 3 / 33%	1 / 3 / 33%	1 / 3 / 33%
MDS Radars	6 / 6 / 100%	5 / 6 / 83%	2 / 6 / 33%	1 / 6 / 17%	1 / 6 / 17%
Satellite / IR Sensors	3 / 4 / 75%	3 / 4 / 75%	1 / 4 / 25%	1 / 4 / 25%	1 / 4 / 25%
ABL Sensors	3 / 3 / 100%	3 / 3 / 100%	3 / 3 / 100%	0 / 3 / 0%	0 / 3 / 0%
SENSOR SUB-TOTAL	21 / 22 / 95%	20 / 22 / 91%	13 / 22 / 59%	9 / 22 / 41%	9 / 22 / 41%
BMDS C2BM	7 / 9 / 78%	7 / 9 / 78%	5 / 9 / 56%	5 / 9 / 56%	5 / 9 / 56%
International C2BM	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%	1 / 2 / 50%
SENSOR SUB-TOTAL	8 / 11 / 73%	8 / 11 / 73%	6 / 11 / 55%	6 / 11 / 55%	6 / 11 / 55%
COMM SYSTEM SUB-TOTAL	0 / 3 / 0%	1 / 3 / 33%	1 / 3 / 33%	2 / 3 / 66%	1 / 3 / 33%
PLATFORM SUB-TOTAL	0 / 5 / 0%	4 / 5 / 80%	4 / 5 / 80%	4 / 5 / 80%	4 / 5 / 80%
SIMULATION TOTAL	42 / 57 / 74%	43 / 57 / 75%	32 / 57 / 56%	27 / 57 / 47%	26 / 57 / 47%
POPULATION TOTAL	170 / 285 / 60%				

Table 9-16. BMD System-Level M&S Representation Summaries

Although the five simulations equate to a statistically insufficient sample to relate to the Department's total M&S population, the results correlate with the numerous studies, reports, and papers cited in earlier chapters, suggesting the Department population representations as a whole, may be more unpredictable than authoritative. However, this study covers one hundred percent of the system-level model representation population, with a significant sample size of 285 representations, which viewed in the Table 9-16, forms patterns. [Fow97] defines a pattern as "an idea that has been useful in one practical and will probably be useful in others" [Fow97], and further suggests that the establishment of a common framework supports the concept of component reuse for information systems.

Table 9-16 identifies potential patterns for several software architecture patterns including product families (e.g. weapons), software product lines (e.g. missiles, lasers), software product line systems (e.g. PAC-3, THAAD missiles), and a software product line architecture. The sixty percent availability rate of authoritative representations (e.g., model product lines) resulted from the causal reasons cited in earlier chapters and for the specific reasons noted in this chapter. This suggests that current Department simulation software development techniques, procedures, and practices are inadequate or incomplete, inconsistently implemented, or a combination of both factors.

The BMDS poses even more challenges to the current status. The BMDS program requires authoritative representations for simulations supporting the engineering of future BMDS block capabilities. The BMDS system engineering process requires authoritative representations for Blocks 2004, 2006, 2008, 2010, 2012, 2014 and beyond to explore and experiment with capabilities, and provide a means to extend the performance window of existing system. Due to live-test limitations, the BMDS System-Level M&S will provide a significant percentage of the future data needed to assess BMD System-wide interoperability and performance. The urgency of improving the BMDS System-Level M&S and adding more credible authoritative representations increased dramatically with the President's decision to deploy a limited Missile Defense capability in 2004. At that time the BMDS System-Level M&S will support an operational Missile Defense for the Nation, allies, and friends. Chapter X provides a software architecture-based alternative to improve the level of authoritative representations in future BMDS System-Level M&S.

F. BMDS INTERNATIONAL COOPERATIVE M&S PROGRAM

The Agency's international cooperative M&S program requires the development of system representations with the necessary detail, attributes [MDW+01], and goodness-to-fit must pass several hurdles from the formal approval of the international agreement permitting the initiative, completion of required disclosure statement, implementing the necessary contracting vehicles supporting the effort, and executing the actual model development effort. The international cooperative M&S development effort must also address information assurance, language, cultural, and national priority issues, in addition to the normal cost, performance, and schedule project management concerns. All or any one of these conditions may provide the underlying conditions for an invalid federate or the creation of intolerable representational anomalies, resulting in substantive interoperability issues.

Table 9-17 illustrates the scope of international cooperative M&S initiatives supported by the BMD System-Level M&S. This international M&S effort is a key component of the missile defense interoperability strategy with friends, allies, and supports evolving international cooperation efforts with new partner nation initiatives such as the Russian Federation Model and Simulation Program (US/RF). Table 9-17 reflects the System-Level

M&S supporting the initiatives, and the type of international agreements authorizing the program. EADSIM supports every international cooperative M&S program, except the embryonic US/RF initiative, while NATO employs EADTB in missile defense interoperability studies. The ARROW-MDSE program directly supports the international terminal defense segment interoperability testing of the U.S. PATRIOT missile defense system with the Israeli ARROW missile defense system.

COUNTRY M&S	SYSTEM-LEVEL M&S				LEGAL BASIS					
	CAPS	EADSIM	EADTB	MDSE	MOA	MOU	D	F	W	S
Australia		X							X	
Canada		X			X					
Germany		X	X		X		X			
Greece		X						X		
Israel		X		X	X					
Japan		X					X			
S. Korea		X						X		
NAMEADSMA		X	X			X				
NC3A		X	X		X					
UAE		X						X		
UK		X	X						X	X
Singapore		X						X		
Spain		X						X		
Netherlands		X						X		X
Turkey	X	X			X					
NATO FS	X	X	X		X					
US/RF	New Model and Simulation Development Program – Legal Agreement Pending									
TOTAL	2	16	5	1	6	1	2	6	1	2

Table 9-17. BMD International Cooperative M&S Program

G. BMDS ELEMENT-LEVEL MODELS & SIMULATIONS PROGRAM

The major Element-Level M&S, listed in Appendix H directly support the program requirements of MDA element program managers. They are also components of the joint BMD System M&S Hierarchy, developed in accordance with the approved element Model and Simulation Support Plan (MSSP). Element-Level M&S employ BMD System-TSEL M&S in the development of their respective system, and support the development of accurate element representations in the BMD System-Level M&S [Kad02a].

H. BMDS-LEVEL THREAT, ENVIRONMENT, SIGNATURE & LETHALITY MODELS & SIMULATIONS

The Legacy BMD System Threat, Environment, Signature, and Lethality (TSEL) M&S are the foundation codes of the missile defense domain. These common-and general-use codes represent the highest fidelity simulations in the missile defense M&S domain and directly support the design, development, and testing of all missile defense element systems. These codes require the highest level of credibility. In addition, the expanded battle space of the new BMD system adds significant new requirements, beyond the current capability of the tools, necessitating major upgrades and new development programs. Appendix I has a short list of the BMDS Threat, Environment, Signature, and Lethality M&S program.

I. BMDS SYSTEM-LEVEL M&S VV&A PROGRAM

1. AGENCY M&S VV&A EFFORTS (1986-2000)

The BMD Core M&S program inherited a legacy of challenges, issues, and problems requiring timely solutions to allow the successful evolution of credible Agency M&S supporting confidence in the simulation results. The agency supported multiple M&S reports, studies, analyses, and reviews to establish the credibility of the M&S supporting the unprecedented technical challenge of the missile defense domain including : [FMA+84, CCL+85, Bre88, SCC+98, HSW+92, EHH97, KBM+97, Li97, SEL97, SEH+97, EHH+98, WAA+98, ZHF98, BBG+99a, BBG+99b, BBG+99c, KGB+99, Li99, ISE99, WAA+99, RTM+99a, RTM+99b, RTM+99c, RTM+99d, RTM+99e, RTM+99f, RTM+99g, BDH00, LEH+00, SBH+01, Wel00, BRC+01, Coy01, EP01, LWC+01, LHC+02, Mil02a, STG+02].

The Agency's internally- and externally-generated reviews for existing simulation systems identified the same type of problems noted in the preceding generation of air defense simulations (e.g., Adage, Carmonette, COMO III [Che87]). The results also confirmed the lack of silver-bullet solutions, and closely paralleled the credibility issues and

systemic, persistent Department M&S problems outlined in the preceding chapters including:

- Lack of Technical and Substantive interoperability
 - Need to improve V&V process,
 - Lack of documentation, including conceptual models,
 - Poor or nonexistent configuration management of models and data,
 - Inadequate understanding by users and decision-makers of model assumptions,
 - Limitations, and capabilities,
 - Insufficient model development practices,
 - Need for improved procedures to update and maintain models,
 - Need to improve how the M&S credibility is established,
 - Need to improve how the validity of the model is determined,
- Simulations need hard data from test programs.

The study effort for the ballistic missile defense M&S research phase started in September 2000 and included a review of the agency documents including multiple Agency M&S briefings, reviews, reports, staffing actions, test results, conversations, and assessments, cited above. The Agency's VV&A issues during the 1990s mirrored the Departmental M&S VV&A issues. Documented V&V of the BMDO core model set during the 1986-1999 timeframe was inconsistent and rare. Two specific studies representative of other Agency M&S studies spanning most of the 1990s include [HSW+92] and [BBG+99a].

In 1992 [HSW+92] reported on the SDIO's Brilliant Pebbles' program effectiveness and the extensive use of simulations in support of system design, manufacturing and deployment; concluding the simulations were still immature, and used unproven assumptions, and noting:

The accuracy of Brilliant Pebbles' simulations has not been validated by test results or verified by formal reviews of the simulations themselves [HSW+92].

The BMDO and Joint Theater Air and Missile Defense Organization (JTAMDO) conducted a joint a, M&S review in 1999 [BBG+99a] of selected BMDO core models. The report identified model validation among the key risk areas and noted:

There is no BMDO-wide process for assuring adequate model validity. The degree of V&V performed is currently up to the organization funding the developer, and no stan-

dards are followed. There is no organized link with the TE [test and evaluation] community for obtaining data for validation or to show that the models can/cannot predict or extend testing. There has only been a minimal effort to compare model results on the same scenario (anchoring) and inadequate checkout of less detailed models against results from the more detailed models [BBG+99a].

The majority of the [BBG+99a] issues included systemic BMDO organizational, process, and technical issues. Accordingly, the joint BMDO/JTAMDO M&S review [BBG+99a] recommended an on going V&V process citing:

V&V is another important activity not actively pursued at this time in a centralized manner, resulting in uncertainties about the robustness of our simulations and non-uniformity in the degree of V&V performed. There should be a planned central effort to validate our models, collect validation data, and assess the developers' processes in order to reduce program risk. A part of this program should be to link T&E activities in an active effort to obtain data for model validation and for making sure the simulations can validly be used to extend test data for model validation and for making sure the simulations can validly be used to extend test data. Anchoring simulations results is another critical activity, which would improve our understanding of the models and of their differences and which would increase our confidence level or motivate corrections [BBG+99a].

[MS02] and [Lev02] address the two key issues of credibility and confidence in missile defense simulations, with [Lev02] noting:

Weapon system confidence is being able to predict the system's performance to within a quantified uncertainty or confidence interval [Lev02].

However, the current suite of BMD System-Level M&S experienced various inconsistent VV&A approaches as the BMD System M&S V&V Program²⁵⁹ matured [HSW+92, BBG+99a].

In 1999 [BBG+99F] identified that CAPS lacked a formal V&V process since first

²⁵⁹ The Agency implemented a centrally managed verification and validation program of the core M&S in FY2002 establishing a V&V process to improve systemic credibility issues and improve user confidence in the Agency core M&S results.

released in 1994, and noted, “there is no indication of the validity of the CAPS engagement logic to represent BMDO systems” [BBG+99f]. A documented CMMS/ FDMS for CAPS was never developed. However, CAPS was formally accredited for specific use on two occasions:

- 1994-1996 for the Theater Missile Defense Cost and Operational Effectiveness Analysis study, accredited by both the Army and the Navy,
- 1997 - Accreditation as the interim Joint Defensive Planner (JDP) by the BMDO [BBG+99a].

MDWAR, the newest addition to the core model suite initiated a V&V program early in the development cycle. The first major MDWAR wargame, C2SIM99, met joint-accreditation requirements by the NMD program office and USSPACECOM for execution in November 1999. Follow-on wargame efforts included the Theater Ballistic Missile Defense Operator-In-The-Loop Phase I war game exercise in July 2000, and Phase II war game exercise in September 2000, with both events meeting joint-accreditation requirements from the NMD program office and USSPACECOM. The developer first published conceptual model documentation for MDWAR components in 1999 [TRW02a, TRW02b].

EADSIM developed a wide missile defense community acceptance throughout the 1990s, although [JAS97c] noted in a comprehensive review that little formal V&V was complete, no documented V&V results, and verification limited by the lack of a detailed design specification against which to check the software code. [JAS97c] acknowledged that some V&V work may be in progress at the time or classified, although AFOTEC reported similar findings in 1994. EADTB credibility according to [JAS97c] depended almost completely on face validation assessments and community acceptance.

The first major study employment of the EADTB simulation was the BMDO Cost and Operational Effectiveness Analysis (COEA) study. A face assessment [BBB+96] review of EADTB for the COEA study was completed October 25, 1996 and identified 96 issues and 34 product quality issues. [BBB+96] detailed major problems including:

- The lack of requirements against which to measure the capabilities,
- Excessive run-time,
- A high learning curve,
- Insufficient testing through November 1995,
- Verification was limited to algorithms, code testing, and performance, with no at-

tempt at design verification.

- An incomplete draft VV&A Plan, dated March 14, 1994 had not been placed on contract.
- V&V contactor had one person assigned.
- No formal conceptual model deliverable.
- Documentation was a high concern with key documents outdated or missing [BBB+96].

[BBB+96] also reviewed EADTB V&V procedures in the areas of ballistic missile flights and radar sensors, and noted the existing EADTB V&V method compared results with other models such as EADSIM or COMO. [BBB+96] recommended EADTB for an application “wring out” but not for the more critical and contentious missile defense campaign-level studies.

The EADTB program established a formal V&V effort for the simulation’s CMS components in 1993 and initiated a follow-on V&V program for the system specific representations (SSR) in 1999. The reasons for the six year difference between the 1993 initiation of the EADTB CMS V&V program and the 1999 EADTB system representations V&V program highlight systemic issues identified in this study limiting the credibility of all Agency System-Level M&S during the 1990s:

- In the competition for resources between developing new system requirements with supporting stakeholders, or support for the potentially high recurring costs of establishing and maintaining a consistent V&V program, the delivery of new capabilities generated user-support, whereas, V&V program efforts provided no new capabilities or satisfied customers, and could be contentious,
- The inherent complexity of V&V confused most stakeholders and decision-makers, detracting from V&V proponent’s ability to compete with other Agency requirements for resources,
- A concern by the MDAP program offices that their system could be misrepresented in a simulation, or the representation used for a purposes it was not designed or intended, such as system performance,
- The time and resources supporting SSR development in the MDAP system engineering correspondingly reduced the MDAP’s resources for it’s primary mission, with little perceived return-on-investment,
- The simulation developers had limited access to the MDAP program offices to research, design and develop credible system representations for Agency-level studies, analyses, or tests.

- Discrete CMMS/FDMS, prescribed by [DoD94, DoD96] were not developed for any of the five simulations during the early life-cycle phases, creating at a minimum the possibility of logical inconsistency in the VV&A process,
- Any efforts to determine the credibility of a simulation or federation for intended use was necessarily *ad hoc*, expensive, and short-lived without a verified and validated conceptual model of the mission space upon which to base the verification and validation process for the FOM or software implementation,
- An Agency trend developed in the 1990s to accredit a simulation without underlying conceptual models of the mission space or V&V plans,
- The Department's accreditation process supported multiple accreditation options and it was acceptable to accredit the simulation by simply using it,
- Face validation or a review by subject matter experts of the simulation for an intended use became the *de facto* standard for establishing credibility and supporting user confidence,
- The V&V of certain SSRs (e.g., threats, systems under development) was extremely complicated and required close coordination with many agencies,
- Contractors had no incentive to V&V simulations, and contractual language supporting V&V as a deliverable were few,
- The previously discussed Year 2000 remediation effort and HLA compliancy work pushed V&V efforts further down in the requirement queue.

2. AGENCY M&S VV&A EFFORTS (2001-2003)

Two BMD System-level M&S, EADTB and MDSE, supported a major-program testing event, the PAC-3 independent operational test and evaluation (IOT&E) with the PATRIOT IOT&E Interoperability Demonstration (PIID) in April 2002. The event was significant in that the Army Test and Evaluation Command accredited the Missile Defense System Exerciser [Bar02a, CRC02a], in a virtual hardware-in-the-loop stimulation tool role, and the Extended Air Defense Test Bed [Bar02b, CRC02b, KS02], as a constructive simulation for a full-up five firing unit air defense battalion in a representative threat environment supplementing a Department IOT&E for the PAC-3 missile defense system. The PIID built on a highly successful PAC-3 Developmental test and M&S program [BRC+01].

The year-long preparation effort in support of the PAC-3 PIID required the full cooperation of the Agency M&S organizations, program offices, simulation developers, VV&A agents, the operational test community and many other stakeholders to develop, verify, validate, accredit or certify the system representations as accurate for the intended use of testing PAC-3 interoperability. The interoperability assessment addressed PAC-3

intra-battalion interoperability, interoperability between the Patriot system and other elements of the Army Air and Missile Defense task Force, and interoperability between Patriot and other Joint Service elements in a Joint environment scenario [Bar02a, Bar02b]. The EADTB and MDSE simulated Theater Missile Defense architecture with tactically representative threats, message traffic, and communications loading. The PIID met all test objectives and successfully completed all test runs-for-record ahead of schedule in April 2002.

Based on past performance and its credibility in the missile defense domain, MDSE was incorporated into the Department's evolving Joint Distributed Engineering Plant (JDEP) program [Dah01a, Dah01b, DT01a, DT01b, Lew02b, DC02, Rad02] supporting improved interoperability of Joint Forces, an example of large-scale reuse by the Department and a potentially significant return-on-investment. Even more importantly, the collaborative multi-stakeholder effort supporting the PIID set the foundation for systematically developing credible system representations in all BMD System-level simulations with element program participation [Kad02a].

The Extended Air Defense Simulation supported the Joint Project Optic Windmill (JPOW) VII Exercise [Obe02] in September 2002, after completing a collaborative VV&A process with the full cooperation of the program offices. This exercise also marked the first employment of the BMD M&S Integrated Product Development Team (IPDT) concept with the objective of developing, verifying, and validating accurate, authoritative representations of element capabilities in the System-Level M&S [Kad02a]. The BMD System-Level M&S V&V program incorporated the lessons-learned from [HSW+92, BBG+99a, Bar02a, CRC02a, Bar02b, CRC02b, Obe02] and several other reports cited earlier, and developed centrally-managed VV&A processes for system-wide events [Kad02a].

J. BMDS SYSTEM M&S FIDELITY

1. Agency M&S Fidelity Background 1986 - 2000

The Agency adopted a "family of systems" (FoS) approach to describe mutual support over a hierarchy ranging from high resolution, limited scope models (e.g., physics level codes) at the base, through higher level overlapping levels of models with lower reso-

lution [SW96, BBB+96], similar to the Defense Systems Management Collage model [PMR94]. [SW96] proposed an analytical taxonomy supporting the FoS approach with six qualitative levels: system, encounter, engagement, mission, campaign, and global. [BBB+96] took a different approach, basically developing a two-dimension array with four major application categories: analysis, test and evaluation, exercises and wargaming, and engineering assessed at four qualitative levels of analysis: campaign²⁶⁰, force-on-force engagement²⁶¹, few-on-few engagement²⁶², and special domain²⁶³ performance.

These qualitative approaches were not sufficient for the high-risk, mission-critical missile defense domain. A later 1999 MITRE Corporation-led review of the BMDO M&S program [RTM+99b] addressed two major BMDO M&S issues central to this dissertation. [RTM99b] recommended the Agency:

- Identify the methodology for representing systems and functions (e.g., emplacement, search and track, engage, missile launch / fly out, terminal guidance, intercept and post-intercept assessment) critical to missile defense at each level in the M&S hierarchy,
- Analyze the consistency of M&S representations of functions up and down the M&S hierarchy [RTM+99b].

Another related internal Agency report, [BBG+99b] noted the quality of each level of simulation depended upon the quality of the input from the lower level simulation or code and recommended:

- An evaluation of the required levels of fidelity between program office high fidelity simulations and factor-driven campaign-level simulation tools,
- The Agency develop an agreed upon range of fidelity for the kill-chain components (e.g., sensors, weapon algorithms, platforms, C4I) instead of a set fidelity level to close the “fidelity gap” between different organizations [BBG+99b].

²⁶⁰ Campaign: Characteristics include aggregated resolution, theater-level scope, many-on-many scenarios, run time speed many times faster than real-time, generally lower fidelity, simulating time span from days to months, includes all architecture components plus ground, air, and naval forces. Addresses logistics factors: resupply, reload, damage and repair, and weapon inventory issues. Examines system performance impact on the course, duration and outcome of the campaign [BBB+96].

²⁶¹ Force-on-Force: Characteristics include minimal aggregation, theater level scope, many-on-many scenarios, medium-to-high fidelity (e.g., treatment may be mixed with three of freedom (3 DoF) modeled for some aspects while way points may be used for other trajectories), and probably does not play intercept or end game performance. Simulation time span is from hours to a few days, running faster than real time to support trade studies, and usually provides data for campaign models [BBB+96].

²⁶² Few-on-Few: Characteristics one-on-one engagements, no aggregation, limited scope, high system detail and fidelity of representation for system under examination. Intercept end game played, simulation time form a span of minutes to a few hours, run time faster than real time to facilitate sensitivity analysis, examines system and subsystem performance. Provides data for Force-on-Force models [BBB+96].

²⁶³ Special Domain: Configured to examine particular phenomena, such as command and control, battle management, communications, with aggregation, scope, fidelity of representation, and run time requirements dependent on specific study or experiment objectives [BBB+96].

2. An Evolving M&S Fidelity Framework 2000 - 2003

The Agency developed a framework providing decision-makers with a background on the BMD System-level model capabilities, flexibility, resolution, accuracy, and precision (Figure 9-4) relative to other simulations in the set and the new block-based capability acquisition policy [Rum02c], as an interim method pending a more definitive model [Gre02a]. In October 2002, the MDA Model and Simulation Working Group (MSWG), composed of members from all the Agency program offices, MDA directorates and executing agents established a fidelity technical working group (TWG) with the following character:

- Define M&S fidelity in the context of the BMDS System-level M&S,
- Propose a time-phased schedule for implementing BMD fidelity standards,
- Determine the requirement for multi-resolution capabilities [Gre02b].

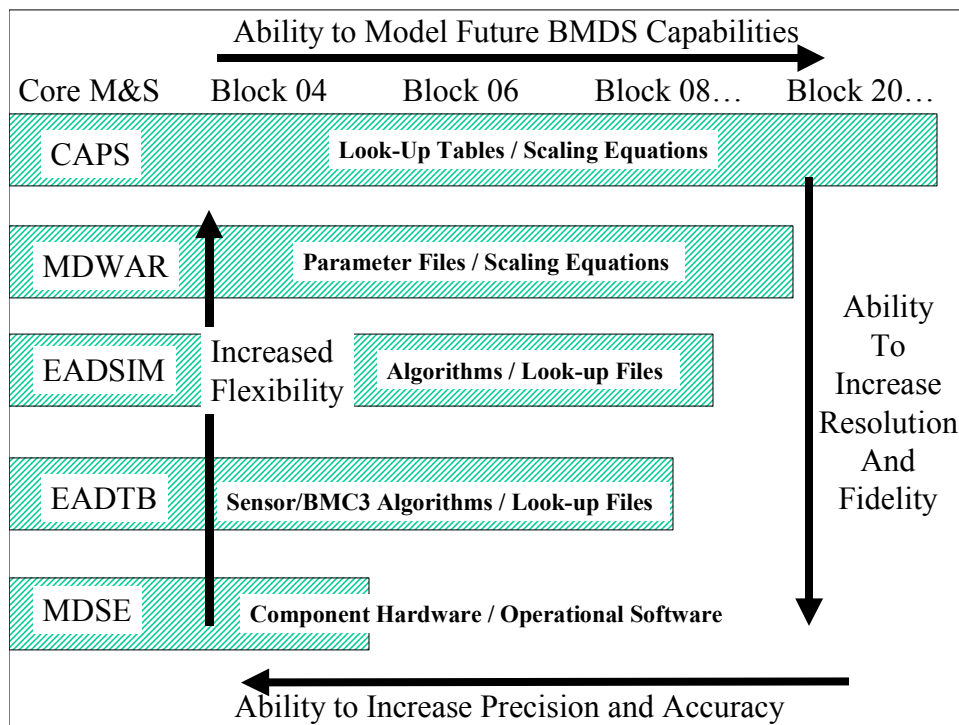


Figure 9-4. MDS System-Level Model and Simulation Capability (After [TEM03])

One of the early product deliverables from the BMDS MSWG Fidelity technical working group (Figure 9-5) provides the methodology the group selected to support their tasks. The TWG's initial effort [Gre02b] addressed the BMDS kill-chain detection function, sub-functions (e.g., search, detect, initiate track, track) and the respective fidelity attributes of the BMD sensor family of radar detection models as they relate to the BMDS System Capability Specification, a major BMD system engineering specification. The [WAD+02] approach also addresses another major Department systemic deficiency where major M&S initiatives and the Department's system engineering processes lack a clear integrated relationship, linkage, and common focus.

Fidelity Levels					
	Low		Medium		High
Search	<u>"Cookie Cutter" Volume</u> - Constant, predefined volume - No operational sectors/volumes - No natural environments - No manmade environments - No power management		<u>Static Search Volume</u> - Predefined, not changeable - Uses operational sectors/volumes - Natural environments - No power management - No manmade environments		<u>Dynamic Search Volume</u> - Can be rapidly defined/changed - Uses operational sectors/volumes - Includes power management - Natural environments - Manmade (included) environments
Detect	<u>Algorithm Based (1)</u> - Radar has reference range/RCS - Look up/estimate target RCS - Apply fourth root scaling law - Line of sight sight with 4/3 earth model to account for refraction - Detection if within radar FCR and scaled detection range	<u>Algorithm Based (2)</u> - Calculate gains/losses/ERP - Determine N or Pfa - Determine necessary SNR to target detection - Line of sight sight with 4/3 earth model to account for refraction - Use Radar Range equation to calculate SNR - Determine bin	<u>Statistical (1)</u> - Calculate gains/losses/ERP - Determine N or Pfa - Use Swirling target model (1/5) - Apply pulse integration (if used) - Line of sight sight with 4/3 earth model to account for refraction - Calculate Pcb between 0 and 1	<u>Statistical (2)</u> - Calculate gains/losses/ERP - Determine N or Pfa - Use Swirling target model (1/5) - Apply pulse integration (if used) - Use clutter and propagation models - Calculate Pcb between 0 and 1	<u>Statistical (3)</u> - Calculate gains/losses/ERP - Determine N or Pfa - Use Swirling target model (1/5) - Apply pulse integration (if used) - Apply aperture weighting(s) - Use detailed clutter and propagation models - Use detailed target orientation and
Initiate Track	<u>"Cookie Cutter"</u> - Track initialized upon detection		<u>Algorithm Based (1)</u> - General "M of N" detections algorithm		<u>Algorithm Based (2)</u> - Actual sensor track initiation algorithm
Track	<u>"Cookie Cutter" Track Model</u> - 9 State track for thrusting targets - 6 State track for ballistic targets - TED State track for reentry targets - Report truth data as state vector - Use predefined covariance estimate	<u>Linear Tracking Model</u> - Use a linear tracking model (i.e., Kalman Filter or Alpha Beta) - Predefined update rate - 9 state filter for thrusting targets - 6 state filter for ballistic targets - Use SNR from the detection model to estimate sensor noise	<u>Linearized Tracking Model (1)</u> - Use a linearized version of a non-linear tracking model - Predefined update rate - 9 state filter for thrusting targets - 6 state filter for ballistic targets - TED State filter for reentry targets - Use SNR from truth	<u>Linearized Tracking Model (2)</u> - Use a linearized version of a non-linear tracking model - Tracking updates determined by power management and operational considerations - 9 state filter for thrusting targets - 6 state filter for ballistic targets	<u>Operational Tracking Model</u> - Use operational tracking algorithm(s) - Tracking updates determined by power management and operational considerations - Estimate target returns and environmental conditions - Calculate state vectors and covaria

Figure 9-5. Example of Radar Fidelity in Core Models (From [TEM03])

3. BMD System-Level M&S Radar Detection Models

There are several critical functions and sub-functions identified in the BMDS kill-chain framework (see Figure 9-2) required for the successful execution of the BMDS mission. However, in this dissertation, we address only a subset of the detection function at a very high level to compare and contrast the development designs for radar detection models in the five BMDS system-level simulations. These radar model representations form patterns used later as the foundation for proposals on product lines and components.

The CAPS simulation, considered “low” fidelity, provides Radar Detection Models with three basic detection capabilities:

- Constant range, referred to as “cookie cutter”,
- Scaled equation with reference parameters,
- Full range equation [Spa00]

CAPS also provides various fidelities supported by four different threshold options:

- Deterministic, using a standard signal-to-noise threshold defined by the user,
- Probabilistic, using a random probability of detection threshold,
- Deterministic, using a probability of detection (Pd) defined by the user,
- Deterministic, with cumulative Pd threshold set by the user [Spa00].

The optional degrees of flexible model methods and fidelity may appear as an unexpected capability in a highly aggregated simulation of CAPS class, except the detect function of the BMD kill chain is critical to the credibility of all BMD system-level M&S. Parameters [Spa01] maintained in database files support most CAPS functions.

The MDWAR simulation, considered “low-medium” fidelity, employs a generic radar sensor model for detecting and tracking simulated threat targets, integrated with the Parallel Discrete Event Simulation (PDES) engine [TRW02a]. The model supports all the events needed to create a generic sensor model and certain specific radars, with design considerations driven by a need for a fast generic sensor simulation modeling components common to all types of sensors [TRW00b]. The MDWAR real-time radar simulator, optimized for large, real-time C2 wargame simulations may be configured to represent a number of different radars, with a design goal of tracking 1,200 air breathing threats (e.g., aircraft, cruise missiles) with 100 radars while maintaining wall-clock time [TRW02b].

The MDWAR design objectives support the requirement to provide users with real-time radar simulations driving fifty to seventy-five player/observer graphic workstations in very large human-in-the-loop wargame scenarios [TRW02b]. In order to achieve the required performance, the radar sensor model depends on the following assumptions:

- All sensor fidelity enhancements must be selectable by .par (e.g., parameter files) so they can be switched off if not required,
- MDWAR models perfect communications between the sensor and the battle manager, as opposed to perception-based methods used in other simulations,
- MDWAR models perfect weather without adverse conditions that may add to the fog and friction of war,
- MDWAR does not model chaff clouds or jammers [TRW02b].

MDWAR does not model radar resolution, but provides perfect radar resolution in which every truth target, if detected, generates a radar return. However, discrimination is imperfect, generated by a time-based probability model [TRW02b]. The radar sensor model has unlimited sensor resources, except real-time constraints for keeping up with the wall-clock time; employs radar cross-section (RCS) data from a classified data table based on target type, radar frequency, and polarization; employs a Swerling target model (e.g., cases 0,1,2,3,4); and computes Pd on a random draw to determine if detection happens or not [TOM99, TRW02b]. MDWAR uses truth-based algorithms, versus a probability model [AGJ95] to determine a hit or miss for kill assessment [TRW02].

EADSIM, considered “medium” fidelity, also provides significant flexibility and options for radar representations, including complex models supporting both deterministic and probabilistic radar range detection [TOM97]. Signature modeling is activity dependent and includes user-specified functions for frequency, wavelength, aspect angle, with the user specifying the granularity of the tabular data [TOM97, TBE02a]. The radar sensor model may use a radar range equation, with jamming to detect targets, with the radar sensor class using a detection probability employing signal-to-noise (SNR) ratio, RCS, and burn-through calculations [TBE02b].

EADSIM employs a detection gate criteria as a means to model sensor detection limitations with a four gate criteria:

- Sensor-to-target range,
- Absolute target speed,

- Sensor-to-target range rate,
- Target altitude [TBE02b].

The four gate criteria share common test criteria with a region defined by minimum and maximum values, and no detections possible outside the defined region [TBE02b]. EADSIM defines RCS in one of three ways: uniform RCS, a single value specified for a system or weapon type; roll symmetric RCS tables, defined by frequency; or laterally symmetric RCS tables, normally employed with aircraft and cruise missiles, when the right and left sides of the system are mirror images [TBE02b].

EADSIM supports radar sensors at different levels of fidelity including a simple sensor approximating the to-level performance of the radar with basic search, detect, and track functions; a compound sensor supporting multifunction radars and advanced functions including multiple autonomous search sectors, cued search, track, interceptor support functions, resource management, sensor constraints and kill assessment [TOM97]. In EADSIM, target detection includes specific sensor-to-target detection tests including:

- There is an unobstructed view of the target,
- There is an unobstructed line-of-sight between the target and the sensor,
- The target is in the sensor's field of view [TOM97].

Specific detection tests may be either deterministic comparing computed signal-to-interference ratios with a user-specified variable, or probabilistic with Swerling target fluctuation models providing a computed probability of detection with a random draw to determine detection [TOM97, TBE02b]. Both methods support the effects of clutter, multipath, atmospheric attenuation, and jamming on the radar sensor [TBE02b]. In all cases EADSIM employs digital terrain models to check for terrain masking and inter-visibility issues [TOM97].

EADTB, considered a “medium-high” fidelity simulation, supports sensor models, which detect objects through the measurement of energy, reflected or emitted by those objects (e.g., the sensed platform supplies the signature, and not the sensing platform). The sensed object provides the signature on request based on the sensing geometry [Ray01b]. The user defines sensor performance (e.g., power, gain, target signature characteristics, en-

vironmental effects) input variables, which the sensor algorithm computes to detect targets [Ray02a].

Although EADTB models both active and passive sensors including electro-optic / infrared passive sensors, and RF passive sensors, radar detection models are active sensors that transmit radio frequency (RF) energy and measure its reflection from the target body [Ray02a]. EADTB sensor models include separate methods for detection and tracking and may include:

- Scanning,
- Jamming,
- Line-of-sight obscuration,
- Antenna patterns,
- Sensitivity and processing,
- Sensor beam or field of view [Ray01b].

The level of detail in an EADTB sensor model varies from a functional sensor incorporating range, target type, and search volume; to a high-detail sensor with many input sensors to model gain patterns, polarization, Doppler spreads, ambiguous ranges, pulse compensation ratios, and main- or side-lobe jamming [Ray02a]. The modular sensor model includes four subcomponents: scanner, transmitter, receiver, and data processor and the management of these resources, while the radar model simulates different radar functions (e.g., search, track, acquisition) and different types of radars (e.g., mechanical, phased array) [Ray02a].

EADTB computes the probability of detection as a function of the probability of detection versus SNR distribution and the difference between the SNR and the detection threshold, followed by a random draw to determine target detection [Ray02a]. EADTB sensors operate in explicitly defined modes supporting multiple scan modes and track modes and accommodate three classes of data: static, dynamic, and mode-dependent [BBB+96]. In addition, EADTB may model different RF phenomena, with over 180 types of radar data input possible.

MDSE currently tests the BMDS at the FoS level (e.g., interoperability), as opposed to the system or subsystem level, which establishes the required level of fidelity and accuracy for the three embedded model categories: BMC2, launcher and fly-out models, and

sensor front end models [McQ97]. The MDSE sensor front-end model provides accurate detection, tracking positions, and timelines with the initial detect function of the kill chain required to be accurate within +/- 10% [Too94]. MDSE provides the RCS for missiles as a function of aspect angle (e.g., a measurement from missile nose to tail), required for initial detection and tracking results [McQ97]. MDSE sensor models generate realistic track position fluctuations for covariance matrices supporting the development of Tactical Digital Information Link (TADIL) message standard [DoD97] J3.6 messages. Although MDSE radar models are not detailed enough to perform the classification or discrimination functions, they are sufficient for use in determining the status or reporting responsibility (R2) message traffic [Too94].

K. BMD SYSTEM-LEVEL M&S REPRESENTATION HETEROGENITY

A review of Table 9-1 through Table 9-16 reveals several specific contributing causes for M&S representation heterogeneity issues in the five BMDS System-Level M&S, and possible contributing causes for M&S representation heterogeneity issues in the Department's large-scale legacy M&S including:

- The absence of a specific system representation for existing systems within a single simulation,
- The absence of a specific system representation for existing systems within a selected set of federated simulations,
- The absence of specific system representations for different configurations of the same existing system within the same simulation (e.g., THAAD PDRR system and the THAAD EMD system),
- The absence of specific system representations in the selected set of federated simulations for the same existing system/platform (e.g., Aegis Weapon System) with different or unique configurations (e.g., SM-3 Block 1A, Navy Theater Wide (NTW), and Navy LEAP ALI),
- Temporal heterogeneity-
 - CAPS is a discrete-event simulation and a hybrid event-driven, time-stepped constructive simulation [Spa00]. In the hybrid event-driven, time-stepped constructive simulation mode, time-compression algorithms reduce run time in scenarios with several periods of intense activity, separated by long periods of relative inactivity prior to the next event. Using time compression, CAPS is able to run an 80-day theater-level campaign in approximately an hour, depending on the available machine resources,

- EADSIM is both event-driven and time-stepped, in a hybrid approach where the C3I processes are event driven and all other processes are time-stepped,
- Geodetic heterogeneity inconsistencies cited in Table 9-15 are a contributing cause of heterogeneous system representation anomalies, including substantive interoperability discussed in Chapter IV. The requirement to accurately model the BMD on worldwide basis dictates a standard geodetic environment, accurately established in all System-Level M&S.

L. BMDS SYSTEM-LEVEL M&S DEVELOPMENT DEMOGRAPHICS

Table 9-18 provides the development demographics for the five BMD System-Level simulations. Official development for all five simulations averaged slightly over three years and ranged from two to five years for the first software release. MDA executive agents²⁶⁴ in Huntsville, AL, and Colorado Springs, CO manage four of the systems developed by contractors. The number of developers cited in Table 9-18 indicates the average annual simulation software development level of effort (LOE) allocated for these systems for the fiscal years 2000-2003, with the exception of EADSIM. The EADSIM program manager determines requirements and accepts resources from over three hundred organizations. Only three of the thirty-six EADSIM developers noted in Table 9-18 directly support MDA requirements.

The simulation software developer metrics for the missile domain M&S is an important metric, as is the location of the major development efforts. The *Fletcher Report* [FMA+84] provided strong justification for mature software development organizations with strong missile domain expertise employing simulations as a design tool noting:

Specifying, generating, testing, and maintaining the software for a battle management system will be a task that far exceeds in complexity and difficulty any that yet has been accomplished in the production of civil or military software systems [FMA+84].

The number of software developers and maintainers for the missile defense domain M&S software identified in Table 9-18 identify sources of the current domain expertise and market presence required to meet the software challenges cited by references earlier in the

²⁶⁴ Missile defense executive agents are government agencies in locations including Huntsville, AL., Colorado Springs, CO., and San Diego, CA., managing MDA-directed and funded projects. Contractors develop most of the products managed by the executive agents.

chapter, and earlier missile defense studies including the *Fletcher Report* [FMA+84], the follow-on *Eastport Study Group* report [CCL+85], and Congressional studies by the Office of Technology Assessment [SCC+88]. The co-location of the simulation software developers with the missile defense program offices and their software developers in Huntsville, AL., and with the developers of the battle management software in Colorado Springs, CO., supported by a distributed network of missile defense expertise play a critical role in the M&S evolutionary acquisition strategy supporting the BMDS.

BMD SYSTEM-LEVEL M&S DEMOGRAPHICS					
	CAPS	MDWAR	EADSIM	EADTB	MDSE
START DEV	1991	1996	1987	1989	1993
RELEASED	1994	1999	1989	1994	1996
EXEC AGENT	MDA	MDA JNIC	ARMY	ARMY	ARMY
DEVELOPER	SPARTA	TRW / RAYTHEON	TELEDYNE (TBE)	HUGHES / RAYTHEON	TELEDYNE (TBE)
STAFFING	4	83	36	65	76
LOCATION	VA	CO	AL	AL	AL
CMM LEVEL	1	3	3	1	3
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]

Table 9-18. BMD System-Level M&S Demographics

The Department requires process maturity from the software acquirers and the software developers. The BMDS government personnel acquiring the simulation software products, must understand the missile defense system evolutionary acquisition strategy [Ald02d] and also develop and define the applicable methods (e.g., Software Acquisition-Capability Maturity Model (SA-CMM), or a similar methodology to determine and report process adherence and performance metrics [AS03]. Similarly, the contractors serving as the builders and vendors for the core simulation software assets must have the software development and performance skills [Gan99] to understand and translate the underlying complex science, engineering, and technology into the missile defense domain M&S software at a minimum of SEI CMM Level 3, or its equivalent. The SEI Capability Maturity Model

(CMM) assessment for three of the five software development organizations for MDWAR, EADSIM, and MDSE is CMM level 3, while a fourth developer for EADTB is pursuing ISO 9000 certification.

M. ANALYSIS OF BMDS M&S SOFTWARE STATE FACTORS

Although these five simulations arrived at their current state by different circumstances and development models, they share several common, important characteristics. First, as noted previously in the M&S synopsis for each simulation, these simulation tools meet a unique requirement (e.g., Human-in-the-loop, Hardware-in-the-loop, architecture development) for the development of the missile defense system. Second, there is a long lineage of experienced stakeholders, a distributed organizational cohort of users, developers, warfighters who understand the underlying missile defense technologies, have developed relationships within the community, interact often on studies, exercises, or analyses, and share the same common mission-focus. Third, these simulations produce results deemed credible enough for major Department decision-makers to support the authorization and allocations of continued Department investments for their evolutionary development. Fourth, these simulation systems represent the repositories of missile defense domain expertise, intellectual capital, and/or presence in the marketplace. Fifth, these simulations are missile defense core assets and repositories for missile defense requirements, knowledge and insights that may not be well documented or understood elsewhere.

Lastly, in an analogous sense, these five legacy simulation systems and support environments survived a Darwinian winnowing process as many other Department simulation systems atrophied, expired, or experienced a stasis of resources during the 1990s. The cohort of organizational sponsors, users, collaborators, and developers for these five simulation systems shared a parallel sociological process as a distributed organizational constituency and have at this point satisfied what Maslow would have identified as the physiological need of survival. The distributed organizational constituency or cohort of supporters, collaborators, and stakeholders, nonexistent when the simulations were first fielded, expanded their knowledge-base concurrently over time with the maturation of the simulations, and contributed significantly to the systems development and longevity. There is an-

other important consideration. All five system-level simulation systems completed the difficult early life cycle phases including the problematic requirements generation process and the continuous justification for funding.

Working within the Departmental transformation process, and supporting an evolutionary block acquisition strategy, the missile defense domain system engineering process, simulation software engineering capability, supporting organizations, and processes need to concurrently mature in an extremely complex and technically demanding environment. Furthermore, when compared to less-complex systems, critical differences in the missile defense software quality requirements exist, which would:

- Permit less opportunity for human intervention,
- Have to handle more objects in its battle space,
- Have to manage a larger battle space,
- Use different weapons and sensor technology,
- Contain vastly more elements,
- Have more serious consequences of failure,
- Have to operate in a nuclear environment,
- Be under active attack by the enemy, and
- Be useless if it failed catastrophically during its first battle [SCC+88].

A review of the Agency's funding documents for the selected core M&S revealed an inconsistent funding profile before 2000 for the reasons noted earlier in the chapter. In addition, resolution of potential Year 2000 problems [Gre97, WG99] and the Department's mandate for HLA interoperability expended funding resources earmarked for simulation development during the 1990's [ISE99]. In the future the core M&S program appears as a relatively stable program according to the Department's Future Years Defense Plan and the Program Objective Memorandum documents, with level and consistent funding levels projected through 2007. However, with no new projected funding additions, future simulation software engineering improvements will compete for a finite set of the Agency's resources.

N. ANALYSIS OF BMDS M&S SOFTWARE DEVELOPMENT PORTFOLIO

Table 9-19 identifies the programming languages and the associated source lines of code (SLOC) of the five software-intensive BMD System-Level Model and Simulations. The BMD System-Level M&S comprise a total of nearly 8.5 million SLOC. This corpus

of code is a domain capital/core asset of the BMD system portfolio. A few programming languages including Ada, C, C++, and FORTRAN equate to approximately ninety-three percent of the total SLOC. The Ada (28%) and C (32%) programming languages are sixty percent of the total SLOC for the five simulations, a metric heavily influenced by the 2.25 million SLOC of Ada code resident in the EADTB simulation. The C or C++ languages are approximately fifty-five percent of the total portfolio SLOC.

A review of Table 9-19 reveals less than two percent of the Ada code written by CMM level 3 software developers. Conversely, the three CMM level 3 software developers identified in Table 9-19, wrote approximately 4.2 million SLOC, or nearly fifty percent of the total 8.5 million SLOC in C or C++. In addition the C and C++ programming languages are also the most commonly used programming languages, with a significant presence in four of the five simulations.

Viewed from the simulation software perspective, MDSE (35%), EADTB (34%), and EADSIM (20%) contain nearly ninety percent of the total 8.5 million SLOC of the five-model portfolio, with EADTB and MDSE approaching the three million SLOC milestones. CAPS, the smallest BMD System-Level M&S employs mostly FORTRAN, C and C++. The C++ and JAVA development environments support the MDWAR effort. EADSIM described by [TOM97] as “object-like nature” and written primarily in C and C++, with some FORTRAN, has the highest population of other COTS and proprietary languages. EADTB is an event-driven simulation written primarily in Ada, with some components developed in C and JAVA, and several other languages. MDSE is the largest language development and maintenance environment including C++, C, FORTRAN, JAVA, and Ada, due to its distributed development environment. Oracle relational database systems house MDSE and EADTB.

In many respects the development cycle of these five simulations listed in Table 9-19 mirror the Department’s software development experience since 1985. CAPS and EADSIM were developed in an incremental, evolutionary manner from an existing capability at relatively low and inconsistent funding levels, separate from the Department’s acquisition thresholds requiring the use of Ada. With the exception of the use of FORTRAN, these systems developed around the languages du jour, primarily C and C++.

Conversely, EADTB and MDSE were relatively large systems with the need to justify significant amounts of Department appropriated funding, which triggered the implementation of the Ada language. Approximately seventy-eight percent of EADTB remains in the Ada development environment, with a significant presence in C and a growing body of JAVA code. In MDSE the amount of Ada code equates to less than five percent of the MDSE system total, while C and C++ correspond to approximately sixty-five percent of the MDSE portfolio, followed by a twenty-eight percent FORTRAN population, the largest FORTRAN presence in the five simulations under study.

BMDS SYSTEM-LEVEL M&S LANGUAGES / SLOC – 2002							
MDL / LANG	ADA	C	C++	FORTRAN	JAVA	Other ²⁶⁵	Total
CAPS		600	650	750			2000
MDWAR			872,434		47,335		919,769
EADSIM		1,338,481	99,705	34,234		225,933	1,698,353
EADTB	2,250,000	490,000			80,000	80,000	2,900,000
MDSE	146,800	891,600	1,051,000	837,300	47,000		2,973,700
TOTAL	2,396,800	2,720,681	2,023,789	872,284	174,335	305,933	8,493,822

Table 9-19. BMD System-Level M&S SLOC – 2002 (After [TEM03])

O. BMDS M&S SOFTWARE INTEROPERABILITY

BMDS System-Level M&S distributed interoperability capabilities includes HLA, DIS, and ALPS. Table 9-20 illustrates the distributed interoperability status of the five simulations under study. Four of the five simulations, for an eighty percent compliance rate, are HLA-compliant, with MDSE, the agency hardware-in-the-loop tool as the sole exception. A review of the simulations supporting the DIS standard also reveals an eighty percent compliance rate, with CAPS being the only simulation under review lacking DIS support. From the interoperability perspective, all five simulations are technically interoperable with either the HLA or DIS protocols, and sixty percent of the simulations, including MDWAR, EADSIM, and EADTB are both HLA and DIS compliant. EADSIM is the only one of the five simulations supporting all three interoperability standards, while CAPS supports only HLA. Overall, the systems under study achieved eighty percent for HLA and DIS interoperability, twenty percent for the ALPS standard, with an overall distributed

²⁶⁵ Other languages include a mix of COTS and proprietary languages: Perl, SQL, HTML, C30, RSRC, H [TEM03]

simulation capability of sixty percent when considering all three M&S distributed M&S protocol standards.

BMDS SYSTEM-LEVEL M&S DISTRIBUTED SIMULATION CAPABILITY						
INTEROPERABILITY	CAPS	MDWAR	EADSIM	EADTB	MDSE	%
HLA	Y	Y	Y	Y	N	80%
DIS	N	Y	Y	Y	Y	80%
ALPS	N	N	Y	N	N	20%
# Avail /Total/ % Avail	1 / 3 / 33%	2 / 3 / 66%	3 / 3 / 100%	2 / 3 / 66%	1 / 3 / 33%	60%
Citation from	[Spa00, Spa01, BBG+99f]	[CM99, TOM99, TRW00a, TRW00b, TRW02b]	[BAC+95, TOM97, BBG+99e, TBE02a, TBE02b]	[BBG+96, Ray01a, Rayo2a, Ray01b]	[Too94, McQ97, TEM00]	

Table 9-20. BMD System-Level M&S Interoperability Status

P. EDUCATION ENABLERS SUPPORTING M&S CREDIBILITY

A few years ago a television commercial showed a scene where a company executive, after receiving a presentation on the problems facing his company, asked the same team to help resolve the issues. In response the presenters noted they only identified problems, and did not fix them. The Department's history with resolving persistent systemic software issues is analogous to this scenario. The Department has done a thorough job identifying the many issues adversely affecting simulation software credibility. There have been many significant policies, programs, processes, and procedures implemented to counter these issues, although, Department simulation software still lacks overall credibility.

One of the underlying reasons for a lack of credible simulations is the Department's lack of depth, understanding, experience, and maturity with disciplined software engineering. We believe a critical pre-condition for resolving this shortcoming is the need for continuing software engineering education [Tuc02] within the Department, acknowledging the software engineering field is still coalescing [Boe02, CT02, HH02a, HH02b, SBM02, SHM02, UHC02]. In a similar situation during the 1980's the Department identified the need to improve the professional development of an industrial-age acquisition workforce and implemented the Defense Acquisition Workforce Improvement Act. Understanding

the changing, challenging dynamics represented by today's information-age working environment, and the future demands of continuous transformation, the Department instituted a Continuous Learning Policy [Ald02f], acknowledging the need to operate as a continuous learning community, continually striving to improve professional knowledge and performance in the Acquisition workforce.

Today, the Nation has an information-based economy and the Department has an information-based national military strategy based on software. Unfortunately, there are few indications that the Department has developed the new capabilities to define, develop, acquire, and manage the myriad of complexities involved in software-intensive systems. The Department recently acknowledged this shortfall and developed a program [Ste02b] to enable selected members of the Department's military and civilian workforce earn master or doctoral degrees in the relevant scientific, technical, and management academic disciplines supporting information assurance.

In a similar program, Dr. Patricia Sanders, the current MDA System Executive Officer and former BMDO Deputy for Test, Simulation, and Evaluation, initiated the BMDO Software Engineering Improvement Program in 2000, in conjunction with the Naval Postgraduate School's (NPS) Software Engineering Distance Learning program, pioneered by Professor Luqi. The NPS Software Engineering Distance Learning program provides Software Engineering programs at the master and doctoral level, for Department software practitioners to study, research, and advance software engineering principles and technology vital to Department researchers and program managers [NPS02]. The BMDO Software Engineering Improvement Program complemented the Agency's Acquisition Certification Program supporting employee Level III Defense Acquisition Corps certification within eighteen months of arrival and an eighty-hour continuous learning program for the Level III workforce every two years.

The melding of information technology employed in the NPS distance-learning model, with the collaborative ability of a distributed learning process shared by multiple major Department acquisition programs in different domains located around the Nation (e.g., the Army's Tank and Automotive Command, (TAACOM) in Michigan, the Navy's Space and Warfare Command (SPAWAR) in California, and the Missile Defense Agency

(MDA) in Washington, DC), employing the adult, cooperative learning model (e.g., study- and work-related research), working closely with and mentored by the NPS Software Engineering faculty appears to be exactly the type of education Transformation model the Department needs. Software engineering research supported by the NPS Distance Learning model also has a reciprocating function, continually updating the academic community with real-world research issues, and providing a source for the latest research to resolve current Department software engineering challenges. A recent example of this successful paradigm was the publication of an NPS Software Engineering Distance Learning Master's thesis *Conceptual Framework Approach for Systems-of-Systems Software Developments* [Caf03] in support of the BMDS evolutionary acquisition strategy²⁶⁶.

Q. SUMMARY

Chapter IX provided an overview of the BMDS M&S domain and included domain background, the domain M&S hierarchy, a BMDS System-Level M&S synopsis, M&S demographics, and a review of the missile defense system representations populating the five BMD System-Level simulations under study in this dissertation. The BMD domain overview section provided the Agency background and highlighted the significant role Department organizational changes played in the current status of BMD System-Level M&S.

Building on the organizational outline and M&S domain overview, the chapter continued with a status review on the domain implementation of the Department's policies for establishing credibility, including a concise VV&A background for each of the BMD System-Level M&S. Summaries of the other M&S in the domain M&S hierarchy provided additional context for the analysis. A top-level review of the BMD System-Level M&S fidelity, and the foundations for radar sensor fidelity followed.

The scope of this research included five large-scale, Missile Defense Agency Domain legacy simulations. The research methodology supported by the NPS software engineering distance-learning model facilitated the timely study of Department primary source

²⁶⁶ The Missile Defense Agency received the Department of Defense 202 M&S Award in the Acquisition category for the MDA "Enterprise Strategy for Modeling and Simulation" from Dr. Ronald Sega, Director of Defense Research and Engineering at a Pentagon ceremony on September 29, 2003. The NPS distance learning software engineering program and software engineering research supported by the NPS Software Engineering Department faculty at Naval Postgraduate School, Monterey directly supported many of the MDA M&S program improvements cited in the award.

material for software-intensive legacy simulation systems. This case study also employed selected product line practice areas (Organization Management, Technical Management, Software Engineering) as a tailored framework for the missile defense domain analysis.

The analysis identified additional root causes for heterogeneous anomalies (e.g., substantive interoperability issues) in the BMD System-Level M&S, although a primary cause was not established. Research indicated that a number of factors converged to create the conditions favorable for the development of heterogeneous system representation anomalies in Department simulations, affectively reducing credibility in the simulation or federation operations, and ultimately reducing confidence in the results. Results from the Agency five-simulation sample closely correlated to the Department's experiences with establishing simulation and federation credibility identified in Chapters II through Chapter VI.

THIS PAGE INTENTIONALLY LEFT BLANK

X. A SOFTWARE ARCHITECTURE-BASED PRODUCT LINE MODEL FOR SIMULATION MODEL REPRESENTATIONS

A. INTRODUCTION

Chapter X introduces the detailed research design for the Simulation Software Architecture-Based Product Line Model including five major elements supporting the method, specification, design, and implementation. The chapter provides an architectural analysis of the Department's existing *de facto* M&S software architecture and introduces the:

- ***Simulation Software Architecture (SSA)***-Based Model as an abstract software architecture-based horizontal foundation supporting multiple viewpoints and views,
- ***Simulation Software Architectural Framework (SSAF)***, a second vertical-slice²⁶⁷ architectural component overlaying the SSA, which includes system and environment components [ABB+02, GA03],
- ***Simulation Product Lines Architecture (SPLA)*** providing the framework to manage the variability²⁶⁸ [BB00, Bos00, DMH01, RSC00, MNJ+02, CBB+03], features [Bos00, KLD02], and differences between products comprises the third element,
- ***Simulation Software Architecture-Based Product Line Model Domain Metadata Repository*** provides the structure for the Domain Metadata Registry modeled from [ISO79c] to ensure interoperability with Department, Federal Government, and private sector metadata registries,
- ***Architecture Readiness Levels (ARL)*** to measure future architectural components and products developed from this methodology, is the fifth and final contribution

The SSA and SSAF establish the architectural foundation for the SPLA, which supports variability and extensibility of the architectural construct. We also adapted the architectural description conceptual framework²⁶⁹ from [IEE00B] into the SSA, SSAF, and SPLA models composing the Simulation Software Architecture-Based Product Line Model.

The chapter also suggests a complementary Domain Integrated Product Development Team (DIPDT) approach for implementing the Simulation Software Architecture-Based Product Line Model, to reduce the cycle time for resolving the multiple dimensions

²⁶⁷ In UML this is called a partition or a set of related classifiers or packages at the same level of abstraction or across layers in a layered architecture [UML99].

²⁶⁸ Variability refers to decisions that will be made by a member of the development team prior to system deployment [CBB+03].

²⁶⁹ The conceptual framework is the term of reference for the architectural description and establishes the terms and concepts for the content and use of architectural descriptions [IEE00b].

causing heterogeneous system representation anomalies, and improving federation interoperability.

B. THE INFLUENCE OF SOFTWARE ARCHITECTURE IN DEPARTMENT M&S

The focus of this research is to improve the quality, accuracy, and consistency of authoritative representations in Department M&S, a major element in addressing the current heterogeneous system representation anomalies (e.g., substantive interoperability), which erodes credibility in the Department simulation process, and injects doubt instead of confidence in the simulation results supporting Departmental- and National-level decision-makers. This research indicates that after fifty years of development the Department M&S domain has not achieved the desired level of credibility and confidence in simulation model representations, in part due to a lack of data sharing and data standardization problems.

There are many reasons cited below for this assessment including:

- Few mechanisms for enabling global data acquisition and interchange, especially across domain application areas,
- The lack of unique global identifiers for standard data elements,
- Inadequate documentation of data element characteristics restricting the usefulness of automation to locate, retrieve, and exchange data,
- A need for uniform guidance for identifying, describing, and developing data elements,
- Locating and retrieving a particular data element is difficult or impossible,
- The absence of a universal means for organizing standard data elements,
- The lack of inter-organization data standards, and to a lesser extent, the lack of intra-organization common data standards,
- A proliferation of customized data interchange representations,
- Imprecise data definitions and descriptions, limiting reuse opportunities, or multiple uses of the data,
- A lack of standard data elements impeding global implementation of Electronic Data Exchange (EDI) [ISO79a].

In contrast to the mature computer hardware-engineering sector, the software engineering sector of the computer software industry is still developing a consensus on a software engineering body of knowledge [Moo99] and agreement on the benefits of software process improvement and definitions of software architecture. Although standards exist for

three of the four components needed for open information processing systems (e.g., hardware, software, and communications), standards for the fourth component of open information processing systems, data specification, is still under development by the international standards community and a relatively new member of the Open Systems Interconnection Environment [ISO79a]. In addition, there are many other salient issues including information operations, information assurance, and the security of personal information from misuse for Department software engineers and software architects to consider.

1. The Department's *De Facto* M&S Software Architecture

Software architecture, even if it is a *de facto* software architecture, strongly influences a system over its life cycle. While computer hardware-related architectures supported by Moore's Law were ascendant over the past fifty years, software-related architectural considerations, if they existed, were of secondary importance to the overall system engineering and development process during much of the period. However, the cost and complexity of software-intensive systems changed the industry's dynamics, as reflected by the publication of software architectural styles [SC96, BK99, HHP00, Bos00, MM01, CBB+03, Fra03] descriptions [IEE00b], and patterns [Fow97] for software-intensive projects provided new views and theories on software architectural principles to the software engineering community. However, the emerging software architecture concepts currently affect little of the Department's M&S software life cycle management practices for large-scale, legacy, software-intensive systems and simulation model representations.

2. A Layered Architecture Model Approach

The Department's M&S Hierarchy provides the standard view of M&S today. However, the current M&S Hierarchy view lacks integration, treats abstraction inconsistently, lacks quantitative measurements, and is more aptly considered a simulation heuristic. From a different view, the current *de facto* Department M&S software architecture view²⁷⁰ appears closely aligned to a layered style²⁷¹. Although there are several software

²⁷⁰ A view is a representation of a set of system elements and the relationships associated with them [CBB+03]. [CBB+03] also views each layer as a virtual machine with constraints on the relationships among the virtual machines.

architectural styles and descriptions cited in Chapter VIII, layering supports the principle of information hiding, theoretically providing a capability to change a lower layer behind the interface and not impact the layers above it.

The Department's M&S Hierarchy continually evolved to support the changing requirements dictated by new strategies, evolving doctrines, funding justification, changing organizations, new weapon systems, and the growing demand for credible simulations. It is clear from the research that the Department's M&S Hierarchy provided a commonly accepted way of providing a high-level communication vehicle to convey an imprecise qualitative definition of a certain level of simulation fidelity (e.g., engineering, engagement, mission, campaign). However, the Department's M&S Hierarchy is not an architectural construct and lacks any clear definition of the individual layers or the interface mechanism between layers.

In addition, the hierarchical structure inaccurately implies that the higher layers of the Department's M&S Hierarchy build on the lower layer(s) in a logical, well-engineered manner. Moreover, since the Department's M&S Hierarchy lacks an architectural connection with the many simulations it represents, changes to either the simulations or the hierarchy lack synchronization or cohesion. Lastly, without an architectural framework, the current Department's M&S Hierarchy appears extremely limited in its ability to meet future requirements for component-based development and rapidly composable simulations.

More specifically, the *de facto* Department M&S software architecture view appears to have two layers, the Conceptual Layer and the Implementation Layer, illustrated in informal notation in Figure 10-1. The Conceptual Layer theoretically supports and maintains the validated CMMS or FDMS conceptual models described in Chapter V. The Implementation Layer is the software implementation and deployment of the M&S system, theoretically verified against the validated conceptual model of the real world under study.

²⁷¹ Layering reflects a division of software into units, with each layer representing a unit [CBB+03]. [CBB+03] also views each layer as a virtual machine with constraints on the relationships among the virtual machines (e.g., an abstract computing device).

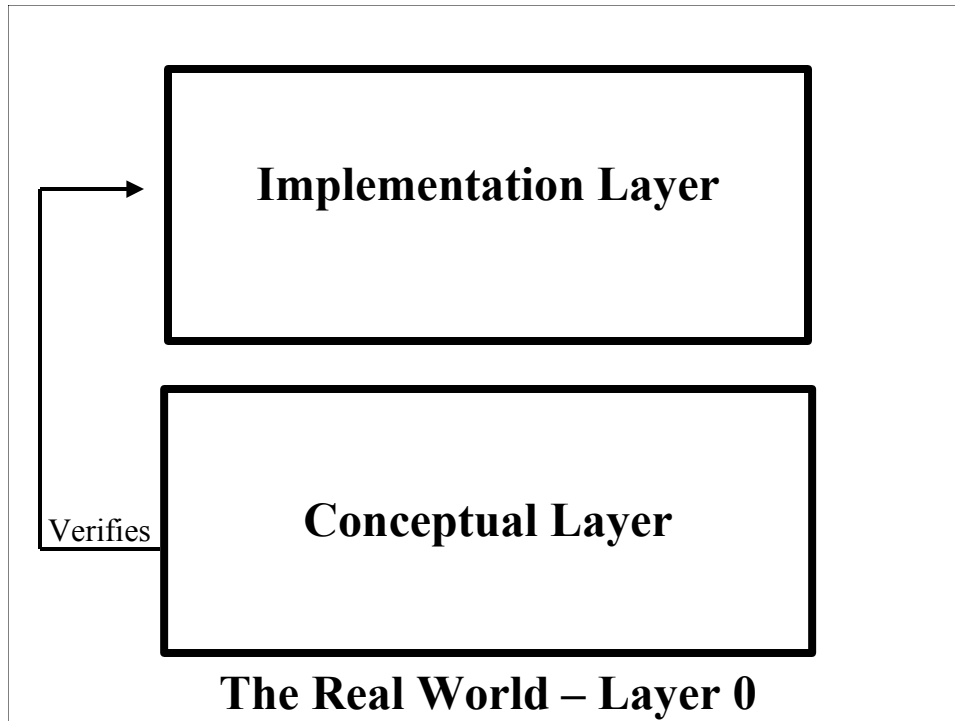


Figure 10-1. Department's *De Facto* Two-Layer M&S Architecture

In reality, at Department and the Agency level, the theoretical development of validated conceptual models representing the first abstraction of the real world, and the subsequent verification of the simulation software implementation with the validated conceptual model occurred more often in literature than in practice as noted in the Department's V&V experience cited in Chapter III through Chapter VI; and the Agency case study in Chapter IX. In addition, the development of a validated conceptual model and verified software implementation indicates a high level of coupling, which may hinder the Department's objective for greater reuse of components and composable M&S frameworks.

C. THE HORIZONTALLY-LAYERED SIMULATION SOFTWARE ARCHITECTURE

1. The Software Architecture Layered Style

The first element of the Simulation Software Architecture-Based Product Line Model is the Simulation Software Architecture (SSA). The proposed simulation software architecture style is the layered style. The layered view of the Department's simulation software architecture, graphically presented as a layer diagram, segments software into each layer with constraints, connected by a public interface. Properly developed layered-style diagrams support the development of software-intensive simulation systems featuring portability, interoperability, and modifiability facilitating reuse, component-based development, and future composability research initiatives. The layered diagram sees common use, although some software architects employ poorly constructed layered diagrams, often using facilities of a higher layer without restrictions.

Properly constructed layered architecture diagrams share a basic quality in which the layers interact according to a strict ordering relationship. Layered diagrams employ an *allowed to use*²⁷² relationship. In the proposed simulation software architecture (SSA), the Referent Layer is beneath the Conceptual Layer and provides the views and viewpoints into the real world (e.g., Layer 0). The implementation of the Conceptual Layer may use all of the public facilities provided by the Referent Layer through the interface. The Conceptual Layer is beneath the Component Layer. The implementation of the Component Layer may use all of the public facilities provided by the Conceptual Layer through the interface. The Component Layer is beneath the Implementation Layer. The Implementation Layer may use all of the public facilities provided by the Component Layer through the interface.

We reviewed several other architectural styles as candidates for the SSA, including pipes and filters [SC96], layers [BBG+00], blackboards [Bos00], object-orientation, distributed and systems, component and connector, interactive systems and several other styles described in Chapter VIII. [Bos00] describes this activity as imposing an architec-

²⁷² The *allowed to use* is a variant of the *depends-on* relation, where if P_1 uses P_2 , P_1 's correctness depends on the correct implementation of P_2 [CBB+03].

tural style on the software architecture, requiring careful consideration in the process and the impact of a complete reorganization of the software architecture. The primary goals of the SSA are to provide an architectural framework to support constantly changing requirements and secondly, reduce simulation software complexity. Quality attributes support each SSA layer. Figure 10-2 illustrates the proposed SSA model.

In addition the review encompassed the decomposition style, uses style (e.g., *depends-on* relationship), generalization style (e.g., *is-a* relationship), and the layered style [CBB+03]. The selected stack layer model for the SSA uses geometric adjacency to represent the *allowed-to-use* protocol, as opposed to symbols (e.g., arrows, lines, diamonds). Other informal layered notations evaluated for the SSA layered style included segmented layers, rings, three-dimensional models, and layers with sidecars [CBB+03], with which we compared and contrasted the architectural attributes against the informal stack notation which represents the layers as a stack of rectangles.

The SSA restricts upward communication by a lower layer of the architecture with the facilities of higher layers to ensure the validity of the desired architectural attributes and to retain the required properties to support the V&V process. Each horizontal slice or layer of the stack supports a vertical slice or abstraction of the product line, discussed later in the chapter, and no two layers contain the same product line abstraction. The thick horizontal lines between the layers indicate the constraints of interlayer communication to the interface protocols at the layer interface, with no other communications to the internal facilities of any other layer.

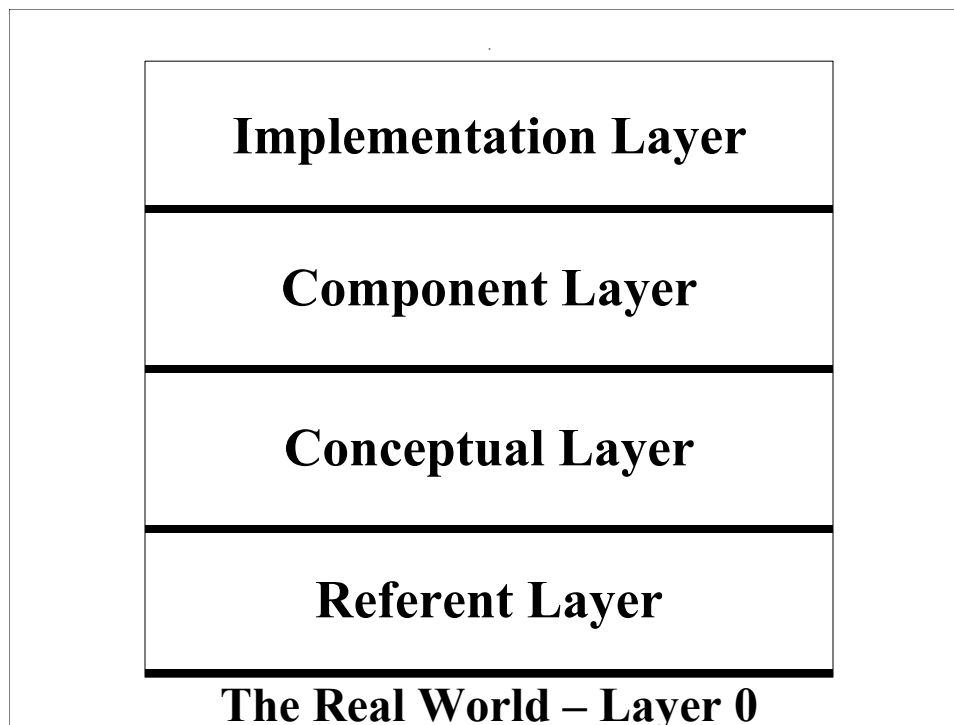


Figure 10-2. Simulation Software Architecture (SSA) Model

2. Viewpoints and Views

A *viewpoint* provides the pattern or framework, based on the stakeholders' intended use of the system, for specifying the view design and development. Viewpoints selected for the simulation software architecture description (SimSAD) may originate from the development effort or defined elsewhere and reused in the SimSAD, termed a library view by [IEE00B]. Stakeholders select one or more viewpoints for the architectural description of the system supporting the intended use of the simulation. A viewpoint sets the modeling methods, procedures and analysis techniques for analyzing, creating, and displaying a view representation. The minimum specification for viewpoint metadata maintained in the Simulation Software Architecture-Based Product Line Model Domain Metadata Repository includes:

- The viewpoint name,
- List of stakeholders supported by the viewpoint,

- The intended use, mission, and concerns addressed by the viewpoint,
- The language, modeling techniques, analysis methods used to develop a view based on the viewpoint,
- The source for the viewpoint or library viewpoint including data sources for validation,
- Formal or informal consistency and completeness tests applied to the models used to develop a view,
- Evaluation or analysis techniques to be applied to the models,
- Patterns, heuristics, and other guidelines supporting the development of a view [IEE00b].

We define a *view* as a representation of a whole system or set of system elements from the perspective of a related set of concerns and the relationships associated with them. The view addresses stakeholders' mission or intended use of the system in the simulation or federation. A view may also include one or more architectural models, which may support one or more views, derived from another associated viewpoint architecture.

Viewpoints and *views* are major contributors to this dissertation. Take the example of a house. The electrician, carpenter, plumber, bricklayer, HVAC (e.g., heating, ventilation, and cooling) contractor, cement contractor, and landscape specialist all have different viewpoints of the same house. All these *viewpoints* are correct since they support the *mission* or *intended use* for the rules, procedures, and guidelines (e.g., appropriate building codes) the different craftsmen must follow to develop the different *views* of the architect, surveyor, builder, and owner. Different views are also important, since the builder wants the potential owner to buy the house, and later on the homeowner hopes the tax appraiser takes a lower-valued *view* of the property, while potential new buyers take a higher-valued *view*.

3. The Referent Layer

The Referent Layer is the architectural framework of the SSA supporting viewpoints. Previous authors cited the need for a referent as a commonly understood standard [Gro99, RGH00], or a properly developed conceptual model supporting fidelity [GFT98, Pac01b] and validation [Pac01b]. We view *system referents* as software architecture constructs, and components of the SimSAD, populating the first layer of the SSA, the Referent

layer. The single SSA Referent Layer in Figure 10-3 replaces all levels of the Department's M&S Hierarchy discussed in Chapter II (see Figure 2-2) and employs viewpoints of the real world as portals into the architectural framework. Viewpoints are new to the Department's simulation software architecture design process. Viewpoints represent a one-to-one mapping from the real world into the architectural sink, the system referent, maintained in the referent Layer. Referents support one or more real-world viewpoints.

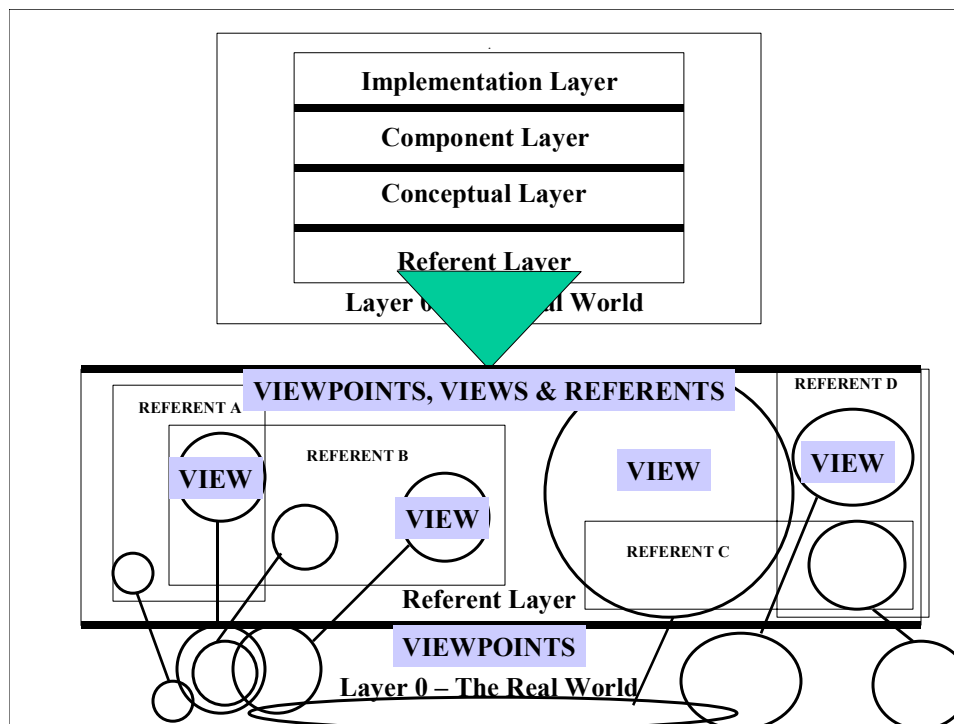


Figure 10-3. SSA Referent Layer Model

The Referent Layer performs the transaction manager role for the SSA. The Referent Layer continually adds new viewpoints to the referent, while existing viewpoints change, evolve, and merge to new meet requirements, until archived pending future use. The Referent Layer provides the first interface to the services provided by the Simulation Software Architecture-Based Product Line Model Metadata Repository. This process applies to all data sources including new system requirements and the core asset development methods discussed later in the chapter, including mining and reverse engineering.

4. The Conceptual Layer

The Conceptual Layer in Figure 10-4 houses the conceptual views of the systems. Where the Referent Layer provided the transaction manager role for the SSA, the Conceptual Layer provides the facilities for conceptual composition. The major component of the Conceptual Layer is the Simulation System Architectural Description (SimSAD). Each SimSAD comprises a specific *system*, which may include product lines, product families, complete enterprises, aggregated entities, subsystems, components, systems, and systems of systems, and their interfaces inhabiting an environment.

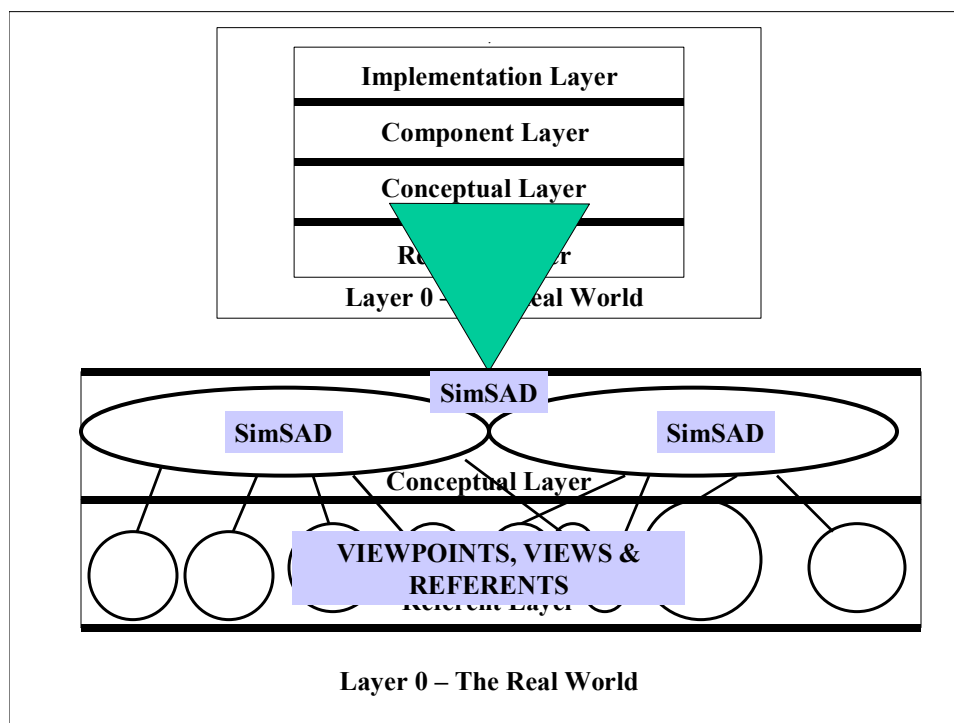


Figure 10-4. SSA Conceptual Layer Model

The Conceptual Layer provides facilities to “pull” viewpoints and referents from the Simulation Software Architecture-Based Product Line Model Metadata Repository provided by the Referent Layer to support the development of views for the SimSAD and conceptual model construction. Conceptual views support the development of specific conceptual models designated for an intended use or mission; or common-use conceptual models;

or conceptual models specifically designed for large-scale reuse. The Simulation Software Architecture-Based Product Line Model Metadata Repository provides configuration control services for all views in the Conceptual Layer, and provides the interface control with the next higher layer, the Component Layer.

5. The Component Layer

The Component Layer in Figure 10-5 houses and manages the component views. Where the Conceptual Layer provided the facilities for architectural conceptual composition, the Component Layer provides facilities to “pull” views from the Simulation Software Architecture-Based Product Line Model Metadata Repository provided by the Conceptual Layer to support the development of design strategies for component models.

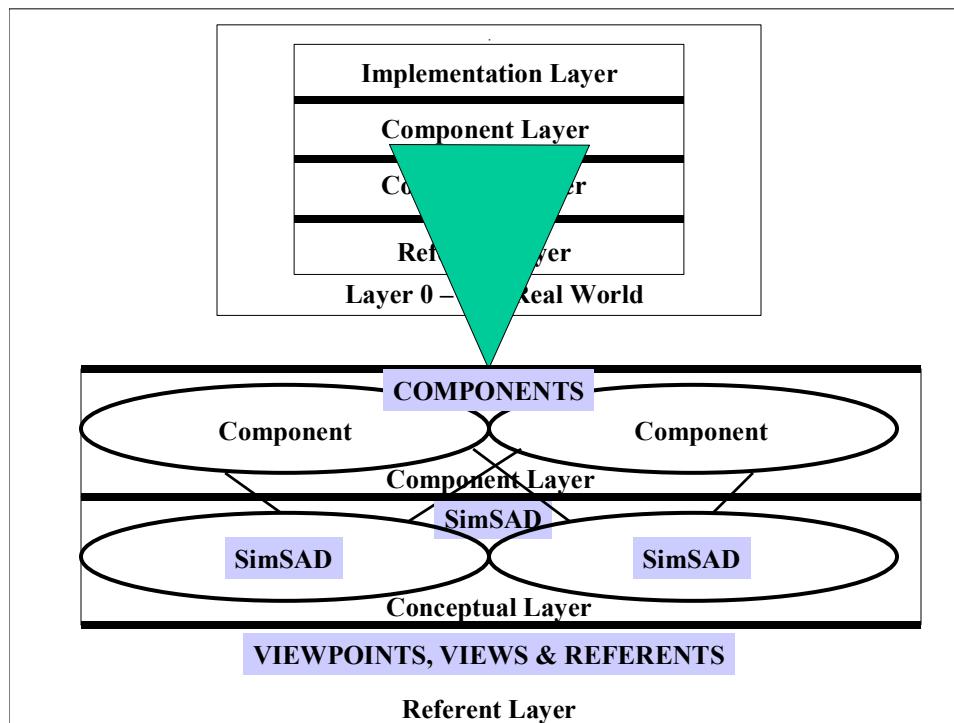


Figure 10-5. SSA Component Layer Model

Conceptual views support the development of patterns of interaction for specific component models designated for an intended use or mission; or common-use component models; or component models specifically designed for large-scale reuse.

The Component Layer provides the architectural foundation for future composable item development by providing standards and conventions for quality attributes [WBL+01], composition rules, and visibility of assets. Components were discussed in Chapter VIII. The Simulation Software Architecture-Based Product Line Model Metadata Repository provides configuration control services for all SimSAD components in the Component Layer, and provides the interface control with the next higher layer, the Implementation Layer, which “pulls” component views as required. The development of application components is beyond the scope of this research.

6. The Implementation Layer

The Implementation Layer in Figure 10-6 houses and maintains the complete SimSAD system view, and provides the visibility of the component views available to implementation layer developers. Where the Conceptual Layer provided the facilities for conceptual composition, the Implementation Layer provides facilities to “pull” component views from the Simulation Software Architecture-Based Product Line Model Metadata Repository provided by the Component Layer to support the development of system model views. Component views support the development of specific system models designated for an intended use or mission; common-use system models; or system models specifically designed for large-scale reuse.

The Implementation Layer provides the architectural foundation for future composable system development. The Simulation Software Architecture-Based Product Line Model Metadata Repository provides configuration control services for all systems in the Implementation Layer and provides the interface control with the next lower layer, the Component Layer, “pulling” component views when required. The development of composable application systems is beyond the scope of this research.

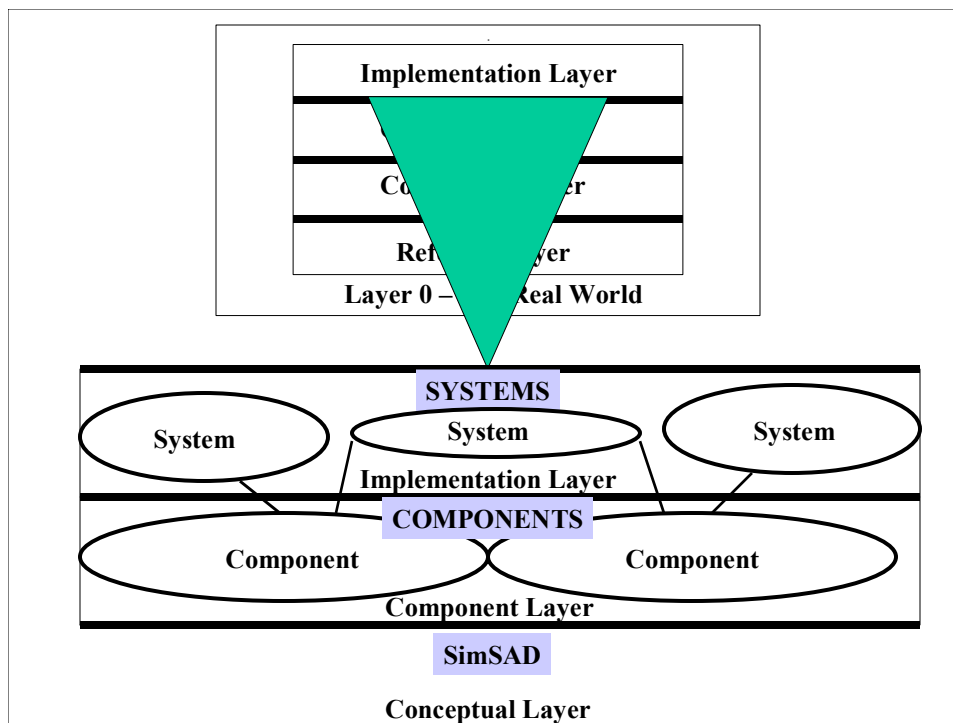


Figure 10-6. SSA Implementation Layer Model

7. The Simulation Software Architecture Framework (SSAF)

The second element of the Simulation Software Architecture-Based Product Line Model is the Simulation Software Architecture Framework (SSAF), which provides the system environment components. The SSAF in Figure 10-7 adds the vertical dimension of the system environment to the horizontal architectural layer presented in the SSA. The requirement for a vertical dimension is an unusual requirement since many Department software systems developed in the past have evolved in a vertical manner, often referred to as “stove-piped” systems (e.g., personnel, logistics, medical, command and control), and lacked horizontal integration and interoperability. However, current M&S Hierarchy models lack structure to vertically integrate the system model representations in a consistent fashion.

Although the Department’ simulation development history experienced a similar pattern, architecturally it developed a different growth pattern. The simulation framework

and its internal models presented a *de facto* self-contained horizontal architecture layer, within the simulation framework (e.g., engine, scenarios) providing the horizontal integration and interoperability. However, the Department's M&S Hierarchy provided few vertical integration enablers supporting a cohesive process for managing quality attributes for the same system at different levels of the M&S Hierarchy. This systemic shortcoming adversely affected the Department's MS credibility. The SSAF presents a method to improve the Department's vertical integration of M&S. The system component of the SSAF supports the SimSAD. The system *environment*²⁷³, includes the physical world and the external objects, conditions, or processes influencing the behavior of the system in past, present, and future dimensions, real or imagined, but clearly defined and documented in the SimSAD.

A system has one or more stakeholders, with interests, intended uses and concerns²⁷⁴. SimSAD stakeholders include simulation developers, architects, simulation sponsors, V&V agents, accreditation agents, and ultimately Agency-, Department-, and National-level decision-makers. A system design also incorporates the intended use, or fulfillment of a mission²⁷⁵ in the system's environment. We use the terms *mission* and *intended use* interchangeably. The architectural description component of the SimSAD includes the compilation of products to document architectures [IEE00b].

²⁷³ The system's environment or context may influence the system's setting, including circumstances of operational, developmental, political and other influences affecting the system. The environment may include other systems' direct or indirect interaction with the system of interest through defined interfaces, and also defines the boundaries and scope for the system of interest [IEE00].

²⁷⁴ Stakeholder Concerns include the system's development, operation, or any aspect of the system deemed critical by the stakeholder, including quality attributes detailed in Chapter VI.

²⁷⁵ A mission involves the operation or use of system by its stakeholders to meet objectives developed for the systems intended use [IEE00b].

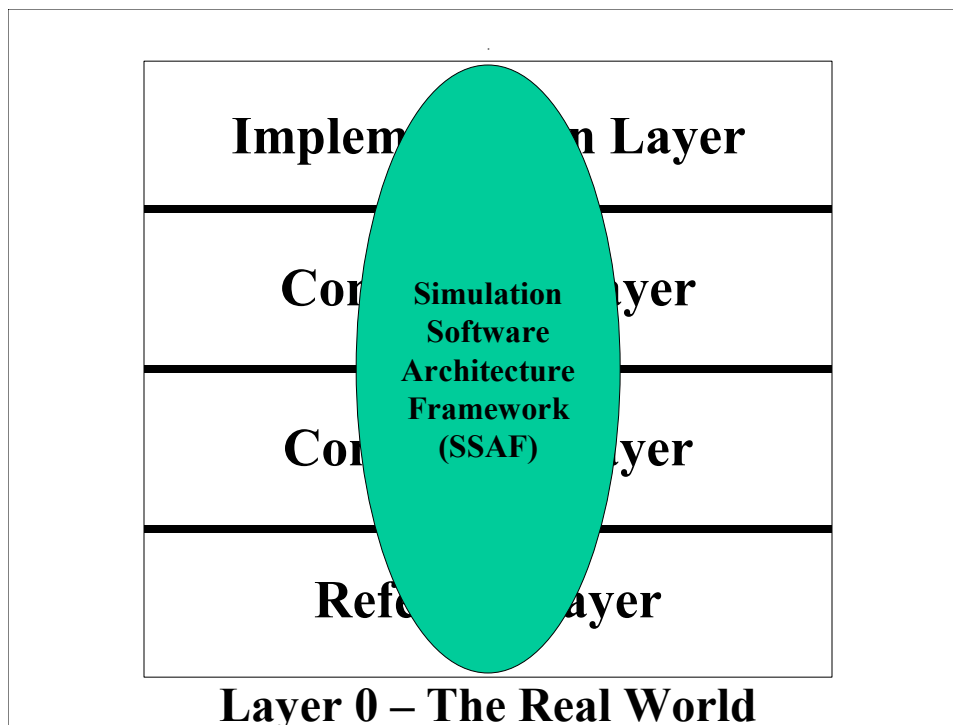


Figure 10-7. Simulation Software Architecture Framework (SSAF)

Recall that all systems have a defined or *de facto* architecture. The SimSAD describes the conceptual architecture of a system and supports the follow-on development of product line descriptions for particular discrete instantiations of that specific architecture. The SimSAD maintains one or more constituent views and viewpoints. In addition to the selected views and viewtypes the SimSAD maintains other information and documentation supporting verification, validation, and accreditation processes, including a system overview, system context, product family relationships, URLs for system validation data, and maintenance information. [IEE00b] provides a concise listing of uses of architectural descriptions, architectural description practices, and recommended documentation supporting a SimSAD, including the specification for a group of systems sharing a common set of features (e.g., product lines). The SimSAD also maintains records and analysis data of known architectural inconsistencies.

8. Software Architecture Reconstruction

Software architecture reconstruction provides a mechanism for the effective reuse of software assets [KC97, CW98, Obr01a, Obr01b, Kov01, OSV02]. Architecture reuse is the cornerstone practice of product line development, supported by three sources for architecture recovery:

- Source code, which is authoritative, but does not hold all the information,
- Documentation, which is normally undependable, incomplete, or outdated,
- Human experts, who may be dependable, but biased [CW98].

The SEI developed the Options Analysis for Reengineering (OAR) methodology [BSW+99, BS00, CN02] for making technical, organizational, and programmatic reengineering decisions. The OAR methodology:

- Combines architectural and reengineering views with defined mappings between the views,
- Classifies and stratifies reengineering options/approaches into distinct layers with a mapping between layers,
- Provides key information for making informed choices about the time and criteria for each option/approach,
- Codifies technical and non-technical risks for each option/approach,
- Relates organizational and programmatic factors to support a unified reengineering option/approach [BSW+99].

[BSW+99] indicate the reengineering process includes three basic steps to evolve an existing legacy into a new system in a disciplined evolutionary process employing the Horseshoe Model, in Figure 10-8, which combines reengineering and architectural views of software analysis and evolution including:

- Analysis of the existing system, and extracting artifacts from source code, the architecture recovery / conformance process,
- The architectural transformation of the legacy systems logical descriptions into new improved descriptions,
- Architecture-Based Development (ABD) of the new system based on the new logical descriptions to instantiate the desired architecture [BSW+99].

The Horseshoe Model [BSW+99] shown in Figure 10-8 includes three levels. These levels according to [BSW+99] consist of a code-structured representation, which includes parsing and analysis of the source code, abstract syntax trees (ASTs), and flow

graphs at the first level, and a functional level representation describing the relationships of functions, data and files at the second level. At the third level, the Horseshoe Model illustrates the concept level representing combined function and code level artifacts used as the base of new architectural components or concepts.

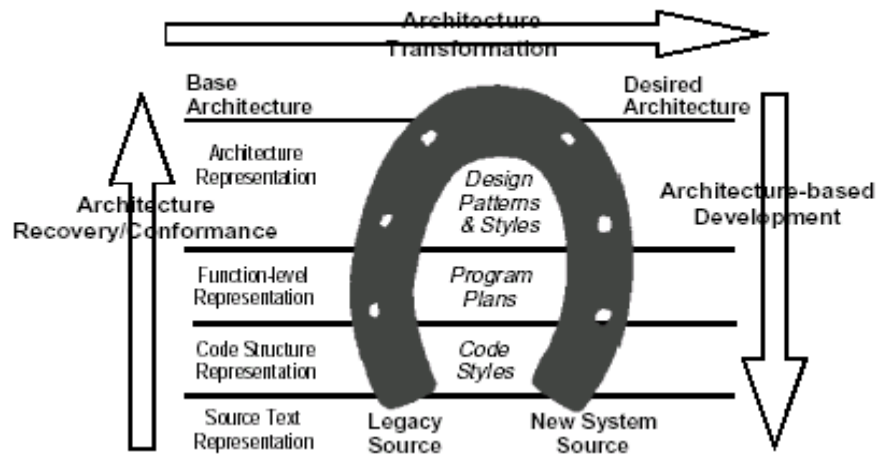


Figure 10-8. The Horseshoe Model Underlying OAR (From [BSW+99])

The source-code level transformations may employ string matching and replacement techniques, or code-structure transformations based on syntax tree-based transformations. Functional level transformations go beyond code-level changes and are concerned primarily with the reworking of the functionality, which may include changing from a functional design to an object-oriented design. The final transformation process, at the architectural level, involves changes to the basic building blocks including the typed of components, patterns of interaction, control mechanism, functional allocation, and data, and normally requires the greatest amount of time and resources according [BSW+99].

Architecture recovery and reconstruction is an iterative and interactive labor-intensive effort, dependent upon the level of specification, documentation, dissemination and control of the legacy architecture artifacts [KOV01, OSV02]. Architecture recovery methods vary from entirely manual reconstruction processes to tool-supported manual reconstruction and semi-autonomous reconstruction techniques, and may include data min-

ing²⁷⁶ and employment of architecture description languages [OSV02]. Pattern analysis supports architecture recovery and reconstruction efforts [OSV02]. [Fow97] addresses two categories of patterns:

- Analysis patterns are groups of concepts that represent a common construction in business modeling, relevant to one or many domains,
- Supporting patterns are patterns in their own right and valuable on their own, although the primary purpose supports the use of analysis patterns, making them real [Fow97].

Manual mining activities recover source information for architecture activities supporting product line development from the source code, documentation, and human experts, since automated tools²⁷⁷ are currently limited. [KOV01, OSV02A] suggest the use and synthesis of existing several tools and techniques into a “workbench” support environment for software analysts reconstructing architectures. A significant number of tools support view extraction, and several tools support view fusing and reconstruction.

[HBL97, KC97, Obr01a, Obr01b, BBC+02] support architecture reconstruction with automated tools; however, [OSV02] note that tool use is limited by a system using several languages, or on cases where the binary code is available, but not the source code. The possible lack of source code is a real possibility in reconstruction efforts with commercial components and the vendors decline to provide the source code [OSV02].

D. THE VERTICALLY-SLICED SOFTWARE PRODUCT LINE ARCHITECTURE

[Bos00] suggests three purposes for architecture: as an individual software system, as product-line architecture²⁷⁸, or as a standard architecture for component development. Research indicates that component-based technologies remain immature, suggesting that an organizationally sponsored (e.g., domain) product line-based approach to implementing simulation software architectures, may serve as a predecessor technology to a follow-on component-based development environment.

²⁷⁶ Mining includes resurrecting and rehabilitating a piece of an existing software system to serve in a new system for which it was not originally intended [Nor03].

²⁷⁷ One existing tool is Dali is an SEI tool for extracting architectural information from an implemented system [CW98]

²⁷⁸ [Bos00] defines the use of a product-line architecture as the common architecture for a set of related products or systems developed by an organization.

[Bos00] also identified two factors for the approach to initiating a product line: evolutionary or revolutionary; and suggested two market-based decisions to develop a new product line or apply a product line methodology to a legacy product. Risk plays a major role in the decision process, especially in mission-critical, high-risk Department domains. Employing an evolutionary product line approach for large-scale, software-intensive legacy simulation systems has a potential in the Department's M&S domain to:

- Minimize investment risk and risks to continuity of operations,
- Support the development of authoritative representations,
- Reduce federation issues caused by substantive interoperability problems,
- Identify the level of quality in the component product, and not just the level of maturity in the processes supporting development,
- Support improved VV&A practices,
- Provide metric/measured credibility of Department M&S supporting improved confidence levels,
- Initiate the transition of Department M&S to a supportable Department software architecture,
- Capitalize on the current core asset investment,
- Improve component syntactical / semantic inconsistencies between Department domains,
- Provide a foundation for an integrated data and object model ontology,
- Provide a common open source for validated algorithms,
- Reduce overall life cycle support costs.

It is possible to implement a software product line architecture shown in Figure 10-9 at various defined levels—for example: enterprise, domain, functional, system, subsystem, or component level. A key step in the systems and software engineering process is the iterative allocation of system requirements to hardware and software. This is also true for software-intensive product lines since the system may include hardware (e.g., hardware in the loop) or live simulation entities.

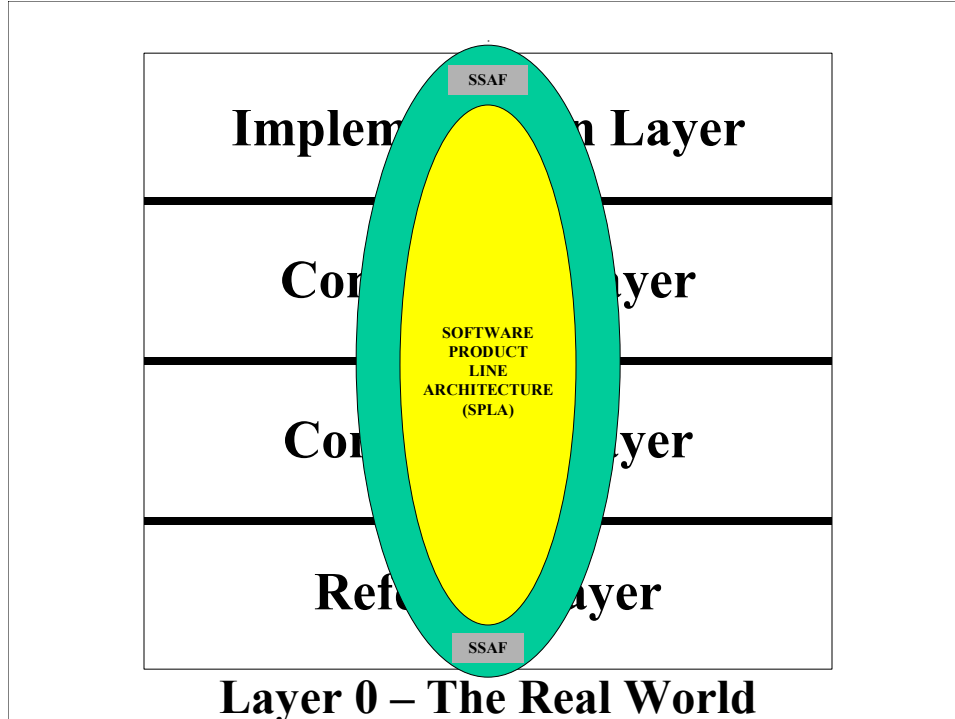


Figure 10-9. Software Product Line Architecture (SPLA)

The domain-level system and software engineering process defines the software product line standard²⁷⁹ and product specification²⁸⁰. These documents provide the allocated level of product line system abstraction²⁸¹ and data abstraction²⁸², including accuracy²⁸³, precision²⁸⁴ and quality factors²⁸⁵ or quality attributes²⁸⁶, allocated as requirements to systems, subsystems, sets, groups, units, components, assemblies, subassembly, and fi-

²⁷⁹ A standard that defines what constitutes completeness and acceptability of items that are used or produced, formally or informally, during the software engineering process [IEE90].

²⁸⁰ The product specification is a critical component of product lines, and is essential to support variability. (1) It specifies the design that production copies of a system or component must implement, or (2) a document that describes the characteristics of a planned or existing product for consideration by potential customers or users [IEE90].

²⁸¹ Abstraction is (1) a view of the object that focuses on the information relevant to a particular purpose and ignores the remainder of the information. (2) The process of formulating a view as in (1) [IEE90].

²⁸² The process of extracting the essential characteristics of data by defining data types and their associated functional characteristics and disregarding representational detail [IEE90].

²⁸³ Accuracy in product lines is either (1) a qualitative assessment of correctness, or freedom from error, or (2) a quantitative measure on the magnitude of error [IEE90].

²⁸⁴ Precision is the degree of exactness or discrimination with a quantity is stated [IEE90].

²⁸⁵ A quality factor is a management-oriented attribute of software that contributes to its quality [IEE98d].

²⁸⁶ A characteristic of software, or a generic term applying to quality factors, quality sub factors, or metric value [IEE98d]. See Chapter VI for a more detailed discussion on quality.

nally the part level composing the product line. Product line quality attributes establish the foundation for product line software quality metrics²⁸⁷ and product metrics²⁸⁸, essential to maintaining control of product line variability.

The software product line architecture also supports a methodology independent classification scheme²⁸⁹, classified scheme items²⁹⁰, and classified components²⁹¹ for several components of data elements including object classes²⁹², properties²⁹³, representations²⁹⁴, value domains²⁹⁵, data element concepts, as well as actual data elements, including keywords²⁹⁶, thesaurus terms²⁹⁷, taxonomy²⁹⁸, and ontology²⁹⁹ taxa³⁰⁰ illustrated in Figure 10-10. Classification attributes³⁰¹ associate various classification schemes with selected components of data elements [ISO79b] including:

- Classified component ID,
- Classified component name.
- Classification scheme type,
- Classification scheme name,
- Classification scheme version,
- Classification scheme item type,
- Classification scheme item value [ISO79b].

²⁸⁷ A function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which the software possesses a given attribute that affects its quality [IEE98d].

²⁸⁸ A metric used to measure the characteristics of any intermediate or final product of the software development process [IEE98d].

²⁸⁹ The arrangement or division of objects into groups based on characteristics that the objects have in common (e.g., origin, composition, structure, application, and function) [ISO79b].

²⁹⁰ Classified scheme items are Components of content in a classification scheme [ISO79b].

²⁹¹ A component of a data element that may be classified in one or more classification schemes [ISO79b].

²⁹² An object class is a set of ideas, abstractions, or things in the real world that can be identified with explicit boundaries and meaning and whose properties and behavior follow the same rules [IEE97a].

²⁹³ A property is a peculiarity common to all members of an object class [ISO79a].

²⁹⁴ A representation describes how the data are represented, and if necessary, a unit of measure or a character set [ISO79a].

²⁹⁵ A value (e.g., data value) domain is a set of permissible values, which may be enumerated or expressed by a description. The value domain provides representation, but does not include the data element concept the values may be associated with nor what the value means [ISO79c].

²⁹⁶ Keywords are basic attributes applied to object classes, properties, representations, and data elements and include the definition, obligation, data type, and comment [ISO79b].

²⁹⁷ Thesaurus terms can be associated with data elements and data element concepts [ISO79b].

²⁹⁸ Taxonomy is a hierarchical organization of concepts (e.g., taxa) based on generalization/specialization and the mathematical notions of sets, subsets, and set membership [ISO79b].

²⁹⁹ Ontology is a network organization of taxa meant to provide a model of some portion of the world, and consists of theories about the sorts of objects, properties of objects, and relations among objects that are possible in that portion of the world [ISO79b].

³⁰⁰ The taxa in taxonomies and ontologies may be related to classified data registration components including object class, property, registration class, and data element concept [ISO79b].

³⁰¹ Classification attribute descriptions include name, definition, obligation, condition, data type, and comments [ISO79b].

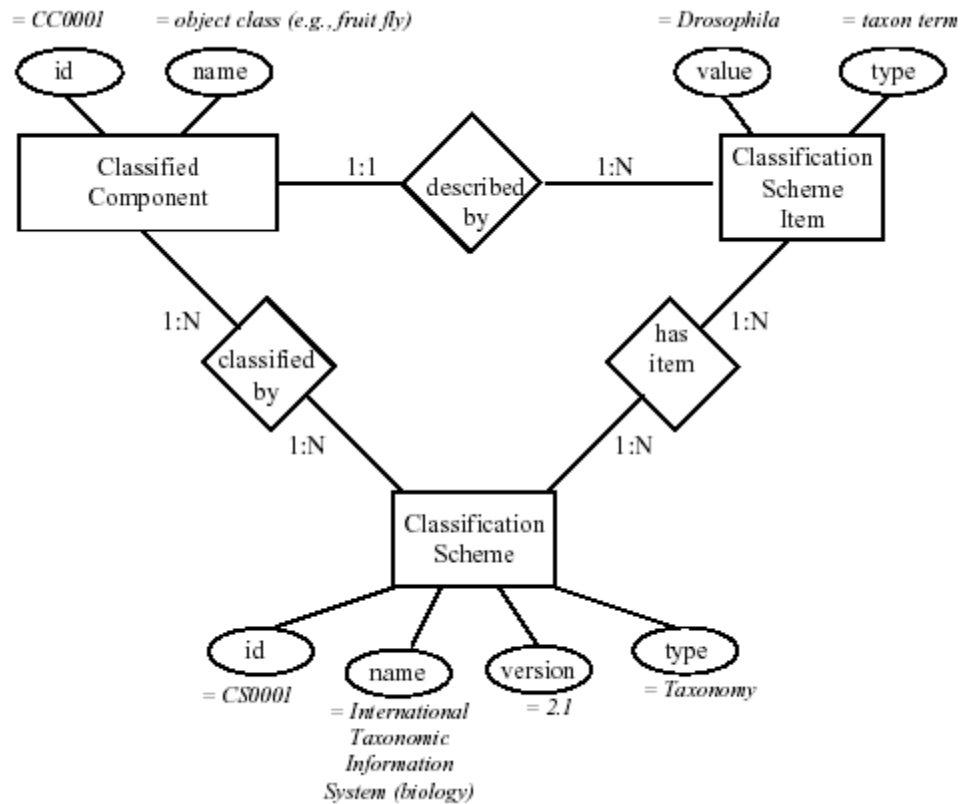


Figure 10-10. Software Architecture Classification (From [ISO79c])

A software product line approach may be employed at one or more levels, depending on many factors such as the application domain, degree of commonality and feature variability with other systems / sub-systems / components, and availability of candidate (reusable) software assets. To illustrate these engineering concepts, consider a missile system in a simulation as an integrated system of subsystems and components performing certain functions for the missile, such as navigation. There are many options, including the establishment of a product line program for a computational system used across a family of missiles, or a product line program established for the navigation subsystems supplied to many missiles.

The best leverage of simulation software architecture-based product lines occurs when they share a common architecture employing quality components for consistent

products. Each level of the M&S software architecture-based product line model would have discrete quantifiable performance and quality attributes.

1. Core Architecture Asset Development

a. Mining for Product Line Core Architecture Assets

Systemic software reuse focuses on large-grained reusable assets such as software architectures, processes, documentation, test cases, and components, versus the previous efforts in software reuse that revolved around small-grained assets of software code, which experienced only modest gains [BFG+00, CN02]. The system architecture, documentation, and components are central core assets used to evolve and build the software product line.

[BSO00, BOS00, CN02] propose existing assets offers an organization the potential to leverage all, or part, of its cumulative system investments, and represents a critical practice area supporting a software product line initiative. However, [BSO00, CN02] cite significant risks achieving success developing software product lines. [BSO00] attributes the high risks to the lack of documentation, the poorly maintained state of many existing systems, and the fact that many systems, initially developed for different purposes, lack support for current software engineering approaches. Research indicates this is generally true in the Department's M&S domain. [BSO00, CN02] identify four basic steps to successfully mine assets: 1) preliminary information gathering, 2) making decisions on whether to mine assets and which type of overall strategy to use, 3) obtaining detailed technical understanding of existing software assets, and 4) rehabilitation of assets.

In mining legacy assets, [BS00, BOS00] suggest the focus should be on mining specific legacy software and evaluating adaptation techniques for product line use, as opposed to either code level transformation or reengineering the system entirely. [BS00] suggests the focus on architectural compatibility and interfaces involves the identification of large-grained legacy system functionality, and mining black-box software elements available for adaptation or wrapping to serve as core assets, or support architectural extraction and reconstruction. Tradeoffs happen continuously in the early mining phases to re-

solve technical challenges, while constantly evaluating legacy software assets for mining potential, and to determine the practicality and cost effectiveness of the mining operation.

b. Reverse Engineering for Product Line Core Architecture Assets

[Til98] introduces a reverse-engineering³⁰² environment framework to improve program understanding³⁰³ based on a descriptive model with categories of support mechanism features established on a foundation of attributes. [Til98] identifies three groupings of support mechanism category: unaided browsing, leveraging corporate knowledge and experience, and computer-aided techniques including reverse engineering. In addition, program understanding according to [Til98] includes the identification, manipulation, and exploration of artifacts of a “particular representation of a subject system via mental pattern recognition by the software engineer and the aggregation of these artifacts to form more abstract system representations” [Til98].

Unaided browsing techniques normally employed in the process include reviewing the source code [HCM02]; however, this method becomes unwieldy as the lines of code exceed normal limits of manual comprehension. Leveraging corporate knowledge and experience is also a valuable method, although maybe of limited value under certain conditions including:

- The lack of available primary source corporate knowledge,
- Third-party system acquisitions,
- Outsourcing.

Computer-aided reverse engineering methodologies³⁰⁴ is a third method of addressing some of the shortcomings of the two previous support mechanisms categories addressed by [Til98]. [JBL97, SLB+98, and BLS+99] contributed a body of knowledge germane to this research initiative with research on the Janus model at the Naval Postgraduate School (NPS). Follow-on NPS reverse-engineering initiatives including research

³⁰² Reverse engineering is the process of understanding, analyzing, and abstracting the system to a form at a higher level of abstraction [Ols95].

³⁰³ Program understanding is a relatively new area of study with an evolving terminology and focus, with an objective to acquire sufficient knowledge about a software system so that it can evolve in a disciplined manner. The essence of program understanding is to identify artifacts and understand the relationships, and is analogous to pattern matching concepts at various abstraction levels [Til98].

³⁰⁴ A reverse engineering environment can manage the complexities of program understanding by helping the software engineer extract high-level information from low-level artifacts, such as source code, reduces the level of tedious, error-prone, manual efforts [Til98].

by [SLB+99, WS99, LBS01] employed all three support mechanism categories identified by [Til98], and concluded that re-engineering using the combination of reverse engineering (e.g., extracting the most useful information) with forward engineering³⁰⁵ techniques addressed by [SLB01] using computer-aided prototyping techniques (CAPS) were cost-effective methods for re-engineering legacy M&S software. Recent NPS research by [You02c] and [Pru03] support both the architectural mining and reengineering activities, and the follow-on simulation software application efforts.

In addition, software evolution initiatives using prototype languages such as the Prototype System Description Language (PSDL) introduced by [Luq89, Luq90] and the Computer Aided Prototyping System (CAPS) development environment [LK88, BL94, Luq92, Luq94, Luq95, Luq98] support the rapid, accurate, and timely development of prototypes. Automated tools [WK98, LBS01, Fav02] provide a capability to invent, correct and refine the conceptual models for new system architectures. This research expands upon the previous NPS work in reverse-engineering and forward-engineering using an entire Department M&S domain model as its focus.

Different reverse-engineering tasks [Til98, YD01, YK02] include program analysis, syntactic pattern matching in the programming language domain; plan recognition, semantic pattern matching [SR02] in the programming language domain; concept assignment semantic pattern matching in the application domain; redocumentation³⁰⁶, aggregation, rejuvenation and reconfiguration of assets, and architecture recovery³⁰⁷. [Til98, BBW+99] also cautions that reverse engineering and the associated terminology, tools [DFP+02], and techniques are still inexact and relatively immature.

Software engineers addressed program understanding in several cognitive models³⁰⁸ [SFM97]. [Til98] explains two common approaches to program understanding, based on the software engineer's level of domain expertise: the functionally based bottom-up deductive approach focused on the cognition of the implementation domain and what

³⁰⁵ Forward engineering is the set of engineering activities that consume the products and artifacts derived from the legacy software and new requirements to produce a new target system [Ols95].

³⁰⁶ Redocumentation is the process of retroactively providing documentation for an existing system [Til98].

³⁰⁷ *Architecture recovery or structural redocumentation* is a term for using reverse engineering to reconstruct the architectural aspects of software [Til98].

³⁰⁸ A cognitive model describes the cognitive processes and knowledge structures used to form a mental presentation of the program being studied [SFM97].

the system does; and the top down, inductive behavioral approach, emphasizing how the system works, based on an existing notion of the system functionality and application domain and employing a goal-driven method of hypothesis postulation and refinement on expected artifacts. Case studies according to [Til98], show that in practice, software engineers switch between these two models employing a third model, the opportunistic approach, depending on the problem at hand. Knowledge management techniques [SG96b, KNR+01, KMS02, WJC02, RJ02, Lie02, RL02, Wel02] provide possible approaches for reusing software engineering-related knowledge and program understanding.

[Til98] also forwards the idea that reverse engineering as the predominant support mechanism used to support program understanding is an activity, which does not change the subject system, since it is an examination process vice an alteration process optimally employed to identify artifacts, discover relationships, and generate abstractions. The process is dependent on several variables including cognitive ability and preferences, domain familiarity and supporting facilities to understand the three categories of artifacts identified by [Til98]: data, knowledge, and information³⁰⁹.

Three canonical reverse-engineering activities support the manipulation of these artifacts: data gathering [Til98], knowledge management, and information exploration, including navigation, analysis cited by [You89]. [BM99 and Moo02] view reverse-engineering methods as a necessary step before starting software engineering activities for an improved system. [TK02] believe a correctly implemented reverse engineering process may directly benefit follow-on reengineering activities.

2. Product Development of the Software Product Line Architecture

a. Reengineering Architecture Assets for Software Product Lines

Software systems have become larger, more complex, more costly and longer lived, which challenges the early software life-cycle models that modeled systems maintained for a short time, until they were retired or replaced. The software engineering challenge cited by [WNS+97] is how to move a large body of legacy code from its current

³⁰⁹ *Data* is the factual information used as the basis for study, reasoning or discussion. *Knowledge* is the sum of what is known, which includes data and information such as relationships and rules progressively derived from the data. *Information* is contextually and selectively communicated knowledge. [Til98]

state to a condition in which it can evolve in a disciplined way. Existing software systems have several inherent problems, which adversely affects maintenance [OS93]:

- Legacy systems are usually complex, unstructured, highly coupled³¹⁰, with low cohesion³¹¹,
- Maintenance can create unpredictable ripple effects,
- Documentation is often missing, incomplete, outdated, or unreliable,
- The system is obsolete with unsupported hardware or software components,
- Experienced software engineers, programmers, and software maintainers are hard to keep,
- Maintenance backlogs continue to grow [OS93].

[OS93] defines reengineering as the bridge to an organization's newly defined processes and environment from the legacy software system. Reengineering is often a better option than redeveloping the system when the following factors are considered:

- Knowledge is imbedded in the software logic,
- Reengineering allows an organization to recoup its investment of time, money and knowledge,
- Legacy software is a valuable organization asset and reengineering extends the life of these systems.
- Reengineered / reused code costs less than redeveloped code [OS93].

Figure 10-11 illustrates the selected software architecture reconstruction approach. Reengineering has the potential to improve an organization's understanding of the software, establish conditions to improve future versions [Ols93, Ols95, BNS97, WBS+97, WNS+97, BSW99, BSW+99, HDK+00, YD01] and better understand past failures [BST+99]. In many existing software systems, the projects were conceived, designed, and developed as unique products, with minimal integration, and very little systematic reuse of assets, which [WBS+97] suggest lose value over time by getting stale and requiring more assets to maintain them.

Replacing these systems and the significant investment they represent with new systems, requiring a new major investment is unlikely, notes [WBS+97] while continuing a fine-grained maintenance strategy, expecting the systems to evolve into maintainable assets, appears overly optimistic. In addition to the traditional software maintenance

³¹⁰ Coupling is the measure of interconnection among modules in a program structure. The lowest possible coupling is desired [Pre97].

³¹¹ A cohesive module ideally performs a single task within a software procedure, requiring little interactions with other parts of the program. High cohesion is desired [Pre97]

activities, an assessment of the legacy system may suggest a replacement strategy, a transformation strategy, either white-box³¹² or black box³¹³, or a combination strategy [WNS+97].

[WNS+97] suggest that time-to-market and widespread distribution technology employing the Web and browsers will drive large-grained use of legacy system components integrated by middleware and object-oriented technologies, as opposed to starting development from scratch. Furthermore, [WBS+97] note that economic realities are making low-level maintenance activities unattractive when compared with the potential gains from large-scale reuse, supported by a focus on architecture and product lines employing middleware and wrapping technologies. However, [BST+99], caution that several organizational, management, and technical pitfalls may befall a reengineering project:

- A flawed or incomplete reengineering strategy,
- Inappropriate use of outside consultants and outside contractors,
- Work force inadequately trained or tied to old technologies,
- Legacy system not under control,
- Requirements elicitation and validation inadequate,
- Software architecture is not a primary reengineering consideration,
- No notion of a separate, discrete reengineering process,
- Inadequate planning or discipline to execute the plan,
- Management lacks long-term commitment,
- Management predetermines technical decisions [BST+99].

In recognition that modern organizations own or use a significant software portfolio, representing a major investment of organization resources, the challenge highlighted by [BSW+99] is to appreciate the value of the organization's software portfolio, and reduce the liability of software asset depreciation over time. The emerging emphasis on software architectures and evolution of software systems provides an impetus to the development of product lines, since systems with interoperable architectures allow the software to operate across defined interfaces, although [BSW+99] notes that legacy systems must be updated to allow them to interact as well-behaved components.

³¹² White-box transformation includes program understanding consisting of activities to recover lost structure or documentation [WNS+97].

³¹³ Black-box transformation includes wrapping or encapsulating the legacy system based on an understanding of the external interfaces, without trying to understand the internal structure [WNS+97].

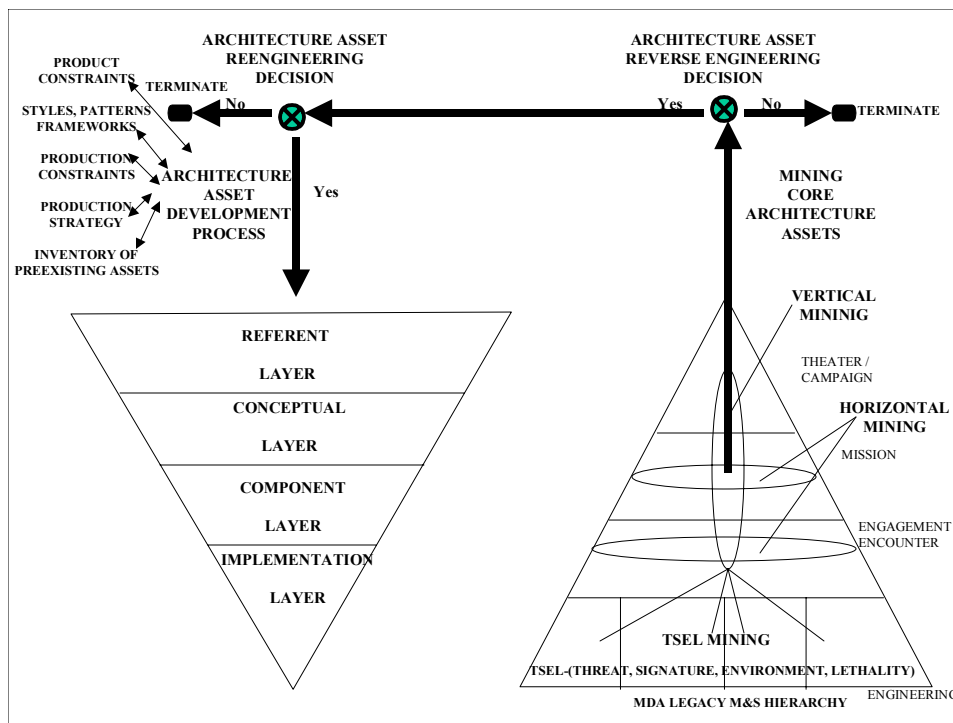


Figure 10-11. Software Architecture Reconstruction

b. Reengineering for Software Product Line Architecture Variability

Recall from Chapter IX that the five BMDS missile domain System-Level M&S possessed two hundred and eighty-five major simulation model representations, with only a sixty percent availability of authoritative model representations. Note, this statistic only accounts for the specific major system model representations detailed in Tables 9-1 through Table 9-14, and does not include other model representations resident in the five simulations, nor does it factor in the different geodetic environments.

It is significant to note that the lack of authoritative model representations is not apparent from a single simulation viewpoint. However, when viewed as part of a domain hierarchy, against the same standard for an authoritative model representation, both the shortfalls and the patterns identified in Table 9-1 through Table 9-14 in Chapter IX become readily apparent. These shortfalls and patterns may contribute to interoperability

anomalies discussed earlier. This condition is the current “as-is” state of the five BMDS missile domain System-Level M&S.

The scope of the future BMDS mission is unprecedented. The illustration in Figure 10-12 notionally represents the future “to-be” requirements for the BMDS missile domain System-Level M&S, with eleven possible dimensions the architectural approach will support. The future requirements for the BMDS missile domain System-Level M&S may add dimensions and significant complexity including possible variant, version, feature, quality, and option dimensions of future systems, in addition to the existing four dimensions of abstraction (e.g., structural, functional, temporal, and qualitative). The future BMDS requirements also add two more dimensions for the BMDS System-Level M&S, the cyclic life cycle support dimension and the quantitative dimension. The cyclic life cycle dimension support for all BMDS Element- and System-Level organizations occurs concurrently during all three life cycle phases (e.g., research and development, transition, and operational support). The quantitative dimension suggests the need for quantitative measurements to augment qualitative identifiers, when possible, for model representation development.

Existing BMDS element systems model representations must closely mirror the existing system and keep current with new capabilities, normally introduced through component software upgrades, suggesting a close relation between the many element system / component software requirement specifications and the component model software build schedules. The current BMDS model representations may be adequate for some future requirements. However, as the number of BMD system level components continue to grow and different manufacturers, or in the future, different countries, provide different capabilities (e.g., radars, boosters, or kill vehicles), the BMDS requires the capability to assess the contribution of that component to the overall BMD system level capability.

Although temporal and life cycle requirements for these future systems are still unknown, the BMDS may require one or more model representations for the six blocks from Block 04 through Block 14 to support temporal dimension requirements. Based on the current population of two hundred and eighty-five model representations, for planning purposes one model representation developed for each future block period would require

the near-term development of approximately one thousand model representations. The blue right-to-left vertical line graphics annotated with element names and cited in the legend, represent this population in Figure 10-12.

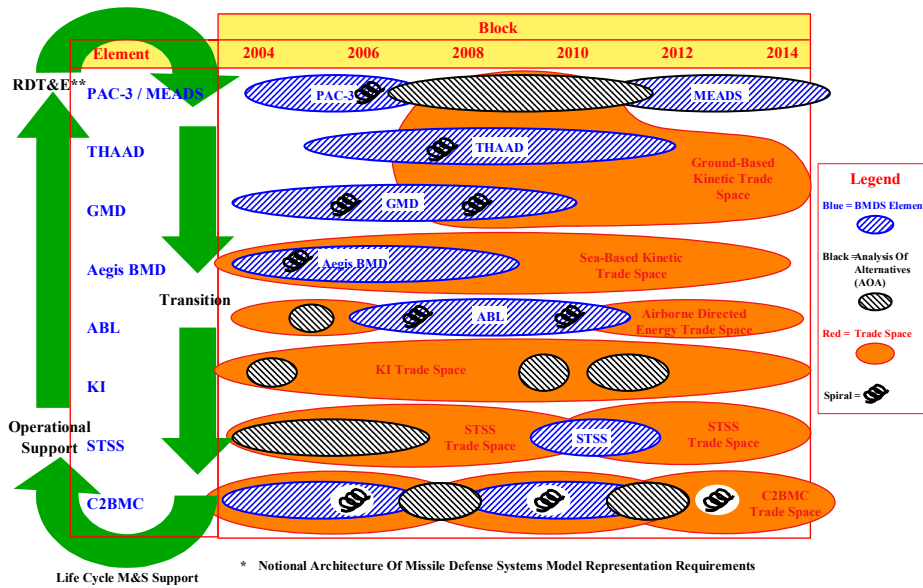


Figure 10-12. Future BMDS Requirements (From [Gre03b])

The BMDS life cycle support dimension is currently undetermined and under study. Recall that the Agency built the current five simulations under study to primarily support only the missile defense research and development phase. The emerging BMDS will also support the transition and operational support phases including new functional requirements for training, logistics, mission planning, system maintenance, and operational planning. Assuming for planning purposes these new functions generate an additional single model representation requirement per simulation per block, an additional one thousand representations may be required. The overall requirement for future BMDS model representations, based on these assumptions indicates a potential need for future BMDS model representations numbering in the thousands.

The BMDS system engineering functions requires flexible, responsive System-Level M&S with the ability to provide here-to-for unknown future variants, versions, features, and option requirements for spiral development within the blocks, analysis of alternatives for new systems, and trade space analysis for future systems or new capabilities. The illustration in Figure 10-12 identifies these notional top-level requirements.

The past performance of the Department's and the Agency's development processes, practices, and techniques cited in Chapters II through Chapter V proved less than satisfactory for previous model representation requirements, and it is questionable whether exhortations for "better, faster, cheaper" development and V&V practices, based on the current practices will be sufficient to meet projected requirements cited above. The research strongly suggest the previous method of individual system-based development and low-levels of reuse, may not support optimal development methodologies for the future BMDS. In this effort we chose the domain management level as the optimal level to execute such a program, since multiple Department enterprise efforts (e.g., JWARS, JSIMS, JMASS) were less than successful, and management efforts below the domain level may lack the resources and perspectives essential to the task

The selected approach involves the introduction of an additional level of abstraction beyond the current software- or component-level into a software architecture framework allowing the visibility of assets and the ability to leverage feature³¹⁴ commonality. The definition of a feature presented by [Bos00] emphasizes the early inclusion of quality attributes discussed in Chapter VI. Features, variants, versions, options, and quality attributes, with associated profiles³¹⁵, will be stored in the Simulation Software Architecture-Based Product Line Model Domain Metadata Repository illustrated in Figure 10-13.

The Simulation Software Architecture-Based Product Line Model Domain Metadata Repository maintains the feature, variant, version, option, and quality attributes at the software product line architecture level, and supports management of the product families (e.g. weapons, sensors), software product lines (e.g. missiles, radars), and software product line systems (e.g. PAC-3, THAAD missiles) at the domain level with a top-down

³¹⁴ [Bos00] defines a feature as a logical unit of behavior that is specified by a set of functional and quality requirements.

³¹⁵ A profile is a set of scenarios that may be used as the basis for specifying a number of quality attributes such as performance or reliability [Bos00].

management approach, and decentralized execution supporting the concept of software architecture-based development.

E. THE SIMULATION SOFTWARE ARCHITECTURE-BASED PRODUCT LINE MODEL DOMAIN METADATA REPOSITORY

1. The Simulation Software Architecture-Based Product Line Model Domain Metadata Repository

The Simulation Software Architecture-Based Product Line Model Domain Metadata Repository in Figure 10-13 provides a public interface and a defined set of services with layer usage dependent on lower layers. The SSA model permits a few exceptions for unique applications. The Simulation Software Architecture-Based Product Line Model Domain Metadata Repository (e.g., Metadata Repository) supports all four layers of the SSA Model.

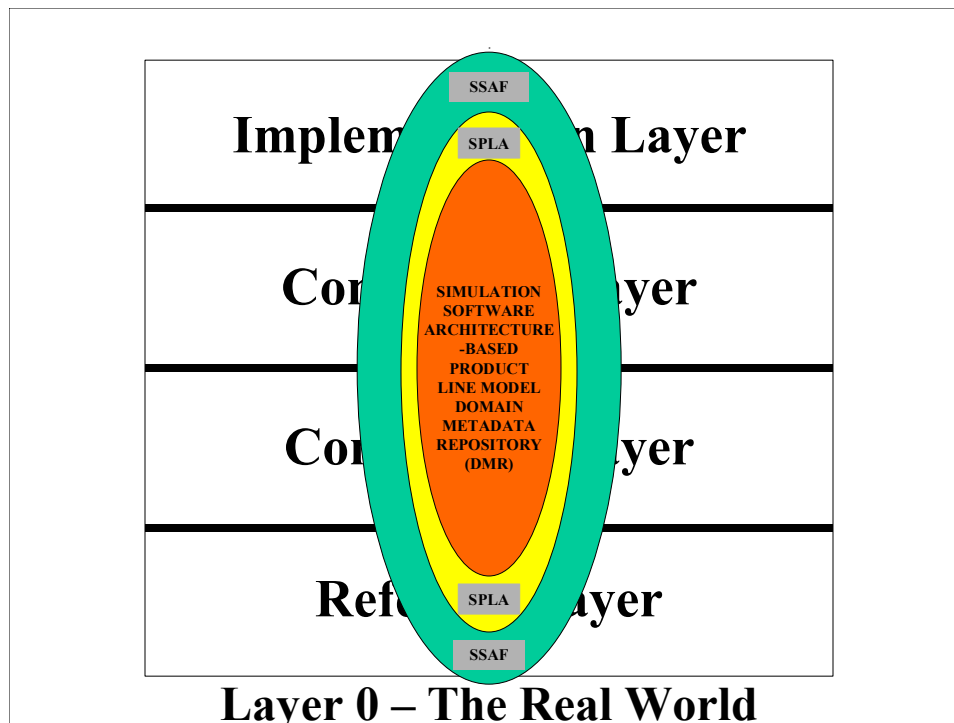


Figure 10-13. SSA-Based Product Line Model Domain Metadata Repository

The primary purpose for the Metadata Repository is to address the root cause of heterogeneous system representation and data anomalies explained in Chapter IV. Major components of the Simulation Software Architecture-Based Product Line Model Domain Metadata Repository tailored from [ISO79c] provides the interfaces shown in Figure 10-14 and includes:

- A *Domain Metadata Registry* holding information describing the structure, format and definition of data [Ste03b]. XML recently emerged as the industry data/metadata interchange format of choice [Dav01] and mandated by the Department's *Joint Technical Architecture* [JTA02a, JTA02b] for domain and application uses of specific markup languages defined by tagged data items [ASC02]. [ASC02] also identifies the requirements for a Department-level XML Registry (DXR) [Ste03a] and Clearinghouse supporting the development of a Federal-level XML Registry (FXR) [Ste03b, XWG02] with components developed with the appropriate XML Namespace [Sal02]. The Domain Metadata Registry will support the following emerging XML standards: XML query (XQuery) data models, algebra, and query language [CFM+03]; the XQuery 1.0 and XPath 2.0 Data Model [FMM+03]; and XQuery 1.0 and XPath 2.0 Formal Semantics [FMM+03],
- A *Domain Metadata Catalog* containing instances of metadata associated with domain data resources, supporting the use of search portals and queries exploring the Domain Metadata Catalog to locate relevant data (e.g., data dictionary),
- A *Domain MetaClass Catalog*, which supports the development of metamodels with a class whose instances are classes,
- A *MetaLanguage Catalog* which specifies some or all of the aspects of a MetaLanguage used in the Domain (e.g., Backus-Naur form, UML, ADL, XML),
- A *Domain Meta-Metamodel Catalog* or model that defines the language for expressing a metamodel,
- A *Domain MetaModel*³¹⁶ *Catalog* defining the language for expressing a mode,
- Domain MetaObject Catalog includes all metaentities in a metamodeling language including metatypes, metaclasses, metaattributes, and metaassociations [DB03],
- *Domain Ontologies Catalog* including data categorization schemes, thesauruses, glossaries, key-word lists, and taxonomies, supporting semantic and syntactic understanding heterogeneous system representation data,
- *Domain Schemas*³¹⁷ *Catalog* representing database tables and relationship structures, XML document type definitions (DTD), and XML schema,
- *Features, versions, variants, options, and quality attributes, with associated profiles,*
- *Architecture Readiness Levels (ARLs)* discussed later in the chapter.

³¹⁶ A metamodel is a model that describes other models [ISO79c].

³¹⁷ Schemas include diagrams, outlines, or models representing data structure, organization, format, structure, relationship, or data.

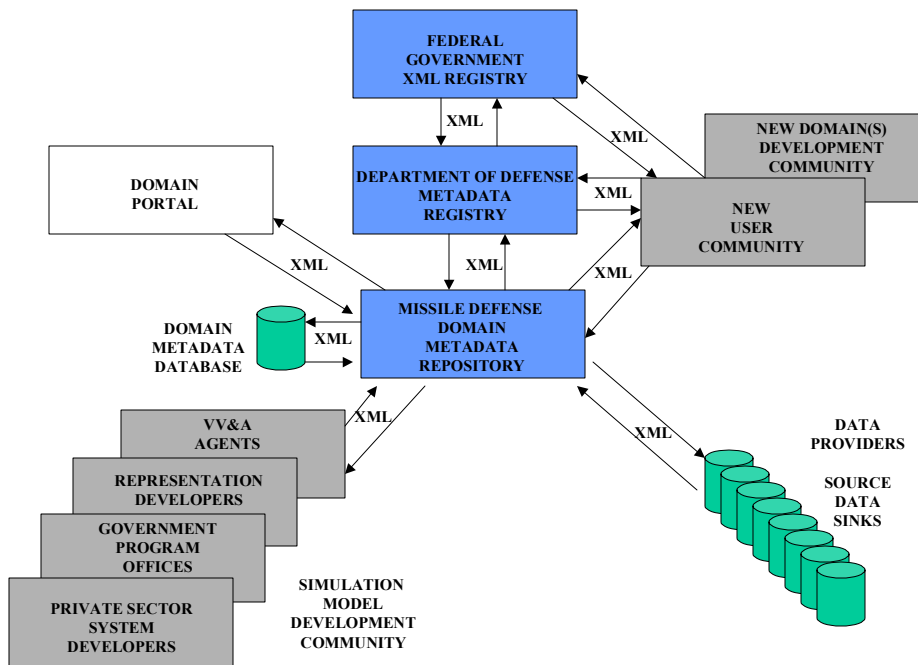


Figure 10-14. BMDS Domain Metadata Repository (DMR)

2. The Simulation Software Architecture-Based Product Line Model Domain Metadata Repository Registry

The Simulation Software Architecture-Based Product Line Model Domain Metadata Repository Registry³¹⁸ (e.g., Domain MRR or DMRR) structure in Figure 10-15 uses a conceptual data model³¹⁹ (e.g., registry metamodel) to maintain information about data elements and associated concepts (e.g., conceptual domains, value domains), including the metadata items³²⁰ described above. The DMRR maintains instances of domain metadata items, which define types of application level data (e.g., in a relational database schema), subsequently populated with real world data. A Unified Modeling Language (UML) subset

³¹⁸ A metadata registry maintains an information system for registering metadata [ISO79c].

³¹⁹ A conceptual data model represents an abstract view of the world. A data model is a graphical and/or lexical representation of data, specifying their properties, structure and inter-relationships [ISO79c].

³²⁰ Metadata items are an instance of a metadata object [ISO79c]

from [ISO79c] in Appendix J. specifies the registry metamodel³²¹ including metamodel constructs³²² for classes, relationships, association classes, attributes³²³, composite attributes³²⁴, and composite data types³²⁵. [ISO79c] uses the term “metamodel construct” for the model construct it *uses*, and the term “metadata objects” for the model constructs it *specifies*.

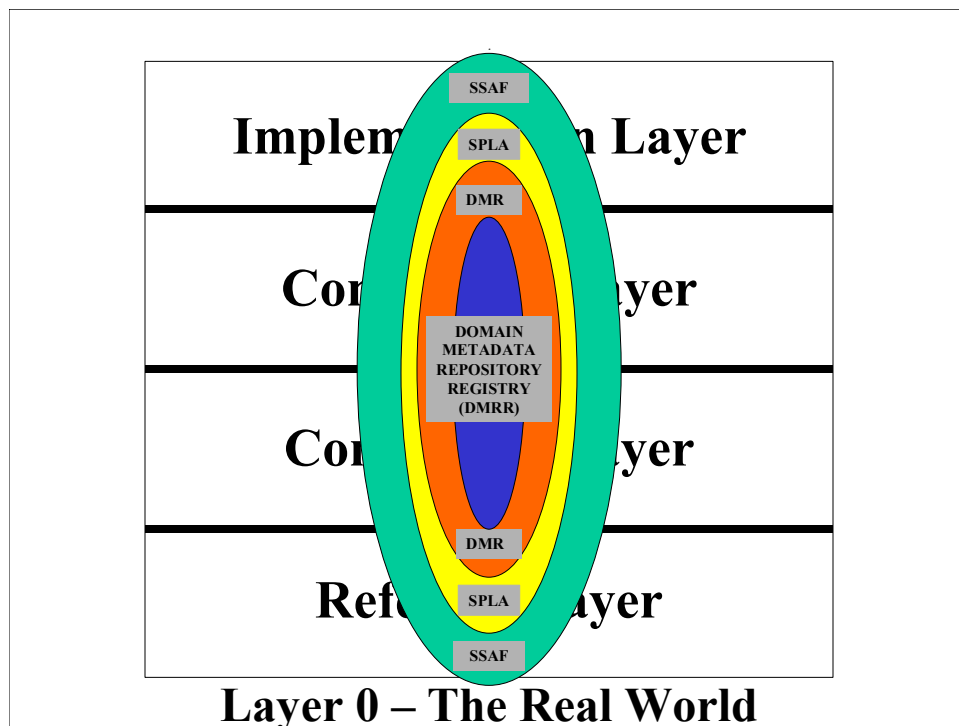


Figure 10-15. BMDs Domain Metadata Repository Registry (DMRR)

The DMRR metamodel specifies types of classes, attributes, and relationships. A specific of classes, attributes, or relationship instance will be a specific type, and at any point in time will contain a specific value. The DMRR metamodel will be extensible, with the ability to add future classes, relationships, and attributes to the conceptual data model.

³²¹ A metamodel specifying a metadata registry [ISO79c].

³²² A unit of measuring for modeling [ISO79c].

³²³ A characteristic of an object or class [ISO79c].

³²⁴ An attribute whose datatype is non-atomic [ISO79c].

³²⁵ A datatype that is also a class [ISO79c].

The DMRR provides a unified view of concepts, terms, value domains, and value meanings³²⁶; promotes a unifying view of the data holdings; and enables reuse and data sharing by providing:

- A means for coordinating data requirements between customers and systems that store, exchange, or manipulate data,
- Assistance to registrars³²⁷ maintaining consistency among different registries,
- A means to store, manipulate, and exchange metadata in support of data attribution, classification, definition, naming, identification, and registration,
- A consistent content supporting interoperability,
- Schema mappings of each tool set,
- Support to translating constructs into the different languages,
- Preservation of the concepts maintained in the original model,
- A conceptual model to base the development of specific logical model (e.g., model of the information system) in an information system (e.g., database design) for the required application [ISO79c].

F. ARCHITECTURE READINESS LEVELS (ARL)

A persistent systemic problem facing decision-makers is the amount of confidence to place on simulation results. Research suggests this issue has many dimensions, as noted in the previous chapters. One of the issues involves the credibility of the model representation or simulation component. Software reuse and component-based strategies consistently faced challenges establishing the credibility of the software component (e.g., predictable composition).

Composable solutions necessitate the need for defined quality properties and a basis for predicting the quality properties. A major challenge of the current software engineering process is applying software quality models and developing reliable software productivity metrics [IEE92] during the early design and analysis phases of the project development. Chapter VIII provided a discussion of software architecture quality analysis methods. Interfaces pose special concerns for future component-based strategies [BBB+00].

Similar problem in the technology, manufacturing, and integration sectors resulted in process models devised as systematic metric and measurement systems, employed to compare the maturity between different types of technology [ACH+02]. Three Readiness

³²⁶ The meaning or semantic content of a value [ISO79c].

³²⁷ A representative of the Registration Authority, which is the organization responsible for maintaining a register [ISO79c].

Level models illustrated in Figure 10-16 support the assessment of technology maturity, production readiness maturity, and integration readiness maturity for a specific technology, product or process technologies: Technology Readiness Levels (TRLs) [Man95], Engineering Manufacturing Readiness Levels (EMRLs) [Fio01], and Integration Readiness Level (IRLs) [MDA02z]. NASA space technology planning employed Technology Readiness Levels (TRLs) for many years [Man95].

TRL	TECHNOLOGY READINESS LEVELS (TRLs) [Man95]	EMRL	ENGR/MFG READINESS LEVELS (EMRLs) [Fio01] SYSTEM / COMPONENT / ITEM (sci)	IRL	INTEGRATION READINESS LEVELS (IRLs) [MDA02Z]
9	ACTUAL SYSTEM 'FLIGHT PROVEN' IN SUCCESSFUL MISSION OPERATIONS		TECHNOLOGIES MUST BE MATURED TO AT LEAST TRL 4 OR 5 BEFORE IT IS READY FOR PRODUCTION	9	TECHNOLOGY PROVEN IN SYSTEM OPERATIONAL TEST / INTEGRATED / PROVEN
8	ACTUAL SYSTEM COMPLETE AND FLIGHT QUALIFIED IN TEST AND DEMONSTRATION			8	ACTUAL SYSTEM COMPLETE & QUALIFIED BY TEST AND DEMONSTRATION
7	SYSTEM PROTOTYPE DEMONSTRATED IN OPERATION ENVIRONMENT			7	INTEGRATION-READY TECH DEMO-COMP/ELEMENT INTEGRATED AND DEMO
6	SYSTEM/SUBSYSTEM MODEL OR PROTOTYPE DEMO IN RELEVANT ENVIRONMENT			6	TECHNOLOGY ADAPTION COMPONENT F3 DEMO IN SYSTEM ARCHITECTURE
5	COMPONENT / BREADBOARD VALIDATED IN RELEVANT ENVIRONMENT	5	IDENTICAL SYSTEM / COMP / ITEM PRODUCED OR S/C/I PROD.	5	SYSTEMS ENGINEERING AND ANALYSIS-FUNCTIONS VAL BY PROTOTYPE
4	COMPONENT / BREADBOARD VALIDATED IN LAB	4	SYSTEM / COMP / ITEM IN PRODUCTION OR LRIP. READY FOR FULL PROD	4	COMPONENT F3 PROVEN FEASIBLE IN LAB
3	ANAL / EXPERIMENTAL CRITICAL FUNC/CHAR PROOF OF CONCEPT	3	SYSTEM / COMP / ITEM IN ADV DEV. READY FOR LOW RATE PRODUCTION	3	SYSTEM APPLICATION AND INITIAL INTERFACE DOCS DEVELOPED
2	TECHNOLOGY CONCEPT OR APPLICATION FORMULATED	2	SYSTEM IN PROTOTYPE DEMO BEYOND BRASS BOARD DEV	2	INTERFACE FORM/FIT/ FUNCTION (F3) DEVELOPED
1	BASIC PRINCIPLES OBSERVED AND REPORTED	1	SYSTEM VALIDATED IN LAB OR EARLY DEVELOPMENT	1	BASIC APPLICATION TO SYSTEM OBSERVED AND REPORTED

Figure 10-16. Current Readiness Level Methods

A major element of the M&S product line architecture approach introduced in this research is the Architecture Readiness Level (ARL) process, the simulation software architecture equivalent to the TRL, EMRL, and IRL processes. The ARL has three major objectives: 1) measure software quality early in the development process as part of the verification process, 2) identify and measure M&S quality attributes as part of the V&V process, and 3) identify and measure the quality of data quality attributes supporting the validation process. The ARL will be employed in all life cycle phases of the simulation product line

development and employment including: 1) mining, 2) reverse engineering, 3) reengineering, 4) development, and 5) VV&A. The ARL supports:

- Model representation quality attributes,
- Simulation quality attributes,
- Data quality attributes
- The Department's VV&A process.

The proposed ARL for improving authoritative system representations incorporates a standard reference for evaluating model representations at any level of abstraction.

ARL	LC	Architecture Readiness Level Summary
9	OP	Verification and Validation completed successfully. Product Operationally Integrated in Final Version.
8	SE- OP	System Implementation Complete. Validated data Available. Implementation Validated
7	SE	Components and Interfaces Composed. Integration Testing Completed. Components and Interfaces meet verified specs.
6	SE	System / Interface Analysis and Design. Design Specs Verified Against Conceptual Model. Validation Data Confirmed.
5	SE- SA	Conceptual Model (CM) Developed. Verification and Validation Plan Developed. Data requirements determined. CM Validated
4	SA	System Views Developed. Implementation Layer established. Product Determined. Metadata employed from Registry.
3	SA	Component Views Developed. Component Layer established. Product Line Determined. Metadata registered.
2	SA	Conceptual Views Developed. Conceptual Layer established. Product Family Determined. Metadata analyzed.
1	SA- RW	Viewpoints / Views Developed from Real World / Core Assets. Referent Layer established. Metadata needs identified.

Table 10-1. Architecture Readiness Level Summary

Recall our earlier discussion about the different viewpoints and views of a house envisioned by the plumber, electrician, and other craftsmen contributing to the construction. Although they may have a primary viewpoints and views of the house based on their craft, they share several constraints such as the architectural drawings; local-, state-, and national-level building codes; the cost and availability of certain materials; and the availability, experience level, and wage requirements of helpers.

To ensure compliance with applicable building codes, the building contractor takes out a permit and schedules several visits by the appropriate building inspectors during the construction. During the interior “roughing in” inspection the building inspector will review and test the work of the electrician, plumber, and HVAC specialist before the drywall is installed to ensure building code compliance by the different contractors. In many cases the building codes evolved after repetitive tragic events caused injuries, deaths, or the loss of property. The proposed ARL for improving authoritative system representations imposes similar constraints and incorporates a standard reference (e.g., building code) for evaluating model representations.

In Table 10-1 we present the Architecture Readiness Level Summary with nine ARL levels numbered one (e.g., immature) through nine (e.g., mature), very similar to the base TRL Model. In addition, the life cycle view of the representation matures from the real-world view (RW), through the four layers of the simulation software architecture (SA) process, into the simulation software engineering (SE) process, and finally into the operation (OP) and maintenance world. Three ARL transition points support the ARL model maturity process at ARL layer 1 (RW to SA), ARL layer 5 (SA to SE), and at ARL layer 8 (SE to OP). Note that in the very similar EMRL Model, technologies need to meet a minimum of EMRL Four or EMRL Five before it is ready for production. The ARL Model takes a very similar tack for the development of authoritative representations.

The next section provides a synopsis of each ARL, including the activities or capabilities associated with the specific ARL:

ARL 1 - The lowest level of the architecture readiness level and transition point from the real world to the Simulation Software Architecture (SSA) referent layer. SSA Referent layer viewpoints from the real world lack definition and understanding. Views

and requirements are immature. Attribute³²⁸ analysis initiated. Core asset analysis is incomplete. Architectural components establishing the view require significant descriptive and classification information. Metadata analysis is pending. Architecture is immature. Quality requirements³²⁹ and measurement³³⁰ process initiated. Validation data requirements are unknown. The SSA is very immature and very high risk.

ARL 2 - The second level of the architecture readiness level, the SSA Conceptual Layer is established. Conceptual viewpoints from the real world have limited definition and understanding. Views and requirements more mature. Architectural components establishing the view require additional descriptive and classification information. Metadata analysis is largely incomplete. Product family analysis initiated. Validation data analysis initiated. Metrics framework³³¹ and software quality metrics³³² process initiated. Core asset analysis is complete. Initial query sent to Domain or Enterprise Metadata Registry for possible item availability and reuse. The SSA remains immature and high risk.

ARL 3 - The third level of the architecture readiness level, the SSA Component Layer is established. Component viewpoints from the Domain or Enterprise Metadata Registry or the SSA Conceptual Layer have more definition and understanding. Metaclass³³³ established. Views and requirements validated. Architectural components establishing the view have adequate descriptive and classification information. Metadata analysis is largely complete and metadata registered. Product line analysis initiated. Product metrics³³⁴ established. Validation data analysis is complete. Metrics framework and software quality metrics process in progress. Core asset utilization options established. Results received from Domain or Enterprise Metadata Repository search for possible item availability and reuse. The SSA is maturing and mitigation actions developed to address the risk(s).

³²⁸ A measurable physical or abstract property of an entity [IEE98d].

³²⁹ A requirement that a software attribute be present in software to satisfy a contract, standard, specification, or other formally imposed document [IEE98d].

³³⁰ The act or process of assigning a number or category to an entity to describe an attribute of that entity. A figure, extent, or amount obtained by measuring [IEE98d].

³³¹ A decision aid for organizing, selecting, communicating, and evaluating the required quality attributes for a software system. A hierarchical breakdown of quality factors, quality subfactors, and metrics for a software system.

³³² A function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which software possess a given attribute that affects its quality [IEE98d].

³³³ A class whose instances are classes. Metaclasses are normally used to build metamodels [UML99].

³³⁴ A metric used to measure the characteristics of any intermediate or final product of the software development process [IEE98d].

ARL 4 - The fourth level of the architecture readiness level, the SSA Implementation Layer is established. System viewpoints from the Domain or Enterprise Metadata Registry or the SSA Component Layer defined and understood. Views and requirements are mature. Architectural components establishing the view have validated descriptive and classification information. Metadata analysis is largely complete and metadata employed from the Metadata Registry. Product line analysis is underway. Validation data availability determined. Metrics framework and software quality metrics process determined. Core assets determined. Final results received from Domain or Enterprise Metadata Repository. The SSA is mature enough to start software engineering planning efforts.

ARL 5 - The fifth level of the architecture readiness level is the transition point from the SSA to conceptual model development in the software engineering process. Architectural assets are available from the Domain or Enterprise Metadata Registry. Product line conceptual model development initiated. Validation data availability confirmed. Metrics framework and software quality metrics transitioned to the conceptual framework. Conceptual Model Verification and Validation Plan developed. Conceptual model developed. Conceptual model validated. The architecture should be matured to at least ARL 5 before it is ready for software engineering.

ARL 6 - The sixth level of the architecture readiness level is the system analysis and design phase of the software engineering process. Interface control documents are complete. Architectural assets identified from the Domain or Enterprise Metadata Registry for component or system development. Product line conceptual model validated. Validated data for validation phase is available. Metrics framework and software quality metrics transitioned to the development phase, and metrics validated³³⁵. Implementation Verification and Validation Plan developed. Analysis and design activities completed and verified.

ARL 7 - The seventh level of the architecture readiness level is the component and interface development phase of the software engineering process. ARL 7 is optional if component-based development methods not employed. Interface control documents are complete. Architectural assets are available from the Domain or Enterprise Metadata Reg-

³³⁵ The act or process of ensuring that a metric reliably predicts or assess a quality factor [IEE98d].

istry for component development. Product line component model validated. Validated data used for component validation phase. Component-level predictive metrics³³⁶ established. Metrics framework and software quality metrics transitioned to the component-level development phase. Implementation Verification and Validation Plan developed. Analysis and design activities completed and verified. Components developed and tested.

ARL 8 - The eighth level of the architecture readiness level is the system implementation phase of the software engineering process. System interface control documents are complete. Architectural assets are available from the Domain or Enterprise Metadata Registry for system development. Product line system model validated. Validated data used for system validation phase. System-level predictive metrics established. Metrics framework and software quality metrics transitioned to the system-level development phase. Implementation Verification and Validation Plan executed. System-level development, test, and integration activities completed and verified. System transitioned from the software engineering phase to the operation and support phase.

ARL 9 - The ninth level of the architecture readiness level is the system operations and support phase of the software life cycle. Product line system model variant maintained. Validated data is archived. System-level predictive metrics exercised. Metrics framework and software quality metrics monitored and analyzed. System-level development, test, and integration documentation maintained as core assets. System transitions from the operation and support phase into the retirement phase.

G. DOMAIN INTEGRATED PRODUCT DEVELOPMENT TEAMS (DIPDT)

Implementing software product lines in the Department may be even more challenging since the Department acquires software system as opposed to developing the software. If credible authoritative model representations populating credible simulations supporting decision-makers confidence in the results is the strategic *end* we wish to achieve; and the Software Architecture-Based Product Line Model is the *way*, we suggest that the Domain Integrated Product Development Team (DIPDT), employing mature Product Line Practices described in Chapter VIII, are the *means* to accomplish this strategic objective.

³³⁶ A metric applied during the development and used to predict the values of a software quality [IEE98d].

Recall in earlier chapters we discussed contract complexity. In Figure 10-17, a notional Missile Defense Agency organizational chart suggests the challenges posed by several layers of government agencies supporting the Missile Defense Agency. Many of these agencies in turn have contractual agreements with one or more contractors to provide services or deliverable products. The contractors in turn may have contractual agreements with one or more sub-contractors. The contracts may differ in many ways such as scope, period of performance, deliverables, contract incentives, and other contractual clauses. The communication channels also vary government-to-government, government-to contractor, contractor-to-contractor, organization-to-organization, and even within the respective organizations.

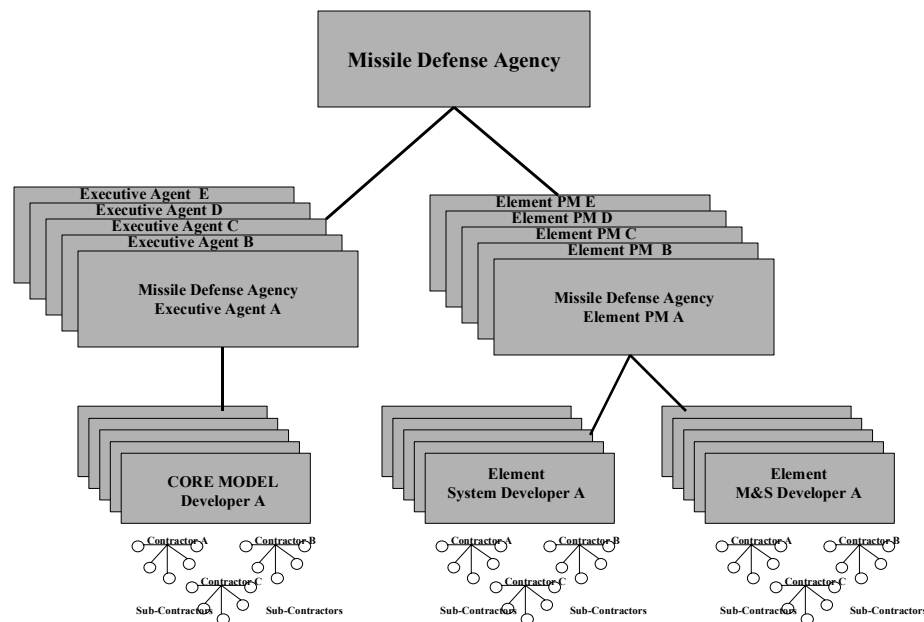


Figure 10-17. MDA Contract Complexity (Notional)

Figure 2-1 and Table 7-1 identified a consistent pattern of Department institutional constraints affecting M&S credibility. The reoccurring, non-technical constraints such as process, culture, organization, and management, in addition to the more technical con-

straints of software development, architecture, and engineering, suggest several pre-conditions require resolution before the Department may realistically plan for credible simulation software and improved confidence in the results. One possible means to improve the pre-conditions identified in the research involves the development of a Domain Integrated Product Development Team (DIPDT).

The Integrated Process Team (IPT) concept is not new to the Department, and many principles of the IPT process support the proposed DIPDT concept. The DIPDT concept illustrated in Table 10-2 also integrates tailored versions of the software product line methodology and practice line areas, cited in earlier chapters. The DIPDT concept evolved with the research as it became readily apparent that no “silver bullet” approach would solve the Department’s systemic problems with achieving credible authoritative representations. Instead, a Domain-Managed, de-centrally executed product team approach evolved from the research of the Agency System-Level Core M&S.

	DIPDT Members	M&S	M&S	M&S	M&S	M&S
	Government/COR/Contractors	A	B	C	D	E
1	Agency M&S Lead	1A	1B	1C	1D	1E
2	Agency System Engineering	2A	2B	2C	2D	2E
3	Agency Test & Evaluation	3A	3B	3C	3D	3E
4	Agency Executing Agent / COR	4A	4B	4C	4D	4E
5	Model Developer PM	5A	5B	5C	5D	5E
6	Model V&V PM	6A	6B	6C	6D	6E
7	Model Accreditation Agent	7A	7B	7C	7D	7E
8	Model Data SME	8A	8B	8C	8D	8E
9	Element A M&S Lead	9A	9B	9C	9D	9E
10	Element A System 1	10A	10B	10C	10D	10E
11	Element A Sys 1 Sub-Sys 1	11A	11B	11C	11D	11E
12	Element A Sys 1 Sub-Sys 2	12 A	12 B	12C	12D	12E
13	Element A Sys 1 Sub-Sys 3	13A	13B	13C	13D	13E

14	Element A Sys 2 Sub-Sys 1	14A	14B	14C	14D	14E
15	Element A Sys 3 Sub-Sys 1	15A	15B	15C	15D	15E
16	Element A System 2	16A	16B	16C	16D	16E
17	Element A System 3	17A	17B	17C	17D	17E
18	Element A System 4	18A	18B	18C	18D	18E
19	Element A System 5	19A	19B	19C	19D	19E
20	Element A System 5	20A	20B	20C	20D	20E
21	Element B M&S Lead	21A	21B	21C	21D	21E
22	Element C M&S Lead	22A	22B	22C	22D	22E
23	Element D M&S Lead	23A	23B	23C	23D	23E
24	Element E M&S Lead	24A	24B	24C	24D	24E

Table 10-2. Domain Integrated Product Development Team (DIPDT)

Table 10-2 illustrates a subset of a notional Agency DIPDT model, showing a virtual organization formed around the software product line. In this case the software product lines are the five core simulations cited in Chapter IX (e.g., CAPS, MDWAR, EADSIM, EADTB, MDSE) shown as Simulation A through Simulation E on the vertical axis or partition. On the horizontal axis of the array a partial list of key stakeholders supporting the software product line, both government and contractor for the actual systems / sub-systems themselves and the model developers. The system engineer, developmental tester [Eic94], and operational tester play significant roles. Key DIPDT members also include the V&V agents and the accreditation agents.

Note line 9 through line 15 of Table 10-2. In this view of the notional VIDPT, Element A is further decomposed into the system and sub-system levels with representatives on each simulation DIPDT in which their system or sub-system is represented. Contracting Officer Representatives (COR) or Contracting Officer Technical Representatives (COTR) play a key role on the DIPDT as matrix support for contract management support. The objective of the DIPDT is to develop a domain integrated product team approach based on the primary source level of the authoritative model representation information through the various levels of reoccurring, non-technical constraints to the simulation product line

level. The simulation product line in Table 10-2 would replace the traditional tree structure in Figure 10-17. We envision that as the DIPDT and the SSA concept mature, and meta-data repositories appear, a wider variety of credible and architecturally sound model representations will be available to support the future development of composable simulations.

H. SUMMARY

Chapter X introduced the detailed research design for the Simulation Software Architecture-Based Product Line Model including five major elements supporting the method, specification, design, and implementation. The chapter provided an architectural analysis of the Department's existing *de facto* M&S software architecture and introduced the Simulation Software Architecture (SSA), the Simulation Software Architectural Framework (SSAF), the Simulation Product Lines Architecture (SPLA), the Simulation Software Architecture-Based Product Line Model Domain Metadata Repository, and Architecture Readiness Levels (ARL).

The Simulation Software Architecture (SSA)-based Model is an abstract software architecture-based horizontal foundation supporting multiple viewpoints and views. The Simulation Software Architectural Framework (SSAF) is a second vertical-slice³³⁷ architectural component overlaying the SSA, which includes system and environment components. The Simulation Product Lines Architecture (SPLA) provides the framework to manage the variability³³⁸ [BB00, Bos00, DMH01, RSC00, MNJ+02, CBB+03], features [Bos00, KLD02], and differences between products and comprises the third element.

The Simulation Software Architecture-Based Product Line Model Domain Metadata Repository provides the structure for the Domain Metadata Registry modeled from [ISO79c] to ensure interoperability with Department, Federal Government, and private sector metadata registries. The SSA and SSAF establish the architectural foundation for the SPLA, which supports variability and extensibility of the architectural construct. We also

³³⁷ In UML this is called a partition or a set of related classifiers or packages at the same level of abstraction or across layers in a layered architecture [UML99].

³³⁸ Variability refers to decisions that will be made by a member of the development team prior to system deployment [CBB+03].

adapted the architectural description conceptual framework³³⁹ from [IEE00B] into the SSA, SSAF, and SPLA models composing the Simulation Software Architecture-Based Product Line Model.

Core asset development is an evolving practice. Mining existing assets, a core asset development option, offers an organization the potential to leverage all, or part, of its cumulative system investments, and represents a critical practice area supporting a software product line initiative. However, there are significant risks achieving success developing software product lines. Some reasons include the high risks due to the lack of documentation, the poorly maintained state of many existing systems, and the fact that many systems, initially developed for different purposes, lack support for current software engineering approaches. Research indicates this is generally true in the Department's M&S domain.

The chapter provides an overview of a reverse-engineering and reengineering environment framework to improve program understanding to support software architecture reconstruction architecture for model representations. Reengineering has the potential to improve an organization's understanding of the software and improve it and must improve on past failures. In many existing software systems, the developers conceived, designed, and developed projects as unique products, with minimal integration, and very little systematic reuse of assets, which lose value over time by getting stale and requiring more assets to maintain them.

Software architecture reconstruction provides a mechanism for the effective reuse of software assets. Architecture reuse is the cornerstone practice of product line development, supported by three sources for architecture recovery:

- Source code, which is authoritative, but does not hold all the information,
- Documentation, which is normally undependable, incomplete, or outdated,
- Human experts, who may be dependable, but biased [CW98].

Architecture recovery and reconstruction is an iterative and interactive labor-intensive effort, dependent upon the level of specification, documentation, dissemination and control of the legacy architecture artifacts. Architecture recovery methods vary from

³³⁹ The conceptual framework is the term of reference for the architectural description and establishes the terms and concepts for the content and use of architectural descriptions [IEE00b].

entirely manual reconstruction processes to tool-supported manual reconstruction and semi-autonomous reconstruction techniques, and may include data mining and employment of architecture description languages. Pattern analysis supports architecture recovery and reconstruction efforts.

The Architecture Readiness Levels (ARL) supports future architectural components and products developed from this methodology, is the fifth and final contribution. The chapter also suggests a complementary Domain Integrated Product Development Team (DIPDT) approach for implementing the Simulation Software Architecture-Based Product Line Model, to reduce the cycle time for resolving the multiple dimensions causing heterogeneous system representation anomalies, and improving federation interoperability.

XI. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

A. REVIEW OF MAJOR RESEARCH AREAS RELEVANT TO THIS WORK

This research focuses on the major research areas introduced in Figure 2-1, and replicated in Figure 11-1 for improving the current heterogeneous system representation anomalies-based issues, which erodes credibility in the Department simulation process, and injects doubt instead of confidence in the simulation results supporting Departmental- and National-level decision-makers.

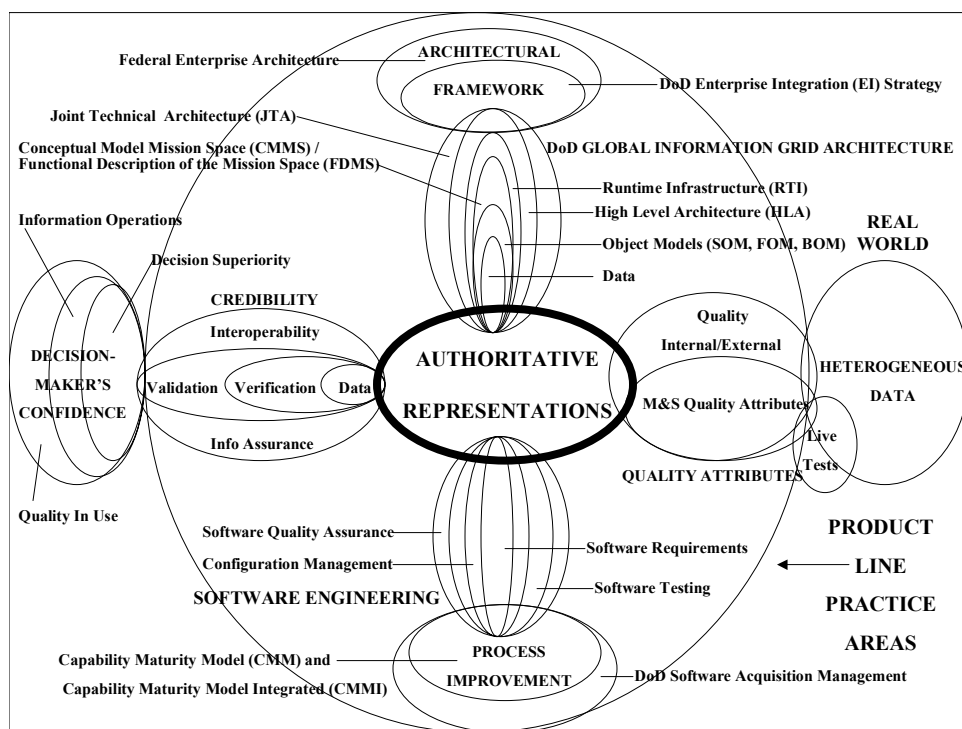


Figure 11-1. Major Areas Researched for this Work

In Chapter II we identified the seven major areas affecting improved credibility in simulation model representations supporting confidence in the Department- and National Level decision-making process:

- An Architectural Framework,
- Software Engineering,
- Process Improvement,

- Quality,
- Verification and Validation,
- Heterogeneous Data,
- Technical and Substantive Interoperability.

Five of the seven areas affecting simulation credibility were methods to improve the simulation product, and two of the seven areas (e.g., heterogeneous data, technical and substantive interoperability) were conditions adversely affecting model credibility. While all of the five simulation product improvement methods experienced some progress, they also encountered several limitations:

- The Department's M&S architectural framework remains immature. The Common Technical Framework including the High-Level Architecture, Conceptual Model of the Mission Space, and improved data quality initiatives did not significantly improve simulation model representation credibility. The High-Level Architecture achieved technical success, and the IEEE adopted it. However, it is too early to tell whether the HLA will be a commercial success, now that the Department no longer financially underwrites it. The Conceptual Model of the Mission Space model implementation proved inconsistent, reducing credibility in the V&V process. Data remains a major problem [DoD01a]. The current object models (e.g., SOM, FOM, BOM) do not address the substantive interoperability issue. In addition the JTA and the HLA remain incompatible.
- The Department's software engineering strategy to replace many large-scale legacy simulations with new, well-engineered, joint simulations (e.g., JMASS, JSIMS, JWAR) experienced significant cost overruns, failed to meet continually extended delivery schedules, and met only a subset of the approved requirements. With major funding reductions or planned cancellation, the future of the three major simulation software engineering initiatives appear limited at best. As a result the Department is reviewing options to extend the life span of the existing large-scale, legacy simulation systems.
- Although process improvement in no way guarantees fewer errors in the final software product, initial indications over the past decade suggest that properly disciplined process improvement has a favorable impact on software development quality and may improve simulation model credibility. Several process methodologies exist including the CMM, CMMI, Software Acquisition Management, and the Product Line Practice Areas. Our research indicates that mature processes are pre-conditions for the development of credible simulation model representations.
- Quality remains an elusive element in the Department's M&S portfolio. In many cases the discussion of M&S quality eventually lead to debates about *fidelity*. Fidelity as a term has many appropriate uses, but some suggest that *fidelity* is an overused term, open to almost any meaning. Our research indicates that quality attributes are also pre-conditions for simulation credibility, and are key architectural components of the proposed Simulation Software Architecture Model.

- The Department's V&V process is the primary method for establishing simulation credibility. However, systemic issues with incomplete or non-existent conceptual models, a general lack of validation data, and inconsistent V&V implementation raises many questions of a simulation's credibility. The accreditation process, dependent upon a sound V&V process, lacks credibility as a result.

The research results presented earlier in Chapters II and IX indicates that after fifty years of development the Department M&S domain has not achieved the desired level of credibility and confidence needed to support the Department decision-making process, and existing methods, techniques, and procedures have not achieved the desired objectives.

A major finding of this research suggest that there is little supporting evidence that the Department M&S software development community will achieve future success: 1) improving heterogeneous system representation anomalies, 2) improving overall simulation credibility; and 3) improving confidence in results provided by Department M&S employing current conceptual model development paradigms. In addition, many of the past methods and recommendations to improve the Department's M&S portfolio have been reiterated and reissued, and it is uncertain if more time and more resources will result in:

- Better verification and validation (V&V) techniques,
- Better discipline in the software development process,
- Improved accreditation processes,
- Improved use of data,
- Improved quality,
- Improved documentation,
- Improved coordination between the users and the developers,
- Improved credibility,
- Improved confidence in the results.

B. EVALUATION OF A SIMULATION SOFTWARE ARCHITECTURE METHOD AGAINST THE CURRENT M&S TECHNIQUES

It is clear from the research that Department and the Service Components acted with due diligence to improve the quality of Department M&S. By the mid-1990s the Department established M&S management organizations, developed policies, implemented plans, allocated significant funding, and executed major new programs with the goals and objectives of improving Department M&S practices. However, notwithstanding these

considerable efforts, major concerns still exist about the credibility of the Department simulation process. There are several major reasons for this lack of credibility in Department M&S.

The first contributor was the software industry-wide hope for a “silver bullet” solution to resolve the Department “software crisis”. However, as hope for a “silver bullet” technical breakthrough to improve Department software, including simulation software, waned in the 1990s, it became apparent that other non-technical solutions were required. In addition, new Department M&S management organizations, such as the DMSO, had to overcome Service parochialism, the unpredictability of the Department budget process, the vagaries of the Department acquisition process, and over forty years of unplanned Department M&S growth. Then in the late 1990s significant funding and manpower resources originally programmed to improve Department M&S were committed to the Year 2000 problem resolution project.

A second underlying cause for the current situation is the Department institutional constraints affecting M&S credibility, and the apparent slow maturity progress of major M&S credibility factors cited in Table 7-1. In 1996 a major Department M&S study [PHP+96] identified three major types of systemic Department M&S challenges, technical, cultural, and managerial; while concurrently the Simulation-Based Acquisition (SBA) initiative identified three similar factors, process, culture, and environment as SBA enablers. More recently the SEI endorsed three related practice areas, technical management, and organizational, to support the successful implementation of software product lines.

This is significant since these factors represent an institutional realization that the resolution to the twenty year old “software crisis” is not a “silver bullet” solution. Instead, the problem and the solution are multi-dimensioned, complex issues that require changes in many areas. This change will take time for the Department to truly assimilate and adapt to during the transition phase into the Information Age. We must however, also be mindful of the many Industrial Age vignettes where nations, armies, navies, or air forces were slow to adapt and the severe consequences of that failure in terms of defeat or loss of life.

Software quality, a third contributing study factor, continues to lag well behind the computer industry’s hardware engineering segment progress on quality, cost, and perform-

ance. While computer hardware engineering is well into the fourth computer era identified by [Pre97] and employs the *production* approach to quality according to criteria established by [Wal94], simulation software quality is best labeled as *transcendental*. In contrast to the mature computer hardware-engineering sector, the software engineering sector of the computer industry is just now achieving a consensus on a software engineering body of knowledge and agreement on the benefits of software process improvement. The implementation of Department M&S quality objectives for improving quality attributes, including fidelity, accuracy, resolution, error, detail, aggregation/disaggregation, and multiresolution modeling techniques have achieved only limited success.

Software architecture is the fourth study factor acknowledged to strongly influence a system over its life cycle. While computer hardware-related architectures supported by Moore's Law were ascendant over the past fifty years, software-related architectural considerations, if they existed, were of secondary importance to the overall system engineering and development process. Today, the cost and complexity of software-intensive systems have changed the industry's dynamics and the publication of software architectural descriptions [IEE00b] and styles [SC96, BK99, CBB+03] for software-intensive projects are introducing software architectural principles to the emerging software engineering community. However, the Department's life cycle management of software-intensive simulation systems lacks software architecture concepts beyond the application of the CTF (e.g., HLA, Conceptual model, data).

The Department legacy M&S portfolio is the fifth major study factor. The Department developed a significant and expensive portfolio of legacy M&S over the past fifty years. Although an operational system has a *de facto* architecture, many of these M&S evolved in an *ad hoc* manner and have become large legacy systems with multiple stakeholders and expensive support infrastructures. A significant number of these systems lack conceptual models and have an *ad hoc* V&V history. In some cases the developer created conceptual models after the fact, if at all, and adherence to disciplined M&S V&V process was a low priority.

Department agencies also have a tendency to use these systems for studies or tests for which they are ill suited, in order to show a return on investment. As a result the ac-

ceptability criteria and caveats developed by the accreditation process for these simulations may be of limited value or counter-productive to the original purpose and intent of the study or test objectives. Complicating the current situation, replacement systems are overdue. The new Joint M&S programs (e.g., JWARS, JSIMS, JMASS) scheduled to replace many legacy systems are all late, over cost, and reducing the original requirements to meet a future operational status.

M&S verification and validation, a sixth study factor, is the foundation for M&S credibility and confidence. However, many consider V & V expensive and time consuming. In this research we reviewed the V&V process in the context of supporting Department-and National-level decisions with improved credibility and confidence, and confirmed previous assessments indicating the quality of Department V&V practices are *ad hoc* and inconsistently applied.

The conceptual model of the mission space (e.g., CMMS or FDMS), a seventh study factor, is the approved foundation for a rigorous Department verification and validation program. The Department bases successful V & V on a verified and validated conceptual model. In practice, conceptual model development and conceptual model validation as a precursor to verifying the software implementation, rarely occurs. The M&S community developed many conceptual model formats [RPG00], however, there is no consensus on a standard conceptual model at this time.

The current Department M&S Hierarchy, the eighth study factor, provides only general, qualified levels for M&S fidelity and resolution, provides only limited support for the user to ascertain a simulation's attributes. The model provides little if any support for quantifiable measures of simulation quality. In addition, metrics based on this model tend to be qualified, subjective, and lack authority. In this research we analyzed the current Department M&S Hierarchical model as a core asset and reviewed alternatives for mining reverse engineering and reengineering these core assets into product line architecture.

Common limitations of the Department's efforts to improve M&S credibility and confidence in results include the following reasons. First, a significant amount of current Department M&S systems were developed in previous decades and are currently in a main-

tenance mode, suggest a strategy of introducing improvements in a planned evolutionary manner.

Second, the M&S policies, procedures, best practices, and funded projects developed in the mid-1990s may need more time to promulgate and affect M&S development processes for new models and simulations.

Third, in the 1990s, the Department experienced a major shift from M&S supporting a bi-polar world with two super powers to a world characterized by asymmetrical warfare and weapons of mass destruction, and support Department transformation initiatives, at a pace of change seldom experienced in the Department.

Fourth, as the Department evolves from a threat-based, requirements-driven organization and transforms itself to a capability-based model while concurrently maintaining the ability to execute the National Military Strategy, the question remains, are we developing the right M&S (e.g., are we building the right thing)?

Last, if we determine that we are building the right thing, have the software engineering capabilities of the Department matured enough to “build it right?” The product line domain architecture supporting specified requirements allocated from the system engineering process presented in Chapter X of this dissertation address these limitations.

Accomplishing this significant undertaking requires a mature population of software engineers and software architects. Grounded in the technical foundations of the field (e.g., computer science, engineering, information technology), and the emerging software architecture discipline, future software engineers and software architects will need to understand the viewpoints of multiple, diverse stakeholders at all levels of the Enterprise and synchronize composable software engineering solutions that meet the cost, performance, and schedule constraints.

Distributed, integrated product development teams software engineers and software architects working with authoritative product line repositories of valid domain data provide one such method for improving future simulation credibility. Software engineering and software architectures are now emerging as critical domain disciplines that must continue to mature the corresponding bodies of knowledge, if the Department and the Nation are to

exploit the full potential of computer technology, and set the foundation for the next great era in computing, the Software Architecture Era.

C. RECOMMENDATIONS FOR FUTURE RESEARCH

The simulation software architecture-based product line approach to simulation credibility, supported by architecture readiness levels and domain integrated product development teams provides five proposed areas for future research. The Simulation Software Architecture (SSA)-Based Model provides an abstract software architecture-based horizontal foundation supporting multiple viewpoints and views. Second, the Simulation Software Architectural Framework (SSAF) adds a second vertical-slice architectural component overlaying the SSA, which includes system and environment components.

These architectural artifacts support the Simulation Product Lines Architecture (SPLA) providing the framework to manage the variability, features and differences between products. The fourth component, the Simulation Software Architecture-Based Product Line Model Domain Metadata Repository provides the structure for the Domain Metadata Registry to ensure interoperability with Department, Federal Government, and private sector metadata registries. The Architecture Readiness Levels (ARL) to measure future architectural components and products developed from this methodology is the fifth area suggested for future research.

1. Extension of the Simulation Software Architecture (SSA)-Based Model into Different Domains

The lack of a true software architecture foundation for the Department's simulation system development is the equivalent of being on the road to Abilene³⁴⁰ without a plan, and nobody knows why we are going. The SSA Model complements and supports the existing HLA, with a focus on improving the conceptual models and data elements of the current Common Technical Framework. Figure 11-2 illustrates the current plan to implement the Missile Domain SSA. The first phase involves decomposing the domain M&S hierarchy. The second phase involves the identification of patterns for future product lines. Chapter

³⁴⁰ Defense Systems Management College teaching point in the Program Management Course reinforcing the need for a program plan.

IX provided the pattern analysis for the first two phases. The final phase involves the core asset mining and architecture reconstruction for the SSA.

The development of views and viewpoints suggest the need for a broad-based research effort in multiple domains and the development of the domain metadata repository to support the development of simulation software architecture description (SimSAD).

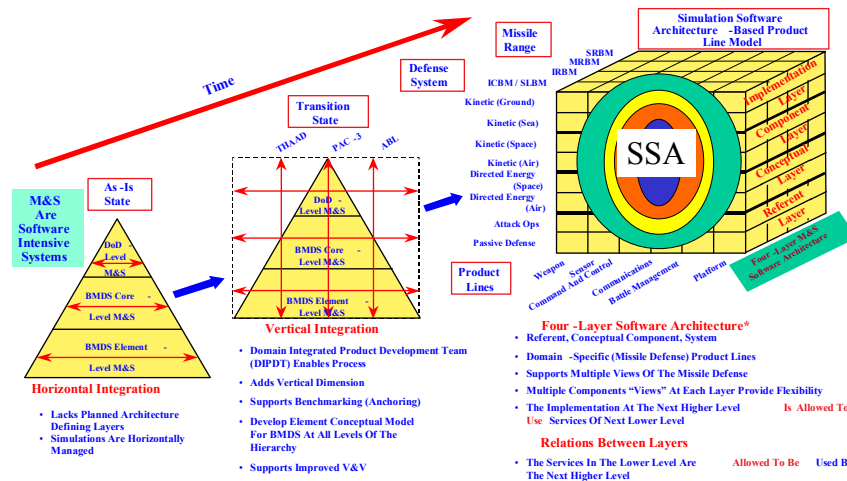


Figure 11-2. Missile Defense Domain SSA Status (After [Gre03b])

2. Simulation Software Architectural Framework

We identified the SSAF supporting the Simulation System Architectural Description (SimSAD), as the system's *environment*, which includes the physical world and the external objects, conditions, or processes influencing the behavior of the system in past, present, and future dimensions, real or imagined, but clearly defined and documented in the SimSAD. The SimSAD provides the foundation for the follow-on SPLA to manage the product line variability of a SimSAD.

In the missile defense domain, future research would determine if the SimSAD provided the scalability to manage the large number of simulation models projected to support a dynamic block evolution of the missile defense system in addition to unplanned experiments and excursions. Future research on the environment component of the SSAF could provide insights into the viability of this approach in supporting a major weapon system acquisition, which must function perfectly the first time in a worldwide environment.

3. Simulation Product line Architecture

The SPLA provides the architectural capability to exploit commonality. There is a potential requirement for thousands of authoritative simulation model representations. In addition, the BMDA capability-based acquisition process may add new elements, while terminating or reducing other element development. This scenario indicates an additional need to maintain accurate configuration management of variant, version, feature, and option requirements of past and future unknown systems. New system developments may also add more complexity since the systems may be unprecedented. Additional research could identify the boundaries for the SPLA.

4. SSA Domain Metadata Repository Model

The SSA Domain Metadata Repository, when fully populated will maintain a *Domain Metadata Registry*; a *Domain Metadata Catalog* containing instances of metadata associated with domain data resources, supporting the use of search portals and queries; a *Domain MetaClass Catalog*, which supports the development of metamodels with a class whose instances are classes; A *MetaLanguage Catalog* which specifies some or all of the aspects of a MetaLanguage used in the Domain; a *Domain Meta-Metamodel Catalog* or model that defines the language for expressing a metamodel; a *Domain MetaModel Catalog* defining the language for expressing a mode; a *Domain Ontologies Catalog* including data categorization schemes, thesauruses, glossaries, key-word lists, and taxonomies; a *Domain Schemas Catalog* representing database tables and relationship structures, XML document type definitions (DTD), and XML schema; *Features, versions, variants, options, and quality attributes, with associated profiles*; and *Architecture Readiness Levels (ARLs)*.

Additional research may evaluate if data is visible, available, and usable when needed and where needed to speed decision-making, including the DoD Metadata Registry identified in Figure 11-3.

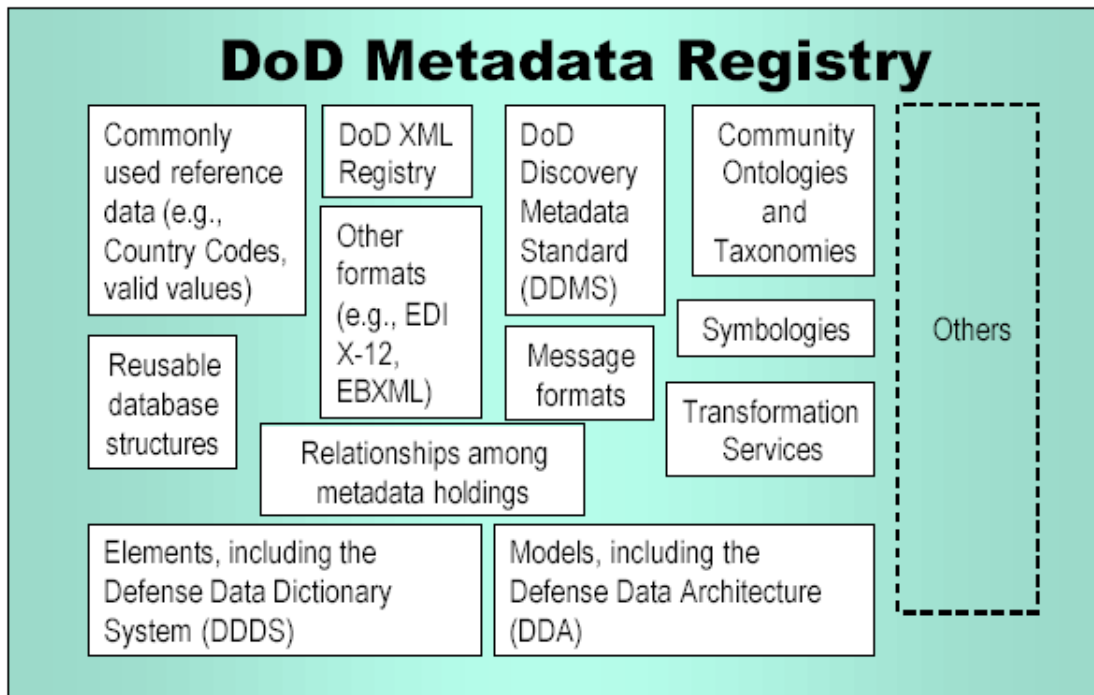


Figure 11-3. Future Areas of Research (From [Ste03b])

5. Architecture Readiness Levels

Architecture Readiness Levels provide the opportunity to anchor simulation credibility and decision-makers' confidence in an architectural foundation with quality attributes. Additional research in improving the V&V process based on the architectural maturity of the model representations has potential to improve decision-makers' confidence in future simulation results.

D. CONCLUDING REMARKS

Chapter I posed the following question: How may the Agency better define, develop, evolve, manage, and maintain authoritative model representations in the five selected large-scale, software-intensive legacy simulations and effectively address simulation interoperability issues (e.g., heterogeneous system representation anomalies) within existing constraints and pre-conditions (e.g., technical, organizational, managerial) affecting credibility in the five selected simulations systems?

The following hypothesis was posed in response to this question: Employing a domain-managed, four-layered simulation software architecture-based product line model with a referent layer, developed in a distributed domain integrated software engineering environment supporting the evolution of five selected Agency legacy simulations can improve the authority of representations in the five selected missile defense simulations. The Software Architecture-Based Product Line Model developed in a distributed development environment by a Domain Integrated Product Development Team employing Architecture Readiness Levels (ARL) to measure quality provides such a methodology. Hypothesis testing included concurrent implementation of the techniques and procedures in the Agency's M&S domain. The Software Architecture-Based Product Line Model for Simulation Model representations provides such a methodology.

APPENDIX A – BALLISTIC MISSILE DEFENSE ORGANIZATION (1996-2001)

A. INTRODUCTION

During the 1996-2001 period, the Ballistic Missile Defense Organization (BMDO) consisted of two major components, a Theater-based missile defense program and the Nation's National Missile Defense program. Figure A-1 identifies the major U.S. missile defense acquisition programs existing in 2001, including international collaborative efforts, major technology programs, and the supporting BM/C2, weapon, sensor, and communication components. Figure A-1 also identifies the service or organization(s) responsible for the management of the program development for each system [Bau02].

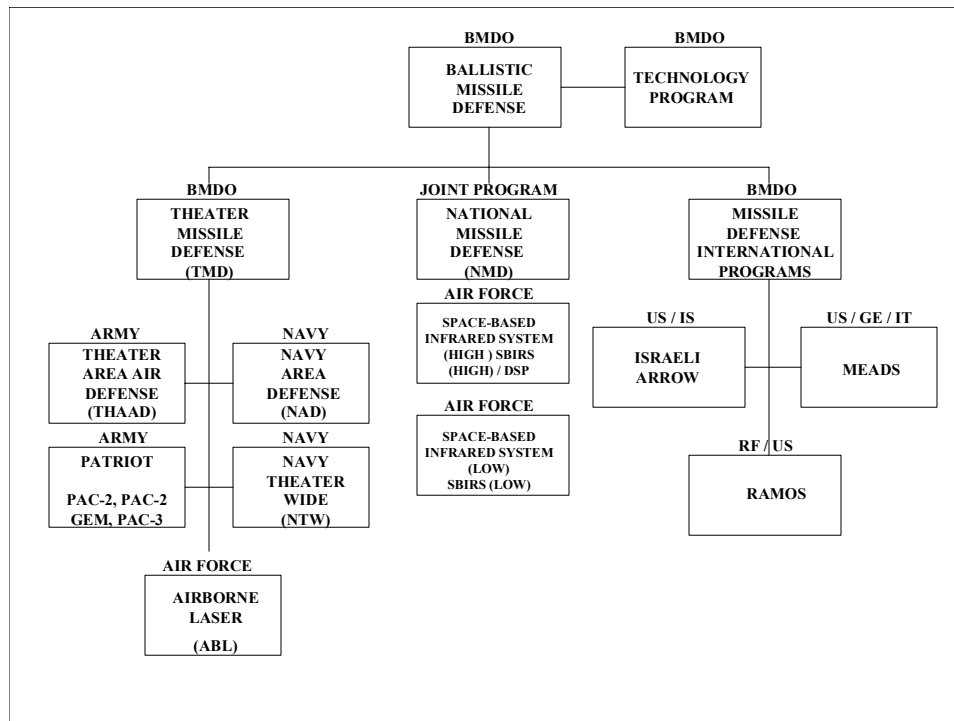


Figure A-1. Missile Defense Organization – 2001 – (After [BMD00a])

B. THEATER MISSILE DEFENSE (TMD) ORGANIZATION – (1996-2001)

No single system can possibly protect the diverse and complex threat posed by enemy ballistic missiles, aircraft, and cruise missiles against U.S. and coalition forces, selected assets and population areas within a theater of operations. The TMD and supporting

the Joint Theater Air and Missile Defense (JTAMD) doctrine plans to detect, classify, intercept, and destroy/negate enemy missiles prior to launch, or while in flight. The BMDO, the Joint Theater Air and Missile Defense Organization (JTAMDO), Combatant Commanders, and other Department Components developed a Family of Systems (FOS) [BMD00b] concept to provide a coherent, cohesive, and effective protection capability. The FOS incorporated four major pillars: Attack Operations³⁴¹, Active Defense of post-launched theater missiles, Passive Defense³⁴² through early warning, and an integrated Battle Management, Command and Control, Communications, Computers, and Intelligence (BMC4I)³⁴³ to provide missile defense in-depth. Figure A-2 illustrates the complex enemy threat and the JTAMD Family of Systems developed to counter them.

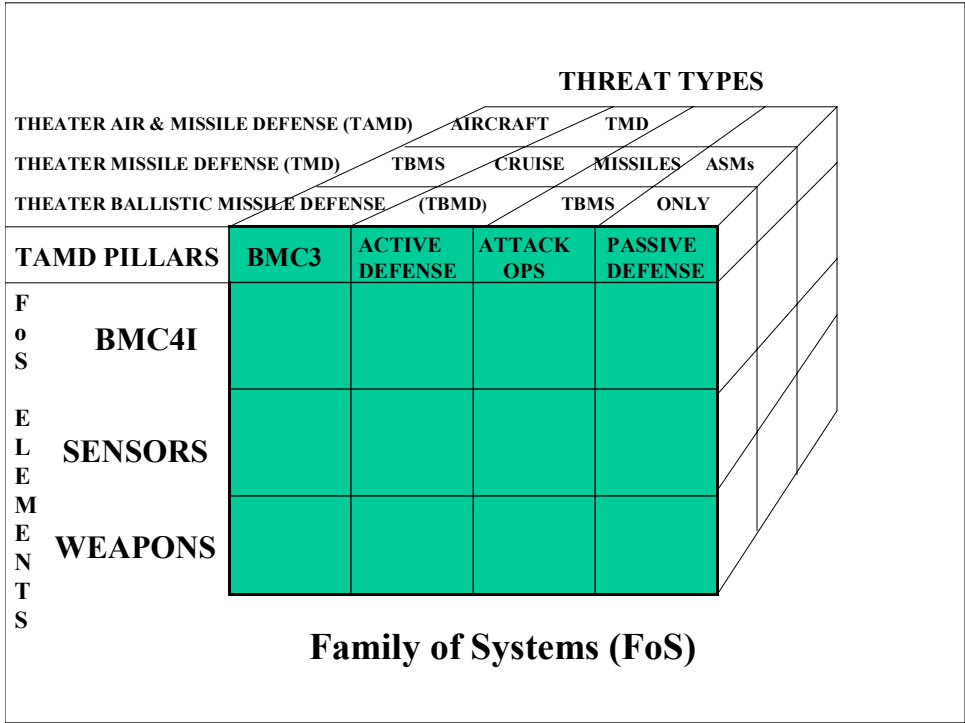


Figure A-2. Family of Systems (After [BMD00a, BMD00b])

³⁴¹ Attack operations includes both pre- and post-launch attack of enemy transportor- erector- launchers (TELs) and supporting infrastructure [BMD00a].

³⁴² Passive defensive measures include actions such as deception, dispersion, or protective construction to minimize the enemy ballistic missile defense effectiveness [BMD00a].

³⁴³ BMC4I includes the controlling and coordinating procedures and systems to integrate an effective defense [BMD00a].

1. PATRIOT – (1996-2001)

The Army's Patriot missile defense system, managed by the Army, has evolved since 1965, first as the Surface to Air Missile Development (SAM-D), followed by Pre-Planned Product Improvements into the Patriot Anti-Tactical Missile – 1 (PAC-1) and –2 (PAC-2) systems. In Operation Desert Storm, the Army quickly upgraded the PAC-2 system from an air defense system with the Quick Response Program (QRP) and Guidance Enhancement Missile (GEM) modifications in 1993 to counter the new SCUD missile threat, emerging as the PAC-2 GEM version of the system. The current PAC-3 system [MDA02v] employs HTK technology and constitutes the lower tier of a layered defensive capability supporting the Theater Missile Defense concept. Major PAC-3 components include the BMC4I, sensors, and weapon system. In 2000, the system was preparing for Independent Operational Test and Evaluation (IOT&E) and a production decision [BRC+01].

2. THEATER AREA AIR DEFENSE (THAAD) PROGRAM – (1996-2001)

THAAD is an Army theater missile defense (TMD) system [MDA02u] designed to engage the full spectrum of theater class ballistic missile threats, employing HTK technology. THAAD was designed to meet critical warfighters needs to intercept longer-range theater-class ballistic missiles at high altitudes and further away from the intended target. The THAAD system is the only defensive system that is effective inside and outside the earth's atmosphere, and may be quickly deployed worldwide on short notice. Major THAAD components include the BMC4I, sensors, and weapon system.

3. NAVY AREA DEFENSE (NAD) PROGRAM– (1996-2001)

The NAD is a lower tier system designed as a tactical missile defense system against endoatmospheric missile threats within the earth's atmosphere. The Aegis weapon system equipped with the phased array, SPY-1B/D radar, and BMC4I was the foundation weapon platform for the NAD. It will evolve into the future sea-based missile defense capability of the BMDS.

4. NAVY THEATER-WIDE (NTW) DEFENSE PROGRAM– (1996-2001)

The NTW missile defense system [BMD99a] was a tactical missile defense system against medium-and intermediate-range theater ballistic missiles outside the earth's atmosphere. The NTW will have an improved interceptor, phased-array SPY-1B/D radar, and BMC4I. With the evolution of the modified Standard Missile (SM-3) and enhancements to the Aegis Weapon System, the NTW defense system has become a component of the sea-based midcourse segment of the BMDS.

5. AIRBORNE LASER DEFENSE PROGRAM (ABL)– (1996-2001)

The ABL system [MDA02o] is a rapidly deployable airborne platform: a 747-400F aircraft, equipped with a long-range laser weapon to acquire, track, and negate TBMS in the boost phase of flight. The ABL will be capable of operating above cloud-level for extended periods of time, providing wide-area geographic and near 24-hour temporal coverage of potential TBM launch areas. The ABL weapon system includes the aircraft, BMC4I, laser, and beam control/fire control segments.

6. MEDIUM EXTENDED AIR DEFENSE SYSTEM (MEADS)– (1996-2001)

The Medium Extended Air Defense System (MEADS) [MDA02w] is an international cooperative terminal missile defense program shared by the U.S., Germany, and Italy to provide 360° protection for maneuver forces against short-range tactical ballistic missiles, cruise missiles, and unmanned aerial vehicles. MEADS will replace HAWK and PATRIOT air defense systems with point and area defense capabilities. The system is composed of a surveillance radar, fire control radar, launcher, missile, and Tactical Operations Center.

7. ARROW WEAPON SYSTEM– (1996-2001)

The Arrow Weapon system [MDA02t], jointly developed by the U.S. and Israel, provides short-and medium-range ballistic missile defense of Israel and U.S. forces deployed in the region.

C. NATIONAL MISSILE DEFENSE (NMD) ORGANIZATION – (1996-2001)

The NMD will provide protection from a limited attack on the United States by ballistic missiles. The NMD program [BMD00c, BMD00i] exploited the products of previous technology and development programs, including the ground-based radar (GBR) prototype, development of the X-band phased-array radar³⁴⁴, passive infrared technology from the Brilliant Pebbles program, and advanced tracking algorithms from the Exoatmospheric Re-entry Vehicle Interceptor System (ERIS) and High Endoatmospheric Defense Interceptor (HEDI) programs. The NMD system [BMD00c, BMD00i] is comprised of the following major components:

1. NMD GROUND–BASED SENSORS- (1996-2001)

Ground-based sensors include the Upgraded Early Warning Radars (UEWRS) [BMD00h, MDA02g], based on existing Pave Paws and Ballistic Missile Early Warning Systems (BMEWS), and the X-Band Ground-Based Radar (GBR/XBR)³⁴⁵ system [BMD00g].

2. NMD SPACE–BASED SENSORS- (1996-2001)

Space-based Sensors include the existing Defense Support Program (DSP) satellite system³⁴⁶ [BMD00h, MDA02g], the planned deployment of two future systems, the Space Based Infrared System (SBIRS) High [BMD00h, MDA02g] to provide missile launch, de-

³⁴⁴ XBR efforts included improved software operations, applications processing, and new radar imaging technologies [BMD00a].

³⁴⁵ The XBR phased-array system conducts track-sweeping operations electronically, which is much faster than mechanically based antennas. With high resolution, the short wavelength of the X-band radar supports early tracking, identification, and discrimination of targets some 2,400 mile away [BMD00a].

³⁴⁶ The DSP system, initially deployed in 1971, includes a constellation of satellites in geosynchronous orbit 23,000 miles above the earth, linked to earth stations to provide missile launch warning and limited tracking capabilities [BMD00a].

tection, surveillance, and track data; and the Space Based Infrared System (SBIRS) Low [BMD00h, MDA02g] to provide more discrete identification, discrimination, and tracking capabilities early in the missile's flight. These systems will also provide an assessment of post-intercept results. SBIRS Low builds on the earlier success of the Midcourse Space Experiment (MSX), which demonstrated the capability of long-wave infrared surveillance and discrimination from space and will support both TMD and NMD systems.

3. NMD GROUND-BASED INTERCEPTOR (GBI) - (1996-2001)

The NMD Ground-Based Interceptor is an enhanced booster stack composed of available commercial-off-the-shelf booster components. The booster delivers the Exoatmospheric Kill Vehicle (EKV), a kinetic energy HTK interceptor into engagement range for a midcourse intercept of the ballistic missile³⁴⁷ warhead [BMD00e, MDA02q]. Battle Management/Command, Control, and Communications (BM/C³) system is actually a composite of three discrete components: battle management, command and control, and communications. Within the NMD system these components consist of the software, hardware, facilities, and communications for planning, tasking, and controlling, ensuring positive human control over all aspects of the NMD system [BMD00a, BMD00f].

³⁴⁷ A ballistic missile is an intercontinental class missile with a range in excess of 5,500 kms [BMD00a].

APPENDIX B – BALLISTIC MISSILE DEFENSE SYSTEM (BMDS) PROGRAM 2002-PRESENT

A. BMD SYSTEM OBJECTIVES

The Agency will develop the Ballistic Missile Defense System incrementally with layered defenses to intercept ballistic missiles of all ranges in all phases of flight—boost, midcourse and terminal [Rum02c] as illustrated in Figure B-1.

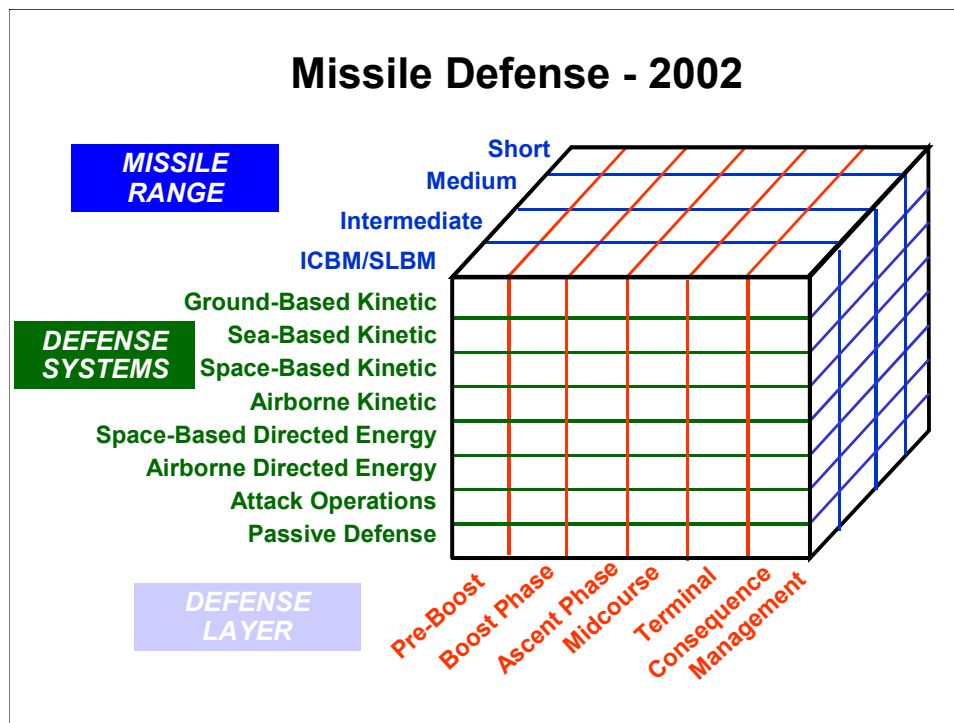


Figure B-1. The BMDS Mission Scope (After [Gre03b])

The existing FoS capabilities will continue to evolve with the BMDS, however the combination of targets, defense systems, and layered defensive options illustrated in Figure B-1 will necessitate major changes to the BMD M&S program. The BMDS is a single system managed as an integrated program, a major programmatic shift from the MDAP-centric systems previously managed by the Services. The BMDS will consist of elements configured into layered defenses to provide autonomous and mutual support, which will provide multiple engagement opportunities against a threat ballistic missile in all phases of flight.

B. BMD SYSTEM-LEVEL CAPABILITY

The Department provided the MDA with new guidance and direction identified in Table B-2 to develop the BMDS.

Previous Direction	New Direction
Treaty Constrained	Withdrew From ABM Treaty June 2002
Requirements-Based	Capability-Based
Major Defense Acquisition Program (MDAP) Focus	Ballistic Missile Defense System (BMDS) Focus
Joint Requirements Oversight Council	Senior Executive Council (SEC) And Missile Defense Support Group (MDSG)
5000 Series Acquisition Life Cycle	3 Phases: RDT&E, Transition, Procurement By Blocks
Standard DoD Contract Authority	Other Transaction Authority (OTA) – National Team
R&D / Acquisition Functions	R&D / Acquisition Functions / IDO
Mission Needs Statement (MNS) / Operational / Capstone Requirements Documents (ORD / CRD)	System Technical Objectives And Goals (TOG) Capability-Based ORD
Developmental / Operational Test	Developmental / Operational Test

Table B-2. Changes to BMDS Direction

The BMDS system engineering process [MDA021] will allocate these system capability specifications cited in Table B-3 to BMDS segments/elements/systems/ subsystems in Table B-4 through a series of documents including the BMDS Technical Objectives and Goals (TOG), the Adversarial Capability Reference Document (ACRD), System Evolution Plan (SEP), Integrated Master Plan, Integrated Master Schedule, and System Capability Specification (SCS), among others. The MDA system engineering function and M&S program must be able to support the development, maintenance, and testing of the unprecedented BMDS and must successfully address:

- Operational realities
- Engineering complexities
- Transitioning from a requirements-based process to a capabilities-based approach
- Implementing an adversarial capability approach instead of a clearly defined threat
- Implementing an evolutionary acquisition strategy using a two-year block approach [MDA021].

<u>DETECTION</u>	<u>IDENTIFICATION / CLASSIFICATION</u>	<u>BATTLE MANAGEMENT</u>	<u>ENGAGEMENT COORDINATION</u>	<u>KILL ASSESSMENT</u>
• Search • Detect • Correlate • Track • Report • Update • Discriminate • Countermeasures • Cue	• Initial ID • ID Updates • Initial Classification • Classification Update • Discriminate • Emerging Threats • Warhead ID	• Track • Situation Awareness • LPE/IPP • Target Nomination • Synchronize • Weapons Management • Engagement at Max Range • Upper/Lower Tier Engagement • Inventory Management	• Target Acquisition • Weapons Assignment • Weapons Release • Mission Control • Cue • Deconfliction	• Track • Discriminate debris • 2^d Engagement Opportunities • Report Status / Results

Table B-3. Required BMDS Capability

C. BMD SYSTEM-LEVEL LAYERED DEFENSE CAPABILITY

1. BOOST PHASE DEFENSE SEGMENT (BDS)

The Boost Phase Defense Segment (BDS) [MDA02i, MDA02j, MDA02n] is designed to counter that part of a threat missile's flight path from launch until it stops accelerating under its own power. At this point in its flight the threat missile has been climbing against the Earth's gravity for 100 to 300 sec. and may achieve an altitude of 200 km. The Boost Phase Defense (BDS) intercept of the missile's flight by the ABL [MDA02n] under development as a system and other future BDS is the best solution, allowing for the defense of a very large area of the Earth, and destruction of the target before the deployment of midcourse countermeasures.

2. MIDCOURSE DEFENSE SEGMENT (MDS)

The Midcourse Defense Segment (MDS) [MDA02i] is that part of the ballistic missile trajectory where the missile has completed the thrust phase and is following a more predictable flight path during the mid-course phase as it freefalls towards its target. This

phase may last as long as 20 minutes in the case of an intercontinental ballistic missile (ICBM). This phase offers the defensive system a longer time to track and intercept, and provides the possibility of having enough time to launch more than one interceptor. However, this phase also supports the deployment of countermeasures against the defensive system.

The Midcourse Defense Segment has two systems: the Ground-Based Midcourse Defense (GMD) [MDA02q] and the Sea-Based Midcourse Defense (SMD) [MDA02r] elements, successor systems to the NMD and NTW programs. The GMD interceptor has two major parts: the booster vehicle and the Exoatmospheric Kill Vehicle (EKV) [MDA02q]. The EKV destroys the target with kinetic energy, closing on the threat at a closing velocity of approximately 15,000 m.p.h. at altitudes more than 100 miles above the earth [MDA02q]. The SMD includes the existing Aegis Weapon Systems (AWS) with the AN/SPY-1B/D radar, and the Standard Missile 3 [MDA02r].

3. TERMINAL DEFENSE SEGMENT (TDS)

The BMDS Terminal Defense Segment (TDS) [MDA02i, MDA02t, MDA03a] covers the final phase of the threat's flight that normally lasts approximately 30 sec. for an ICBM class of vehicle reentering the earth's atmosphere at over 2,000 M.P.H. [MDA02j]. Countering the TDS threats are the PAC-3 missile [MDA02v], a high velocity HTK technology against aircraft, cruise missiles, and short-range ballistic missiles, the Arrow Weapon System [MDA02t] and the MEADS [MDA02w]. The THAAD ground-based system complements the PAC-3 and is capable of engaging short-and medium-range ballistic missiles inside and outside the Earth's atmosphere [MDA02u].

4. SENSOR SEGMENT

The Sensor Segment [MDA02i] has two primary projects: Space Sensors and International Cooperation. The Space Sensors project supports the Block 2010 Space Based Infrared System (SBIRS) increment 3 and Space Tracking and Surveillance System (STSS) and will also support the BMDS Boost, Midcourse, and Terminal Segments with a constellation of infrared sensing satellites. The International Cooperative project, the Russian-

American Observation Satellite (RAMOS) [BMD00a] is a cooperative research and development effort to observe the earth's atmosphere and ballistic missile launches. The Targets and Countermeasures Program [MDA02s] provides threat representative targets and countermeasure challenges to the evolving BMD system and elements.

5. BATTLE MANAGEMENT / COMMAND AND CONTROL (BM/C2)

The BM/C2 [MDA02i] architecture will provide a highly complex and advanced C2 element to effectively manage the system segments and execute the battle management function by achieving interoperability³⁴⁸ with the system elements, external interfaces, and integrate with the national military command structure. BM/C2 improvements include algorithms and techniques to improve the control of the future missile defense system.

<u>WEAPONS</u>	<u>SENSORS</u>	<u>BMC2</u>	<u>COMMS</u>
<u>MISSILES</u> PAC-2 PAC-2 GEM PAC-2 GEM + PAC-3 THAAD PDRR THAAD EMD GMD GBI 1 W/EKV GMD GBI II W/EKV AEGIS SM-2 BLOCK IV NAVY LEAP ALI SMD SM-3 ARROW MEADS	<u>RADARS</u> PAC-2 AN/MPQ-53 PAC-3 AN/MPQ-65 THAAD UOES/TPS-X THAAD GBR COBRA DANE UEWR SEA-BASED XBR GBR/XBR AEGIS SPY-1B ARROW RADAR MEADS FIRE CONTROL MEADS SURVEILLANCE	<u>BMC2</u> BMDs C2BM GMD BMC2 AEGIS BMC2 PAC-2 BMC2 PAC-3 BMC2 ABL BMC4I SBL BMC2 JTAGS/ALERT MGS THAAD BMC3 MEADS BMC2 ARROW BMC2 TAOC CRC	<u>COMMS</u> JDN/LINK-16 LINK-16 JRE JPN JECN JCTN JMMN DISN IBIS
<u>DIRECTED ENERGY</u> AIRBORNE LASER (COIL) SPACE-BASED LASER	<u>SATELLITES</u> DSP SBIRS-HIGH SBIRS-LOW RAMOS	<u>PLATFORMS</u> YAL-1A ABL (747-400) THAAD LAUNCHER PAC-2 LAUNCHER PAC-3 LAUNCHER AEGIS	
<u>KINETIC ENERGY</u> SEA-BASED KINETIC SPACE-BASED KINETIC	<u>LASER</u> BEACON ILLUMINATOR TRACKING ILLUMINATOR ACTIVE LASER RANGER		

Table B-4. Major Ballistic Missile Defense Systems

³⁴⁸ Interoperability is the ability of systems, units, or forces to provide services to or to accept services from other systems, units, or forces and to use the services so exchanged to operate effectively together [MDA02b].

D. BMD SYSTEM ACQUISITION STRATEGY

The new Agency mission is to develop, test, and prepare for deployment of a missile defense system that will be able to engage all classes and ranges of ballistic missile threats with complementary land-, sea-, air-, and space-based interceptors, directed energy weapons, sensors, battle management, command and control, and communications systems. The BMDS capability identified in Figure B-5 will be characterized by:

- A constantly changing system configuration,
- Continuous upgrades to software,
- A constantly changing threat,
- Limitations imposed by the physical environment which must be understood,
- Interoperability and integration challenges, and
- The political will of the Nation.
- The Agency's has a three-phased evolutionary acquisition program [MDA02k]: development, transition, and procurement and operations [Rum02c]. A research, development, test, and evaluation program will support the fielding of missile defense capabilities to the field in two-year increments, with each successive block having a greater capability than the previous blocks.

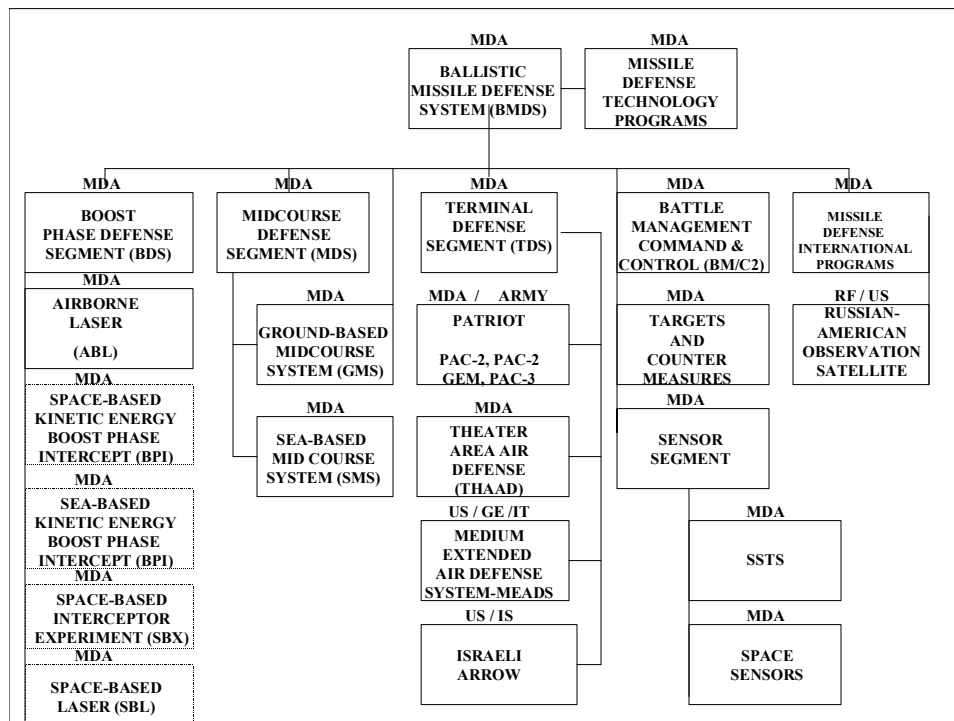


Figure B-5. BALLISTIC MISSILE DEFENSE SYSTEM (BMDS) – 2002

E. BMD SYSTEM TECHNOLOGY PLANNING

Technology played a critical part in the previous missile initiatives. Future required advances in technology include many areas: sensors³⁴⁹, weapons³⁵⁰, and BMC3. Critical sensor area research includes the Advanced Radar Technology program, which focuses on anticipating and negating countermeasures, improving discrimination functions, and aids the identification of the target in a potentially cluttered environment. The resolution of many complex countermeasure issues requires the development of more advanced discrimination algorithms and infrared sensor technologies.

Future weapon technologies include research into Space-Based Lasers, Atmospheric and Exo-atmospheric Interceptor programs. These programs will incorporate advanced HTK technologies, reduce costs, improve seeker capabilities and materials of the interceptor to discriminate and correctly identify the correct target in time to close and destroy the target.

³⁴⁹ Sensors and detectors working individually or in networks may include passive sensors in the visible (optical), infrared, and ultraviolet ranges; active sensors including radars or ladars (laser detection and ranging); and may be airborne, ground-, sea-, or space-based. Sensors on interceptors and satellites are normally passive, while active sensors requiring high levels of energy are normally surface-based [BMD00a].

³⁵⁰ Weapon systems may be airborne, ground-, sea-, or space-based, and must deliver enough energy into the target to destroy or negate it. These technologies currently include HTK interceptor systems and directed energy (laser) systems [BMD00a]

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C– COMMANDERS ANALYSIS AND PLANNING SIMULATION (CAPS)

The CAPS simulation illustrated in Figure C-1 consists of the following components configured in separate executable files: a graphical user interface (GUI), a simulation engine, and several utility programs, which convert data formats and support analysis. A user may enter the simulation environment or run the simulation through the GUI or via a shell script. The GUI program interfaces with the other CAPS components via their input and output files, detects program termination and displays the results.

CAPS may simulate missile defense campaigns up to several months in length, with a scripted striking force, and a defending force that attempts to prevent damage to defended areas from the striking force ballistic missile threat (Figure C-2). Developed threat scenarios may consist of a series of raids against selected targets in specified regions of the world. A raid may consist of a mix of ballistic missile and air breathing threats (e.g., aircraft, cruise missiles), termed threat objects, available to the striking force. Threats may be validated threats or user-created/modified.

CAPS TODAY IS MDA'S

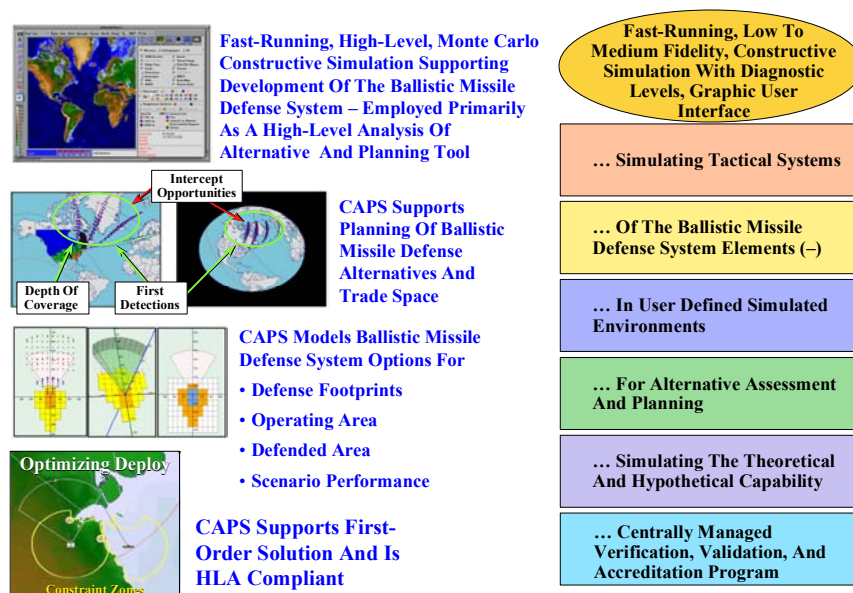


Figure C-1. CAPS Overview (From [Gre03a])

System libraries for many current systems and regional topographic data are provided, and a user may create additional systems tailored to the individual study requirement. Topographic data libraries for selected regions are of medium-high resolution, while the entire world is provided at lower resolutions.

Documentation for CAPS includes a User's Manual supporting analysis procedures, the Methodology Manual [Spa00] describes the methods and algorithms in the simulation engine, the Database Guide [Spa01] provides the data structure and content of the CAPS database, and the CAPS Training Brief. The CAPS system also provides several sources of information, including a statistical summary on the effectiveness of the defensive and offensive forces, event logs from previous runs, and a status file that allows follow-on execution of the campaign supported by CAPS utility programs for plotting and data extraction.

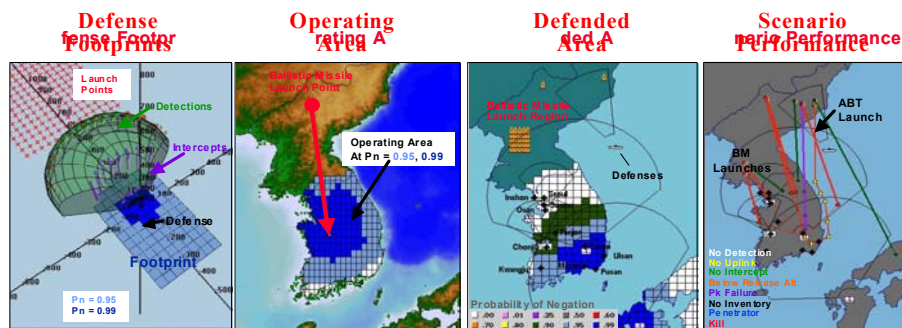


Figure C-2. CAPS Intended Uses (From [Gre03a])

The defending force deploys sensors and interceptors in space and the air, on land, and at sea, to detect and engage the threat objects before they can damage defended areas. Before each scenario, the defending force for each analysis option is created with the GUI, or modified from a previous scenario. The CAPS simulation engine reads all input files,

initializes the campaign system clock and all simulation components, before entering the main simulation loop.

Within the main simulation loop, CAPS is a discrete-event simulation and a hybrid event-driven, time-stepped constructive simulation [Spa00]. In the hybrid event-driven, time-stepped constructive simulation mode, time-compression algorithms reduce run time in scenarios with several periods of intense activity, separated by long periods of relative inactivity prior to the next event. (Figure C-3).

Using time compression, CAPS is able to run an 80-day theater-level campaign in approximately an hour, depending on the available machine resources. CAPS was described by [BMD99a] as a low-fidelity theater-level analysis tool, capable of modeling system performance and interceptor usage for land-, air-, and sea-based missile defense systems employed against TBMs.

In CAPS the primary simulation loop contains three segments (Status Update, Acquisition and Tracking, Engagements) and acts as the executive [Spa00]. Within this loop the CAPS phenomenological models that simulate the campaign environment, threat object, sensors, weapons, and command authority functions simulate the events that occur within the current step [Spa00].

The first segment in the CAPS Simulation Engine processes the status of all defending units and all of the threat objects through a call to the CAPS Composite Threat Model. This process activates threat objects in the current time slot, updates a state vector with the threat object location and velocity, and deactivates threat objects which reach their target during the current time slot. The status of all defensive units are then updated by calls to: 1) the CAPS Deployment Model for changes to sensor or interceptor locations; 2) the CAPS Airborne Interceptor Model, for the latest status of airborne interceptors; and 3) the CAPS Interceptor Model that provides the inventory adjustment status for interceptors

The last task of this segment is a call to either the Phillips Laboratory Airborne Simulation Tempest Revision (PLASTR) model or the ISAAC model, developed by the Air Force, that simulate the suite of sensors and the laser weapon of the ABL. This task is necessary due to the ABL's autonomous search, acquisition, and tracking by the onboard sensors and engagement by the ABL laser weapon [Spa00].

- Graphical User Interface
- Input And Output Databases Are Managed Through Tools Provided

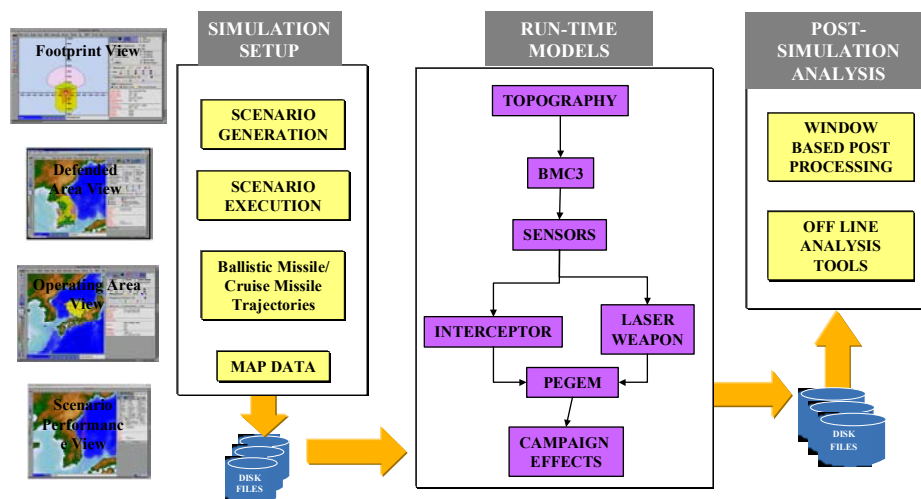


Figure C-3. CAPS Architecture (From [Gre03a])

The second segment determines the acquisition and tracking of threat objects by defending unit sensors, according to the status of each sensor and each threat object. This segment is accomplished by calling the CAPS Radar Search and Acquisition Model to provide the search, acquisition, detection, and tracking functions; followed by a call to the CAPS Command, Control, and Communications Model for communication link status; the CAPS Composite Threat Model to update state vectors for active threat objects; the CAPS Radar Signature Model to determine whether the defensive sensor systems actually detect threat objects in their field of view; and lastly, a call to the CAPS Threat Status Model to update the status of all active tracks and to initiate new tracks [Spa00].

The final segment processes the engagement status of the threat objects based on the results of the previous two segments, calling the CAPS Battle Management Model to initiate new engagements, followed by a call to the CAPS Interceptor Engagement Model to update engagements in process or when it creates a new engagement. Both models call

the CAPS Command, Control, and Communications Model for the latest status about communication links.

The CAPS model is mostly deterministic, with the exception of the stochastic radar detection model [BBG+99f]. CAPS currently runs under UNIX, Windows, and Motif on various host computers, and has been tested on Sun SparcStations, Silicon Graphics Indigo work stations, and personal computers running the Linux, Windows NT and Windows 98/2000 [Spa00].

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D –MISSILE DEFENSE WARGAME AND ANALYSIS RESOURCE (MDWAR)

The MDWAR synthetic battlespace or “gameboard” supports operators with a real-time constructive simulation of futuristic air and missile defense C2 systems in an integrated manner, allowing autonomous and interactive missile defense C2 simulations with variable resolution M&S for air, ground, space, and naval forces and their respective weapon systems, sensors and C2 [TRW00b]. MDWAR, illustrated in Figure D-1 supports both HLA and DIS protocols allowing distributed execution and interoperability with real C4I systems [TRW00b]. MDWAR provides models of all NMD core elements and specific TMD models for all of the four JTAMD pillars of the active defense.

MDWAR TODAY IS MDA’s...

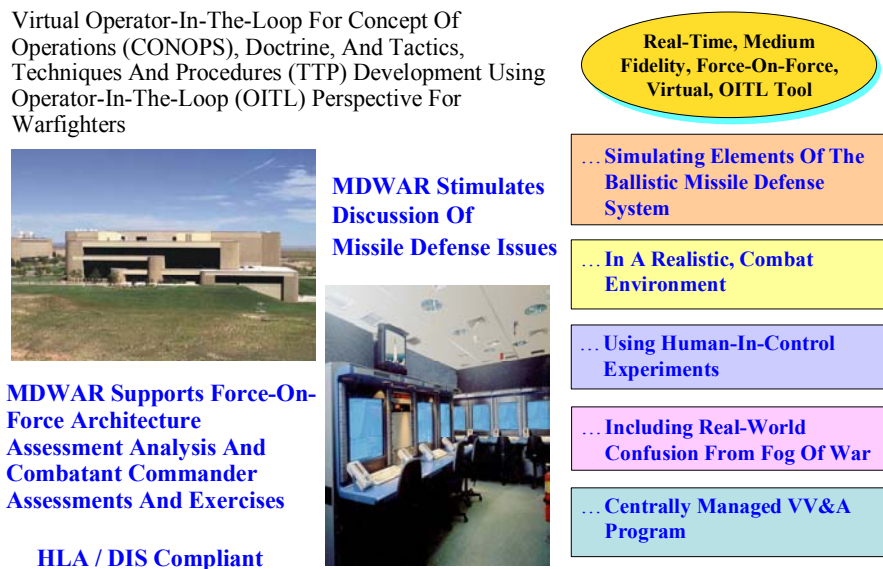


Figure D-1. MDWAR Intended Uses (From [Gre03a])

A goal of the MDWAR C2 war game process is to satisfy approved test objectives, identify the human usability of the missile defense simulation, and use this feedback to develop and refine the CONOPS, doctrine, TTPs, acquisition strategy and military utility of the evolving BMDS. The operator feedback supports the evolution of weapon designs, in-

teroperability, and operational procedures, balancing the effectiveness of the missile defense system with the capabilities of human operators to manage the system. A product of the MDWAR experiment process is the engineering data to support system designers, operational planners, and decision makers with future design decisions, CONOPS, TTPs and missile defense doctrine.

MDWAR simulates a large number of objects (~1,000) in real-time, with sufficient detail and resolution to allow for discrete differences between alternative CONOPS options for operators, analysts, and decision-makers, and supports missile defense experiments in five different modes:

- CONOPS Development – This is the primary mode. The remaining four modes are derivatives:
- Combatant Commander (formerly CINC) assessment and field exercise,
- Staff and operator familiarization,
- Missile defense architecture assessment,
- Test and Evaluation [TRW00b].

MDWAR replaced the Advanced Real Time Gaming Universal Simulator (ARGUS) model due to requirements for higher fidelity weapon and sensor models, more complex and realistic games with player positions similar to those in the field, and the need to reduce the turn-around time between games and analyze lessons learned. The MDWARS architecture (Figure D-2) is a distributed architecture, which allocates the supporting tasks to computational resources separate from the processors required to support the mission-space models. MDWARS is data driven, versus ‘hard-wiring’ or internetworking, employing parameter files for missile, radar containing quantity, performance, location, and other data, upon initialization [TRW00b].

In order to enhance the performance of the mission space models and the processors, an optimistic parallel discrete simulation (PDES)³⁵¹ framework, the Agency selected the Synchronous Parallel Environment for Emulation and Discrete Event Simulation (SPEEDES) as the simulation executive. To achieve performance objectives, SPEEDES executes in a host environment featuring processor clustering, network computing, and par-

³⁵¹ PDES simulations are categorized by how time is managed, how time management functions are distributed, if distributed, and how the distributed components interact [TRW00b].

allel computing architectures employing Symmetric Multiprocessing (SMP)³⁵² and Cache-Coherent Non-Uniform Access (CC-NUMA)³⁵³ systems [TRW00b].

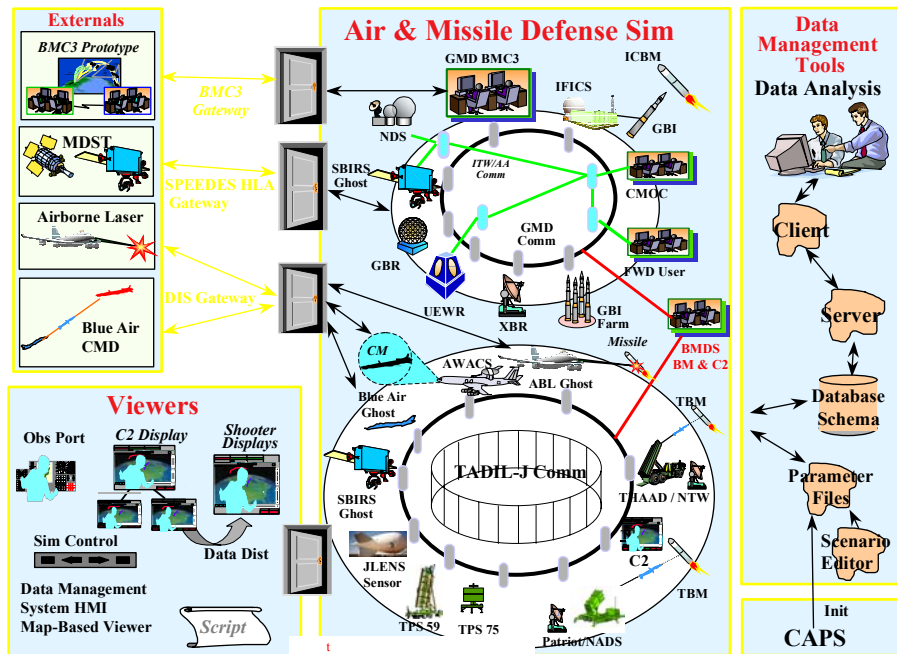


Figure D-2. MDWAR's Architecture (From [Gre03a])

The MDWAR architecture is composed of three sub-components called Top-Level Computer Software Components (TLCSC), which are executed as linked applications:

- Missile and Air Defense Simulator (MADSim) TLCSC – This TLCSC is composed of the mission-space representations, the PDES framework, and the interfacing gateways and simulator controls.
- Viewers and Editors TLCSC – This component includes simulation control, analysis monitoring, and player displays supporting the human interface to MADSim during game execution.
- MDWAR Resource Repository TLCSC – This component maintains all the data and documentation for all phases of the game, after action review (AAR) interfaces to the simulator's object-oriented database, and analysis support tools [TRW00b]

³⁵² On a SMP, communication between processors is through shared memory and a very high-speed bus [TRW00b].

³⁵³ In a CC-NUMA architecture, the processors access shared memory through a cross-point switch, whereas the SMP shares both memory and data buses [TRW00b]

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E – EXTENDED AIR DEFENSE SIMULATION (EADSIM)

Since 1987 EADSIM, shown in Figure E-1, has evolved and today provides capabilities supporting fixed and rotary-wing aircraft, ballistic missiles, cruise missiles, sensors, command and control, communication jammers³⁵⁴, and communication networks. EADSIM supports three standard interface protocols-- ALPS, DIS, and HLA-- and may be run either in simulated or real time, and supports message passing, event passing, and control passing, via these standard protocols [BAC+95]. EADSIM is compatible with Unix operating systems on Silicon Graphics workstations, Sun workstations, and Windows NT [BBG+99e].

EADSIM TODAY IS MDA's . . .



Figure E-1. EADSIM's Intended Use (From [Gre03a])

EADSIM architecture in Figure E-2 includes the following general models: air defense, offensive air operations, attack operations, multi-stage ballistic missiles, air breathers, sensors, jammers, satellites, early warning, generic noncombatants, communications, electronic warfare, and specific areas of interest [BAC+95]. The simulations flexibility is a key reason for the major role EADSIM plays in the BMDS cooperative international M&S program. EADSIM performs three basic functions: simulation set-up, scenario execution by a set of run-time models running in a

³⁵⁴ Jammers are radio transmitters accompanying attacking RVs and tuned to broadcast as the defensive radar to add 'noise' to signals reflected from the RV and received by the defensive radar [MDA02b].

multi-process mode, and post-processing and analysis. The EADSIM simulation also features an integrated graphic user interface for set-up and post-processing support [BAC+95]. EADSIM explicitly models each platform. Modeled platforms represent a system type (e.g., PAC-3) and may have any number of elements such as sensors, weapons, defined by the user [JAS97c].

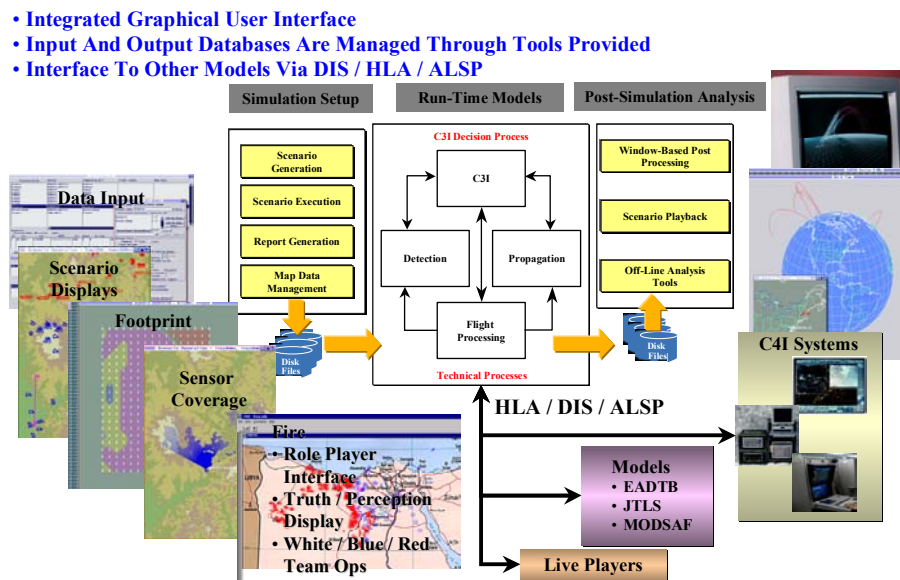


Figure E-2. EADSIM's Architecture (From [Gre03a])

EADSIM is a Monte Carlo model with a capability able to support some deterministic analysis. A set of run-time models represent selected aspects of the scenario, support scenario execution, and perform data exchanges during execution [BAC+95]. There are four EADSIM run-time processes: C3I, flight processing, detection, and propagation [TOM97]. Major functions in EADSIM include the following: battle management, communications, movement, propagation, IR and radar signatures, radar sensors, and natural environments. All models in EADSIM rely on data files for storage and retrieval of scenario information, and reflect the level of model abstraction maintained in the data files [Tom97].

APPENDIX F – EXTENDED AIR DEFENSE TESTBED (EADTB)

EADTB provides a capability to invoke pre-compiled algorithms with an interpreted code, which provides the user with a significant capability to control model functionality and define the weapon or system under study [BBB+96]. EADTB, a two-sided model (Red and Blue forces), illustrated in Figure F-1, models many types of air defense weapon systems, with algorithms supporting the varying degree of representation detail required; and features the ability to model perception versus truth data, that allows the incorporation of limited, flawed, or misinterpreted information to influence the decision process [BBB+96]. Missile defense functional areas modeled by EADB include: BM/C2, movement, missile launching, lethality, vulnerability, signature generation, communications, and the natural environment [Ray02a]. With its variable levels of detail, it is able to support studies of systems, components, tactical doctrine development, and mission planning [BMD99a].

EADTB TODAY IS MDA's . . .

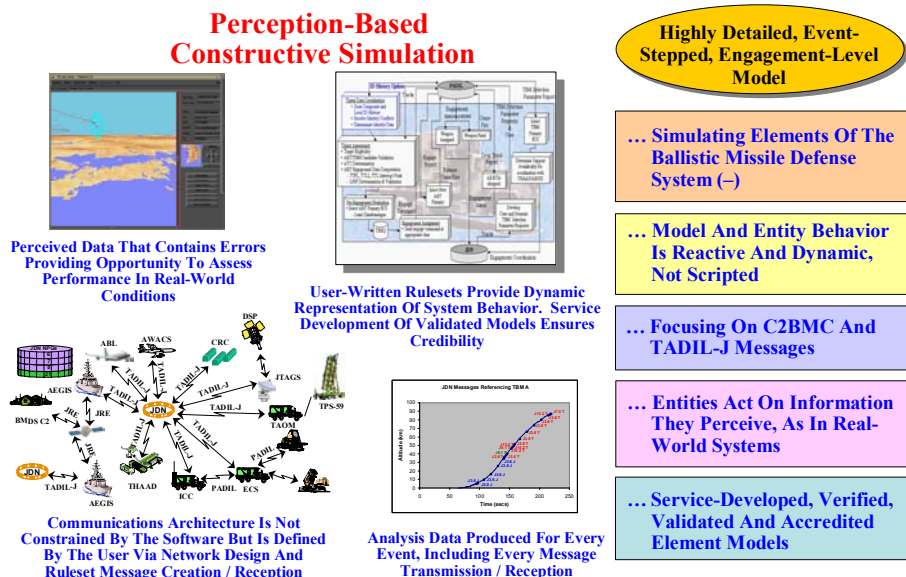


Figure F-1. EADTB's Intended Uses From [Gre03a])

SSRs, called rule sets, are a major feature of EADTB's capability and flexibility, allowing EADTB to function at various levels of detail and aggregation [BBB+96, MR01]. The EADTB Common Model Set (CMS) maintains SSR algorithms, and provides background processes and routines called by SSRs [BBB+96]. User-developed SSRs, built from EADTB components (e.g. Thinker, Platform, Communications, and Sensor) represent specific weapon systems supported by the modeled natural environment in which these components operate.

Each SSR has at least one or more Thinker components [Wai94, WS94], which provides the intelligence for the SSR, the constructive surrogate for the human command and control. Thinker rule sets act on perception³⁵⁵ data as opposed to truth³⁵⁶ data. Thinker algorithms are available to rule sets through a set of specific symbols. The Thinker receives data from onboard sensors and by way of the communications network. However EADSIM processes "truth" data by sending communication, sensors, and other elements information transmitted to the Thinker as "perceived" data. Integral to the Thinker are the user-defined Rule sets [ZCE+01].

The rule set logic combines with the representation of the command and control system and available communication networks to simulate the C2 process. The Rule set invokes a series of algorithms to complete functions including threat assessment, missile guidance, correlation, fusion, and weapon assignment. EADTB supports a wide range of potential SSRs supported by user-defined rule set logic, which may represent both manual and automatic processes. The EADTB interpreted rule set in EADTB consisting of a set of symbols and syntax rules provide all C2 decision-making logic, and supports simulation scenario changes without recompiling or linking. The flexibility inherent in EADTB and the rule sets also creates the greatest challenges, since it requires so much input from the users of the tool.

A SSR also has one Platform component. The Platform component contains algorithms to allow it to model movement and other physical characteristics of different Plat-

³⁵⁵ 'Perception' data constitutes the results of perception transformations (observations, derivation, or estimates) by given entities of the states of other entities within the scope of the simulations / exercise as manifest in local perceived-information data structures (possibly corrupted, inaccurate, imprecise, or imperfectly known) [Wai94]

³⁵⁶ 'Truth' data is generated as part of the original, intrinsic unequivocal representation within the scope of the simulations / exercise and which constitutes the truth regarding the states of the entities as known to themselves or the executive [Wai94].

form types, different kill mechanisms, Platform signatures, and Platform vulnerabilities. The Platform component has five constituent capabilities or physical characteristics that correlate to the real world: 1) move / motion capabilities, 2) the ability to cause damage, or 3) be damaged, 4) the ability to carry and/or launch an object, and 5) the ability to generate a signature³⁵⁷ for reflection or emission [Ray02a].

A Communication component allows the transfer of data between Thinkers, but does not access the contents of the message. Determining who may communicate is dynamic since the operational state of communication devices may change. However, EADTB delivers only error-free messages from the receiving communication unit to the receiving Thinker. Two communication models support message transfers, a low-detail model called Direct Comms for statistically modeling message delay, and the Standard Comms model which models the links, queues, networks, and propagation factors [Ray02a]. The two methods for dissemination messages are point-to-point and broadcast.

Sensor components model the detection of objects by measuring energy levels reflected or emitted by the object. Target detection is a function of the sensor performance, target-signature characteristics, and environmental effects. The user defines the sensor performance by selecting the sensor power and gain. Sensor models may be active sensors such as radar³⁵⁸ or passive sensors including EO/IR passive³⁵⁹ and RF passive sensors³⁶⁰ [Ray02a]. A sensor, controlled by rule set commands, and tasked by a Thinker, may transmit and/or receive over a common set of user-defined extended frequencies.

The level of sensor detail ranges from functional sensors that only process range, target type, and search volume; to fine-grained sensor models that can model gain, polarization, Doppler spreads, ambiguous ranges, and jamming (main lobe or side lobe) [Ray02a]. The Sensor component is comprised of a set of models used to simulate the following sensors: radar, RF passive, RF noise jammer, EO/IR passive, and functional sensor; which are further decomposed into four subcomponents: scanner, transmitter, receiver, and data processor [Ray02a].

³⁵⁷ The signature may be in the radio frequency (RF), electro optical (EO) or infrared (IR) wavelengths.

³⁵⁸ Radars transmit RF energy and measure its reflection from the body of the target [Ray02a].

³⁵⁹ Sensors that measure the energy emitted by the body of the target [Ray02a].

³⁶⁰ Sensors that measure the energy emitted from an active sensor transmitter carried on the target [Ray02a].



The Environment algorithms model the portion of the environment that affects objects by three feature groups: 1) terrain and sea surface, 2) atmosphere, and 3) celestial bodies. The simulation supports DTED, DFAD and user-derived data sets [Ray02a]. The environment influences sensor and communication representations, sensor detection, and the performance of object movement representations.

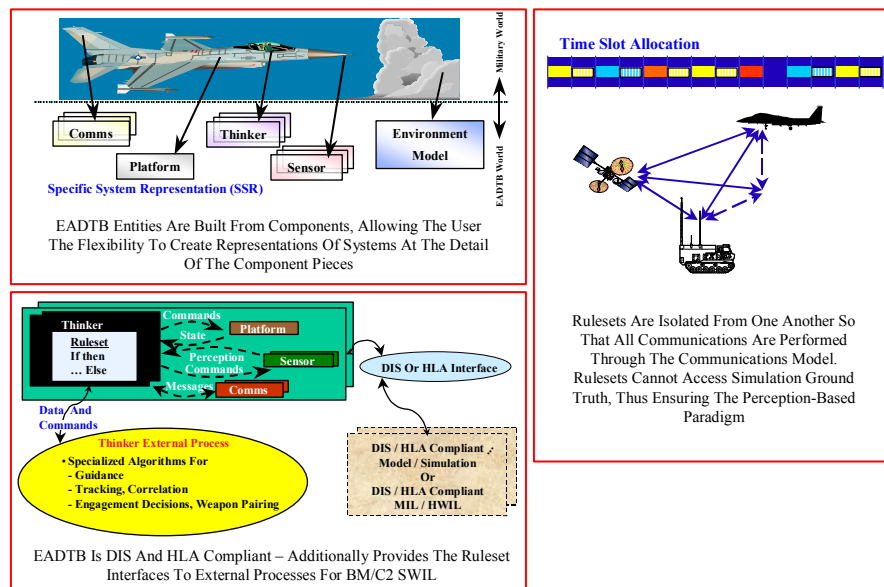


Figure F-2. EADTB's Architecture (From [Gre03a])

EADTB analysts build scenarios incrementally, developing elements such as aircraft, communication devices, and sensors. When deployed on a digitized terrain map, the systems become platforms, and then combined into lay downs, and lastly, merged into scenarios. EADTB architecture shown in Figure E-2, runs on a quad-processor Covex 3840, supporting Silicon Graphics workstations and successfully ported to Silicon Graphics and Origin 2000 workstations. EADTB is currently DIS and HLA compliant [SAP98, Wai99, STB+00, WS00].

APPENDIX G – MISSILE DEFENSE SYSTEM EXERCISER (MDSE)

Starting with a proof of principle in 1993, and evolving with an incremental build methodology, MDSE, shown in Figure G-1, exercises real tactical hardware and software in dynamic, interactive interoperability tests allowing the hardware to be operated earlier in the development cycle and over a wider range of conditions than may ever be possible in live tests. A simulated secure voice radio environment between tactical operators at each site support MDSE HWIL tests, while test operators have secure voice communications coordination with recording and playback capabilities. MDSE uses the DIS protocol [McQ97].

MDSE TODAY IS MDA's . . .



Figure G-1. MDSE's Intended Uses (From [Gre03a])

As a virtual model, MDSE's objective is to stimulate the actual missile defense hardware and software to the maximum extent possible and employ embedded constructive simulations only when necessary due to cost, availability, or physics. Areas where constructive simulations in MDSE's architecture (Figure G-2) are employed include: the infra-red and radar sensor front ends; transmitter, receiver and signal processing; interceptor fly out; and portions of the BMC3 [McQ97]. MDSE also emulates the Joint Data Network

(JDN) [YS00] with a gateway in order to pass Tactical Digital Information Link-J (TADIL-J) [DoD97, BSB+02], Tactical Information Broadcast System (TIBS), Tactical and Related Applications (TRAP), and Tactical Data Distribution (TDDS) tactical messages to exercise participants [Too94].

MDSE then stimulates the sensor data processing systems of the tactical software and hardware to develop a flow of information through the tactical system interfaces. The stimulation supports an assessment of FoS interoperability issues [BB98a, BB98b], including reporting responsibility (R2), track management, cueing effectiveness, engagement coordination, and tactical battle management and C2 implementation in the participating systems [Too94]. The stimuli for the track processing, based on realistic representations of pre-scripted theater-level attack scenarios, pass over a geographically distributed network to a remote-environment interface connected to the tactical system [Was01].

MDSE is composed of three segments interconnected by the Test Communications Network (TCN), the MDSE Control Segment (TCS), the Tactical System Driver Segments (TSDs), and the Tactical Communications Environment Segment (TCES). The TCN consists of a combination of MDSE component and segment local area networks and wide area networks interconnected by T1 encrypted lines in a star topology between the TEC at the hub to all the remote environments (RE) at the test sites [McQ97]. The TCN allows MDSE to monitor and affect the health and welfare status of the MDSE network.

propagation of natural objects such as weather, terrain, natural and hostile environments and auxiliary interfaces such as the global positioning system (GPS).

A major function of the TE node is to provide tactical systems only that information which the systems would naturally perceive through their sensors or which affect their operational performance. By segregating simulation truth data from tactical system perception, MDSE ensures that a tactical systems performance is not biased by information that would not be available to the tactical system under the simulated scenario, since the tactical system must react to the TE node stimulus as it would to a real-world situation [McQ97].

The MDSE TSD segments accept truth data from the TCS Theater Environment (TE), determine whether the sensors can see the object(s) as scripted in the scenario, and if appropriate, convert the truth data into signals compatible with the systems sensor. The TSD segment consists of the tactical driver, which interfaces between the REs and the tactical system or simulator. The MDA program office responsible for the tactical system development provides the system driver through which MDSE interfaces, stimulates, and controls the test execution for the tactical system.

The system drivers provide the physical interface to the system under test, receiving data from the MDSE TCS segment, and convert the data to a format usable by the tactical system. System drivers may also provide simulations of tactical components of the weapon system or sensor, required by the scenario, but may be missing or unavailable for a HWIL test [McQ97]. The hardware for the development and operational environments includes Sun Ultra's, Sun Servers, Silicon Graphics Indigos, and X-terminals.

APPENDIX H – ELEMENT LEVEL M&S

Table H-1 is a listing of the major Element-level M&S systems to provide a context of the constructive and HWIL Elements M&S systems within the MDA M&S Hierarchy.

BMDS ELEMENT LEVEL CONSTRUCTIVE M&S					
PATRIOT	THAAD	GMS	SMS	ABL	KEB
PAC2SIM	ETESIM	LIDS	JHU-6 DOF	ISAAC	TRITON E2E
PAC3SIM	TISES	SENTRY	KWEVAL	LAMDA	KWEVAL
DDSE	MEDUSA	SSTB	HYDROCODES	SIMPROCESS	
MFSIM	THHADSIM	TESS	MEDUSA	BMAP	
FMS-D	STAT	TEX	SSTRESS	ICONOS	
	TEAM-2	RELEX	ACCSIS	ABLPROP	
	SEEKER	SIMTAS	FIRMTRACK	HITS	
	ASTAB	XBR DIGSIM	ARTEMIS-T		
		UEWR DIGSIM	MARS		
		UTB			
		NMDSIM			
		WPNSIM			
BMDS ELEMENT LEVEL HARDWARE-IN-THE-LOOP (HWIL)					
FMS	SIL	ISTC_1	GSEL	TACCSF	
GTSF	TEC	ISTC_2	RAY HIL/CIL		
MSS-3	TRTB	PCIL	CSEDS		
LMMFC	TTC		ETEDDS		
	TMTD				
[PAT01]	[THA02]	[TCD+00]	[AEG02]	[ABL01]	[KEB02]

Table H-1. MAJOR BMDS ELEMENT LEVEL M&S

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I – THREAT, ENVIRONMENT, SIGNATURE, and LETHALITY (TSEL) MODELS

Appendix I provides brief descriptions of the BMDS System-Level Threat, Signature, Environment, and Lethality M&S. These models support the development and testing of the Element-Level M&S listed in Appendix H. and the BMDS System-Level Core M&S.

A. BMDS SYSTEM-LEVEL THREAT M&S

1. TM93

TM93 is a modeling and simulation architecture used by the SPC to provide intelligence credible representations of threat missile, aircraft, and cruise missile systems. The TM93 Threat Model is the official set of tools used within the Special Program Center (SPC) to generate threat files of ballistic threats and the responsibility for assuring that the threat scenarios used in simulations, studies, and analyses adequately represent the best assessment of the intelligence community.

The SPC has the responsibility for the integration and execution of the various tools used to generate standard threat tapes. The TM93 Threat Model is the set of integrated tools used to produce standard threat tapes and includes tools to allocate strategic and theater ballistic threats and air-breathing threats. The allocation tools feed tools, which generate high fidelity trajectories of all threat objects including boosters, tanks, PBV's, RV's, debris, etc. Post-processing tools perform quality control checks and to feed the final documentation process. All threats include a complete set of hardcopy documentation of the scenario. TM93 provides representations of single events through mass strategic attacks. TM93 models threat trajectories using an ECI coordinate frame, J4 oblate earth model, and standard atmosphere.

The TM93 documentation includes a User's Manual. TM93 uses a UNIX based operating system. However, certain applications only operate on IRIX/Silicon Graphics platforms. A conversion tool exists allowing data conversion to DIS PDUs. TM93 will meet HLA standards in the future.

2. STRATEGIC AND THEATER ATTACK MODELING PROCESS (STAMP)

STAMP is a workstation-based threat generator and engineering analysis tool that models ballistic missile flights from launch to impact. STAMP is a missile flight generator and engineering analysis tool that models missile flight from launch to impact and presents extensive flight characteristics and trajectory descriptions using a wide array of graphical and tabular outputs. It features: an easy to use operator interface using Windows.

STAMP is missile-database driven, developed and approved by intelligence agencies and contains the parameters and values needed to model strategic, theater, and space-launch missiles consistent with intelligence estimates. These databases are in a standard format, read directly into STAMP. The main output from STAMP consists of state vectors in a specified format.

STAMP has three primary applications: (1) missile analysis; (2) scenario generation; and space-launch vehicle modeling. The modeling and analysis tool assists with missile definition development, checkout, and depiction of missile characteristics and flight trajectories to include penaids and sub munitions. The scenario modeling and analysis tool generates simple or massive attack scenarios and provides graphical and statistical characteristics of the scenarios. The space-launch vehicle-modeling tool allows users to specify the desired satellite orbit. STAMP will determine the missile flight path to achieve this orbit.

STAMP has a wide range of missile modeling capabilities that satisfy the needs of most users for accurate and timely missile performance information in functional formats. The plotting and data outputs from STAMP have evolved over several years of threat analysis support and have been continually refined to meet user needs. The primary applications are: (1) Missile analysis, (2) Scenario generation, and (3) Space-launch vehicle modeling.

Documentation includes the STAMP User's Manual, STAMP Design Document, Missile Definition Data Base (MDDDB) Files, Worldwide Ballistic Missile Baseline Performance Characteristics, ACE User's Manual, and NTF Interface Requirements Specifica-

tion. STAMP source code includes Ada and C, and it runs on UNIX platforms (e.g., Sun and Silicon Graphics), with two support graphics packages (XMGR and AXIS).

The following are inputs to STAMP: Missile Definition Data Bases (MDDDB); facility files; scenario specifications; modeling parameters; earth model; Digital Integrated Combat Evaluator (DICE) system files; Modular Vehicle Simulation (MVS) aeronautical data. In addition to the graphic and tabular outputs, STAMP can also generate every object threat files and MEM threat input files.

STAMP links to the following models: TISES (THAAD System Simulation); PTOS (PATRIOT Tactical Operations Simulation); BESIM (Brilliant Eyes Simulation); PERM (Penaid Effectiveness and Requirements Model); MEM (Mission Effectiveness Model); 6 DOF free flight models (ATAP & MVS); Commercial Satellite Analysis codes (STK and OMNI); chemical and biological codes. The Space Warfare Center also uses STAMP to model all ballistic and space-launched missiles for input to their sensor.

B. BMDS SYSTEM-LEVEL SIGNATURE AND ENVIRONMENT M&S

1. SYNTHETIC SCENE GENERATION MODEL (SSGM)

The SSGM generates composite Infrared (IR) scenes, X-band radar cross sections and IR LIDAR cross sections and range/Doppler returns using high fidelity, validated standard component phenomenology models. The SSGM consists of computer software and input databases that provide the capability to generate two-dimensional time-sequenced scenes as well as other supporting data for use in the design, development, and testing of surveillance and weapons systems.

The SSGM computes viewer-perspective apparent-radiance pixel maps and point-source intensity data from the output of state-of-the-science phenomenology codes and authenticated input databases. This SSGM performs via an interactive software system that assists users in creating the scene-sequence scenario and content, selects or generates the required databases, and renders the databases to generate the desired output. In addition, provided tools allows users to display and analyze the output images. The standard wavelength regime extends from the UV to the far infrared, plus the extension to include hard-body-modeling capability at radar frequencies. The space-time samplings are those of the sensor systems and are not otherwise constrained.

The SSGM developers restructured the software in Release 6.0 to make it easier to add and remove capabilities from the overall system. The SSGM uses the concept of a “component” to describe the different phenomenology model capabilities. The SSGM has five types of components: general, foreground, background, laser foreground, and non-imaging. General components include the Observer, Environment and Trajectory; which provide information to the models as required. The various phenomenology models include foreground, background, and laser foreground or non-imaging components.

SSGM radiance output scenes consist of quiescent and enhanced natural and perturbed backgrounds with embedded foreground elements such as targets and target related events. Background components include terrain, horizon, earth limb, aurora, space, and nuclear events. Foregrounds include boosting missile plumes and hardbodies, and post-boost hardbody objects including RVs and decoys. Clouds maybe modeled as either back-

ground or foreground components. The hardbody LIDAR model is the laser foreground component. All radar components are non-imaging.

SSGM contains numerous first-principles physics models. As such, it contains a number of assumptions and approximations. The principal limitation is the spectral coverage and spatial resolution. Most SSGM imaging components cover the 0.2-25 μ m wavelength range. Plumes and nuclear phenomena are restricted to a 2-25 μ m range. Most imaging components support a spatial resolution of $\geq 10^{-9}$ radians. Celestial backgrounds are limited to $\geq 4 \times 10^{-5}$ radian resolution. Terrain and cloud databases have inherent resolutions that vary from 30 - 18500 m and 30 - 1000 m, respectively. These effectively bound sensor footprint resolution. Primary applications include:

- Sensor Development
- Sensitivity & Trade-Off Analyses
- Threat Signature Development
- Nominal Threat Signature, Launch through impact
- Realistic range and distribution of signatures
- Algorithm Development
- Fusion
- Detection, discrimination and tracking
- Data Analysis
- Test and Evaluation
- Test planning and excursion evaluation
- Hardware-in-the-loop simulations and emulators
- High-level simulations
- Support for Wargames: signatures and backgrounds
- System Architecture Studies

The SSGM main architecture shown in Table I-1 includes OOA/OOD techniques and C++. Phenomenology component codes include a mix of FORTRAN, C and C++. Interface between the component codes and SSGM involves C++ “wrappers”. The GUI written in C++, uses X-Windows and X-Motif call. Component model source code is generally not available for modifications, as any modification might negate the validation status of the model. The Hardware includes Silicon Graphics workstations or servers running IRIX.

Scenarios can take seconds to days to execute to run to completion. This is principally a function of the number and complexity of the targets, types of background (e.g., clouds, earthlimb), phenomenology models selected and number of scenes to be generated.

Current Inputs include Scenario Description File (all other necessary databases are internal). This includes sensor, target and engagement description, but does not include specific target trajectories. Current Outputs include

- At-aperture radiance scenes
- Target and plume radiant intensities
- Target and sensor metrics
- Target heating and ablation history
- Detailed information on atmosphere, clouds, and terrain

Atmosphere		Nuclear	
SHARC MODTRAN SAMM SAG APART MOSART FASTLIMB CIRRIS-1A		IRSim HiSEMM PEM	
		Celestial	
		IRAS Star Catalog CBSKY4 CBZODY CBAMP	
		TROs	
Clouds		OSC OPTISIG Xpatch DELTAS	
CLDSIM			
Aurora			
AURORA		Missile Plume	
Terrain			
GENESSIS			
Vents/Spills	Debris	SIRRM CHARM CAMERA SPF-2	
Fuel Vent Vent-Trail	KIDD-OPTISIG DEBRA-KIDD		

Table I-1. SSGM Constituent Model Matrix

2. THE BATTLESPACE ENVIRONMENT SIGNATURE TOOLKIT (BEST)

The Battlespace Environment Signature Toolkit (BEST) program will be developed to replace the legacy Environment and Signature models shown in Table I-1 and Table I-2, establishing the foundation for the next-generation missile defense Scene Generation capability for future advanced sensor development based on requirements identified by [CKB+94, Hec99, Hie99, SB99, Ber00a, BMC+00, BSB+00, Bur00, BVW+00, CWM+00, DKC+00, DNC+00, DSH00, DWE+00, FAK+00, KMM+00, MSB+00, PB00, SST+00]. The MDA awarded a contract in June 2002 for the development of the Battlespace Environment and Signatures Toolkit (BEST) development program supporting the design, development, testing and evaluation of active and passive sensors, signature processing, and algorithms with empirical and physics-based signature phenomenology and battle space environment codes for all three phases of missile flight.

Meeting the requirement to counter the full range of future potential threats, the MDA developed an Adversarial Capability methodology which will be introduced into the MDA System Core Models, augmenting the existing DIA validated Design-to-Threat, providing BMDS system engineers with new capabilities for trade space analysis.

BMDS Legacy Environment and Signature Codes		
Model/Code	Function	Executing Agent
Target Related Hardbody Objects		
OSC	Resolved Passive EO/IR Signatures	SMDC
OPTISIG	Unresolved Passive EO/IR Signatures	SMDC
XPATCH	Radar Signatures (Physical Optics)	AFRL/SN
FISC	Radar Signatures (Method of Moments)	AFRL/SN
RTS	Radar Signatures	NRL
DELTAS	Laser Retroreflectance	SMDC
Missile Plumes & Trails		
SFM, VIPER, TDK	Flowfield, Nozzle & Combustion	AFRL/PR
SPF, GIFS	Complex Plume Flowfield	AFRL/PR
PERCORP,SPP	Liquid & Solid Motor Performance	AFRL/PR
SIRRM	Plume Radiance (< 50 km)	AFRL/PR
CHARM	Plume Flowfield & Radiance (> 50 km)	AFRL/PR
SPURC	Plume UV/VIS Radiance (> 50 km)	AFRL/PR
SOCRATES	Monte Carlo Flowfield & Gas Dynamics	AFRL/PR
HALT, PARCS	Laser backscatter & Plume RCS	AFRL/PR

Battlespace Environment		
MODTRAN	Atmospheric Transmission & Radiance (<50 km)	AFRL/VS
SHARC	Quiescent Atmosphere, Auroral & Terminator Radiance and Clutter (>50 km)	AFRL/VS
SAMM	Atmospheric Transmission & Radiance (all altitude)	AFRL/VS
SAG, MSIS, BEAM, SIG	Atmospheric State and Clutter	AFRL/VS, NRL & MSIC
CBSD	Planets, Moons, Asteroids, Zodiacal Emission, Stars	AFRL/VS
GENESSIS	Terrain & Ocean Surface Radiance	AFRL/VS
CLDSIM	Cloud Radiance, Transmission, & Glints	AFRL/VS
IRSIM, HiSEMM, NucSim, NORSE	Nuclear Perturbed Atmosphere	DTRA
Trajectory		
STAMP	Strategic Missile Trajectories	DIA/NAIC
DICE	Theater Missile Trajectories	DIA/MSIC
Integrated Models		
SSGM	Targets, Trajectories, and Environments	NRL
PLEXUS	Expert System for Atmospheric & Celestial Phenomenology	AFRL/VS
CHAMP	Missile Plume & Hardbody	AFRL/MN

Table I-2. BMDS Signature and Environment Functions

Because of the distributed nature of BMD weapon systems and the critical importance of each element to achieve overall system effectiveness, interoperability and collective system performance assessments are an important area of evaluation. All BMD sensor and weapon systems require signature information upon which to base decisions. Existing MDA signature codes have served the engineering analysis community over the years, but they are difficult and expensive to modify, difficult for the user to learn and operate correctly, and most importantly cannot effectively support the future M&S needs of MDA. Present and anticipated requirements will demand greater interoperability, consistency, and faster execution speeds..BEST will enhance signature tools and complete the phenomenology set effectively providing a robust signature toolkit for all MDA signature needs. It will:

- Provide a more complete representation of battlespace signatures.
- Ensure consistency in signature results thereby assisting verification and validation and accreditation.
- Meet the future challenge of simulation-driven design, evaluation, and acquisition.
- Provide users the flexibility they require
- Support interoperability (within the signature codes and with external simulations).

BEST will replace the patchwork assemblage of signature codes with a consistent, modular code that is accessible through one interface. BEST will support the entire range of missile defense signature needs. The modular nature of BEST will support plug-and-play of functionally equivalent components. This will permit contractors to insert proprietary components and to even use the components delivered with BEST in other applications. The following illustration depicts the component-based structure of BEST.

A secondary goal of BEST is to provide the government with signature tools as part of a suite of government-approved tools to independently evaluate and assess contractor performance.

C. BMDS SYSTEM-LEVEL LETHALITY M&S

The BMD System lethality model program consists of PEELS and PEGEM as a suite of tools continually evolving to identify any residual threat after a successful intercept. The BMD System lethality programs directly support the development of the MDA element level M&S and system level testing.

1. POST ENGAGEMENT GROUND EFFECTS MODEL (PEGEM)

The Post-Engagement Ground Effects Model (PEGEM) is a simulation tool to study ground effects caused by chemical, biological, and conventional weapons/agents distributed in both unitary (bulk) and submunition (canister or bomblet) form. The model provides chemical/biological agent ground contamination and conventional weapons blast zones. It also provides data for unit effectiveness and casualty estimation.

PEGEM is a suite of models to produce a simulation incorporating various phases of ground effects assessment, such as Munition Propagation and Atmospheric Transport. In a typical case, the analyst specifies a chemical or biological weapon event scenario including all threat details and the locations and times of the various events on the user-defined grid. Intercept lethality information flows from the output of an endgame lethality model. The lethality model provides PEGEM a prediction of the fraction of payload surviving following an intercept event. For canister submunition payloads, the locations of surviving submunitions within the target payload are given. PEGEM uses this information to propagate the potential residual threat(s) to the ground.

PEGEM documentation includes a User's Manual, Technical Manual, and other supporting documentation. Written in ANSI-C, PEGEM runs on Windows 95, Windows NT and UNIX

2. PARAMETRIC ENDO- EXO-ATMOSPHERIC LETHALITY SIMULATION (PEELS)

PEELS is a high fidelity terminal endgame lethality simulation, which models hit-to-kill and fragment kill of ballistic missile from kinetic energy weapon attack. It operates as a non-real-time simulation, intended to support weapon system designers as well as system level analysts. The models primary purpose is to analyze lethal effectiveness of non-nuclear kill interceptors in an anti-tactical ballistic missile role.

PEELS is a terminal endgame lethality simulation, which models hit-to-kill (HTK) and fragment kill of ballistic missiles from Kinetic Energy Weapons (KEW) attack. PEELS predicts damage, probability of kill (P_k), and submunitions kill fraction dependent on target design using lethality criteria developed for a number of threats. The design facilitates force-on-force systems studies where six-degree-of-freedom (6-DOF) fly out models predict endgame state vectors.

Algorithms in the PEELS code also allow parametric investigations of lethality through variation of endgame variables such as miss distance, crossing angle, warhead burst point, intercept altitude, target and interceptor velocity and angles of attack, target aim point, and fragment pattern density. The model evaluates HTK and fragment kill effects against both high explosive (HE) and chemical unitary warheads, and against both HE and chemical submunitions warheads. Primary applications areas include: Parametric Feasibility Analysis Studies—warhead concept evaluation; System Assessment—scenario specific lethality coupled with a 6-DOF model; Pre-Test Analysis—help define test matrix—identify stressing cases to be tested.

Documentation includes: An Executive Summary and an eight-volume set of documentation. Volume 0, Executive Summary; Volume I, PEELS Technical Manual (Algorithm and model detail); Volume 2, PEELS Targets Manual and Volume 3, TMD Lethality Criteria (Information describing the geometric, structural, and payload design of the targets and lethality criteria); Volume 4, PEELS Validation and Verification Manual; Volume 5, User's Manual; Volume 6, Accreditation Report; Volume 7; Maintenance and Distribution Manual; Volume 8, Source Code Manual.

PEELS runs on SGI UNIX based system, VAX, or PC systems, and programmed in FORTRAN-77. PEELS operates in two basic modes of operation: 1) the *Database Mode* and 2) the *Direct Mode*. The *Database Mode* uses a pre-processor to create databases accessed during execution, for rapid system level applications where lethality is only one part of many computations. The *Direct Mode* runs if the target model does not exhibit axial-symmetry or the desired fragment type is not in the database. This typically requires four to five times more computer processing time.

Database Mode requires a series of databases accessed during execution. These contain information about the solid geometry of the target as well as parameters associated with fragment lethality. An *Index Database* composed of shot-line intersection with internal target components maintains data considered critical to target lethality. The index database points to a Lethality Database. The lethality database maintains empirical data. This contains information that defines the minimum fragment material and energy characteristics necessary to achieve lethality along each shot line.

APPENDIX J – THE DOMAIN METADATA REPOSITORY REGISTRY (DMRR)

A. COMMON DMRR METADATA FACILITIES

From [ISO79c]³⁶¹ we extract the metadata facilities. This is a critical component for achieving Registry interoperability with the domain, inter-domain integration, and at the enterprise level. The common DMRR facilities illustrated in Figure J-1 support administered items, identified once within the registry. The named administered items have at least one context, maybe more, and within each context, names and definitions specified in one or more languages. The administered items have classification in zero or more classification schemes [ISO79c].

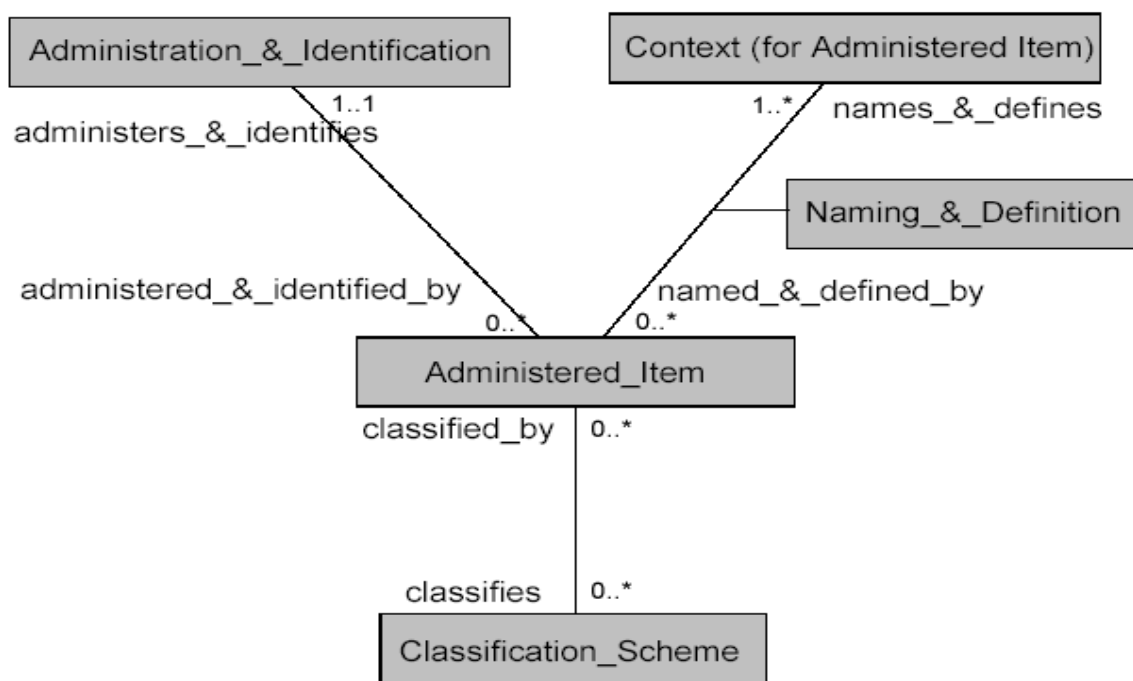


Figure J-1. BMDS DMRR Common Facilities (From [ISO79c])

³⁶¹ © International Organization for Standardization (ISO). This material is reproduced from ISO/IEC 11179-3, Second Edition, 2003-02-15 with permission of the American National Standards Institute on behalf of ISO. No part of this material may be copied or reproduced in any form, electronic retrieval system or otherwise or made available on the Internet, a public network by satellite or otherwise without the prior written consent of the American National Standards Institute, 25 West 43rd Street, New York, NY, 10036.

B. DMRR TYPES OF ADMINISTERED

[ISO079c] specifies and describes the types of Administered Items³⁶² listed in Figure J-2, and allows definition of additional types of Administered Items.

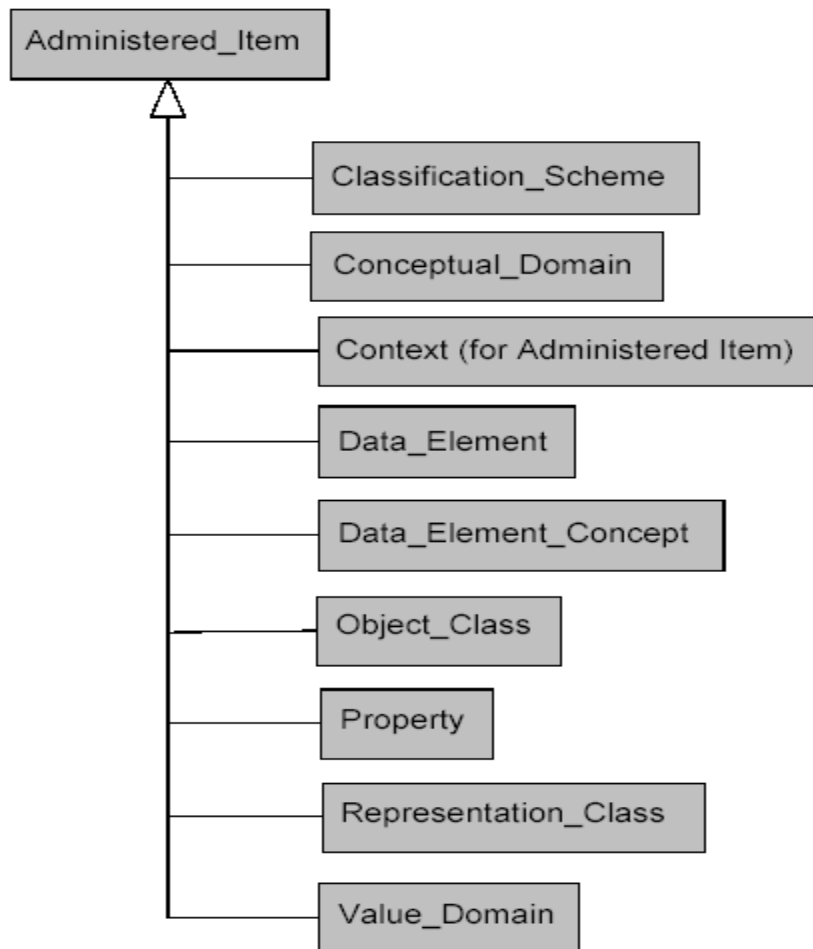


Figure J-2. BMDS DMRR Types of Administered Items (From [ISO79c])

³⁶² An Administered Item may be one of the types listed in Figure J-2. Each instance of the Administered Item encapsulates its own Administration Record. An Administered Item is submitted by an Organization represented by the relationship *submission* in Figure J-4. A Registration Authority registers Administered Item represented by the relationship *registration* in Figure J-4. An Organization administers the Administered Item represented by the relationship *stewardship* in Figure J-4. A Reference Document may describe zero or more Administered Item represented by the relationship *reference* in Figure J-4. Each instance of the Administered Item through an Administered Record has a unique Administered Item Identifier [ISO97c].

C. DMRR HIGH-LEVEL METAMODEL

Figure J-2 provides a high-level overview of the central regions of the metamodel. [ISO79c]

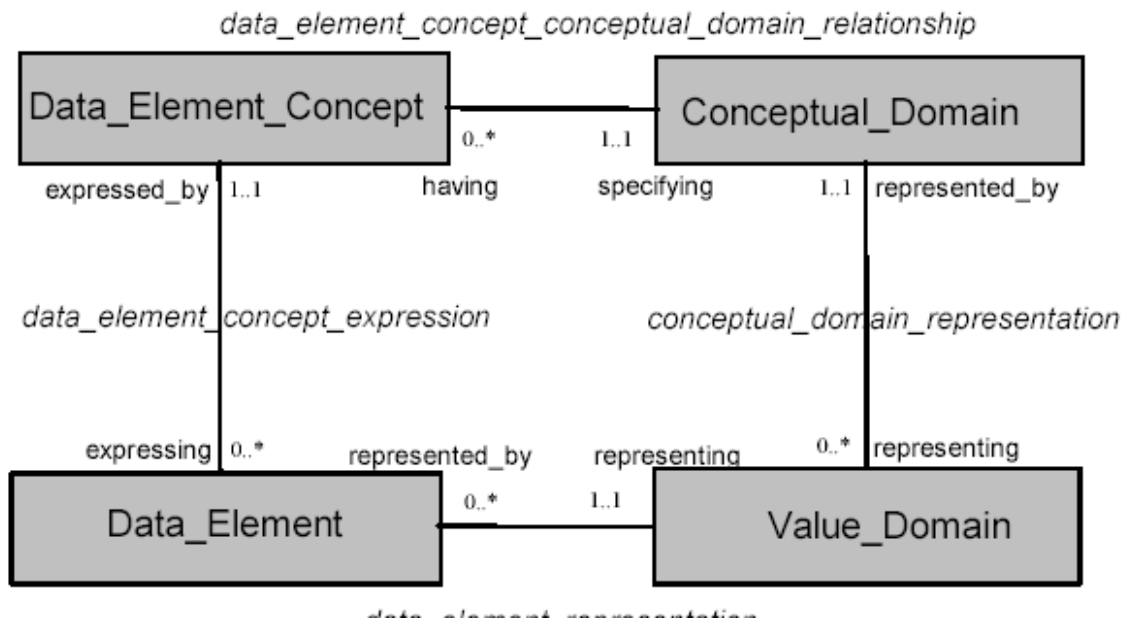


Figure J-3. BMDS DMRR High-Level Metamodel Overview (From [ISO79c])

D. DMRR ADMINISTRATION AND IDENTIFICATION METAMODEL REGION

The Administration and Identification in Figure J-4 and Figure J-5 supports the administrative areas of the Administered items in the DMRR including:

- Identification and registration of items submitted to the registry,
- Organizational information and responsible agencies, including Registration Authorities³⁶³,
- Organizational contact information,
- Supporting documentation,
- Relationships among administered items [ISO79c]

³⁶³ A Registration Authority is any organization authorized to register metadata. A Registration Authority is a subtype of organization and inherits all of its attributes and relationships. An Administered Item has a Registration Authority that is its owner, shown by the relationship in Figure J-4. A Registration Authority may register many Administered Items [ISO97c].

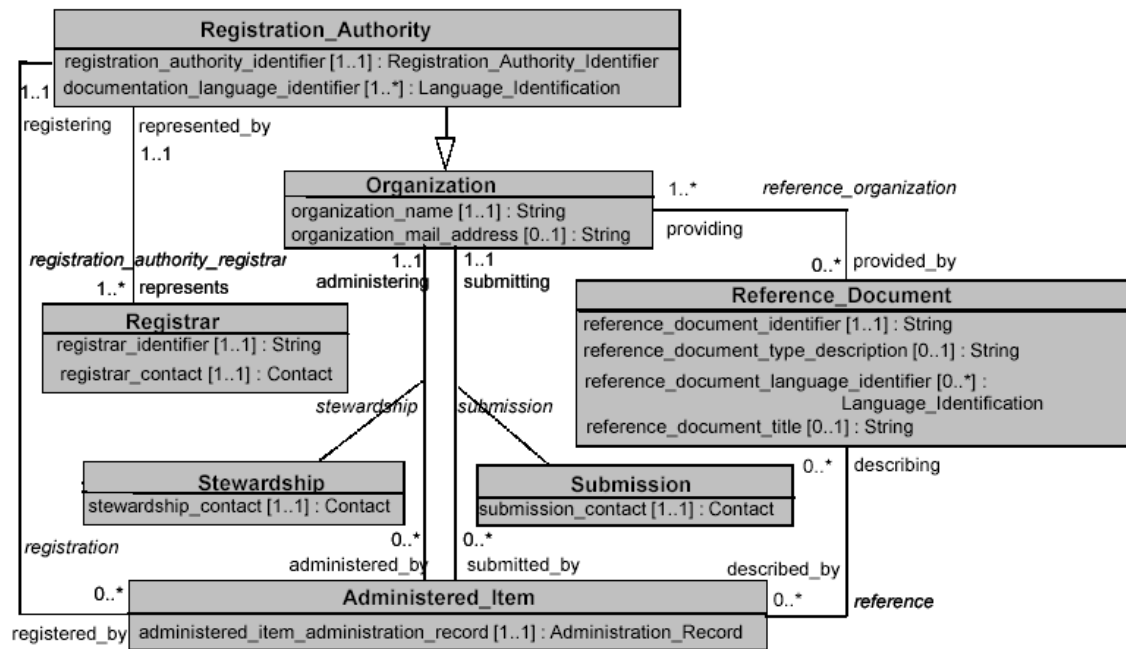


Figure J-4. BMDS DMRR Admin / Identification Metamodel Region (From [ISO79c])

E. DMRR CLASSES USED AS COMPOSITE DATATYPES

[ISO97f] prescribes the registration of Administered Items. Figure J-5 shows the composite data types used on composite attributes.

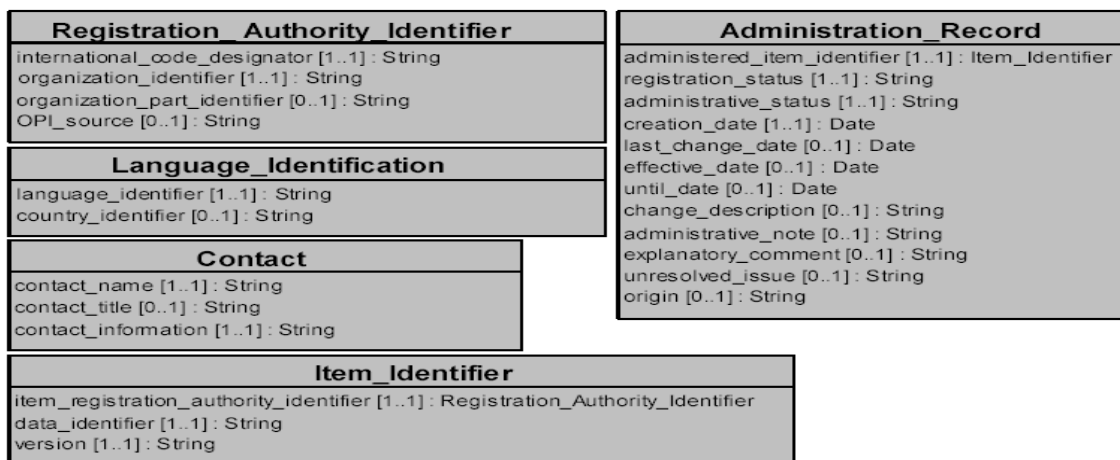


Figure J-5. BMDS DMRR Classes / Composite Datatypes in the Administration and Identification Region (From [ISO79c])

F. DMRR NAMING AND DEFINITION METAMODEL

Figure J-6 represents the Naming and Definition Region used to manage the names and definitions of administered items and the contexts for the names. An administered item may have many names that will vary based on discipline, locality, and technology employed [ISO79c]. [ISO79d]³⁶⁴ provides rules and guidelines for the formulation of data definitions, and [ISO79e] provides naming and identification principles for administered items within a context.

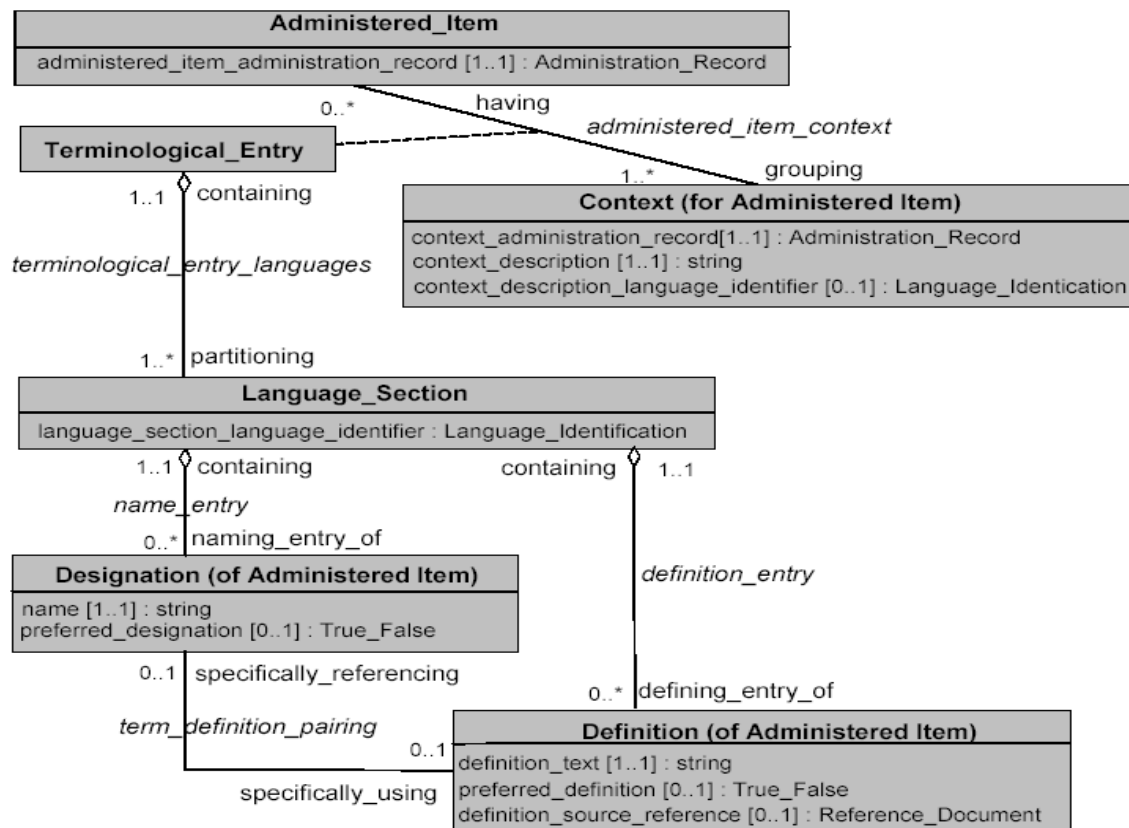


Figure J-6. BMDS DMRR Naming and Metamodel Region (From [ISO79c])

³⁶⁴ © International Organization for Standardization (ISO). This material is reproduced from ISO/IEC 11179-4, First Edition, 1996-07-01 and ISO/IEC 11179-5 First Edition, 1995-12-01 with permission of the American National Standards Institute on behalf of ISO. No part of this material may be copied or reproduced in any form, electronic retrieval system or otherwise or made available on the Internet, a public network by satellite or otherwise without the prior written consent of the American National Standards Institute, 25 West 43rd Street, New York, NY, 10036.

G. DMRR CLASSIFICATION METAMODEL REGION

The Classification Region illustrated in Figure J-7 provides a facility to register and administer *Classification Schemes*³⁶⁵ and their constituent *Classification Scheme Items*. In addition, a *Classification Scheme* may classify *Administered Items* within the registry. Some *Classification Schemes* will be more applicable to classifying objects in the real world than classifying metadata objects in the registry [ISO79c].

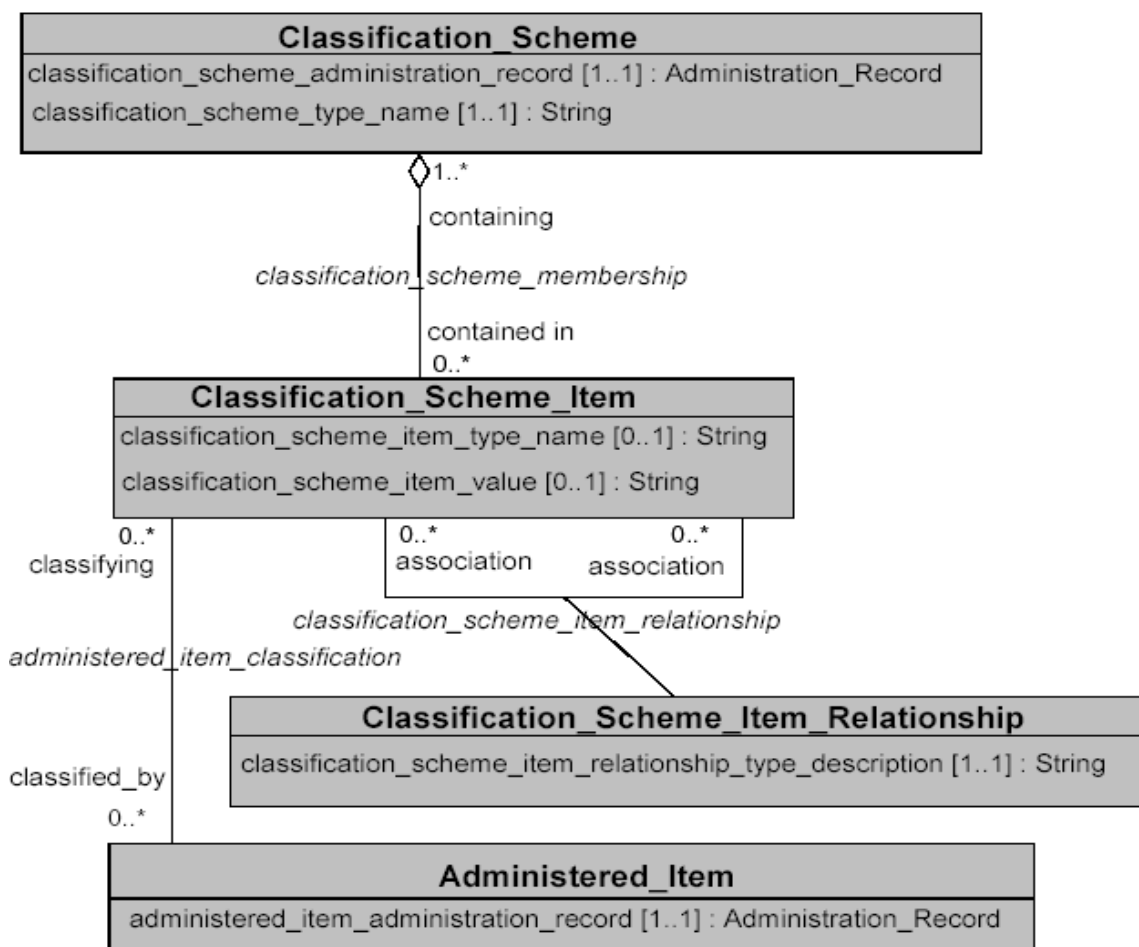


Figure J-7. BMDs DMRR Classification Metamodel Region (From [ISO79c])

³⁶⁵ A *classification scheme* may be a taxonomy, a network, an ontology, or any other terminological system; or the classification scheme may consist of a list of controlled vocabulary words and terms. A classification scheme is a sub-type of Administered Item, inheriting its attributes and relationships, supporting identification, naming, definition, and classification functions [ISO79c].

H. DMRR DATA ELEMENT CONCEPT METAMODEL REGION

The data element concept region illustrated in Figure J-8 maintains the information on the concepts upon which the data elements are developed. The metadata objects in this region concentrate on semantic concepts. The metadata objects in this region are object classes and properties, combined to form *Data Element Concepts*. The concepts are independent of any internal or external physical representation. [ISO79c]. A Data Element Concept may be represented in the form of a data element, described independently of a particular representation, with zero or one object class and zero or one *property*. A property³⁶⁶ is a characteristic common to all members of an object class.

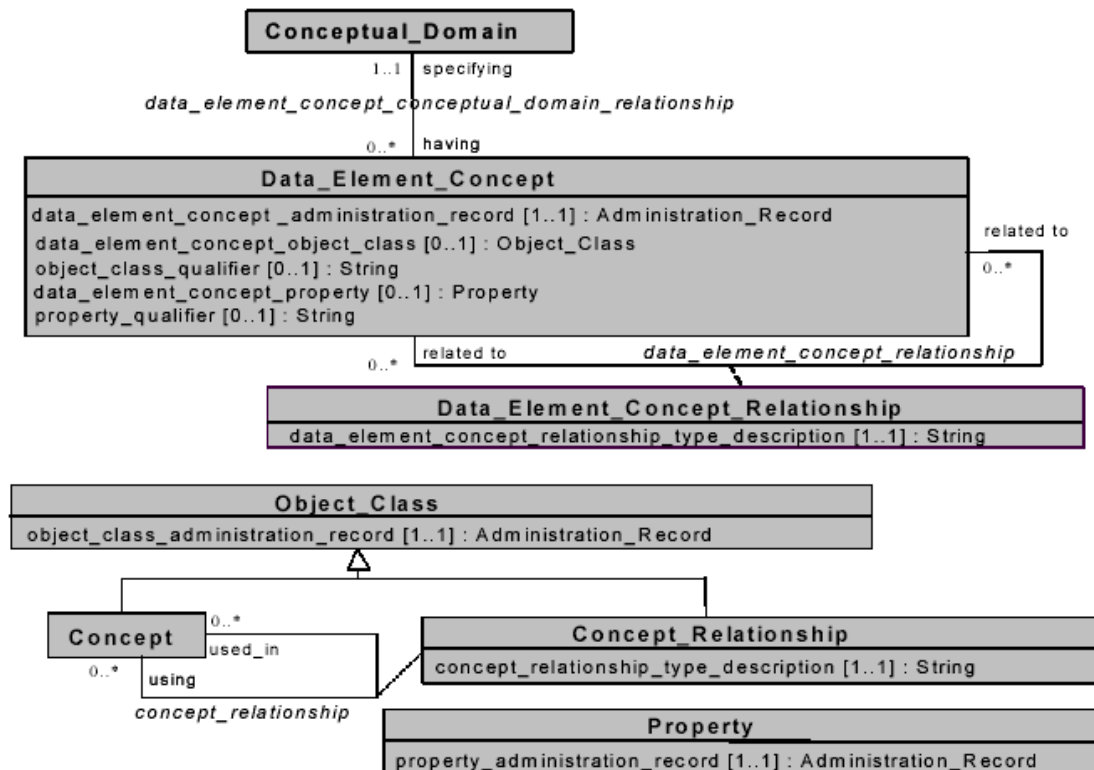


Figure J-8. BMDS DMRR Conceptual Metamodel Region (From [ISO79c])

³⁶⁶ A property may be any feature that humans naturally use to distinguish one individual object from another. It is the human perception of a single characteristic of an object class in the real world. It is conceptual without an associated means of representation for communication [ISO79c].

I. DMRR CONCEPTUAL AND VALUE DOMAIN METAMODEL REGION

The region of the metamodel in Figure J-9 supports the administration of the *Conceptual Domains* and *Value Domains*, viewed as logical code sets and physical code sets. Conceptual Domains support *Data Element*³⁶⁷ *Concepts*³⁶⁸, and Value Domains support Data Elements as illustrated in Figure J-9. A Conceptual Domain is a set of Value Meanings, which may be expressed or enumerated by a description, and as an Administered Item maintains *Administrative Record* information, allowing identification, naming, definition, and optional classification within a *Classification Schema* [ISO79c].

A Conceptual Domain may be associated with other *Conceptual Domains* through a *Conceptual Domain Relationship*, either as a composition with other Conceptual Domains, or as a component member of a larger Conceptual Domain. A Conceptual Domain may specify a constraint such as a linear measure as its dimensionality. A specified dimensionality requires that any Value Domain based on the Conceptual Domain specify a *Unit of Measure*³⁶⁹. An Enumerated Conceptual Domain, a sub-type of Conceptual Domain may contain a finite number of enumerated notions (e.g., codes representing the names of states). Conversely, a Non-enumerated Conceptual Domain lacks a finite set of Value Meanings. As a sub-type of the Conceptual Domain, both the Enumerated Conceptual Domain and Non-enumerated Conceptual Domain inherit the attributes and relationships of the former [ISO79c].

In an Enumerated Conceptual Domain, each member has a *Value Meaning*, in the registry distinguishing it from other members, and is independent of their representation in any corresponding *Value Domain*³⁷⁰. A specific *Value Meaning* may have more than one

³⁶⁷ A data element is a basic unit of data of interest to an organization. It is a unit of data defined, identified, represented, and contains permissible values specified by means of a set of attributes [ISO79c].

³⁶⁸ Data Element Concepts is a concept represented in the form of a data element, and described independently of any specific representation. It may have zero or one Object Class and zero or one Property [ISO79c].

³⁶⁹ A *Unit of Measure* is the unit associated when Data Element values are specified. The Unit of Measure Name designates the unit. When specified the unit must maintain consistency with any dimensionality specified in the corresponding Conceptual Domain. Optionally, a *Unit of Measure* may have specifications for the number of decimal spaces supported in the associated Data Element values. The precision is a default that may be overridden for any particular data element [ISO79c].

³⁷⁰ A Value Domain is a key component of a Conceptual Domain, providing representation for the Conceptual Domain. An example of a Value Domain is ISO 3166 describing the codes of the representation of countries within a set of seven Value Domains: short English name, official English name, short French name, official French name, alpha-2 code, alpha-3 code, and numeric code [ISO79c].

means of representation by *Permissible Values*³⁷¹, from a distinct *Enumerated Value Domain*³⁷². A Value Domain may be associated with other *Value Domains* through a *Value Domain Relationship*, either as a composition with other Conceptual Domains, or as a component member of a larger Conceptual Domain, using the *value domain relationship type description* [ISO79c].

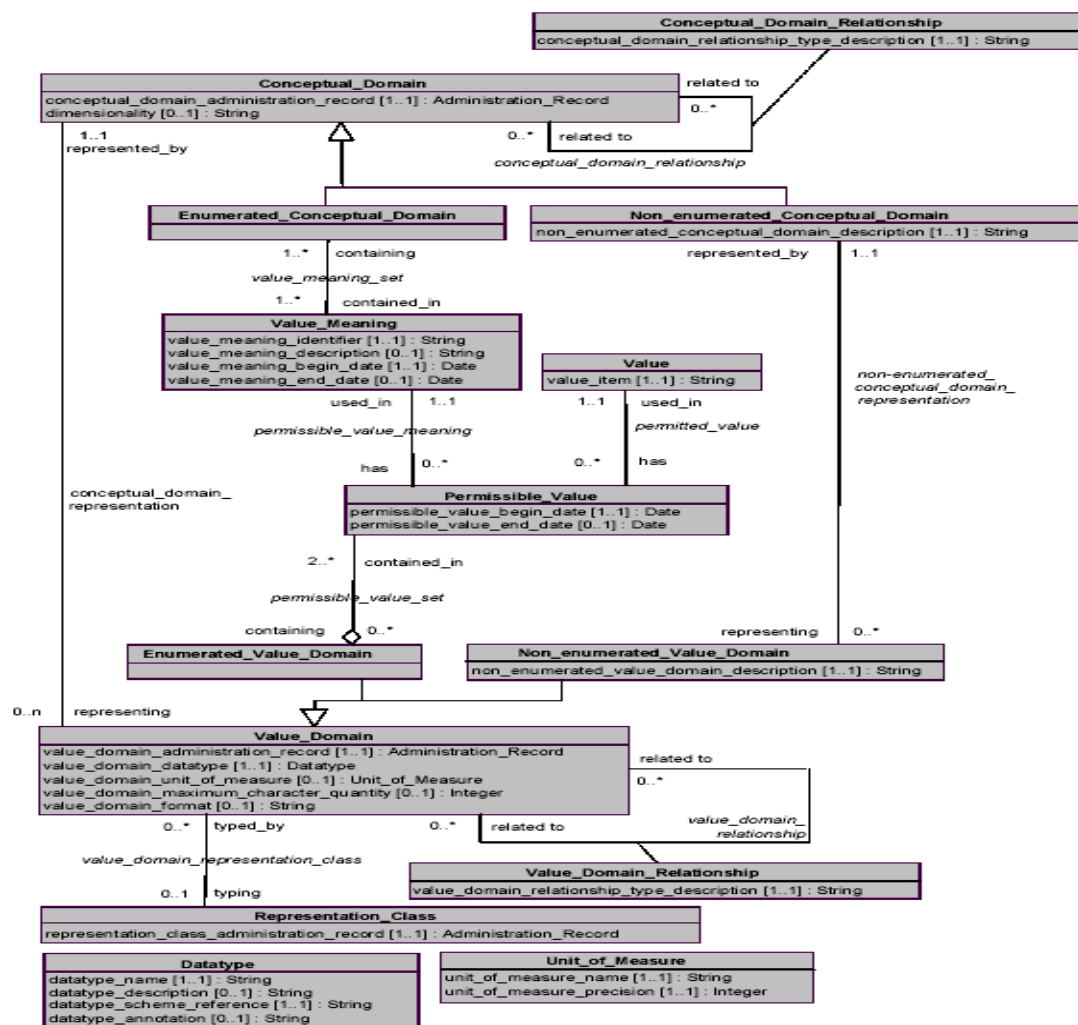


Figure J-9. Conceptual And Value Domain Metamodel Region (From [ISO79c])

³⁷¹ A permissible value is an expression of a *Value Meaning* within an *Enumerated Value Domain* and one of a set of values comprising an *Enumerated Value Domain*. Each permissible value is associated with a value meaning [ISO79c].

³⁷² An Enumerated Value Domain is an expression of an explicit set of two or more permissible values, and a Non-Enumerated Value Domain is a value domain expressed by a description, specification or rule, rather than a set of permissible values. Both the Enumerated Value Domain and the Non-Enumerated Value Domain are a sub-type of the value domain and inherit the attributes and relationships of the value domain [ISO79c].

J. DMRR DATA ELEMENT METAMODEL REGION

The Data Element metamodel region, shown in Figure J-10, addresses the administration of data elements. Data Elements provide the formal representation of information (e.g., a fact, a preposition, an observation) about some concrete or abstract thing. Data Elements are reusable and sharable representations of Data Element Concepts. When a Data Element Concept receives a value, a Data Element forms. A Data Element is the association among a Data Element Concept, a Value Domain, and a Representation Class [ISO79c].

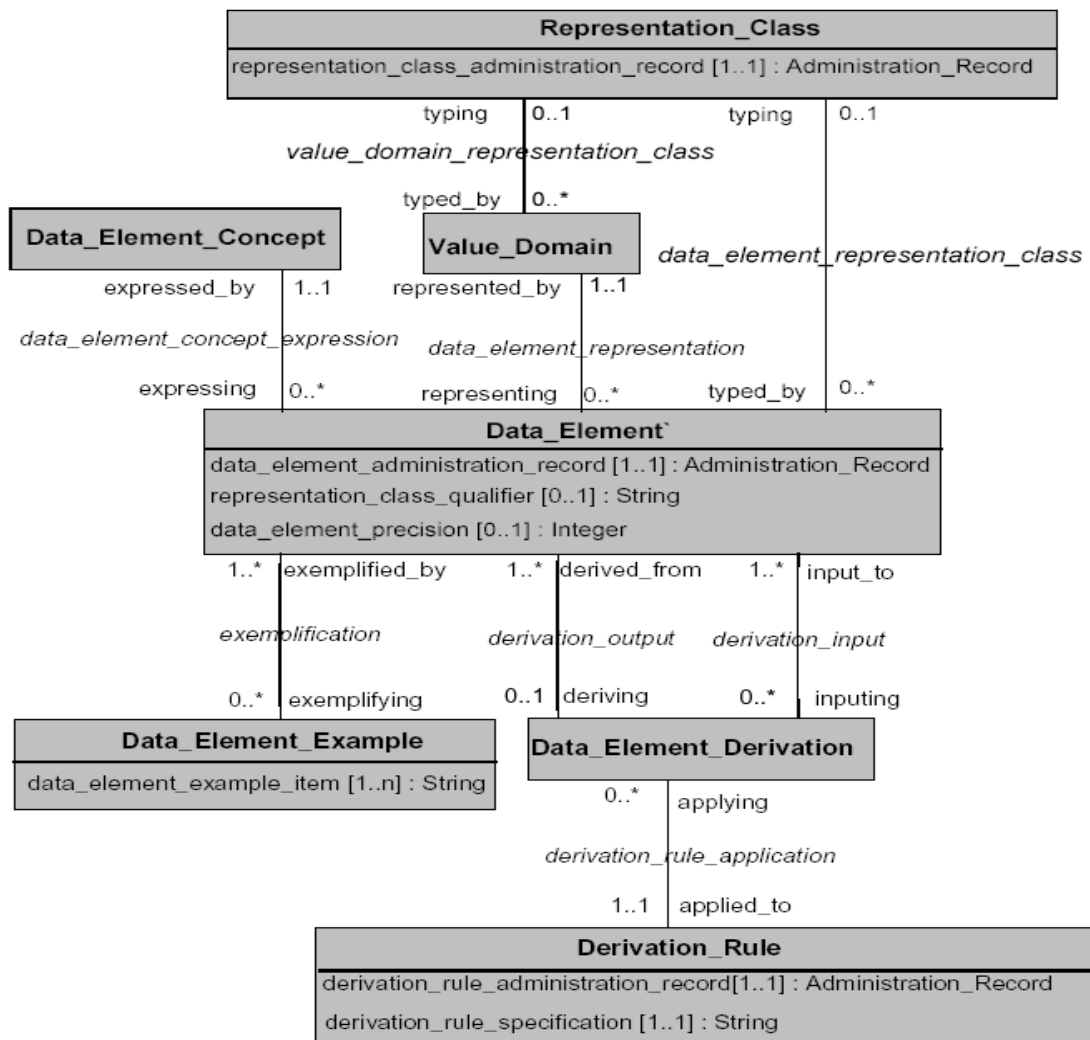


Figure J-10. BMDS DMRR Data Element Metamodel Region (From [ISO79c])

The Data Element to Representation Class association may occur directly or by way of the Value domain. Registration of a Data Element as an Administered Item requires association with a Data Element Concept and a Value Domain. A representation class qualifier, if specified, qualifies the name of the data element. Data Element Precision specifies the number of decimal spaces permitted by any associated data element values. Unless specified differently, the unit of measure precision from the associated Value Domain applies. A data element may also have Data Element Examples³⁷³ and a Derivation Rule³⁷⁴ [ISO79c].

A Data Element may have associations with several Value Domains, resulting in a different Data Element Concept for each association. The Value Domain provides representation, but no links to the Data Element Concept values, or what the values mean. A Value Domain may associate with multiple Data Elements. The Representation Class³⁷⁵ is the Classification Scheme for representation, supporting a set of classes to distinguish among the various elements in the registry. Objectives of the Representation Class include providing a complete, discrete set of high-level definitions for data element/value domain categorization, supporting the integration of business rules for applications. Representational Class use enhances semantic control over contents of value domains, supporting the development of rules for representation classes allowing the enforcement of content within and among domain values [ISO79c]. [ISO79e] and [ISO79c] provide an informational list or representational class terms.

K. DMRR CONSOLIDATED METAMODEL

A consolidated metamodel in Figure J-11 illustrates the combined Data Element Concept, Data Element, Conceptual Domain, and Value Domain of the [ISO79c] Model.

³⁷³ Data Element Examples provide representative samples of the Data Element [ISO79c].

³⁷⁴ A Derivation Rule specifies a derivation for the Data Element [ISO79c].

³⁷⁵ A Representation Class is a mechanism for conveying the functional and/or presentational category of an item to a user [ISO79c].

LIST OF REFERENCES

- [AB02] Asklund, U., Bendix, L., "A Study of Configuration Management in Open Source Software Projects," *IEEE Proceedings on Software*, Vol. 149, Issue 1, February 2002, pp. 40-46.
- [ABB00] Anderson, R., Brackney, R., Bozek, T., *Advanced Network Defense Research*, RAND National Defense Research Institute, Santa Monica, CA, August 2000.
- [ABB+02] Atkinson, C., Bayer, J., Bunse, C., *Component-based Product Line Engineering with UML*, Addison-Wesley, London, England, 2002.
- [ABC+97] Abowd, G., Bass, L., Clements, P., Kazman, R., Northrop, L., Zaremski, A., *Recommended Best Individual Practice for Software Architecture Evaluation*, (CMU/SEI-96-TR-025), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, January 13, 1997.
- [ABC00] Abts, C., Boehm, B., Clark, E., *COCOTS: A COTS Software Integration Lifecycle Cost Model – Model Overview and Preliminary Data Collection Findings*, USC Center for Software Engineering, Los Angeles, CA, 2000.
- [ABD+93] Anderson, R., Bankes, S., Davis, P., *Toward a Comprehensive Environment for Computer Modeling, Simulation, and Analysis*, RAND Corporation, Santa Monica, CA, 1993.
- [ABL01] ABL/SPO, *BMDO Airborne Laser (ABL) Program Development and Risk Reduction Test and Evaluation Master Plan*, (Draft) MSC/TMT (ABL/SPO), Kirtland AFB, NM, October 2001.
- [ABM00] Atkinson, C., Bayer, J., Muthig, D., "Component-Based Product Line Development: The Kobra Approach," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 289-310.
- [ACC+94] Allgood, B., Clough, A., Cunha, G., *Process Technologies Method and Tool Report (Volume I)*, Software Technology Support Center, Hill AFB, UT, March 1994.
- [ACD+02] Aversano, L., Canfora, G., DeLucia, A., "Integrating Document and Workflow Management Tools using XML and Web Technologies: A Case Study," *Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02)*, IEEE Computer Society, 2002.

- [ACH+02] Andrew, C., Chaplain, C., Hazard, M., *Capturing Design and Manufacturing Knowledge Early Improves Acquisition Outcomes*, GAO-02-171, United States General Accounting Office, Washington, DC, July 2002.
- [ACP+99] Amos, P., Coyle, G., Purdy, E., *Acquisition Modeling and Simulation Comprehensive Core Body of Knowledge*, Ver. 1.0, Acquisition Functional Working Group, Acquisition Council of the DoD Executive Council for Modeling and Simulations (EXCIMS), Washington, DC, December 23, 1999.
- [ADD+00] Ardis, M., Dudak, P., Dor, L., "Domain Engineered Configuration Control," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 479-494.
- [Ade01] Adelantado, M., "Multiresolution Modeling and Simulation of an Air-Ground Combat Application," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30 2001.
- [AEG02] Aegis BMD Program, *Aegis Ballistic Missile Defense Modeling and Simulation Plan*, (Draft) Ver 2.0, Aegis BMD Program Office, July 31, 2002.
- [AFI00] Department of the Air Force, *Modeling and Simulation (M&S) Support to Acquisition*, AF Instruction 16-1002, Department of Defense, Washington, DC, June 1, 2000.
- [AFW+94] Anding, D., Franklin, T., Wilson, D., *Methodology for Incorporation of Environmental Effects into DIS*, accessed on 7 February 2002 from http://vader.nrl.navy.mil/e2dis_www/.
- [AGJ95] Almeida, R., Graver, D., Jacobs, P., "Simple Probability Models for Assessing the Value of Information in Defense Against Missile Attack," *Warfare Modeling*, John Wiley & Sons, 1995, pp. 393-405.
- [AH97] Arnette, J., Hamilton, J., "Open Architectures for Diverse Missions: Opening Windows for the Department of Defense," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, November, 1997.
- [AIA93] American Institute of Aeronautics and Astronautics, *Guide for the Preparation of Operational Concept Documents*, ANSI/AIAA G-043-1992, American Institute of Aeronautics and Astronautics, Washington, DC, 1993.
- [AKN02] Akkanen, J., Kiss, A., Nurminen, J., "Evolution of a Software Component Experience with a Network Editor Component," *Proceedings of the Sixth*

European Conference on Software Maintenance and Reengineering (CSMR '02), IEEE Computer Society, 2002.

- [Ald02a] Aldridge, E., *Cost-as-an-Independent Variable (CAIV) and Spiral Development Implementation Plans*, Under Secretary of Defense memorandum, Washington, DC, January 19, 2002.
- [Ald02b] Aldridge, E., *Performance Based Logistics*, Under Secretary of Defense memorandum, Washington, DC, February 13, 2002.
- [Ald02c] Aldridge, E., *Missile Defense Support Group*, Under Secretary of Defense memorandum, Washington, DC, February 13, 2002.
- [Ald02d] Aldridge, E., *Evolutionary Acquisition and Spiral Development*, Under Secretary of Defense memorandum, Washington, DC, April 12, 2002.
- [Ald02e] Aldridge, E., *Ballistic Missile Defense Program Implementation Guidance*, Under Secretary of Defense memorandum, Washington, DC, February 13, 2002.
- [Ald02f] Aldridge, E., *Continuous Learning Policy for the Department of Defense Acquisition, Technology, and Logistics Workforce (DoD AT&LWF)*, Under Secretary of Defense memorandum, Washington, DC, September 13, 2002.
- [Ald03] Aldridge, E., *Missile Defense: Past to Future*, Under Secretary of Defense addresss, Washington, DC, February 13, 2002.
- [Ale01] Alexander, R., "Improving the Quality of Object-Oriented Programs," *IEEE Software*, Vol. 18, No. 5, September-October 2001, pp. 90-91.
- [All02] Allen, P., "Measures of IO Effects Using the Joint Staff Analysis Model," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [AMD98] Acquisition Management Directorate, *A Guide to Auditing Defense Acquisition Programs Critical Program Elements*, Department of Defense, Office of the Inspector General, Washington, DC, September 1998.
- [And00] Anderson, J., *Highway Infrastructure: FHWA's Model for Estimating Highway Needs Is Generally Reasonable, Despite Limitations*, GAO/RCED-00-133, United States General Accounting Office, Washington, DC, June 2000.
- [And01] Anderson, F., *Risk Management Guide*, Defense Acquisition University Press, Fort Belvoir, VA, February, 2001.

- [And02] Anthony, J., "Using Extensible Markup Language for Composability of C4I Devices in Warfighters Simulation 2000," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [AOV+00] America, P., Obbink, H., Ommering, R., "CoPAM: A Component-Oriented Platform Architecting Method," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 167-180.
- [AP93] Department of the Army, *Verification, Validation and Accreditation of Army Models and Simulations*, Army Pamphlet 5-11, Department of Defense, Washington, DC, October 15, 1993.
- [AR97] Department of the Army, *Management of Army Models and Simulations*, Army Regulation 5-11, Department of Defense, Washington, DC, July 10, 1997.
- [AR00] Ayres, G., Richter, K., *Modeling and Simulation for Manufacturing*, Institute for Defense Analysis, Alexandria, VA, October 2000.
- [Arm64] Armer, P., "The Use of Computers" *Analysis for Military Decisions*, R- 387-PR, reprinted by the Military Operations Research Society, The RAND Corporation, 1964, 250-263.
- [ARS+96] Addy, E., Robinett, S., Sabolish, M., "V&V Within Reuse-Based Software Engineering," *Reuse '96 Working Group*, accessed on August 8, 2001 from: <http://research.ivv.nasa.gov/~eaddy/reuse96/>,.
- [AS94] Atkinson, S., Sorenson, R., *Documentation Technology Report*, Software Technology Support Center, Hill AFB, UT, April 1994.
- [AS01] Adelantado, M., Siron, P., "Multiresolution Modeling and Simulation of an Air-Ground Combat Application," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [AS02a] Aldridge, E., Stenbit, J., *Clinger-Cohen Act Compliance Policy*, Office of the Secretary of Defense memorandum, Washington, DC, March 8, 2002.
- [AS02b] Aldridge, E., Stenbit, J., *Joint Technical Architecture Version 4.0*, Office of the Secretary of Defense memorandum, Washington, DC, April 12, 2002.

- [AS02c] Aldridge, E., Stenbit, J., *Policy for Registration of Extensible Markup Language (XML)*, Office of the Secretary of Defense, Washington, DC, April 22, 2002.
- [AS03] Aldridge, E., Stenbit, J., *Software Acquisition Process Improvement Programs*, Office of the Secretary of Defense, Washington, DC, April 22, 2002.
- [ASC94] Aeronautical Systems Command, *Structural Modeling Handbook*, U.S. Air Force, November 1994.
- [ASC02b] Aldridge, E., Stenbit, J., Christie, T., *Operation of the Defense Acquisition System*, Under Secretary of Defense for Acquisition, Technology and Logistics, Washington, DC, April 5, 2002.
- [ASC02c] Aldridge, E., Stenbit, J., Christie, T., *Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs*, Under Secretary of Defense for Acquisition, Technology and Logistics, Washington, DC, April 5, 2002.
- [AT00] Albergo, E., Thomen, D., "The Future of Simulation Based Acquisition," *MSIAC M&S Journal Online*, December 1999 - March 2000.
- [ATL01] Department of Defense, *Intellectual Property: Navigating Through Commercial Waters*, Ver.1.1, Under Secretary of Defense for Acquisition, Technology and Logistics, Washington, DC, October 15, 2001.
- [Atw91] Atwood, D., *DoD Data Administration*, Department of Defense Directive Number 8320.1, Washington, DC, September 26, 1991.
- [AV98] Ayerbe, A., Vasquez, I., "Software Products Quality Improvement with a Programming Style Guide and a Measurement Process," *Proceedings of the Twenty-Second Annual International Computer Software and Applications Conference, COMPSAC '98*, Vienna, Austria, August 19-21, 1998, pp. 172-178.
- [AVL+97] Alexander, S., Volles, E., Levene, N., *Guide for Managing Information Technology (IT) as an Investment and Measuring Performance*, Department of Defense, Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, Washington, DC, February 10, 1997.
- [AWG97] Architecture Working Group (AWG), *C4ISR Architecture Framework*, Version 2.0, Assistant Secretary of Defense, Command, Control, Communications, Computers and Intelligence (ASD(C3I)), Washington, DC, December 18, 1997.

- [AWG98] Architecture Working Group (AWG), *Levels of Information Systems Interoperability (LISI)*, Assistant Secretary of Defense, Command, Control, Communications, Computers and Intelligence (ASD(C3I)), Washington, DC, March 30, 1998.
- [AWQ+93] Atwood, N. Winsch, B., Quinkert, K., *Catalog of Training Tools for Use in Distributed Interactive Simulation (DIS) Environment*, Loral Systems Co, Orlando, FL, July 1993.
- [BA99] Boehm, B., Abts, C., *Software Development Cost Estimation Approaches -A Survey*, University of Southern California, Los Angeles, CA, 1999.
- [BAC+95] Bravy, S., Adelson, R., Carniol, S., *Extended Air Defense Simulation (EAD-SIM) Face Assessment Modeling and Simulation Directorate*, Ballistic Missile Defense Organization, Washington, DC, June 12-16, 1995.
- [Bai92] Bailey, M., "The Scientific Method of Choosing Model Fidelity," *Proceedings of Conference on Variable-Resolution Modeling*, RAND National Defense Research Institute, Washington, DC, 5-6 May 1992, pp. 448-455.
- [Bal97] Balci, O., "Principles of Simulation Model Verification, Validation, and Testing," *Transactions of the Society for Computer Simulation International*, Volume 14, Issue 2, March, 1997, pp. 3-12.
- [Bal98] Balci, O., "Verification, Validation, and Accreditation," *Proceedings of the 1998 Winter Simulation Conference*, pp. 41-48.
- [Bar98] Barad, M., "Timed Petri Nets as a Verification Tool," *Proceedings of the 1998 Winter Simulation Conference*, pp. 547-554.
- [Bar02a] Barr, B., *Accreditation Statement for the Missile Defense System Exerciser (MDSE) to be used in support of the Interoperability Demonstration during the Patriot Advanced Capability-3 Configuration (PAC-3) Initial Operational Test and Evaluation (IOT&E)*, memorandum, CSTE, United States Army Test and Evaluation Command, Alexandria, VA, April 16, 2002.
- [Bar02b] Barr, B., *Accreditation Statement for the Extended Air Defense Testbed (EADTB) simulation used in support of the Interoperability Demonstration during the Patriot Advanced Capability-3 Configuration (PAC-3) Initial Operational Test and Evaluation (IOT&E)*, memorandum, CSTE, United States Army Test and Evaluation Command, Alexandria, VA, April 22, 2002.
- [Bar02c] Barry, D., "The Inevitable Pain of Software Development: Why There is No Silver Bullet," *Monterey 2002, Radical Innovations of Software and Systems*

Engineering in the Future Workshop Proceedings, Venice, Italy, October 7-11, 2002, pp. 28-47.

- [Bas00] Basili, V., *Using Experiments to Build a Body of Knowledge*, University of Maryland, 2000.
- [Bau02] Baucom, D., *History of Ballistic Missile Defense*, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/history.html>.
- [BB98a] Bodeker, D., Butker, D., "The Use of Models & Simulations (M&S) in Ballistic Missile Defense Organization (BMDO) Family of Systems (FoS) Test and Evaluation," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 1998.
- [BB98b] Butker, D., Bodeker, D., *Integration of Models & Simulations into the Evaluation of Theater Missile Defense Battle Management, Command, Control and Communications (BM/C3)*, accessed on January 4, 2002 from http://stinet.dtic.mil/cgi-bin/fulcrum_main.pl.
- [BB99] Baik, J., Boehm, B., *Empirical Analysis of CASE Tool Effects on Software Development Effort*, University of Southern California, Los Angeles, CA, 1999.
- [BB00] Bachman, F., Bass, L., *Managing Variability in Software Architecture*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, January 2000.
- [BBB+96] Bravy, S., Barnett, T., Bent., G., *Extended Air Defense Testbed (EADTB) Face Assessment* Modeling and Simulation Directorate, Ballistic Missile Defense Organization, Washington, DC, October 25, 1996.
- [BBB+99] Berteau, D., Bovin, D., Bowes, W., *Report of the Defense Science Board Task Force on Acquisition Reform Phase IV*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, July, 1999.
- [BBB+00] Bachman, F., Bass, L., Buhman, C., *Technical Concepts of Component-Based Software Engineering*, Volume II (CMU/SEI-2000-TR-008), 2nd ed., Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 2000.

- [BBC+00] Bachman, F., Bass, L., Chastek, G., *The Architecture Based Design Method*, (CMU/SEI-2000-TR-017), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, January 2000.
- [BBC+02] Bachman, F., Bass, L., Clements, P., *Documenting Software Architecture: Documenting Behavior*, (CMU/SEI-2002-TN-001), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, January 2002.
- [BBF+01] Brown, T., Burns, S., Fruitman, M., *Software and Systems Process Improvement Programs Vary in Use of Best Practices*, GAO/AIMD-01-116, United States General Accounting Office, Washington, DC, March 2001.
- [BBF+02] Bird, M., Booth, H., Fruitman, M., *Information Technology: Enterprise Architecture Use Across the Federal Government Can Be Improved*, GAO/02-6, United States General Accounting Office, Washington, DC, February 2002.
- [BBG+99a] Bent, G., Bravy, S., Gordon, B., *JTAMDO M&S Review Executive Summary*, Volume I, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, July 28, 1999.
- [BBG+99b] Bent, G., Bravy, S., Gordon, B., *JTAMDO Assessment Support Review*, Volume II, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, September 28, 1999.
- [BBG+99c] Bent, G., Bravy, S., Gordon, B., *JTAMDO M&S Review Draft: Appendices*, Volume III, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, September 9, 1999.
- [BBG+99d] Bent, G., Bravy, S., Gordon, B., *JTAMDO M&S Assessment: Joint Warfare System (JWARS)*, Volume IV, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, November 30, 1999.
- [BBG+99e] Bent, G., Bravy, S., Gordon, B., *JTAMDO M&S Assessment: (EADSIM)*, Volume V, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, December 1, 1999.
- [BBG+99f] Bent, G., Bravy, S., Gordon, B., *JTAMDO M&S Assessment: Commanders Analysis and Planning Simulation (CAPS)*, Volume VI, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, December 1, 1999.
- [BBG+00] Bachmann, F., Bass, L., Garlan, D., *Software Architecture Documentation in Practice: Documenting Architectural Layers*, (CMU/SEI-2000-SR-004),

Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, March 2000.

- [BBG+03] Butz, P., Bloom, G., Gross, M., *Is There a Shortage of Scientists and Engineers? How Would We Know?* RAND, Santa Monica, CA, 2001.
- [BBK00] Bachman, F., Bass, L., Klein, M., *An Application of The Architecture - Based Design Method to the Electronic House*, (CMU/SEI-2000-SR-009), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, September 2000.
- [BBK02] Bachman, F., Bass, L., Klein, M., *Illuminating the Fundamental Contributors to Software Architecture Quality*, (CMU/SEI-2002-TR-025), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 2002.
- [BBM+01] Buehler, K., Buehler, S., McNeely, J., Hicks, E., Ressler, R., "Using the M&S/C4ISR Conceptual Reference Model as a Framework for Integrating DII COE Segments into C4ISR Stimulation," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [BBN+01] Buehler, K., Buehler, S., Hicks, E., "Using the M&S/C4ISR Conceptual Reference Model as a Framework for Integrating DII COE Segments into C4ISR Stimulation," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [BBT+03] Barkakati, N., Bird, M., Collier, B., *A Framework for Assessing and Improving Enterprise Architecture Management*, Ver. 1.1, GAO-03-584G, United States General Accounting Office, Washington, DC, April 2003.
- [BBW+99] Bergey, J., Smith, Nelson, *Options Analysis for Reengineering (OAR) Issues and Conceptual Approach*, (CMU/SEI-99-TN-014), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, September 1999.
- [BBW+00] Bass, L., Buhman, C., Wallnau, K., *Market Assessment of Component Based Software Engineering*, Vol. 1, (CMU/SEI-2001-TN007), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 2000.
- [BC96] Brownsword, L., Clements, P., *A Case Study in Successful Product Line Development*, (CMU/SEI-96-TR-016), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, October 1996.

- [BC02] Berzins, V., Chaki, N., “MTDOAG: Module Test Driver and Output Analyzer Generator,” *Monterey 2002, Radical Innovations of Software and Systems Engineering in the Future Workshop Proceedings*, Venice, Italy, October 7-11, 2002, pp. 48-56.
- [BCB02] Bers, J., Carlson, Boswell, S., “Monitoring, Measuring, and Analyzing Federation Performance,” *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [BCC+97] Bass, L., Chastek, G., Clements, P., Cohen, S., Northrop, L., Withey, J., *Product Line Practice Workshop Report*, (CMU/SEI-97-TR-003), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 1997.
- [BCC+98] Bergey, J., Clements, P., Cohen, S., Donohoe, P., Jones, L., Krut., B., Northrop, L., Tilley, S., Smith, D., Withey, J., *DoD Product Line Practice Workshop Report*, (CMU/SEI-98-TR-007), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 1998.
- [BCC+00] Brown, Y., Canal, K., Cullen, R., *Software Risk Management: A Practical Guide*, Department of Energy, February 2000.
- [BCC+01a] Braddock, J., Chatham, R., Christie, J., *Report of the Defense Science Board Task Force on Training Superiority & Training Surprise*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, January, 2001.
- [BCC+01b] Barbaci, M., Chang, C., Coalier, F., *Guide to the Software Engineering Body of Knowledge (SWEBOK), Trial Version*, Ver. 0.95, Software Engineering Coordinating Committee, IEEE Computer Society-ACM Committee, May 2001.
- [BCC+02] Brandt, K., Carr, F., Conrad, G., *C4ISR/SIM Technical Reference Model Study Group Zfinal Report (C4ISR/Sim TRM) Final Report*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 2002.
- [BCD+00] Bass, L., Clements, P., Donohoe, P., McGregor, J., Northrop, L., *Fourth Product Line Practice Workshop Report, Technical Report CMU/SEI-2000-TR-002 ESC-TR-2000-002*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, February 2000.

- [BCE+99] Brown, D., Cusimano, G., Enzinger, F., *Seeking the Proper Balance Between Simulation and Flight Test*, AIAA Information Paper, American Institute of Aeronautics and Astronautics, Reston, VA, October 1, 1999.
- [BCE+00] Borah, J., Cutts, D., Ecklund, E., *Object Architecture for Semantically Interoperable Systems (OASIS) Final Report*, The Aegis Technologies Group, Huntsville, AL, 2002.
- [BCH+02] Borstler, J., Carrington, D., Hislop, G., "Teaching PSPS: Challenges and Lessons Learned," *IEEE Software*, Vol. 19, No. 5, September/October 2002, pp. 48-56.
- [BCK98] Bass, L., Clements, P., Kazman, R., *Software Architecture In Practice*, Addison-Wesley, Reading, MA, 1998.
- [BCN96] Banks, J., Carson, J.S., Nelson, B.L., *Discrete Event System Simulation*, Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 1996.
- [BCN+98] Bass, L., Chastek, G., Clements, P., Northrop, L., Smith, D., Withey, J., *Second Product Line Practice Workshop Report*, (CMU/SEI-98-TR-015), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 1998.
- [BCP+01] Beauchamp, J., Colantonio, D., Pickup, S., *Defense Plans: Status of DoD's Efforts to Improve Its Joint Warfighting Requirements Process*, GAO-02-100R, United States General Accounting Office, Washington, DC, November 9, 2001.
- [BCS00] Batory, D., Cardone, R., Smaragdakis, "Object-Oriented Frameworks and Product Lines," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 227-248.
- [BD02] Bansiya, J., Davis, C., "A Hierarchical Model for Object-Oriented Design Quality Assessment," *IEEE Transaction on Software Engineering*, Vol. 28, No. 1, January 2002, pp. 4-17.
- [BDA+00] Bourgue, P. Dupuis, R., Abran, A., *Annual Status Report Guide to the Software Engineering Body of Knowledge*, Software Engineering Coordinating Committee, IEEE Computer Society-ACM Committee, February 21, 2000.
- [BDH00] Blieberger, A., Dey, M., Heere, J., *Defense Acquisitions, Comprehensive Strategy Needed to Improve Ship Cruise Missile Defense*, GAO/NSIAD-00-149, United States General Accounting Office, Washington, DC, July 2000.

- [BDM00] Bigelow, J., Davis, P., McEver, J., "Informing and Calibrating a Multiresolution Exploratory Analysis Model with High Resolution Simulation: The Interdiction Problem as a Case History," *Exploratory Analysis and a Case History of Multiresolution, Multiperspective Modeling*, RAND National Defense Research Institute reprint, Santa Monica, CA, 2001.
- [BE99] Boehm, B., Egyed, A., *Optimizing Software Product Integrity through Life-Cycle Process Integration*, University of Southern California, Los Angeles, CA, 1999.
- [Bec94] Becker, F., *Developing a Navy Strategy for Modeling and Simulation: Review of VV&A Studies and Practices*, Center for Naval Analysis, Alexandria, VA, November 1994.
- [Bec99] Beckman, K., *Directory of Industry and University Collaborations with a Focus on Software Engineering Education and Training, Version 7*, CMU/SEI-99-SR-001," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, February 1999.
- [Bee02] Beeker, E., "Federating Time-Stepped and Discrete Event Simulations for Analysis," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [BEL+02] Barbacci, M., Ellison, R., Lattanze, A., *Quality Attribute Workshops, 2nd Edition*, (CMU/SEI-2002-TR-019), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 2002.
- [Ber94] Berard, E., *Metrics for Object-Oriented Software Engineering*, The Object Agency, Inc., accessed on February 13, 2002 from <http://www.ipipan.gda.pl/~marek/objects/TOA/moose.html>.
- [Ber98a] Berry, D., "Formal Methods: The Very Idea*, Some Thoughts about Why They Work When They Work," *Proceedings of the 1998 on Engineering Automation for Computer Systems*, Naval Postgraduate School, Monterey, CA, April 1999, pp. 3-8.
- [Ber98b] Berzins, V., "Lightweight Inference for Automation Efficiency," *Proceedings of The 1998 Monterey Workshop on Engineering Automation for Computer Systems*, Naval Postgraduate School, Monterey, CA, April, 1999, pp.19-29.
- [Ber00a] Berg, S., *Battlespace Environment and Signature Toolkit Stakeholder Needs* (Draft), Version 0.27, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, September 20, 2000.

- [Ber00b] Bernstein, L., "A Software Engineering Course for Trustworthy Software," *Software Tech News*, accessed on May 18, 2000 from <http://www.dacs.dtic.mil/awareness/newsletters>.
- [Ber01] Berzins, V. *XML Technology Assessment Progress Report (7/17/2000 – 12/31/2000)*, NPS-SW-01-002, Naval Postgraduate School, Monterey, CA, January 2001.
- [BF00] Bryant, R., Flynn, J., "Seeing through the Fog-Attaining 'Full Spectrum Dominance' with the System-of-Systems Through Operational Engineering," *Naval Engineers Journal*, Vol. 112, No. 4, July 2000.
- [BFG+00] Bergey, J., Fisher, M., Gallagher, B., Jones, L., Northrop, L., "Basic Concepts of Product Line Practice for the DoD (Technical Note CMU/SEI-2000-TN -001)," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, February 2000.
- [BFJ99] Bergey, J., Fisher, M., Jones, L., *The DoD Acquisition Environment and Software Product Lines, Technical Note* (CMU/SEI-99-TN-004), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 1999.
- [BGK+00] Brown, C., Grant, G., Kotchman, D., "Building a Business Case for Modeling and Simulation," *Acquisition Review Quarterly*, Fall 2000, pp. 311-327.
- [BGK+02] Bhargavan, K., Gunter, C., Kim, M., "Verism: Formal Analysis of Network Simulations," *IEEE Transaction on Software Engineering*, Vol. 26, No. 2, February 2002, pp. 129-145.
- [BH99] Bray, O., Hess, M., "Re-engineering a Configuration Management System," *IEEE Software*, Vol.12, No.1, January 1999.
- [BHH+99] Bagert, D., Hilburn, T., Hislop, G., *Guidelines for Software Education Version 1.0*, CMU/SEI-99-TR-032), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, PA, October 1999.
- [BHK01] Biffi, S., Halling, M., Kohle, M., "Investigating the Effect of a Second Software Inspection Cycle," *Proceedings of the First Asia Conference on Quality Software, 2000*, Hong Kong, China, October 30-31, 2000, pp.194-203.
- [Bia00] Bianca, D., "Simulation and Modeling For Acquisition, Requirements and Training (SMART)," *Strategy Research Project*, U.S. Army War College, Carlisle Barracks, PA, April 10, 2000.

- [Bid00] Biddle, M., "A Proposed Scheme for Implementing Aggregation and Disaggregation in HLA," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [Bir97] Birkel, P., *SEDRIS Geospatial Reference Model*, MITRE Corporation, July 10, 1997.
- [Bir98] Birkel, P., "SNE Conceptual Reference Model," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18 1998.
- [BIL97] Berzins, V., Ibrahim, O., Luqi, "A Requirements Evolution for Computer Aided Prototyping," *Proceedings of the 9th International Conference on Software Engineering*, Madrid, Spain, June 17-20, 1997, pp. 38-47.
- [BJ98] Buss, A., Jackson, L., "Distributed Simulation Modeling: A Comparison of HLA, CORBA, and RMI," *Proceedings of the 1998 Winter Simulation Conference*, " 1998, pp. 819-825.
- [BK99] Bass, L., Kazman, R., *Architecture-Based Development*, (CMU/SEI-99-TR-007), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 1999.
- [BK01a] Burke, A., Kerner, J., *Mapping of Joint Technical Architecture (JTA) Version 3.1 Core Standards to Technical Reference Model (TRM) Version 1.0 Services and Interfaces*, The Aerospace Corporation, El Segundo, CA, March 2, 2001.
- [BK01b] Bernstein, L., Klappholz, D., "Getting Software Engineering into Our Guts," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, July 2001, pp.25-26.
- [BK02a] Brade, D., Koster, A., *Risk-based Validation & Verification Levels Definition*, Joint Accreditation Support Activity, M&S Credibility Workshop II, Reno, NV, March 4, 2002.
- [BK02b] Bernstein, L., Klappholz, D., "Innovative Software Engineering Education," *Software Tech News*, Vol. 5, No. 2, April 2002, pp. 13-15.
- [BKB00] Bass, L., Klein, M., Bachmann, F., *Quality Attribute Design Primitives*, Software Engineering Institute, (CMI/SEI-2000-TN-017), Carnegie Mellon University, Pittsburgh, PA, December 2000.

- [BKB01] Bass, L., Klein, M., Bachmann, F., *Quality Attribute Design Primitives and the Attribute Driven Design Method*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2001.
- [BKR95] Bracken, J., Kress, M., Rosenthal, R., Ed., *Warfare Modeling*, John Wiley & Sons, 1995.
- [BKK02] Bernstein, L., Klappholz, D., Kelley, C., "Eliminating Aversion to Software Process in Computer Science Students and Measuring Results," *Proceedings of the 15th Conference on Software Engineering and Training (CSEET'02)*, 2002.
- [BKW97] Barbacci, M., Klein, M., Weinstock, C., *Principles for Evaluating the Quality Attributes of a Software Architecture*, (CMU/SEI-96-TR-036), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 1997.
- [BKW+02] Black, C., Kidwell, R., Whitson, S., "Breaking New Ground: Simulation to C4I Intra-Army Interoperability Certification," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [BL91] Berzins, V., Luqi, *Software Engineering With Abstractions*, Addison-Wesley Publishing Company, Reading, MA, 1991.
- [BL94] Badr, S., Luqi, "Automation Support for Concurrent Software Engineering" *Proceedings of 6th International Conference on Software Engineering and Knowledge Engineering*, Jurmala, Latvia, June 20-23, 1994, pp. 46-53.
- [BLB+01] Berzins, V., Luqi, Bhattacharya, S., "JAVA Wrappers for Automated Interoperability," *Interim Progress Report, Engineering Automation for Reliable Software*, Naval Postgraduate School, Monterey, CA, September 2001, pp. 26-45.
- [BLR+02] Bentley, T., Larsen, K., Roden, A., "HLA Lessons Learned from the RDEC Federation Calibration Experiment," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [BLS+99] Berzins, V., Luqi, Shing, M., Saluto, M., Williams, J., *Re-engineering the Janus (A) Combat Simulation System*, Naval Postgraduate School, Monterey, CA, January, 1999.
- [BM92] Brendley, K., Marti, J., "A Distributed Network Approach to Variable Resolution Modeling," *Proceedings of Conference on Variable-Resolution Mod-*

eling, RAND National Defense Research Institute, Washington, DC, 5-6 May 1992, pp. 248-255.

- [BM93] Basili, V., Musa, J., “The Future of Software Engineering: A Management Perspective,” *Software Management*, 4th Edition, IEEE Computer Society Press, Los Alamedas, CA, 1993, pp. 9-15.
- [BM99] Baxter, I., Mehlich, “Reverse Engineering is Reverse Forward Engineering,” *Proceedings of the 4th Workshop on Reverse Engineering*, IEEE Computer Society, 1997.
- [BM02] Baker, L., Michel, L., “Education, Information Technology, and the ‘Software Crisis’,” *Software Tech News*, Vol. 5, No. 2, April 2002, pp. 5-8.
- [BMC+00] Buford, J., Mikkelson, C., Burson, C., McFarlin, S., *U.S. Army Aviation and Missile Command (AMCOM) Aviation & Missile Research, Development & Engineering Center AMRDEC Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, January 12, 2000.
- [BMD99a] Ballistic Missile Defense Organization, *Summary of Report to Congress on Utility of Sea-Based Assets to National Missile Defense*, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, June 1, 1999.
- [BMD99b] Ballistic Missile Defense Organization, *BMDO Core M&S Decision Brief*, BMDO/AQ, Office of the Secretary of Defense, Washington, DC, September 1, 1999.
- [BMD00a] Ballistic Missile Defense Organization Fact Sheet, *Harnessing The Power of Technology: The Road to Ballistic Missile Defense From 1983-2007*, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, September 2000.
- [BMD00b] Ballistic Missile Defense Organization, *The Family of Systems Concept*, BMDO Fact Sheet 214-00-11, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, November 2000.
- [BMD00c] Ballistic Missile Defense Organization, *National Missile Defense*, BMDO Fact Sheet 123-00-06, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, November 2000.
- [BMD00d] Ballistic Missile Defense Organization, *National Missile Defense Multi Service Launch System*, BMDO Fact Sheet 102-99-03, Ballistic Missile De-

fense Organization, Office of the Secretary of Defense, Washington, DC, November 2000.

- [BMD00e] Ballistic Missile Defense Organization, *National Missile Defense Weapon Element*, BMDO Fact Sheet 125-00-11, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, November 2000.
- [BMD00f] Ballistic Missile Defense Organization, *Battle Management, Command, Control, and Communications*, BMDO Fact Sheet 122-00-11, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, November 2000.
- [BMD00g] Ballistic Missile Defense Organization, *National Missile Defense X-Band Radar*, BMDO Fact Sheet 119-00-11, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, November 2000.
- [BMD00h] Ballistic Missile Defense Organization, *National Missile Defense Early Warning System*, BMDO Fact Sheet 113-00-11, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, November 2000.
- [BMD00i] Ballistic Missile Defense Organization, *National Missile Defense Architecture*, BMDO Fact Sheet 106-00-11, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, November 2000.
- [BMD01] Ballistic Missile Defense Organization, *Extended Air Defense Testbed (EADTB) Kill Chain Capability*, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, October 22, 2001.
- [BML02] Brade, D., Maguire, R., Lotz, H., "Arguments-based Credibility Levels," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [BNT93] Barrow, D., Nilson, S., Timberlake, D., *Software Estimation Technology Report*, Software Technology Support Center, Hill AFB, UT, March 1993.
- [BNS97] Bergey, J., Northrop, L., Smith, D., *Enterprise Framework for the Disciplined Evolution of Legacy Systems*, (CMU/SEI-97-TR-007), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, October 1997.
- [Boc00] Bockle, G., "Model-Based Requirements Engineering for Product Lines," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 193-204.

- [Boe91] Boehm, B., "Risk Management for Software Development," *ESCOM Tutorial*, 24 March 4, 1993.
- [Boe93] Boehm, B., "A Spiral Model of Software Development and Enhancement," *Software Management*, 4th Edition, IEEE Computer Society Press, Los Alamedas, CA, 1993, pp. 120-131.
- [Boe00] Boehm, B., *Spiral Development: Experience, Principles, and Refinements*, USC-SEI Workshop on Spiral Development Experience and Implementation Challenges, USC, February 9-11, 2000.
- [Boe02] Boehm, B., "Software Engineering is a Value-Based Contact Sport," *IEEE Software*, Vol. 19, No. 5, September/October 2002, pp.95-96.
- [Boo01] Booch, G., *Quality Software and the Unified Modeling Language*, accessed on July 1, 2001 from <http://www.rational.com/products/whitepapers/285.jsp>.
- [Boo02] Booch, G., *Measures of Goodness*, accessed on July 22, 2002 from <http://www.rational.com/products/whitepapers/393.jsp>.
- [Bor02] Borah, J., "Conceptual Modeling:- The Missing Link of Simulation Development," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [Bos00] Bosch, J. *Design & Use of Software Architectures: Adopting and Evolving a Product-Line Approach*, Addison-Wesley/ACM Press, Reading, MA, 2000.
- [BOS00] Bergey, J., O'Brien, L., Smith, D., *Mining Assets for Software Product Lines*, (CMU/SEI-2000-TN-008), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 2000.
- [BOS02] Bergey, J., O'Brien, L., Smith, D., *An Application of an Iterative Approach to DoD Software Migration Planning*, (CMU/SEI-2002-TN-027), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 2002.
- [Bow02] Bowyer, K., "Star Wars Revisited: Ethics and Safety-Critical Software," *IEEE Technology and Society Magazine*, Spring 2002.
- [BP02] Bunker, P., Palazzolo, M., "High Fidelity Unmanned Ground Vehicle Modeling," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.

- [BPC+96] Browning, M., Pirnie, A., Carabello, J., *Information Management Performance Measures: Developing Performance Measures and Management Controls for Migration Systems, Data Standards, and Process Improvement*, Report by the U.S. National Academy of Public Administration for the Department of Defense, January 1996.
- [BPD02] Burnell, L., Priest, J., Durrett, T., "Teaching Distributed Multidisciplinary Software Development," *IEEE Software*, Vol. 19, No. 5, September/October 2002, pp. 86-93.
- [BPM+95] Butler, S., Perry, F., McCoy, D., *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification (I&RTS)*, Version 2.0 Joint Interoperability and Engineering Organization, Defense Information Systems Agency, Washington, DC, October 23, 1995.
- [BPT96] Burgeson, J., Piowar, T., Try, P., *Natural Environmental Effects in Military Models and Simulations: Part III-An Analysis of Requirements Versus Capabilities*, Science and Technology Corp. Scientific Report No. 2, Hampton, VA, August 1996.
- [Bra96] Bravy, S., *Extended Air Defense Testbed (EADTB) Face Assessment*, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, October 25, 1996.
- [Bra98] Branch, A., *Implementing the Joint Synthetic Test and Evaluation Battlespace*, Navy Air Systems Command, Patuxent River, MD, September 10, 1998.
- [Bra00a] Bravy, S., *BMDO/DEI Modeling and Simulation Hierarchy Study (Version 6)*, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, January, 2000.
- [Bra00b] Bravy, S., *Model and Simulation Assessment: Vector-In-Commander (VIC)*, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, December 18, 2000.
- [Bra02] Brand, A., "Ground-Based Midcourse Defense Reference FOM," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL. March 2002.
- [BRC+01] Bunyard, J., Richardson, H., Capps, W., *Patriot Test and Evaluation Lessons Learned: 1965-2001, Final Report*, SAIC AAA Division, Alexandria, VA, September 2001.

- [Bre88] Brew, T., *Strategic Defense Initiative Program: National Test Bed Acquisition Has Not Proceeded As Planned*, GAO/NSIAD-88-180FS United States General Accounting Office, Washington, DC, June 1988.
- [Bre02a] Breitler, A., "Statistics of Validation," *M&S Credibility Workshop II* presentation, Reno, NV, March 4, 2002.
- [Bre02b] Brennan, R., *Protecting the Homeland, Insights From Army Wargames*, RAND Arroyo Center, Santa Monica, CA, 2002.
- [BRJ99] Booch, G., Rumbaugh, J., Jacobsen, I., *The Unified Modeling Language Users Guide*, Addison-Wesley, Reading, MA, 1999.
- [Bro74] Brooks, F., "The Mythical Man-Month," *Software Management*, 4th Edition, Reifer, D., Ed., IEEE Computer Society Press, Los Alamedas, CA, 1993, pp. 151-159.
- [Bro87] Brooks, F., "No Silver Bullet: Essence and Accident of Software Engineering," *Computerworld*, Vol. 20, No., 4, April 1987, pp. 10-19.
- [Bro88] Brooks, J., One Hundredth Congress House of Representatives memorandum, Washington, DC, May 20, 1988.
- [Bro91] Brock, J., *Supercomputing in Industry*, GAO/T-IMTEC-91-5, United States General Accounting Office, Washington, DC, March, 1991.
- [Bro92] Brown, R., "Joint Modeling and Simulation System," *Proceedings of Conference on Variable-Resolution Modeling*, RAND National Defense Research Institute, Washington, DC, 5-6 May 1992, pp. 432-440.
- [Bro96] Brown, N., *Industrial-Strength Management Strategies*, accessed on January 5, 1997 from <http://www.stsc.hill.af.mil/www/xt96aug/xt96d08e.html>.
- [Bro98] Brown, N., *Little Book of Configuration Management*, Software Program Managers Network, Arlington, VA, November 1998.
- [Bro99] Brock, J., *Defense Information Management: Continuing Implementation Challenges Highlight the Need for Improvement*, GAO/AIMD-99-93, United States General Accounting Office, Washington, DC, February 25, 1999.
- [Bru98] Brune, K., "Product Line Practice at Work in Government Organizations," [News@SEI](#), *Vol 1, No 3*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Fall 1998, pp. 4-5.

- [BS95] Berzins, V., Shing, M., “Summary of the ’95 Workshop: Specification-Based Software Architecture,” *Proceedings of Monterey Workshop 1995*, Monterey, CA, September 12-14 1995, pp. 107-112.
- [BS00] Bergey, J., Smith, D., *Guidelines for Using OAR Concepts in DoD Product Line Acquisition Context* (CMU-SEI-2000-004), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, March 2000.
- [BS01] Bossuyt, B., Snyder, B., “Software Testing Tools: Metrics For Measurement of Effectiveness On Procedural and Object-Oriented Source Code,” Master thesis, *Progress Report: Assessment of Defense Modeling And Simulation Office (DMSO) Conceptual Models of the Mission Space*, Naval Postgraduate School, Monterey, CA, December 31, 2001.
- [BSB+98] Bernstein, R., Stelling, J., Binford, J., *A Road Map for Simulation Based Acquisition*, Executive Committee on Modeling and Simulation (EXCIMS) Acquisition Council Draft for Coordination, Washington, DC, December 4, 1998.
- [BSB+00] Boyd, A., Schweitzer, S., Basany, K., Klem, B., Hayhurst, T., Lawrie, D., Casey, E., *Space Based Infrared System (SBIRS) Low Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, February 4, 2000.
- [BSB+02] Burroughs, A., Sorroche, J., Barrett, N., “Tactical Digital Information Link-Technical Advice and Lexicon for Enabling Simulation (TADIL-TALES), *Proceedings of the 2002 Fall Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-13, 2002.
- [BSK+00] Birkler, J., Smith, G., Kent, G., *An Acquisition Strategy, Process, and Organization for Innovative Systems*, RAND, Santa Monica, CA, 2000.
- [BSL+00] Berzins, V., Shing, M., Luqi, *Object-Oriented Modular Architecture for Ground Combat Simulations*, Naval Postgraduate School, Monterey, CA, 2000.
- [BSO00] Bergey, J., Smith, D., O'Brien, L., *Mining Existing Assets for Software Product Lines* (CMU-SEI-2000-TN-008), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Pittsburgh, PA, May 2000.
- [BST+99] Bergey, J., Smith, D., Tilley, S., Weiderman, N., Woods, S., *Why Reengineering Projects Fail*, (CMU/SEI-99-TR-010), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 1999.

- [BSV+97] Brannin, P., Spencer, R., Vincent, D., *Requirements Planning and Impact on Readiness of Training Simulators and Devices*, 97-138, Department of Defense, Office of the Inspector General, Washington, DC, April 30, 1997.
- [BSW99] Bergey, J., Smith, S., Weideman, N., *DoD Legacy System Migration Guidelines*, (CMU/SEI-99-TN-013), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, September 1999.
- [BSW+99] Bergey, J., Smith, S., Weideman, N., *Options Analysis for Reengineering (OAR) Issues and Conceptual Approach*, (CMU/SEI-99-TN-014), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, September 1999.
- [BTE+93] Braddock, J., Thurman, M., England, G., *Report of the Defense Science Board Task Force on Simulations, Readiness, and Prototyping: Impact of Advanced Distributed Simulation on Readiness, Training, and Prototyping*, Office of the Undersecretary of Defense For Acquisition, Washington, DC, January 1993.
- [Bun00] Bundy, G., *Army M&S Architecture Working Group*, presentation by AMSO Technical Advisor, March 21, 2000.
- [Bur00] Burwell, D., *Battlespace Environment and Signature Toolkit (BEST) Vision*, Version 0.4, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, September 8, 2000.
- [Bus01] Bush, G., *Executive Order on Critical Infrastructure Protection*, President of the United States, The White House, Washington, DC, October 16, 2001, accessed on March 25, 2002 from <http://www.whitehouse.gov/news/releases/2001/10/20011016.html>.
- [Bus02a] Bush, G., *Securing the Homeland Strengthening the Nation*, President of the United States, The White House, Washington, DC, 2002.
- [Bus02b] Bush, G., *The Department of Homeland Security*, President of the United States, The White House, Washington, DC, June 2002.
- [Bus02c] Bush, G., *National Strategy for Homeland Security*, President of the United States, The White House, Washington, DC, July 2002.
- [Bus02d] Bush, G., *National Security Strategy of the United States of America*, President of the United States, The White House, Washington, DC, September 20, 2002.

- [Bus02e] Bush, G., *President Announces Progress on Missile Defense Capabilities*, President of the United States, The White House, Washington, DC, December 12, 2002, accessed on December 23, 2002 from <http://www.whitehouse.gov/news/releases/2002/12/20021217.html>.
- [Bus02f] Bush, G., *Statement by the President*, President of the United States, The White House, Washington, DC, December 17, 2002, accessed on December 23, 2002 from <http://www.whitehouse.gov/>.
- [BVW+00] Bender, F., Varnavas, J., Wilson, M., Fama, N., Smith, P., Hartley, S., *ODA Notes - Combined*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, 2000.
- [BW97] Burns, A., Wellings, A., *Real-Time Systems and Programming Languages*, Addison-Wesley, Harlow, England, 1997.
- [BW01] Briand, L., Wust, J., "Modeling Development Effort in Object-Oriented Systems Using Design Properties," *IEEE Transactions on Software Engineering*, Vol. 27, Issue 11, November 2001, pp. 963-986.
- [BW02] Bergey, J., Wood, W., *Use of Quality Attribute Workshops (QAWs) in Source Selection for a DoD System Acquisition: A Case Study*, (CMU/SEI-2002-TN-013), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 2002.
- [Byn00] Bynum, M., *The IT Workforce Shortage and Education*, ITAA Public Policy Report, accessed on March 3, 2000 from <http://www.ita.org/govt/pubs>.
- [BZP+02] Brutzman, D., Zyda, M., Pullen, M., "Extensible Modeling and Simulation Framework (XMSF) Challenges for Web-based Modeling and Simulation," *Extensible Modeling and Simulation Framework Symposium*, George Mason University, VA, September 4, 2002.
- [CA02] Cazard, L., Adelantado, M., "HLA Federates Design and Federations Management: Towards a Higher Level Object-Oriented Architecture Hiding the HLA Services," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [Caf03] Caffall, S., *Conceptual Framework Approach for Systems-of-Systems Software Developments*, thesis, Naval Postgraduate School, Monterey, CA, March 2003.

- [Cal97] Calbom, L., *Credit Reform: Review of OMB's Credit Subsidy Model*, GAO/AIMD-97-145, United States General Accounting Office, Washington, DC, August, 1997.
- [Cam02] Campbell, G., *A Software Product Line Vision for Defense Acquisition*, (CMU/SEI-2002-TN-002), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, December 2002.
- [Cau95] Caughlin, D., "Verification, Validation, and Accreditation (VV&A) of Models and Simulations Through Reduced Order Metamodels," *Proceedings of the 1995 Winter Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, pp. 1405-1412.
- [Car88] Carlucci, F., Office of the Secretary of Defense memorandum to the Chairman, Committee on Government Operations, U.S. House of Representatives on DoD's progress to implement the GAO recommendations on DoD simulations, Washington, DC, June 22, 1988.
- [Car99] Carney, D., *A Little Red Book of Truths to Enlighten and Guide on the Long March Toward the COTS Revolution*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, December 6, 1999.
- [CB96] Chastek, G., Brownsword, L., *A Case Study In Structural Modeling*, (CMU/SEI-96-TR-035), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, December 1996.
- [CBB+03] Clements, P., Bachmann, F., Bass, L., *Documenting Software Architectures, Views and Beyond*, Addison-Wesley, Boston, MA, 2003.
- [CC96] Clinger, R., Cohen, W., *Clinger/Cohen Act (formerly the Information Technology Management Reform Act of 1996)*, S.1124, 104th Congress of the United States, 2d Session, Washington, DC, January 1996.
- [CCB+02] Clark, J., Capella, S., Bailey, C., "The Development of an HLA Compliant High Performance Computing Run Time Infrastructure," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [CCK+96] Clark, D., Costanzo, C., Karangelen, N., *Study on the Application of Modeling and Simulation to the Acquisition of Major Weapon Systems*, American Defense Preparedness Association, Washington, DC, September 27, 1996.
- [CCL+85] Cohen, D., Chatterjee, D., Lau, R., *Eastport Study Group, Summer Study 1985, A Report to the Director Strategic Defense Initiative Organization*,

Department of Defense, Strategic Defense Initiative Organization, Washington, DC, February 14, 1986.

- [CDC+00] Cross, J., Denning, P., Chang, C., *Computing Curricula 2001*, The Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery, (Draft), March 6, 2000.
- [CDK+01] Chastek, G., Donohoe, P., Kang, K., Thiel, S., *Product Line Analysis: A Practical Introduction*, (CMU/SEI-2001-TR-001), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 2001.
- [CDS02] Cohen, S., Dunn, E., Soule, A., *Successful Product Line Development and Sustainment: A DoD Case Study*, (CMU/SEI-2002-TN-019), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, September 2002.
- [CFG+94] Cruickshank, R., Felber, H., Gaffney, J., *Software Measurement Guidebook*, Ver.2.01, Software Productivity Consortium, Herndon, VA, August 1994.
- [CFM+96] Cohen, S., Friedman, S., Martin, L., Royer, T., Solderitsch, Webster, R., *Concept of Operations for the ESC Product Line Approach*, (CMU/SEI-96-TR-018), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 8, 1996.
- [CFM99] Chu, L., Feehan, D., McCaffrey, A., *Nuclear Weapons: DOE Needs to Improve Oversight of the \$5 Billion Strategic Computing Initiative* GAO/RCED-99-195, United States General Accounting Office, Washington, DC, June 1999.
- [CFM+03] Chamberlain, D., Frankhauser, P., Marchiori, M., *XML Query (XQuery) Requirements*, World Wide Web Consortium (W3C), WSC working draft May 2, 2003.
- [CGF+00] Cohen, S., Gallagher, B., Fisher, M., *Third DoD Product Line Practice Workshop Report* (CMU-SEI-2000-024), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, July, 2000.
- [CGG98] Cat, K., Gardner, K., Gryder, T., "Military Modeling Framework (MMF) A Framework Supporting Composability and Multi-Resolution Modeling," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18 1998.
- [CH01] Carr, F., Hieb, M., "M&S Interoperability within the DII COE: Building a Technical Requirements Specification," *Proceedings of the 2001 Spring*

Simulation Conference, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30 2001.

- [Cha00] Chapman, R., "Conceptual Modeling Framework for Complex Synthetic Systems - an Example from F-15C Distributed Mission Training," *Proceedings of the 2000 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL. March 26-31, 2000.
- [Cha01] Chamberlain, S., "Enterprise Identifiers for Global Naming Across the C4I-Simulation Boundary," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30 2001.
- [Che86a] Chelimsky, E., *Retirement Forecasting: Evaluation of Models Shows Need for Information on Forecast Accuracy*, GAO/PEMD-87-6A, United States General Accounting Office, Washington, DC, December, 1986.
- [Che86b] Chelimsky, E., *Retirement Forecasting: Technical Descriptions of Cost, Decision and Income Models*, GAO/PEMD-87-6B, United States General Accounting Office, Washington, DC, December, 1986.
- [Che87] Chelimsky, E., *DoD Simulations: Improved Assessments Would Increase the Credibility of Results*, GAO/PEMD-88-3, United States General Accounting Office, Washington, DC, December, 1987.
- [Che98] Chew, J., "Verification, Validation, & Accreditation and Test & Evaluation," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18 1998.
- [CHK+90] Caro, I., Higuera, R., Kockler, F., *Mission Critical Computer Resources Management Guide*, Defense Systems Management College, Fort Belvoir, VA, 22060-5426, 1990.
- [CHS+02] Cooper, D., Hite, R., Schinasi, K., *Defense Acquisitions: DoD Faces Challenges in Implementing Best Practices*, GAO-02-469T, United States General Accounting Office, Washington, DC, February 27, 2002.
- [CJB00] Coriat, M., Jourdan, J., Boisbourdin, F., "The SPLIT Method, *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 147-166.
- [CKB+94] Charo, A., Kolsrud, Bowersox, P., *Remotely Sensed Data: Technology Management and Markets*, U.S. Congress, Office of Technology Assessment, U.S. Government Printing Office, Washington, DC, September 1994.

- [CKM+98] Covington, R., Kelly, J., Mathews, V., *Formal Methods Specification and Verification Guidebook for Software and Computer Systems, Volume I: Planning and Technology Insertion, Release 2.0*, [NASA/TP-98-208193], 1998, accessed on August 9, 2001 from http://eis.jpl.nasa.gov/quality/Formal_Methods/document/NASAgbl.doc.
- [Cla93] Clark, D., "The Application of Modeling Frameworks in the Assessment of Radar Systems," *Colloquium on Computer Modeling and Simulation of Radar Systems*, Institution of Electrical Engineers, London, UK, 1993, pp 1-6.
- [Cla97] Clark, B., *The Effects of Software Process Maturity on Software Development Effort*, Ph.D. dissertation, University of California, Los Angeles, CA, 1997.
- [Cla01] Claxton, J., *Test and Evaluation Management Guide*, Fourth Edition, Defense Acquisition University Press, Fort Belvoir, VA, November 2001.
- [Cle97] Clements, P., *Report of the Reuse and Product Lines Working Group of WISR8*, (CMU/SEI-97-TR-010), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.
- [Cle99] Clements, P., "Software Product Lines: A New Paradigm for the New Century," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, February 1999.
- [CLP+01] Crocker, R., Lee, J., Panwar, M., *Information Technology: Architecture Needed to Guide Modernization of DoD's Financial Operations*, GAO-01-525, United States General Accounting Office, Washington, DC, May 2001.
- [CLR97] Cormen, T., Leiserson, Rivest, R., *Introduction to Algorithms*, joint production of the MIT Press, Cambridge, MA, and McGraw-Hill Book Company, New York, NY, 1990.
- [CM99] Cimera, G., Mai, A., *Wargame 2000 Software Version 4.0 Vision Document*, Joint National Integration Center, Missile Defense Agency, January 16, 2002.
- [CM02] Caffall, D., Michael, J., "Systems-of-Systems Design From an Object Oriented Paradigm," *Monterey 2002, Radical Innovations of Software and Systems Engineering in the Future Workshop Proceedings*, Venice, Italy, October 7-11, 2002, pp. 146-157.
- [CM03] Carr, F., Myers, L., "Interoperability and Reuse through a Modeling and Simulation Common Operating Environment," *Proceedings of the 2003*

Spring Simulation Conference, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2003.

- [CN96] Clements, P., Northrop, L., *Software Architecture: An Executive Overview*, (CMU/SEI-96-TR-003), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, February 1996.
- [CN02] Clements, P., Northrop, L., *Software Product Lines: Practices and Patterns*, Addison-Wellesley, Boston, MA, 2002.
- [Coh94] Cohen, W., *Computer Chaos: Billions Wasted Buying Federal Computer Systems*, Senate Governmental Affairs Committee, October 12, 1994.
- [Coh99] Cohen, S., *Guidelines for Developing a Product Line Concept of Operations*, (CMU/SEI-99-TR-008), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 1999.
- [COH00] Coyle, P., O'Bryon, J., Hullegas, A., "Survey of M&S Use in Acquisition Programs," *Training and Simulation 2000*, Armed Forces Journal, 2000, pp. 32-39.
- [Coh02] Cohen, S., *Product Line State of the Practice Report*, (CMU/SEI-2002-TN-017), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, September 2002.
- [Con86a] Conahan, F., *Measuring Military Capability, Progress, Problems, and Future Direction*, GAO/NSIAD-86-72, United States General Accounting Office, Washington, DC, February 1986.
- [Con86b] Connor, H., *Measuring Military Capability—Progress, Problems, and Future Direction*, 129224, United States General Accounting Office, Washington, DC, March 5, 1986.
- [Con93] Conahan, F., *Simulation Training: Management Framework Improved, But Challenges Remain*, GAO/NSIAD-93-122, United States General Accounting Office, Washington, DC, May, 1993.
- [Con01] Conwell, C., *Process for Evaluating Need for IV&V for Risk Management*, accessed on 21 February 2002 from http://www.sisostds.org/webletter/siso/iss_78/art_428.htm.
- [Coo01a] Cook, D., "Computers and M&S - Modeling and Simulation 101," *Crosstalk: The Journal of Defense Software Engineering*, Vol. 14, No. 1, Software Technology Support Center, Hill AFB, UT, January 2001, pp. 8-10.

- [Coo01b] Cook, J., "Supporting Rapid Prototyping through Frequent and Reliable Deployment of Evolving Components," *12th International Workshop on Rapid System Prototyping*, IEEE Computer Society, 2001, pp.194-199.
- [Cos01] Cosgrove, J., "Software Engineering and the Law," *IEEE Software*, Vol. 18, No. 3, May-June 2001, pp. 14-16.
- [Coy00a] Coyle, P., "TE Readiness—The Myth of Excess Capacity," *ITEA Journal*, March-April 2000, pp. 6-9.
- [Coy00b] Coyle, P., "Evolutionary Acquisition: Seven Ways to Know If You are Placing Your Program at Risk," *PM*, November-December 2000.
- [Coy01] Coyle, P., "The Coyle Report: An Internal Pentagon Study Criticizing the National Missile Defense Test Program," *Minority Staff Report*, Committee on Government Reform, U.S. House of Representatives, June 26, 2001.
- [CR98] Cloud, D., Rainey, L., Ed., *Applied Modeling and Simulation: An Integrated Approach to Development and Operation*, The McGraw-Hill Companies, Inc., New York, NY, 1998.
- [Cra01a] Crain, W., *Department of Defense (DoD) Modeling and Simulation Master Plan Final Report*, Office of the Director of Defense Research and Engineering, Washington, DC, January 3, 2001.
- [Cra01b] Crain, W., "Decision on Findings and Recommendations of the DoD Modeling and Simulation (M&S) Working Group (MSWG) Integration Task Force (ITF)," Office of the Director of Defense Research and Engineering, Washington, DC, August 30, 2001.
- [CPS+96] Casper, L., Powers, E., Steffens, T., "Knowledge-Based Warfare: A Security Strategy for the Next Century," *Joint Forces Quarterly*, Autumn, 1996, pp. 81-89.
- [CR98] Cloud, D., Rainey, L., eds., *Applied Modeling and Simulation: An Integrated Approach to Development and Operation*, McGraw-Hill, New York, NY, 1998.
- [CRB01] Chattopadhyay, S., Roy, C., Bhattacharya, S., "Modeling Constraints as Methods In Object Oriented Data Model," *Proceedings of Monterey Workshop 2001*, Monterey, CA, June 18-22, 2001, pp. 220-227.
- [CRC02a] Coleman Research Corporation, *Patriot Advanced Capability – 3 (PAC-3) Configuration – 3 Initial Operational Test and Evaluation Accreditation*

Request for the Missile Defense System Exerciser, Coleman Research Corporation, El Paso, TX, April 8, 2002.

- [CRC02b] Coleman Research Corporation, *Patriot Advanced Capability – 3 (PAC-3) Configuration – 3 Initial Operational Test and Evaluation Accreditation Request for the Extended Air Defense Testbed*, Coleman Research Corporation, El Paso, TX, April 11, 2002.
- [CS97] Coyle, P., Sanders, P., *Simulation, Test, and Evaluation Process (STEP) Guidelines*, Office of the Under Secretary of Defense (Acquisition & Technology), 4 Dec 1997.
- [CS99] Christie, A., Staley, M., “Simulation of a Software Requirements Development Process,” [News@SEI](#), Vol2, No 2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Spring 1999, pp. 6-7.
- [CS02] Clarke, R., Schmidt, H., *A National Strategy to Secure Cyberspace*, President’s Critical Infrastructure Board, The White House, Washington, DC, September 18, 2002.
- [CSC94a] Computer Sciences Corporation, *A Comparative Analysis of Tri-Service Accreditation Policies and Practices: Vol. I of the Accreditation Requirements Study Report*, JTCG/AS-93-SM-20, Camarillo, CA, February 1994.
- [CSC94b] Computer Sciences Corporation, *Information Requirements in Support of Accreditation: Vol. II of the Accreditation Requirements Study Report*, JTCG/AS-93-SM-20, Camarillo, CA, February 1994.
- [CSS+01] Crnkovic, I., Schmidt, H., Stafford, J., eds., *Component Certification and System Prediction*, Proceedings 4th ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, May 14-15 2001.
- [CT02] Cheston, G., Tremblay, J., “Integrating Software Engineering in Introductory Computing Courses,” *IEEE Software*, Vol. 19, No. 5, September/October 2002, pp. 64-71.
- [CTW98] Chevez, A., Tornabene, C., Wiederhold, G., “Software Component Licensing: A Primer,” *IEEE Software*, September/October 1998, pp. 47-53.
- [CVL+97] Crow, J., DiVito, B., Lutz, R., *Formal Methods Specification and Guidebook for the Verification of Software and Computer Systems, Volume II: A Practitioner’s Companion*, [NASA-GB-001-97], Release 1.0, May 1997, accessed on August 9, 2001 from http://eis.jpl.nasa.gov/quality/Formal_Methods/document/.

- [CW98] Clements, P., Weiderman, N., *Report on the Second International Workshop on Development and Evolution of Software Architecture for Product Families*, (CMU/SEI-98-SR-003), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 1998.
- [CW02] Carr, F., Wilson, K., "Simulation Services for ABCS 7.0: Enhancing War-fighter Capabilities for the Future," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [CWK99] Carriere, S., Woods, S., Kazman, R., *Software Architectural Transformation*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 1999.
- [CWM+00] Campbell, S., Worthington, T., Mahafza, B., *X-Band Radar Project Management Office (XBR PMO) Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, 6 April, 2000.
- [CWS02] Cartwright, J., Wynne, M., Stenbit, J., *Joint Distributed Engineering Plant*, Office of the Secretary of Defense memorandum, Washington, DC, August 5, 2002.
- [CY01] Christie, B., Young, P., "Integrated Development Environment (IDE) for the Construction of a Federation Interoperability Object Model (FIOM)," Master thesis, *Progress Report: Assessment of Defense Modeling and Simulation Office (DMSO) Conceptual Models of the Mission Space*, Naval Postgraduate School, Monterey, CA, December 31, 2001.
- [Cyn92] Cynamon, J., "Chapter II - The Basics," *Simulation Validation Workshop Proceedings (SIMVAL II)*, Military Operations Research Society, Alexandria, VA, 31 March – 2 April 1992, II-3-II-16.
- [DA00] Diem, J., Allred, N., *Modeling and Simulation Employment of Common C4I Interfaces-The Challenges and Successes*, Whitfil Central Technical Support Facility, Ft. Hood, TX, 2000.
- [Dag00] Dager, J., "Cumming's Experience in Developing a Software Product Line," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp.23-46.
- [Dah99] Dahmann, J., *The High Level Architecture and Beyond: Technology Challenges*, Office of the Director of Interoperability (OSD/AT&L), Department of Defense, 1999.

- [Dah01a] Dahmann, J., *Joint Distributed Engineering Plant (JDEP), JDEP Strategy Final Report* Office of the Director of Interoperability (OSD/AT&L), Department of Defense, March 2001.
- [Dah01b] Dahmann, J., *Role of Distributed System of System Testing in the Systems Development and Acquisition Process: The Joint Distributed Engineering Plant (JDEP) in the Context of Simulation-Based Acquisition*, Office of the Director of Interoperability (OSD/AT&L), Department of Defense, September 2001.
- [Dah02] Dahmann, J., *Applying Concepts of Simulation-Based Acquisition to System-of-Systems*, (Draft), MITRE Corporation, 2002.
- [Dan02] Daniels, J., "Modeling with a Sense of Purpose," *IEEE Software*, Vol. 19, No. 1, January-February 2002, pp. 8-10.
- [DAU01] Defense Acquisition University, *System Engineering Fundamentals*, Defense Acquisition University Press, Fort Belvoir, VA, January 2001.
- [Dav91] Davis, R., *Army Training: Computer Simulations Can Improve Command Training in Large-Scale Exercises*, GAO/NSIAD-91-67, United States General Accounting Office, Washington, DC, January, 1991.
- [Dav92a] Davis, P., "An Introduction to Variable-Resolution Modeling and Cross Resolution Model Connection," *Proceedings of Conference on Variable-Resolution Modeling*, RAND National Defense Research Institute, Washington, DC, 5-6 May 1992, pp. 1-43.
- [Dav92b] Davis, P., *Generalizing Concepts and Methods of Verification, Validation, and Accreditation (VV&A) for Military Simulations*, RAND National Defense Research Institute, Santa Monica, CA, 1992.
- [Dav92c] Davis, P., "A Framework for Verification, Validation, and Accreditation," *Simulation Validation Workshop Proceedings (SIMVAL II)*, Military Operations Research Society, Alexandria, VA, 31 March – 2 April 1992, pp. VI-1-VI-22.
- [Dav93] Davis, P., *An Introduction to Variable-Resolution Modeling and Cross Resolution Model Connection*, RAND National Defense Research Institute, Santa Monica, CA, 1993.
- [Dav95a] Davis, P., "An Introduction to Variable-Resolution Modeling," *Warfare Modeling*, John Wiley & Sons, 1995, pp. 5-35.

- [Dav95b] Davis, P., *Aggregation, Disaggregation, and the 3:1 Rule in Ground Combat*, RAND National Defense Research Institute, Santa Monica, CA, 1995.
- [Dav96] Davis, R., *Bottom-up Review: Analysis of DoD War Game to Test Key Assumptions*, GAO/NSIAD-96-170, United States General Accounting Office, Washington, DC, June 1996.
- [Dav98] Davis, P., *Experiments in Multiresolution Modeling (MRM)*, RAND National Defense Research Institute, Santa Monica, CA, 1998.
- [Dav00] Davis, P., "Exploratory Analysis Enabled By Multiresolution, Multiperspective Modeling," *Exploratory Analysis and a Case History of Multiresolution, Multiperspective Modeling*, RAND National Defense Research Institute reprint, Santa Monica, CA, 2001
- [Dav01] Davis, E., "Evaluation of the Extensible Markup Language (XML) as a Means for Establishing Interoperability Between Multiple DoD Databases," Master thesis, *Progress Report: Assessment of Defense Modeling And Simulation Office (DMSO) Conceptual Models of the Mission Space*, Naval Postgraduate School, Monterey, CA, December 31, 2001.
- [DB91] Davis, P., Blumenthal, D., *The Base of Sand Problem: A White Paper on the State of Military Combat Modeling*, RAND Corporation, Santa Monica, CA, 1991.
- [DB99] Davis, P., Bigelow, J., *Experiments in Multiresolution Modeling (MRM)*, RAND National Defense Research Institute, Santa Monica, CA, 1999.
- [DB03] Davis, P., Bigelow, J., *Motivated Metamodels, Synthesis of Cause-Effect Reasoning and Statistical Metamodeling*, RAND, Santa Monica, CA, 2003.
- [DBM99] Davis, P., Bigelow, J., McEver, J., *Analytical Methods for Studies and Experiments on 'Transforming the Force,'* RAND National Defense Research Institute, Santa Monica, CA, 1999.
- [DBM00] Davis, P., Bigelow, J., McEver, J., *Effects of Terrain, Maneuver Tactics, and C4ISR on the Effectiveness of Long-Range Precision Fires: A Stochastic Multiresolution Model (PEM) Calibrated to High-Resolution Simulation*, RAND Corporation, Santa Monica, CA, 2000.
- [DBW87] Davis, P., Bankes., S., Weissler, R., *Formalization of Tradeoff Rules and Techniques For Comprehending Complex Rule-Based Models*, The RAND Corporation, Santa Monica, CA, November, 1987.

- [DC97] Davis, P., Carrillo, M., *Exploratory Analysis of "The Halt Problem"* RAND National Defense Research Institute, Santa Monica, CA, 2001
- [DC02] Dahmann, J., Clark, R., "Joint Distributed Engineering Plant Technical Framework: Applying Industry Standards to System-of-Systems Federation for Interoperability," *Proceedings of the 2002 Fall Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-13, 2002.
- [DCC+93] Deffer, F., Chao, D., Cho, J., *Software Reuse*, United States General Accounting Office memorandum (B-251542), Washington, DC, January 28, 1993.
- [DDH01] Donley, M., Duncan, R., Hillegas, A., *Modeling & Simulation Survey Briefing*, briefing to the Director, Operational Test and Evaluation, Department of Defense, Washington, DC, February 2001.
- [DDW01] Dilliard, G., Dixon, R., Watson, P., "A Product Line Approach For Solving The Army's Live Instrumented Training Needs," *Proceedings of the 2001 European Simulation Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Middlesex, UK, June 25-27, 2001.
- [De100] de Leon, R., *Standardization of Department of Defense (DoD) Chemical and Biological (CB) Models, Simulations, and Associated Data*, Deputy Secretary of Defense memorandum, Washington, DC, November 1, 2000.
- [Dem95a] Demarco, T., *Why Does Software Cost So Much?* Dorset House Publishing, New York, NY, 1995.
- [Dem95b] DeMarco, T., "On Systems Architectures," *Proceedings of Monterey Workshop 1995*, Monterey, CA, September 12-14 1995, pp. 44-57.
- [DFP+02] DiLucca, G., Fasolino, A., Pace, P., "WARE: A Tool for Reverse Engineering of Web Applications," *Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02)*, IEEE Computer Society, 2002.
- [DGH+98] Davis, P., Gompert, D., Hillestad, R., *Transforming the Force: Suggestions for DoD Strategy*, n-3400-DARPA, RAND, Santa Monica, CA, 1992.
- [DH92a] Davis, P., Huber, R., *Variable-Resolution Combat Modeling: Motivations, Issues, and Principles*, RAND National Defense Research Institute reprint, Santa Monica, CA, 2001

- [DH92b] Davis, P., Hillestad, R., (eds.), *Proceedings of Conference on Variable Resolution Modeling*, Washington, DC, 5-6 May, 1992, RAND National Defense Research Institute, Santa Monica, CA, 2001
- [DKC+00] Defourneaux, J., Keeney, T., Crafts, G., Zelickson, J., *MDDC Notes - Combined*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, 2000.
- [DL93] DeMarco, T., Lister, T., "State of the Art vs. State of the Practice," *Software Management*, 4th Edition, IEEE Computer Society Press, Los Alamedas, CA, 1993, pp. 496-500.
- [DLB94] Dampier, D., Luqi, Berzins, V., "Automated Merging of Software Prototypes" *Journal of Systems Integration*, Vol. 4, No. 1, February 1994, pp. 33-49.
- [DM03] Daum, B., Merten, U., *System Architecture with XML*, Morgan Kaufmann Publishers, Amsterdam, NE, 2003.
- [DMF+95] Dyckman, L., McGee, W., Frankel, R., *Global Warming: Limitations of General Circulation Models and Costs of Modeling Efforts*, GAO/RCED-95-164, United States General Accounting Office, Washington, DC, July, 1995.
- [DMH01] Dincel, E., Medvidovic, N., van de Hoek, A., "Measuring Product Line Architectures," *Proceedings of the Fourth International Workshop on Product Family (PFE-4)* Bilbao, Spain, October 2001.
- [DMP+97] Dyckman, L., McGee, W., Pagan, J., *Air Pollution: Limitations of EPA's Motor Vehicle Emissions Model and Plans to Address Them*, GAO/RCED-97-210, United States General Accounting Office, Washington, DC, September 1997.
- [DMS98a] Defense Modeling and Simulation Office (DMSO) Information Paper, *The Conceptual Model of the Mission Space*, Department of Defense, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, Washington, DC, June 11, 1998.
- [DMS98b] Defense Modeling and Simulation Office (DMSO) Information Paper, *The Conceptual Model of the Mission Space and Data Engineering Toolset*, Department of Defense, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, Washington, DC, June 11, 1998.

- [DMS99] Defense Modeling and Simulation Office, *High Level Architecture Federation Development and Execution Process (FEDEP) Model*, Ver.1.5, Office of the Director of Defense Research and Engineering, Washington, DC, December 8, 1999.
- [DMS00a] Defense Modeling and Simulation Office (DMSO), *Conceptual Models of the Mission Space (CMMS) Meta-Data Representation Taxonomy Data Interchange Format (DIF) Version 1.4*, Defense Modeling and Simulation Office, Alexandria, VA, February 1, 2000.
- [DMS00b] Department of Defense, *Verification, Validation and Accreditation (VV&A) Automated Support Tools, A State of the Art Report, Part I*, Defense Modeling and Simulation Information Analysis Center, Washington, DC, December 15, 2000.
- [DN02] Dobrica, L., Niemela, "A Survey on Software Architecture Analysis Methods," *IEEE Transactions on Software Engineering*, Vol. 26, No. 7, July 2002, pp. 638-653.
- [DNC+00] Dent, B, Nash, K., Cooney, M., Bradford, B., *THAAD Notes - Combined*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, 2000.
- [Dod97] Dodaro, G., *Financial Management: Review of the Military Retirement Trust Fund's Actuarial Model and Related Computer Controls*, GAO/AIMD-97-128, United States General Accounting Office, Washington, DC, September, 1997.
- [Dod98] Dodaro, G., *Executive Guide: Measuring Performance and Demonstrating Results of Information Technology Investments*, GAO/AIMD-98-89, United States General Accounting Office, Washington, DC, March 1998.
- [DoD94] Department of Defense, Under Secretary of Defense for Acquisition and Technology, *DoD Modeling and Simulation Management*, DoD Directive 5000.59, Under Secretary of Defense for Acquisition and Technology Washington, DC, January 4, 1994, with Change 1, January 20, 1998.
- [DoD95] Department of Defense, *DoD Modeling and Simulation Master Plan* DoD 5000.59-P, Under Secretary of Defense for Acquisition and Technology, Washington, DC, October 1995.
- [DoD96] Department of Defense, *DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)*, Department of Defense Instruction

No. 5000.61, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, Washington, DC, April 29, 1996.

- [DoD97] Department of Defense, *Department of Defense Interface Standard: Tactical Digital Information Link (TADIL) J Message Standard*, MIL-STD-6016A, Joint Interoperability and Engineering Organization, Department of Defense, Reston, VA, April 30, 1999.
- [DoD98] Department of Defense, *Department of Defense Modeling & Simulation (M&S) Glossary*, DOD 5000.59-M, Under Secretary of Defense for Acquisition and Technology, Washington, DC, Jan 15, 1998.
- [DoD00a] Department of Defense, *DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A) (DRAFT)*, Department of Defense Instruction No. 5000.61, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, Washington, DC, June 27, 2000.
- [DoD00b] Deputy Secretary of Defense Memorandum for Under Secretary of Defense (Acquisition Technology, and Logistics), *Standardization of Department of Defense (Chemical and Biological (CB) Models, Simulations, and Associated Data*, Department of Defense, Washington, DC, November 4, 2000.
- [DoD00c] Department of Defense, *Enterprise Integration Implementation Strategy*, Office of the Assistant Secretary of Defense Command, Control, Communications, and Intelligence, Washington, DC, accessed March 13, 2000 from <http://www.c31.osd.mil/other/ei4toc.html>.
- [DoD00d] Department of Defense, *What is the COE? – About the COE*, Defense Information Systems Agency, Washington, DC, accessed on May 17, 2000 from <http://dii-sw.ncr.disa.mil/coe/docs/flyers/whatis.html>.
- [DoD01a] Department of Defense, *DoD Modeling and Simulation (M&S) Master Plan (DRAFT)*, Department of Defense Instruction No. 5000.71, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, Washington, DC, May 24, 2001.
- [DoD01b] Department of Defense, *DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A) (DRAFT)*, Department of Defense Instruction No. 5000.61, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, Washington, DC, October 5, 2001.
- [DoD02a] Department of Defense, *DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A) (DRAFT)*, Department of Defense

Instruction No. 5000.61, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, Washington, DC, July 10, 2002.

- [DoD02b] Department of Defense, *Mandatory Procedures for Major Defense Acquisition Programs and Major Automated Information Systems*, DoDD 5000.2-R, Under Secretary of Defense for Acquisition and Technology, Washington, DC, April 5, 2002.
- [DoD02c] Department of Defense, *DoD Chemical and Biological (CB) Modeling and Simulation (M&S) Master Plan (MSMP) (DRAFT)*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, 2002.
- [DoD02d] Department of Defense, *C4ISR Information Superiority Modeling and Simulation Master Plan*, Office of the Assistant Secretary of Defense Command, Control, Communications, and Intelligence, Washington, DC, March 12, 2002.
- [DoD02e] Department of Defense, *Information Assurance, Department of Defense Directive 8500.1*, Office of the Deputy Secretary of Defense, Washington, DC, October 24, 2002.
- [DoD02f] Department of Defense, *Operation of the Defense Acquisition System*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, April 5, 2002.
- [DoD02g] Department of Defense, *The Defense Acquisition System*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, October 30, 2002.
- [DoD02h] Department of Defense, *Operation of the Defense Acquisition System*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, October 30, 2002.
- [DoD02i] Department of Defense, *Interim Defense Acquisition Guidebook*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, October 30, 2002.
- [DoD03a] Department of Defense, *Department of Defense Modeling and Simulation Strategic Plan 2003-2012, (DRAFT)*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, April 2003.

- [DoD03b] Department of Defense, *The Defense Acquisition System*, Number 5000.1, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, May 12, 2003.
- [DoD03c] Department of Defense, *Operation of the Defense Acquisition System*, Number 5000.2, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, May 12, 2003.
- [DON99] Department of the Navy, *Verification, Validation, and Accreditation (VV&A) of Models and Simulations*, SECNAV Instruction 52000.40, Department of Defense, Washington, DC, April, 1999.
- [Don00] Donohoe, P., ed., *Software Product Lines, Experience and Research Directions*, Kluwer Academic Publishers, Norwell, MA, 2000.
- [DOT89] Director, Operational Test and Evaluation, *Policy for the Application of Modeling / Simulation in Support of OT&E*, Director, Operational Test and Evaluation, Washington, DC, 1989.
- [DOT99] Director, Operational Test and Evaluation, *Missile Defense and Related Programs, Director, Operational Test & Evaluation (DOT&E) FY'99 Annual Report*, Director, Operational Test and Evaluation, Washington, DC, February 2000.
- [DOT00] Director, Operational Test and Evaluation, *Annual Report FY2000 Department of Defense*, Director, Operational Test and Evaluation, Washington, DC, February 2001.
- [DOT01a] Director, Operational Test and Evaluation, "Assuring M&S Credibility for Defense Acquisition and T&E: Survivability, Lethality, and Mission Effectiveness," *Department of Defense M&S Workshop Quick-Look Outbrief*, Reno, NV, February 12-15, 2001.
- [DOT01b] Director, Operational Test and Evaluation, *FY2001 Annual Report for the Office of the Director, Operational Test & Evaluation*, Director, Operational Test and Evaluation, Washington, DC, February 2002.
- [Dou01] Dougherty, F., *Functional Description of the Mission Space Knowledge Acquisition Product Style Manual*, IMC Inc, Sterling, VA, March 1, 2001.
- [Dow92] Downes-Martin, S., *Seamless Simulation Literature Survey*, Institute for Defense Analyses, Alexandria, VA, April, 1992.
- [DPB+01] Darilek, R. Perry, W., Bracken, J., *Measures of Effectiveness for the Information-Age Army*, RAND, Santa Monica, CA, 2001.

- [DPG02] News Transcript, *Background Briefing on the Defense Planning Guidance*, the Pentagon, Washington, DC, May 10, 2002.
- [DPR+94] Daich, G., Price, G., Ragland, B., *Software Test Technology Report*, Software Technology Support Center, Hill AFB, UT, August 1994.
- [DRC99] Dynamics Research Corps, *CMMS Knowledge Acquisition Tool (KAT)*, DMSO CMMS Documentation, accessed on September 15, 2001 from <http://www.dmsomil/index.php?page=75>.
- [Dro95] Dromey, G., "A Model for Software Product Quality," *IEEE Transaction on Software Engineering*, Vol. 21, No. 2, February 1995, pp. 146-162.
- [Dro96] Dromey, G., "Cornering, the Chimera," *IEEE Software*, Vol. 13, No. 1, January 1996, pp. 33-43.
- [DRS99] Dorda, S., Robert, J., Seacord, R., *Theory and Practice of Enterprise Java-BeanTM Portability*, (CMU/SEI-99-TN-005), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 1999.
- [DS00] Doerr, B., Sharp, D., "Freeing Product Line Architectures from Execution Dependencies," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 313-330.
- [DSB+99] Dahmann, J., Salisbury, M., Barry, P., "HLA and Beyond: Interoperability Challenges," *Proceedings of the 1999 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 1999.
- [DSB+01] Deitz, P., Sheehan, J., Harris, B., *Military Domain Representation Framework (MDRF): A General Framework & Methodology for Analyzing Weapon System Effectiveness*, Defense Modeling and Simulation Office presentation, August 16, 2001.
- [DSH00] Patin, M., Savage, M., Higley, J., *Sensor Simulation Test Bed (SSTB) Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, April 6, 2000.
- [DT01a] Dahmann, J., Tindall, J., "Distributed Environments for Integration and Interoperability Testing of Systems of Systems: A Strategy for Extending the Joint Distributed Engineering Plant (JDEP)," *Proceedings of the 2001*

Spring Simulation Conference, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30 2001.

- [DT01b] Dahmann, J., Tindall, J., *Joint Distributed Engineering Plant (JDEP) Strategy Final Report*, The MITRE Corporation, March 2001.
- [DT02] Duggins, S., Thomas, B., "An Historical Investigation of Graduate Software Engineering Curriculum," *Proceedings of the 15th Conference on Software Engineering and Training (CSEET'02)*, 2002.
- [Dun96] Duncan, W., *A Guide to the Project Management Body of Knowledge*, Project Management Institute, Newtown Square, PA, 1996.
- [DWE+00] Dobson, C., Witczak, J., Edwards, A., *National Missile Defense Weapon System Program Management Office (NMD WSPMO) Interview*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, January 13, 2000.
- [EB01] Eckler, A., Burr, S., Ed., *Mathematical Models of Target Coverage and Missile Allocation*, Military Operations Research Society, Alexandria, VA, 1972, reissued 2001.
- [Ebe01] Ebert, C., "Hands-On Experience Implementing a Product Line," *IEEE Software*, January/February 2001, pp 105-106.
- [EBG+94] Etter, D., Brancatti, T., Gardner, J., *Naval Research Advisory Committee Report on Modeling and Simulation NRAC 94-3*, Office of the Assistant Secretary of the Navy (Research, Development, and Acquisition) Washington, DC, November 1994.
- [ECP02] Eirich, P., Coolahan, J., Purdy, E., "A Collaborative Environment Architecture for Future Combat Systems (FCS) Modeling and Simulation," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [Edd92] Eddy, F., "Object-Oriented Modeling: Holistic and Variable-Resolution Views," *Proceedings of Conference on Variable-Resolution Modeling*, RAND National Defense Research Institute, Washington, DC, May 5-6, 1992, pp. 44-51.
- [Ede97] Eden, S., "Knowledge-Based Warfare Implications," *Military Review*, March-April 1997.

- [EFL+97] Ellis, J., Fisher, D., Lonstaff, T., *Report to the President's Commission on Critical Infrastructure Protection*, (CMU/SEI-97-SR-003), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, January 1997.
- [EGF+02] Eick, S., Graves, T., Karr, A., "Visualizing Software Changes," *IEEE Transactions on Software Engineering*, Vol. 28, Issue, April 2002, pp. 396-412.
- [EHH97] Edwards, L., Hall, B., Hopp, T., *National Missile Defense, Schedule and Technical Risks Represent Significant Development Challenges*, GAO/NSIAD-98-28, United States General Accounting Office, Washington, DC, December 1997.
- [EHH+98] Edwards, L., Hall, B., Hand, D., *National Missile Defense, Even With Increased Funding Technical and Schedule Risks Are High*, GAO/NSIAD-98-153, United States General Accounting Office, Washington, DC, June 1998.
- [Eic94] Eichblatt, Emil J., "Use of Simulation to Evaluate Tactical Missile Performance," *ITEA Journal*, September-October 1994, pp. 30-34.
- [EKH00] Ellis, L., Kern, P., Hollis, W., *Planning Guidelines for Simulation and Modeling Acquisition, Requirements and Training (SMART)*, Department of the Army, Washington, DC, September 15, 2000.
- [EKH01] Ellis, L., Kern, P., Hollis, W., *Simulation and Modeling Acquisition, Requirements and Training (SMART) Reference Guide*, Department of the Army, Washington, DC, April 2001.
- [EMR02] Elbaum, S., Malishevsky, A., Rothermel, G., "Test Case Prioritization: A Family of Empirical Studies," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 2, February 2002, pp. 159-182.
- [EP01] Eaton, A., Pace, D., *THAAD-Patriot Interoperability Possibilities, U.S. Army Anti-Tactical Missile (ATM) Program Review Panel Report*, The John Hopkins University Applied Physics Laboratory, Laurel, MD, September 2001.
- [ES02] Emami, R., Sulesky, A., "Composable Modeling and Simulation Workshop," *DMSO Workshop on Composable M&S*, July 9, 2002
- [Ett99] Etter, D., *Joint Simulation System (JSIMS) Recommendations*, memorandum, Director of Defense Research and Engineering, Washington, DC, November 19, 1999.

- [Eva96] Evans, M., *Software Acquisition Best Practices*, Integrated Computer Engineering presentation, Inc., Campbell, CA, 1996.
- [Eva00] Evans, M., *SPMN Director Identifies 16 Critical Software Practices*, Integrated Computer Engineering, Inc., Campbell, CA, 2000.
- [FAB+99a] Ferguson, J., Ahern, D., Bate, R., *Capability Maturity Model-Integrated Systems/Software Engineering (CMMI-SE/SW), Staged Representation*, Vol., I, Ver., 0.2b Carnegie Mellon University and Software Engineering Center, Pittsburgh, PA, 1999.
- [FAB+99b] Ferguson, J., Ahern, D., Bate, R., *Capability Maturity Model-Integrated Systems/Software Engineering (CMMI-SE/SW), Staged Representation*, Vol., II, Ver., 0.2b Carnegie Mellon University and Software Engineering Center, Pittsburgh, PA, 1999.
- [FAB+99c] Ferguson, J., Ahern, D., Bate, R., *Capability Maturity Model-Integrated Systems/Software Engineering (CMMI-SE/SW), Continuous Representation*, Vol., I, Ver., 0.2b Carnegie Mellon University and Software Engineering Center, Pittsburgh, PA, 1999.
- [FAB+99d] Ferguson, J., Ahern, D., Bate, R., *Capability Maturity Model-Integrated Systems/Software Engineering (CMMI-SE/SW), Continuous Representation*, Vol., II, Ver., 0.2b Carnegie Mellon University and Software Engineering Center, Pittsburgh, PA, 1999.
- [FAK+00] Ford, J., Archer, J., Klar, R., Silvey, S., Cross, J., Giola, T., Walker, J., *ISTC Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, 5 April, 2000.
- [Fal97] Fallin, H., *Simulation Support Plan Guidelines*, Department of the Army, Washington, DC, May 1997.
- [Fal01] Falat, M., *Requirements for Process Evaluation Methods and Their Application* (Draft), DoD Software Evaluation IPT Chartered by the Deputy Under Secretary of Defense for Science and Technology, Washington, DC, February 15, 2001.
- [Fav02] Favre, Jean-Marie, "A New Approach to Software Exploration: Back-packing with G^{SEE} ," *Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02)*, IEEE Computer Society, 2002.

- [Fay98a] Fay, J., "Measuring the Fidelity of a Vehicle Simulation I: Level of Detail," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [Fay98b] Fay, J., "Measuring the Fidelity of a Vehicle Simulation II: List of Input Parameters," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [FBK+01] Fisler, K., Batory, D., Krishnamurthi, S., Liu, J., "A Model Checking Framework for Layered Command and Control Software," *Proceedings of Monterey Workshop 2001*, Monterey, CA, June 18-22, 2001, pp. 103-115.
- [FC99] Florac, W., Carleton, A., *Measuring the Software Process: Statistical Process Control for Software Process Improvement*, Addison-Wesley Longman, Inc., Reading, MA, 1999.
- [FCF+96] Ferguson, J., Cooper, J., Falat, M., *Software Acquisition Capability Maturity Model (SA-CMM)*, Ver. 1.01, (CMU/SEI-96-020), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, December 1996.
- [FDR+98] Fisher, M., Damer, R., Reed, L., *Software Acquisition Improvement Framework (SAIF) Definition*, (CMU/SEI-98-TR-003), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 1998.
- [FEA02a] Federal Enterprise Architecture Working Group, *E-Gov Enterprise Architecture Guidance (Common Reference Model)*, Draft Ver. 2.0, Federal Enterprise Architecture Working Group, Washington, DC, July 25, 2002.
- [FEA02b] Federal Enterprise Architecture Program Management Office, *Component-Based Architecture (Guidance and Recommendations) Working Draft* Ver. 1.3, Solution Architecture Working Group, Washington, DC, October 24, 2002.
- [FED01] Defense Modeling and Simulation Office, *High Level Architecture Federation Development and Execution Process (FEDEP) Model*, (DRAFT 3), Office of the Director of Defense Research and Engineering, Washington, DC, November 12, 2001.
- [FEE+99] Frankel, M., Eftakari, R., Evers, W., *The Defense Science Board Task Force on Tactical Battlefield Communications Final Report*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, 30 December, 1999.

- [Fer01] Ferguson, J., "Crouching Dragon, Hidden Software: Software in DoD Weapon Systems," *IEEE Software*, Vol. 18, No. 4, July-August 2001, pp. 105-107.
- [FG99] Frelinger, D., Gabrielle, M., *Remote Sensing Operational Capabilities, Final Report*, MR-1172.0-OSTP, RAND Science and Technology Policy Institute, Santa Monica, CA, October 1999.
- [FG02] Frolund, S., Guerraoui, R., "e-Transactions: End-to-End Reliability for Three-Tier Architectures," *IEEE Transactions on Software Engineering*, Vol. 28, No. 4. April 2002, pp. 378-395.
- [FGW+99] Fields, C., Gold, T., Welch, L., *Report of the Defense Science Board on Advanced Modeling and Simulation for Analyzing Combat Concepts in the 21st Century*, " Department of Defense, Under Secretary of Defense for Acquisition and Technology, Washington, DC, June 2, 1999.
- [FH01] Figart, G., Hays, C., "Verification, Validation, and Accreditation of Hardware-In-The-Loop Systems," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [FHR00] Faulk, S., Harmon, R., Raffo, D., "Value-Based Software Engineering (VBSE), *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 205-224.
- [Fio01] Fiorino, T., *Engineering Manufacturing Readiness Levels*, White Paper, Andrulis Corporation, October 30, 2001.
- [Flo02a] Flourney, R., "Leveraging Human Behavior Modeling Technologies to Strengthen Simulation-Based C2 System Acquisition," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [Flo02b] Floodeen, D., *Improved Engineering Duty Officer Capability Through Software Engineering*, Naval Postgraduate School briefing to RADM Paige, Missile Defense Agency, May 8, 2002.
- [FMA+84] Fletcher, J., McMillan, Adams, D., *Report on the Study on Eliminating the Threat Posed by Nuclear Ballistic Missiles (U), Battle Management, Communications, and Data Processing (U)*, Volume V, February 1984.

- [FMF02] Fitzgibbons, B., McLean, T., Fujimoto, R., "RTI Benchmark Studies," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [FMG94] Formal Methods Group, Department of Defense, Fort George C. Mead, "Formal Methods Experience and Recommendations," *Proceedings of Monterey Workshop 1994*, Monterey, CA, September 7-9 1994, pp. 12-115.
- [FMM+03] Fernandez, M., Malhotra, A., Marsh, J., *Xquery 1.0 and Xpath 2.0 Data Model*, World Wide Web Consortium (W3C), WSC working draft May 2, 2003.
- [FNP+02] Fritman, M., Noble, O., Parker, R., *Information Resource Management: Comprehensive Strategic Plan Needed to Address Mounting Challenges*, GAO/02-292, United States General Accounting Office, Washington, DC, February 2002.
- [FOP+01] Fields, C., Odeen, P., Poste, G., *Protecting the Homeland: Report of the Defense Science Board 2000 Summer Study, Executive Summary*, Volume I, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, February, 2001.
- [Fow97] Fowler, M., *Analysis Patterns: Reusable Object Models*, Addison-Wesley, Menlo Park, CA, 1997.
- [Fow01] Fowler, M., "Reducing Coupling," *IEEE Software*, July-August 2001, pp. 102-104.
- [FPC97] Florac, W., Park, R., Carleton, A., *Practical Software Measurement: Measuring for Process Management and Improvement*, CMU-SEI-97-HB-003), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 1997.
- [Fra94] Frame, J., *The New Project: Tools For An Age of Rapid Change, Corporate Reengineering, and Other Business Realities*, Jossey-Bass Inc., San Francisco, CA, 1994.
- [Fra03] Frankel, D., *Model Driven Architecture: Applying MDA to Enterprise Computing*, Wiley Publishing, Inc., Indianapolis, IN, 2003.
- [FRC+02] Frey, P., Radhakrishnan, R., Carter, H., "A Formal Specification and Verification Framework for Time Warp-Based Parallel Simulation," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 1, January 2002, pp. 58-78.

- [Fri01] Fries, A., "Another 'New' Approach For 'Validating' Simulation Models," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [FS94] Fife, D., Springsteen, B., *Software Capability Evaluation of Contractors*, Institute for Defense Analysis, Alexandria, VA, 1994.
- [FT98] Frost, R., Thomen, D., "Simulation Based Acquisition," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13, 1998.
- [FV96] Fedchak, E., Vienneau, R., *A History of Software Measurement at Rome Laboratory*, Data and Analysis Center, Rome Laboratory, Rome, NY, November 25, 1996.
- [FY97] Foster, L., Yelmgren, K., "Accuracy In DoD High Level Architecture Federations," *Proceedings of the 1997 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-12, 1997.
- [GA03] Garland, J., Anthony, R., *Large-Scale Software Architecture, A Practical Guide Using UML*, John Wiley & Sons Ltd., West Sussex, England, 2003.
- [GAB97] Gallagher, B., Alberts, C., Barbour, R., *Software Acquisition Risk Management Key Process Area (KPA) – A Guidebook Version 0.20*, (CMU/SEI-97-HB-002), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 3, 1997.
- [Gac95] Gacek, C., "Exploiting Domain Architectures in Software Reuse," *ACM-SIGSOFT SSR*, 1995.
- [Gal99] Gallagher, B., *Software Acquisition Risk Management Key Process (KPA)– A Guidebook*, Ver. 1.02, (CMU/SEI-99-HB-001), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, October 1999.
- [Gan95] Gansler, J., *Defense Conversion*, MIT Press, 1995.
- [Gan98] Gansler, J., *Modeling and Simulation (M&S) in Defense Acquisition*, Under Secretary of Defense memorandum, Washington, DC, October 26, 1999.
- [Gan99] Gansler, J., *Software Evaluations for ACAT I Programs*, Under Secretary of Defense memorandum, Washington, DC, October 26, 1999.

- [Gan00] Gansler, J., *High Level Architecture (HLA) for Simulation Memorandum of Agreement (MOA)*, Under Secretary of Defense memorandum, Washington, DC, November 3, 2000.
- [GAO95] Garlan, D., Allen, R., Ockerbloom, J., "Architectural Mismatch or Why it's so Hard to Build Systems out of Existing Parts," *Proceedings of the 17th International Conference on Software Engineering*, Seattle, WA, April 1995.
- [Gar02] Garrison, K., "Component-based Architecture and Modeling and Simulation," *DMSO Workshop on Composable M&S*, July 9, 2002.
- [Gau98] Gauker, E., (Ed.), *A Catalog of Department of Defense Medical Models and Simulations*, Ver. 2.0, Joint Services Medical Readiness Modeling & Advisory Group, downloaded from DTIC, August 24, 1998.
- [GB99] Gluch, D., Brockway, J., *An Introduction to Software Engineering Practices Using Model-Based Verification* (CMU-SEI-99-TR-005), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 1999.
- [GBB+99] Gold, T., Bayer, M., Bovin, D., *Report of the Defense Science Board Task Force on DoD Warfighting Transformation*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, September 1999.
- [GBL01] Ge, J., Berzins, V., Luqi, "Computer Aided Prototyping in a Distributed Environment," *Interim Progress Report, Engineering Automation for Reliable Software*, Naval Postgraduate School, Monterey, CA, September 2001, pp. 46-51.
- [Geb93] Gebicke, M., *Army Training: Commanders Lack Guidance and Training for Effective Use of Simulations*, GAO/NSIAD-93-211, United States General Accounting Office, Washington, DC, August 1993.
- [Geb94a] Gebicke, M., *Military Training: Lessons Learned and Their Implications for the Future*, GAO/NSIAD-94-128, United States General Accounting Office, Washington, DC, August 1993.
- [Geb94b] Gebicke, M., *Joint Simulation Training*, GAO/NSIAD-94-249R, United States General Accounting Office, Washington, DC, August 19, 1994.
- [Geb95] Gebicke, M., *Military Training: Cost Effective Development of Simulations Presents Significant Challenges*, GAO/NSIAD-96-44, United States General Accounting Office, Washington, DC, November 1995.

- [GF97] Gross, D., Freeman, R., "Measuring Fidelity in HLA Simulation," *Proceedings of the 1997 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-12, 1997.
- [GF00] Goldenson, D., Fisher, M., *Improving the Acquisition of Software Intensive Systems* (CMU-SEI-2000-TR-003), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August, 2000.
- [GF02] Glasgow, P., Fossett, C., "M&S, VV&A and T&E, A Report by the Committee on M&S and VV&A in T&E," *Military Operations Research Society (MORS) Workshop*, Albuquerque, NM, October 15-17, 2002.
- [GFM01] Goalwin, P., Feinberg, J., Mayne, P., "A Detailed Look at Verification, Validation, and Accreditation (VV&A) Automated Support Tools," *Proceedings of the 2001 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 9-14, 2001.
- [GFT98] Gross, D., Freeman, R., Tucker, W., "A Case Study in Fidelity Differential Measurements," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL. March 9-13, 1998.
- [GGR+94] Ganska, R., Grotzky, J., Rubenstein, J., *Requirements Analysis and Design Technology Report*, Software Technology Support Center, Hill AFB, UT, March 1994.
- [GGR+95] Ganska, R., Grotzky, J., Rubenstein, J., *Requirements Engineering and Technology Report*, Software Technology Support Center, Hill AFB, UT, October 1995.
- [GGU+97] Gimble, T., Guli., Ugone, M., *Implementation of the DoD Joint Technical Architecture*, 98-023, Department of Defense, Office of the Inspector General, Washington, DC, November 18, 1997.
- [GHL+01] Gustavson, P., Hancock, J., Lutz, B., *BOM Study Group Final Report*, SISO-REF-005-2001, Simulation Interoperability Standards Organization (SISO), Orlando, FL. May 15, 2001.
- [GHO+01] Gustavson, P., Hancock, J., O'Connor, *BOM Methodology Strawman (BMS) Specification*, Ver.0.7, SISO-REF-006-2001, Simulation Interoperability Standards Organization (SISO), Orlando, FL. May 15, 2001.
- [Gib94] Gibbs, W., "Software's Chronic Crisis," *Scientific American*, September 1994, pp. 86-95.

- [Gia92] Giadrosich, D., "Validating Models and Simulations," *Simulation Validation Workshop Proceedings (SIMVAL II)*, Military Operations Research Society, Alexandria, VA, March 31–April 2, 1992, pp. IV-1-IV-16.
- [GIG00] DoD Chief Information Officer, *Global Information Grid*, information brief to the Joint Requirements Oversight Council (JROC), the Pentagon, Washington, DC, March 16, 2000.
- [GJ00] Gaver, D., Jacobs, P., *DISC-O-TIC: A Discrete-Time Analytical Meta Model for Use in Combat Systems Studies that Utilize High Resolution Simulation Models*, Preliminary Report, NPS-IJWA-01-009, Naval Postgraduate School, Monterey, CA, December 2000.
- [GKB+97] Gerry, E., Kameny, I., Braddock, J., *Army Simulation Implementation and Use*, Army Science Board, the Deputy Under Secretary of the Army (Operations Research) Washington, DC, April 1997.
- [GKB01] Ge, J., Kin, B., Berzins, V., "A Software Agent Framework for Distributed Applications, *Progress Report: Assessment of Defense Modeling And Simulation Office (DMSO) Conceptual Models of the Mission Space*, Naval Postgraduate School, Monterey, CA, December 31, 2001.
- [Gla01] Glass, R., "Frequently Forgotten Fundamental Facts about Software Engineering," *IEEE Software*, Vol. 18, No. 3, May-June 2001, pp. 110-112.
- [GLT+02] Griffin, A., Lacetera, J., Tolk, A., "C4ISR/SIM Technical Reference Model Study Group Final Report," *Proceedings of the 2002 Fall Simulation Conference*, Simulation Interoperability Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 2002.
- [GM01] Ghosh, S., Mathur, A., "Certification of Distributed Component Computing Middleware and Applications," Crnkovic, I., Schmidt, H., Stafford, J., eds., *Component Certification and System Prediction*, Proceedings 4th ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, May 14-15, 2001.
- [GMS+96] Glasgow, P., Muessig, P., Sikora, J., *Department of Defense Verification, Validation and Accreditation (VV&A) Recommended Practices Guide*, Office of the Director of Defense Research and Engineering Defense Modeling and Simulation Office, Washington, DC, November 1996.
- [GMW89] Guerrero, P., McGee, W., Wisnoski, E., *Air Pollution: Improved Atmospheric Model Should Help Focus Acid Rain Debate*, GAO/RCED-90-14, United States General Accounting Office, Washington, DC, November 3, 1989.

- [GMW00a] Gansler, J., Money, A., Woodward, J., *DoD Policy Change—Cancellation of the Technical Architecture Framework for Information Management (TAFIM)*, Office of the Secretary of Defense memorandum, Washington, DC, January 7, 2000.
- [GMW00b] Gansler, J., Money, A., Woodward, J., *Promulgation of DoD Technical Reference Model (DoD TRM) Version 1.0*, Office of the Secretary of Defense memorandum, Washington, DC, March 21, 2000.
- [Gor96] Gorman, M., *Database Objects The Foundation Stones of Enterprise Database*, Whitmarsh Information Systems Corporation, Bowie, MD, 1996.
- [Goz00] Gozel, R., *Fitting Firepower Score Models to the Battle of Kursk Data*, Master thesis, Naval Postgraduate School, Monterey, CA, September 2000.
- [GP96] Gargaro, A., Peterson, A., *Transitioning a Model-Based Software Engineering Architectural Style to Ada 95*, (CMU/SEI-96-TR-017), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 1996.
- [GPH+01] Gilmore, M., Gupta, R., Hollomon, D., *Critical Infrastructure Protection: Significant Challenges in Developing National Capabilities*, GAO-01-323, United States General Accounting Office, Washington, DC, April 2001.
- [Gre97] Greaney, K., *The Year 2000 Problem: Catalyst or Cataclysm for Future Information Operations?*, USAWC strategic research project, U.S. Army War College, Carlisle, PA, March 26, 1997.
- [Gre99] Greenhill, J., *Distributed Interactive Simulation (DIS)*, MS Thesis, Virginia Commonwealth University, Richmond, VA, October 1999.
- [Gre02a] Greaney, K., *Modeling and Simulation Immersion Day*, briefing by COL Kevin J. Greaney, Director, Models and Simulations, Missile Defense Agency, Washington, DC, March 21-22, 2002.
- [Gre02b] Greaney, K., *Model & Simulation Fidelity Technical Working Group*, memorandum, Missile Defense Agency, Washington, DC, October 24, 2002.
- [Gre03a] Greaney, K., *Defense Science Board (DSB) Phase III Study Missile Defense Agency (MDA) Model & Simulation (M&S) Program*, Vol. I, Missile Defense Agency, Washington, DC, March 27, 2003.

- [Gre03b] Greaney, K., *Defense Science Board (DSB) Phase III Study Missile Defense Agency (MDA) Element M&S Program*, Missile Defense Agency, Huntsville, AL, April 9, 2003.
- [GRG+02] Gustavson, P., Root, L., Goss, S., "CODE-Net: XML, SOAP and Simulation Development," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002
- [GRH01] Gross, D., Roza, Z., Harmon, S., "Report Out of the Fidelity Experimentation Implementation Study Group," *Proceedings of the 2000 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 26-31, 2000.
- [GRH+01] Gustavson, P., Root, L., Hancock, J., "The Base Object Model (BOM) Primer: A Distilled Look at a Component Reuse Methodology for Simulation Interoperability," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2001.
- [Gri00] Griss, M., "Implementing Product-Line Features by Composing Aspects," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 271-288.
- [Gro99] Gross, D., "Report from the Fidelity Implementation Study Group," *Proceedings of the 1999 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 14-19, 1999.
- [GRS02] Guilford, S., Rugg, G., Scott, N., "Pleasure and Pain: Perception Bias and Its implications for Software Engineering," *IEEE Software*, May-June 2002, pp. 63-69.
- [GS99] Gibson, L., Shea, D., *Management Challenges of Simulation Based Acquisition*, Center for Naval Analysis, April 28, 1999.
- [GS01] Garlan, D., Schmert, B., "Component-Based Software Engineering in Pervasive Computing Environments," Crnkovic, I., Schmidt, H., Stafford, J., eds., *Component Certification and System Prediction*, Proceedings 4th ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, May 14-15 2001.
- [Gua98] Guarino, N., "Formal Ontology and Information Systems," *Formal Ontology in Information Systems, Proceedings of FOIS '98*, Trento, Italy, June 6-8 1998, pp. 2-15.

- [GUS+00] Gimble, T., Ugone, M., Santoni, C., *Summary of Audits of Acquisition of Information Technology*, D-2000-162, Department of Defense, Office of the Inspector General, Washington, DC, July 13, 2000.
- [GW96] Gravitz, P., Jordan, W., "Utilizing IDEF0 For Examination of the Individual Activities of the DIS VV&A Process Model," *11th DIS Workshop*, September 26-30, 1994.
- [GWB+99] Gold, T., Welch, L., Braddock, J., *Report of the Defense Science Board Task Force on Advanced Modeling and Simulations for Analyzing Combat Concepts in the 21st Century*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, May 1999.
- [GYL03] Graffagnini, J., Youngblood, S., Lewis, R., *An Overview of the Verification, Validation, and Accreditation (VV&A) Process for the HLA FEDEP*, accessed on March 7, 2003 from <http://www.dmsomil/public/transition/vva/currentinitiatives>.
- [GZ98] Graves, C., Zack, J., *The Application of High Resolution Dynamical Numerical Models as a Tool to Infer Climate Statistics: January Simulations*, Technical Report, St. Louis University, May 28, 1998.
- [HAC+99] Heebner, D., Adolph, C., Christie, T., *Report of the Defense Science Board Task Force on Test and Evaluation*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, September 1999.
- [Had98] Haddix, F., "Prescribing Fidelity and Resolution for CMMS," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL. March 9-13, 1998.
- [Hal95] Hall, M., Ed., *Process Improvement: The DSMC Approach*, Defense System Management College Press, Fort Belvoir, VA, 1995.
- [Hal01a] Hall, D., "Statistical vs. Analytical Significance: How Much V&V is Enough?," accessed June 7, 2001 from <http://www.nawcwpns.navy.mil/~jasa/library/vva/papers/Statistics/Stats.html>.
- [Hal01b] Halle, R., "XML Schema Integration," Berzins, V. ed., *XML Technology Assessment Progress Report (7/17/2000 – 12/31/2000)*, NPS-SW-01-002, Naval Postgraduate School, Monterey, CA, January 2001, pp. 440-443.
- [Ham96] Hamilton, J., *Multilevel Simulation of Discrete Network Models*, Ph.D. dissertation, Texas A&M University, TX, August 1996.

- [Ham00a] Hamre, J., *DoD Chief Information Officer Board*, Deputy Secretary of Defense memorandum, Washington, DC, March 31, 2000.
- [Ham00b] Hamre, J., *Global Information Grid*, Deputy Secretary of Defense memorandum, Washington, DC, March 31, 2000
- [Ham01a] Hamilton, J., "Multilevel Simulation of Discrete Network Models," Joint Forces Program Office briefing, accessed October 11, 2001 from <http://www.drew.hamilton.com/download.htm>.
- [Ham01b] Hammer, J., *The Process Enterprise: An Executive Perspective*, Second Edition, Hammer and Company, 2001.
- [Har97] Hartley, D., *Defense Information Infrastructure (DII) Common Operating Environment (COE), Version 3.1, Baseline Specifications*, Center for Computer Systems Engineering, April 1997.
- [Har98a] Hartley, D., *Defense Information Infrastructure (DII) Common Operating Environment (COE) Configuration Management Plan, Version 2.0*, Center for Computer Systems Engineering, April 1, 1998.
- [Har98b] Harmon, S., "Dependencies of Fidelity upon Simulation Interoperability," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [Har99a] Harn, M., *Computer-Aided Software Evolution Based on Inferred Dependencies*, Computer Science Department, Naval Postgraduate School, Monterey, CA, 1999.
- [Har99b] Harmon, S., *An Update to the FEX Fidelity Framework*, September 15, 1999, SISO Document Library, accessed on April 1, 2002 from http://www.sisostds.org/doclib/obtain_doc.cfm?record_id+REF_1001075.
- [Har01a] Hartman, F., *Integration Task Force Assessment Briefing*, Defense Modeling and Simulation Office briefing, July 10, 2001.
- [Har01b] Hartman, F., *M&S Integration Task Force Decision Update Briefing*, Defense Modeling and Simulation Office briefing, November 20, 2001.
- [Har01c] Harth, K., *Research on An Enabling Infrastructure for Distributed Simulation*, Master thesis, State University of New York, Buffalo, NY, February 1, 2001.

- [Hay03a] Haynes, W., *The Defense Transformation for the 21st Century Act*, Department of Defense proposed bill to Congress, General Counsel of the Department of Defense, Washington, DC, April 10, 2003.
- [Hay03b] Haycock, B., *Review of Draft Service Component and Technical Reference Models*, FEA Program Manager memorandum, January 31, 2003.
- [HB98a] Hodge, D., Bradley, B., *Army Standard Platform Object*, Technical Report TR-634, U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, July 1998.
- [HB98b] Hodge, D., Bradley, B., *Army Standard Unit Object*, Technical Report TR-638, U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, July 1998.
- [HB98c] Hefley, W., Curtis, B., *People CMM-Based Assessment Method Description*, Ver.1.0, (CMU/SEI-98-TR-012), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 1998.
- [HB99] Hieb, M., Blalock, J., "Data Alignment between Army C4I Databases and Army Simulations," *Proceedings of the 1999 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 1999.
- [HBD+01] Hillegas, A., Backschies, J., Donley, M., *The Use of Modeling and Simulation (M&S) Tools in Acquisition Program Offices: Results of a Survey*, Hicks & Associates, McLean, VA, January 31, 2001.
- [HBF92] Harshbarger, E., Bennett, B., Frelinger, D., "An Approach to Hierarchies of Models: Process Independence," *Proceedings of Conference on Variable-Resolution Modeling*, RAND National Defense Research Institute, Washington, DC, May 5-6, 1992, pp. 231-247.
- [HBL97] Harn, M, Berzins, V., Luqi, *Software Evolution via Reusable Architecture*, Computer Science Department, Naval Postgraduate School, Monterey, CA, 1997.
- [HBM96] Hillestad, R., Bennett, B., Moore, L., *Modeling for Campaign Analysis: Lessons for the Next Generation of Models*, the RAND Corporation, Santa Monica, CA, 1996.
- [HC93] Hammer, M., Champy, J., *Reengineering The Corporation: A Manifesto for Business Revolution*, Harper-Collins, New York, NY, 1993.

- [HCG+01a] Hartman, F., Collier, S., Gardner, D. *Integration Task Force Assessment*, Defense Modeling and Simulation Office, Department of Defense, Office of the Director of Defense Research and Engineering, Washington, DC, July 10, 2001.
- [HCG+01b] Hartman, F., Collier, S., Gardner, D. *Integration Task Force Report*, Defense Modeling and Simulation Office, Department of Defense, Office of the Director of Defense Research and Engineering, Washington, DC, August 20, 2001.
- [HCH+02] Hoek, Carzaniga, A., Heimbigner, D., "A Testbed for Configuration Management Policy Programming," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 1, January 2002, pp. 79-99.
- [HCM02] Hendrix, D., Cross, J., Maghsoodloo, S., "The Effectiveness of Control Structure Diagrams in Source Code Comprehension Activities," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 5, May 2002, pp. 463-477.
- [HCP96] Hamilton, J., Cook, D., Pooch, U., "A Software Engineering Perspective on Distributed Simulation," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, February 1996.
- [HCP+02] Hoare, P., Carson, J., Penny, R., "Designing an HLA Federation Object Model for Atmospheric Dispersion Model," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [HDG00] Hamilton, J., Deal, J., Grobmeier, J., "Network Modeling and Simulation with Mixed Traffic Data Sources," Western Multiconference 2000, accessed on October 11, 2001 from <http://www.drew-hamilton.com/download.htm>.
- [HDK+00] Holman, B., Denman, J., Knoepfle, G., *Defense Logistics: Actions Needed to Enhance Success of Reengineering Initiatives*, GAO/NSIAD-00-89, United States General Accounting Office, Washington, DC, June 2000.
- [HDK+02] Hamill, T., Deckro, R., Kloeber, J., "Risk Management and the Value of Information in a Defense Computer System," *Military Operations Research*, Vol. 7, No. 2, Military Operations Research Society, Alexandria, VA, 2002, pp. 61-81.
- [HEC91] Hathaway, B., Easton, B., Cockerham, G., *Nuclear Weapons: A Model for Evaluating the Tritium Reservoir Exchange*, GAO/NSIAD-91-86, United States General Accounting Office, Washington, DC, May 1991.

- [Hec94] Heckathorn, H., *Environmental Effects For Distributed Interactive Simulations (E2DIS)*, proceedings of the IRIS Specialty Group on Targets, Backgrounds and Discrimination, Monterey, CA, February 1994.
- [Hec99] Heckathorn, H., *BEST: Battlespace Environments & Signatures Toolkit Requirements Gathering Meeting*, briefing slides, Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, December 2, 1999.
- [HEE+00] Howard, B., Eftekari, R., Evatt, J., *Report of the Defense Science Board Task Force on DoD Supercomputing Needs*, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, October 11, 2000.
- [Hen92] Henderson, D., "The Multidimensional Space of Validation," *Simulation Validation Workshop Proceedings (SIMVAL II)*, Military Operations Research Society, Alexandria, VA, 31 March – 2 April 1992, pp. IV-17 to IV-19.
- [HGE03] Hamilton, J., Greaney, K., Evans, G., *Defining A Process for Simulation Software Vulnerability Assessments*, Missile Defense Agency, Washington DC, February 2003.
- [HH00a] Haddix, F., Hopkins, B., *Conceptual Models of the Mission Space Data Representation and Interchange Specification (CMMS-DRIS) Model Representation Data Interchange Format (DIF)*, Ver.1.4, The University of Texas at Austin, Austin, TX, February 1, 2000.
- [HH00b] Hollenbach, J., Hartnett, R., "The Joint Strike Fighter (JSF) Distributed Product Description (DPD)," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [HH02a] Hilburn, T., Humphrey, W., "The Impending Changes in Software Education," *IEEE Software*, Vol. 19, No. 5, September/October 2002, pp. 22-24.
- [HH02b] Hilburn, T., Humphrey, W., "Teaching Teamwork," *IEEE Software*, Vol. 19, No. 5, September/October 2002, pp. 72-78.
- [HHG+02] Hall, D., Hulett, D., Graves, R., *Universal Risk Project Final Report*, International Council on Systems Engineering (INCOSE), February 2002.

- [HHK+99] Hibern, T., Hirmanpour, I., Khajenoori, S., *A Software Engineering Body of Knowledge*, Ver.1.0, (CMU/SEI-99-TR-004), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 1999.
- [HHP00] Hatley, D., Hruschka, P., Pirhai, S., *Process For System Architecture and Requirements Engineering*, Dorset House Publishing Co., New York, NY, 2000.
- [HHS+98] Hancock, J., Harkrider, S., Shanks, G., *Reference FOM Study Group Final Report*, Ver.1.0, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9, 1998.
- [Hie99] Hiers, R., *Advanced Missile Signature Center (AMSC) Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, December 8, 1999.
- [Hil01] Hillelsohn, M., "Event-Driven Learning as a Process Improvement Strategy," *Crosstalk: The Journal of Defense Software Engineering*, Vol. L, 14, No. 1, Software Technology Support Center, Hill AFB, UT, January 2001, pp. 22-26.
- [Hin97] Hinchman, J., *Information Management and Technology*, GAO/HR-97-9, United States General Accounting Office, Washington, DC, February 1997.
- [Hin01] Hina, D., "Evaluation of the Extensible Markup Language (XML) as a Means for Establishing Interoperability Between Heterogeneous Department of Defense (DoD) Databases," Berzins, V. ed., *XML Technology Assessment Progress Report (7/17/2000 – 12/31/2000)*, NPS-SW-01-002, Naval Postgraduate School, Monterey, CA, January 2001, pp. 35-136.
- [His00] Hislop, D., *From "Programming" to Software Engineering*, briefing by U.S. Army Research Office, Mathematical and Computer Sciences Division, January 2000.
- [HJ95] Hillestad, R., Juncosa, M., "Cutting Some Trees to See the Forest: On Aggregation and Disaggregation in Combat Models", *Warfare Modeling*, John Wiley & Sons, 1995, pp. 37-62.
- [HJB+98] Hamann, D., Jarvinen, J., Birk, A., "A Product-Process Dependency Definition Method," *1998 Proceedings of the 24th Euromicro Conference*, Vol. 2, Vasteras, Sweden, pp. 25-27.
- [HJG94] Heckathorn, H., de Jonckheere, P., Grigsby, S., *Towards Realistic Environmental Effects in Distributed Interactive Simulations*, proceedings of

Test Technology Symposium VII, Johns Hopkins University, Applied Physics Laboratory, March 29, 1994.

- [HJS97] Hollenbach, J., Jefferson, M., Sheehan, J., *M&S Data Engineering Technical Framework (M&S DE-TF)*, DMSO-DE-97-0002, Department of Defense, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, Washington, DC, February 13, 1997.
- [HKK93] Humphrey, W., Kitson, D., Kasse, T., "The State of Software Engineering Practice: A Preliminary Report," *Software Management*, 4th Edition, IEEE Computer Society Press, Los Alamedas, CA, 1993, pp. 78-89
- [HKL96] Howie, C., Kunz, Law, K., *Software Interoperability*, Data & Analysis Center for Software, Rome, NY, November 25, 1996.
- [HL01] Hoffman, H., Lehner, F., "Requirements Engineering as a Success Factor in Software Projects," *IEEE Software*, Vol.18, No. 4, July/August 2001, pp. 58-66.
- [HLA98] High-Level Architecture, "High-Level Architecture Rules Version 1.3 (IEEE P1516/D1)," Department of Defense, Washington, DC, February 5, 1998.
- [HM95] Hillestad, R., Moore, L., *The Theater-Level Campaign Model: A New Research Prototype for a New Generation of Combat Analysis Models* (Draft), RAND's Project Air Force and Arroyo Center, November 1995.
- [HM96] Hitz, M., Montazeri, "Chidamber and Kemerer's Metrics Suite: A Measurement Theory Perspective," *IEEE Transaction on Software Engineering*, Vol. 22, No, 4, April 1996, pp. 267-271.
- [HM98] Hone, G., Moulding, M., "Application Domain Modeling to Support the Verification and Validation of Synthetic Environments," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL. March 9-13, 1998.
- [HM00] Hamilton, J., Murtagh, J., "Enabling Interoperability Via Software Architecture," Command and Control Research and Technology Symposium, accessed on October 11, 2000 from <http://www.drew-hamilton.com/download.htm>.
- [HM02] Hrischuk, C., Woodside, C., "Logical Clock Requirements for Reverse Engineering Scenarios from Distributed Systems," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 4, April 2002, pp. 321-339.

- [HMB+00] Hein, C., Madiseti, V., Bard, A., *VHDL Modeling Terminology and Taxonomy*, Version 3.1, RASSP Taxonomy Working Group, May 26, 2000.
- [HMD99] Hamilton, J., Murtagh, J., Deal, J., "A Basis for Joint Interoperability," *1999 Command & Control Research & Technology Symposium*, US Naval War College, Newport RI, June 29 - July 1, 1999.
- [HMR02] Diaz-Herrera, J., Murphy, M., Ramsey, D., "A Collaborative Program to Retrain Lockheed Martin Aero Engineers," *IEEE Software*, Vol. 19, No. 5, September-October 2002, pp. 30-34.
- [HMT+01] Hieb, M., Myers, L., Tolk, A., "Interim Report of the C4ISR/Sim Technical Reference Model Study Group," *Proceedings of the 2001 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 2001.
- [HN97] Hines, D., Norton, W., "Simulation, Test, and Evaluation Process," *ITEA Journal*, Sep - Oct 1997, pp. 29-34.
- [HNC+00] Hansen, M., Nesbit, R., Cross, S., *Report of the Defense Science Board Task Force on Defense Software*, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, November 2000.
- [HNK+01] Hamil, J., Navarro, M., Knight, F., Ressler, R., "Verification and Validation (V&V) of C4ISR Simulation for Testing and Training," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30 2001.
- [HNP97] Hamilton, J., Nash, D., Pooch, U., *Distributed Simulation*, CRC Press, Boca Raton, FL, 1997.
- [HO97] Hayes, W., Over, J., *The Personal Software Process (PSP): An Empirical Study of the PSP on Individual Engineers*, (CMU/SEI-97-TR-001), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, December 1997.
- [HOB95] Hillestad, R., Owen, J., Blumenthal, D., "Experiments in Variable Resolution Combat Modeling," *Warfare Modeling*, John Wiley & Sons, 1995, pp. 63-86.
- [Hob03] Hobbs, R., "Sharing Stories: Using Narrative for Simulations Interoperability," *Proceedings of the 2003 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, April 2003.

- [Hoe96] Hoenig, C., *Information Technology: Best Practices Can Improve Performance and Produce Results*, GAO/T-AIMD-96-46, United States General Accounting Office, Washington, DC, February 26, 1996.
- [HOF+00] Hoover, H., Olekshy, T., Froehlich, G., "Developing Engineered Product Support Applications," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 451-466.
- [Hol01] Hollenbach, J., "Collaborative Achievement of Advanced Acquisition Environments," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [HOM+97] Haddix, F., O'Brien, L., Might, R., Sheehan, J., Whitman, S., *Conceptual Models of the Mission Space (CMMS) Technical Framework, USD/A&T-DMSO-CMMS-0002, Revision 2.1*, Defense Modeling and Simulation Office, Department of Defense, Office of the Director of Defense Research and Engineering, Alexandria, VA, February 13, 1997.
- [HP99a] Hobbs, I., Parker, G., *Meeting the Federal IT Workforce Challenge*, CIO Council Education and Training Committee, June 1999.
- [HP99b] Hissam, S., Plakosh, S., *COTS in the Real World: A Case Study in Risk Discovery and Repair*, (CMU/SEI-99-TN-003), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 1999.
- [HPS98] Hartley, D., Perry, F., Salisbury, G., *Defense Information Infrastructure (DII) Common Operating Environment (COE) Configuration Management Plan*, Version 2.0, Defense Information Systems Agency, Washington, DC, April 1, 1998.
- [HRA+01] Hart, G., Rudman, W., Augustine, N., *Road Map for National Security: Imperative for Change*, The United States Commission on National Security / 21st Century, Washington, DC, February 15, 2001.
- [HS00a] Hieb, M., Sprinkle, R., "Simulation Infrastructure for the DII COE Architecture: The Army Vision," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 2000.
- [HS00b] Haddix, F., Sheehan, J., *Conceptual Models of the Mission Space (CMMS): Data Repository and Interface Specification (DRIS) Data Interchange Format*, University of Texas at Austin presentation, 2000.

- [HS00c] Herzum, P., Sims, O., *Business Component Factory: A Comprehensive Overview of Components-Based Development for the Enterprise*, John Wiley & Sons, 2000.
- [HS02] Holzman, G., Smith, M., "An Automated Verification Method for Distributed Systems Software Based on Model Extraction," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 4, April 2002, pp. 364-377.
- [HSB98] Harris, B., Smits, R., Bondaruk, J., *Data Dictionary Requirements Analysis and Validation & Verification Plan*, Army Research Laboratory, Aberdeen Proving Grounds, MD, May 21, 1998.
- [HSW+92] Hathaway, B., Spencer, K., Walter, C., *Strategic Defense Initiative: Estimates of Brilliant Pebbles' Effectiveness are Based on Many Unproven Assumptions*, GAO/NSIAD-92-91, United States General Accounting Office, Washington, DC, March 1992.
- [HSV00] Hein, A., Schlick, M., Vinga-Martins, R., "Applying Feature Models in Industrial Settings," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 47-70.
- [Hug97a] Hughes, W., Ed., *Military Modeling for Decision Making*, Third Edition, The Military Operations Research Society, Inc., Alexandria, VA, 1997, pp. vii-xv.
- [Hug97b] Hughes, W., "Overview," *Military Modeling for Decision Making*, Third Edition, The Military Operations Research Society, Inc., Hughes, W., Ed., Alexandria, VA, 1997, pp. 1-68.
- [Hum98a] Humphrey, W., "Three Dimensions of Process Improvement Part I: Process Maturity," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, February 1998.
- [Hum98b] Humphrey, W., "Three Dimensions of Process Improvement Part II: The Personal Process," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, March 1998.
- [Hum98c] Humphrey, W., "Three Dimensions of Process Improvement Part III: The Team Process," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, April 1998.
- [Hum99] Humphrey, W., "Pathways to Process Maturity: The Team Software Process and Team Software Process," *SEI Interactive*, Vol. 2, No. 2, June 1999.

- [Hus00] Hussmann, H., "Towards Practical Support for Component-Based Programming," *Proceedings of the 2000 Monterey Workshop on Modeling Software System Structures in a Fast Moving Scenario*, Santa Margherita Ligure, Italy, June 13-16, 2000, pp. 178-186.
- [HW96] Hermann, R., Welch, L., *Report of the Defense Science Board 1996 Task Force on Command, Control Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) Integration*, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, February 1997.
- [HW97] Hall, T., Wilson, D., "Views of Software Quality: a Field Report," *IEEE Proceedings on Software Engineering*, Vol. 144, No. 2, April 1997, pp.111-118.
- [HW02] Hyett, M., Wuerfel, R., "Implementation of the Data Distribution Management Services in the RTI-NG," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [HWB+98] Herman, R., Welch, L., Brandes, R., *Defense Science Board Task Force on Satellite Reconnaissance*, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, January 1998.
- [HY01a] Harmon, S., Youngblood, S., "Leveraging Fidelity to Achieve Substantive Interoperability," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [HY01b] Harmon, S., Youngblood, S., "Fidelity and Human Behavior Representations," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [IC99] Ioana, R., Collofello, J., "A Decision Support System for Software Reliability Engineering Strategy Selection," *Proceedings of COMPSAC*, Scottsdale, AZ, October 1999.
- [IC01] Ioana, R., Collofello, J., "Integrating Process Simulation and Reliability Models," *Crosstalk: The Journal of Defense Software Engineering*, Vol. 14, No.1, Software Technology Support Center, Hill AFB, UT, January 2001, pp. 15-18.

- [IDA01] Institute for Defense Analysis, *Simulation in Support of Live Fire Test and Evaluation*, accessed on December 13, 2001 from <http://www.ida.org/LFTSimulation/documents/ResearchSummaries.htm>
- [IEE86] IEEE Std 730-1989, *IEEE Standard for Software Verification and Validation Plans*, Institute of Electrical and Electronics Engineers, New York, NY, 1986.
- [IEE89] IEEE Std 730-1989, *IEEE Standard for Software Quality Assurance Plans*, Institute of Electrical and Electronics Engineers, New York, NY, 1989.
- [IEE90] IEEE Std 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, Institute of Electrical and Electronics Engineers, New York, NY, 1990.
- [IEE92] IEEE Std 1045-1992, *IEEE Standard for Software Productivity Metrics*, Institute of Electrical and Electronics Engineers, New York, NY, 1992.
- [IEE93] IEEE Std 1044-1993, *IEEE Standard Classification for Software Anomalies*, Institute of Electrical and Electronics Engineers, New York, NY, 1993.
- [IEE95] IEEE Std 1278.X-1995, *Unapproved IEEE Standards Draft for Distributed Interactive Simulation—Fidelity Description Requirements*, Institute of Electrical and Electronics Engineers, New York, NY, 1995.
- [IEE96] IEEE/EIA 12207.0-1996, *IEEE/EIA Standard Industry Implementation of International Standard ISO/IEC 12207:1995 (ISO/IEC 12207) Standard for Information Technology-Software Life Cycle Processes*, Institute of Electrical and Electronics Engineers and Electronic Industries Alliance, New York, NY, March 1998.
- [IEE97a] IEEE/EIA 12207.1-1997, *IEEE/EIA Guide for Information Technology Software Life Cycle Processes- Life Cycle Data*, Institute of Electrical and Electronics Engineers and Electronic Industries Alliance, New York, NY, April 1998.
- [IEE97b] IEEE/EIA 12207.1-1997, *IEEE/EIA Guide for Information Technology Software Life Cycle Processes-Implementation Considerations* Institute of Electrical and Electronics Engineers and Electronic Industries Alliance, New York, NY, April 1997.
- [IEE97c] IEEE/EIA 12207.1-1997, Annex J., *Supplement to IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1 1997* Institute of Electrical and Electronics Engineers and Electronic Industries Alliance, New York, NY, April 1997.

- [IEE98a] IEEE Std 1278.1a-1998, *IEEE Standard for Distributed Interactive Simulation-Application Protocols*, Institute of Electrical and Electronics Engineers, New York, NY, 1998.
- [IEE98b] IEEE Std 1012-1998, *IEEE Standard for Software Verification and Validation*, Institute of Electrical and Electronics Engineers, New York, NY, 1998.
- [IEE98c] IEEE Std 1062 1998 Edition, *IEEE Recommended Practice for Software Acquisition*, Institute of Electrical and Electronics Engineers, New York, NY, 1998.
- [IEE98d] IEEE Std 1061-1998, *IEEE Standard for a Software Quality Metrics Methodology*, Software Engineering Standards Committee of the IEEE Computer Society, Institute of Electrical and Electronics Engineers, New York, NY, 1998.
- [IEE98e] IEEE Std 1516/D1, April 1998, *U.S. Department of Defense High-Level Architecture Rules*, Version 1.3, Institute of Electrical and Electronics Engineers, New York, NY, April 20, 1998.
- [IEE99a] IEEE Std 1517-1999, *IEEE Standard for Information Technology Software Life Cycle Processes-Reuse Processes*, Institute of Electrical and Electronics Engineers, New York, NY, 1999.
- [IEE99b] IEEE Std 1420.1b-1999, *IEEE Trial-Use Supplement to IEEE Standard for Information Technology-Software Reuse-Data Model for Reuse Library Interoperability: Intellectual Property Rights Framework*, Institute of Electrical and Electronics Engineers, New York, NY, 1999.
- [IEE00a] IEEE Std 14143.1-2000, *Implementation Note for IEEE Adoption of ISO/IEC 14143-1:1998 Information technology-Software Measurement-Functional Size measurement-Part 1: Definition of Concepts*, Institute of Electrical and Electronics Engineers, New York, NY, 2000.
- [IEE00b] IEEE Std 1471-2000, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, Institute of Electrical and Electronics Engineers, New York, NY, 2000.
- [IEE00c] IEEE Std 1516-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules* Institute of Electrical and Electronics Engineers, New York, NY, 2000.

- [IEE00d] IEEE Std 1516.1-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification*, Institute of Electrical and Electronics Engineers, New York, NY, 2000.
- [IEE00e] IEEE Std 1516.2-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specification*, Institute of Electrical and Electronics Engineers, New York, NY, 2000.
- [IEE01] IEEE Std 1540-2001, *IEEE Standard for Software Life Cycle-Risk Management*, Institute of Electrical and Electronics Engineers, New York, NY, 2001.
- [ISE99] Independent Scientific and Engineering Group, *TMD Modeling and Simulation Review*, Briefing to RADM David Altwegg (Ret) PEO TSC, Washington, DC, December 28, 1999.
- [ISO79a] ISO/IEC 11179-1, *Information Technology – Specification and Standardization of Data Elements – Part 1: Framework for the Specification and Standardization of Data Elements* First Edition, 1999-12-01, International Standards Office / International Electrotechnical Commission, 1999.
- [ISO79b] ISO/IEC 11179-2, *Information Technology – Specification and Standardization of Data Elements – Part 2: Classification for Data Elements* First Edition, 2000-02-15, International Standards Office / International Electrotechnical Commission, 2000.
- [ISO79c] ISO/IEC 11179-3, *Information Technology – Metadata Registries – Part3: Registry Metamodel and Basic Attributes*, Second Edition, 2003-02-15, International Standards Office / International Electrotechnical Commission, 2003.
- [ISO79d] ISO/IEC 11179-4, *Information Technology – Specification and Standardization of Data Elements – Part 4: Rules and Guidelines for the Formulation of Data Definitions*, First Edition, 1995-07-01, International Standards Office / International Electrotechnical Commission, 1995.
- [ISO79e] ISO/IEC 11179-5, *Information Technology – Specification and Standardization of Data Elements – Part 5: Naming and Identification Principles of Data Elements* First Edition, 1995-12-01, International Standards Office / International Electrotechnical Commission, 1995.
- [ISO79f] ISO/IEC 11179-6, *Information Technology – Specification and Standardization of Data Elements – Part 6: Registration of Data Elements* First Edition, 1997-04-01, International Standards Office / International Electrotechnical Commission, 1997.

- [ISO00] ISO/IEC 15288, CD 2, *Life-Cycle Management-System Life Cycle Processes*, First Edition, 2001-06-15, International Standards Office / International Electrotechnical Commission, 2000.
- [ISO01] ISO/IEC 9126-1, *Software Engineering-Product Quality-Part I: Quality Model*, First Edition, 2001-06-15, International Standards Office / International Electrotechnical Commission, 2001.
- [JA97] Jacquet, J., Abran, A., "From Measurement to Software Process Measurement Methods: A Process Model," *Third International Symposium and Forum on Software Engineering Standards, ISESS '97*, Walnut Creek, CA, June 2-6, 1997.
- [Jac98] Jackson, L., *A Component Approach to Object Model Standards for Simulations*, U.S. Army TRADOC Analysis Center, Monterey, CA, 1998.
- [JAS95] Joint Accreditation Support Activity (JASA), *Configuration Management Requirements Study*, (JTCG/AS-95-M-005), Naval Air Warfare Center, China Lake, CA, January 1995.
- [JAS96] Joint Accreditation Support Activity (JASA), *Accreditation Support Package For Enhanced Surface To Air Missile Simulation (ESAMS)*, (JTCG/AS-95-M-011), Naval Air Warfare Center, China Lake, CA, April 23, 1996.
- [JAS97a] Joint Accreditation Support Activity (JASA), *VV&A From A To Z: A SMART approach to VV&A for Acquisition M&S*, (JTCG/AS-97-M-008), Naval Air Warfare Center, China Lake, CA, October 1997.
- [JAS97b] Joint Accreditation Support Activity (JASA), *How To VV&A Without Really Trying*, (JTCG/AS-97-M-009), Naval Air Warfare Center, China Lake, CA, November 1997.
- [JAS97c] Joint Accreditation Support Activity (JASA), *Accreditation Support Package For Extended Air Defense Simulation (EADSIM)*, (JTCG/AS-97-M-012), Naval Air Warfare Center, China Lake, CA, January 17, 1997.
- [Jaz02] Jazayeri, M., "Software Components: Some Challenges Ahead," *Monterey 2002, Radical Innovations of Software and Systems Engineering in the Future Workshop Proceedings*, Venice, Italy, October 7-11, 2002, pp. 227-231.
- [JBC+88] Jones, A., Brooks, F., Chatelier, P., *Report of the Defense Science Board Task Force on Computer Applications to Training and Wargaming*, Office of the Undersecretary of Defense for Acquisition, Washington, DC, May 1988.

- [JBC+99] Joy, B., Benhamou, E., Cerf, V., *Information Technology Research: Investing in Our Future*, President's Information Technology Advisory Committee Report to the President, Washington, DC, February, 1999.
- [JBS+93] Johns, L., Blair, P., Shaw, A., *Data Format Standards for Civilian Remote Sensing Satellites*, OTA-BP-ISC-114, Office of Technology Assessment, Congressional Board of the 103d Congress, May, 1993.
- [JBL97] Jackson, L., Buss, A., Larimer, L., "From Legacy Model to Future Federate: The Janus Simulation Object Model," *Proceedings of the 1997 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-12, 1997.
- [JDD98] Jenkins, R., Deshpande, Y., Davison, G., "Verification and Validation and Complex Environments: A Study in Service Sector," *Proceedings of the 1998 Winter Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13, 1998.
- [JDL01] Jezequel, J., Deveaux, D., Le Traon, Y., "Reliable Objects: Lightweight Testing for OO Languages," *IEEE Software*, Vol. 18, No. 4, July-August 2001, pp. 76-83.
- [Jen02] Jensen, J., "Software Engineering Graduate Opportunities at the Naval Postgraduate School," *Software Tech News*, Vol. 5, No. 2, April 2002, pp. 10-12.
- [JHB98] Jackson, L., Hodge, D., Bradley, B., *Army Standard Location Object*, "U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, October 1998.
- [JHU03] John Hopkins University, *First U.S. Application of New Simulation Standard*, John Hopkins University, accessed on January 15, 2003 from <http://www.sciencedaily.com/releases/2003/01/030114072151.htm>.
- [JMP02] Janett, A., Miller, D., Prager, S., "Capturing Environment Data Requirements – The Common Data Model Framework," *Proceedings of the 2002 Fall Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-13, 2002.
- [JMS98] Johnson, M., McKeon, M., T. Szanto, *Systems Acquisition Managers Guide For Implementing Simulation Based Acquisition*, DSMC Research Fellows, draft, June 30, 1998.
- [JNR98] Jaber, K., Nada, N., Rine, D., *Product Line Stakeholder Viewpoint Approach and Validation Model*, 1999.

- [Jon95] Jones, C., *What are Function Points?* Software Productivity Research, Inc., accessed on December 23, 1996 from <http://www.spr.com/library/funcmet.htm>.
- [Jon99] Jones, L., "Product Line Acquisition in the DoD: the Promises, the Challenges," *Crosstalk: The Journal of Defense Software Engineering*, Vol. 12, No 8, Software Technology Support Center, Hill AFB, UT, August 30, 20031999, pp.17-21.
- [Jon00] Jones, C., "Spotting the Best, and Worst, Software Development Practices," *Application Development Trends*, Vol. 7, No. 7, July 2000, pp. 47-54.
- [Jor01] Jordan, J., "Simulation, Verification and Validation: An Operational Perspective," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [JP98] Department of Defense, *Joint Doctrine for Information Operations*, Joint Pub 3-13, The Joint Chiefs of Staff, Washington, DC, October 9, 1998.
- [JS02] Jones, L., Soule, A., *Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice*, (CMU/SEI-2002-TN-012), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, July 2002.
- [JSK02] Johnson, B., Seltser, B., Kingsbury, N., *Assessing the Reliability of Computer-processed Data*, GAO-03-273G, United States General Accounting Office, Washington, DC, October, 2002.
- [JTA02a] Department of Defense, *Joint Technical Architecture, Version 4.0*, JTA Development Group, Office, Washington, DC, July 17, 2002.
- [JTA02b] Department of Defense, *Joint Technical Architecture List of Mandated and Emerging Standards for JTA Version 4.0*, JTA Development Group, Washington, DC, July 17, 2002.
- [JWC01] Juristo, N., Windl, H., Constantine, L., "Introducing Usability," *IEEE Software*, January-February 2001, pp. 20-21.
- [Kad02a] Kadish, R., *Element Models and Simulations (M&S) Verification, Validation, and Accreditation (VV&A)*, MDA/TE memorandum, Missile Defense Agency, Washington, DC, June 10, 2002.

- [Kad02b] Kadish, R., *Missile Defense Deployment Announcement Briefing*, News Transcript, Department of Defense, Washington, DC, December 17, 2002, accessed on 4 January 2003 from http://www.defenselink.mil/news/Dec2002/t12172002_t1217missiledef.html.
- [Kad02c] Kadish, R., *The Missile Defense Program Fiscal Year 2003 Budget*, Senate Appropriations Committee, Defense Subcommittee, Written Statement, Washington, DC, December 17, 2002, accessed on 4 January 2003 from <http://www.acq.osd.mil/bmdo/bmdolink/html>.
- [Kan01] Kanoun, K., "Real-World Design Diversity: A Case Study on Cost," *IEEE Software*, Vol. 18, No. 4, July/August 2001, pp. 29-33.
- [Kan03] Kan, S., *Metrics and Models on Software Quality*, 2d edition, Addison-Wesley, Boston, MA, 2003.
- [KBA+02] Kingsbury, N., Bausell, C., Ashery, A., *Economic Models of Cattle Prices: How USDA Can Act to Improve Models to Explain Cattle Prices*, GAO-02-246, United States General Accounting Office, Washington, DC, March 2002.
- [KBM+97] Kuhta, S., Baird, T., MacFarlane, S., *Theater Missile Defense*, GAO/NSIAD-98-37, United States General Accounting Office, Washington, DC, October 1997.
- [KBO99] Department of Defense, *Knowledge Based Organization Strategic Plan* (Draft), United States Joint Forces Command, Norfolk, VA, November 19, 1999.
- [KC97] Kazman, R., Carriere, S., *Playing Detective: Reconstructing Software Architecture from Available Evidence*, (CMU/SEI-97-TR-010), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, October 1997.
- [KDM+94] Katz, S., Dabrowski, C., Miles, K., *Glossary of Software Reuse Terms*, NIST Special Publication 500-222, Gaithersburg, MD, *IEEE Software*, December 1994.
- [KEB02] Kinetic Energy Boost, *Kinetic Energy Boost Modeling and Simulation (M&S) Plan*, Ver.0.2, (Draft), Kinetic Energy Boost Element, October 17, 2002.
- [Kec99] Keck, E., "Planning an ADS-Supported Test," *Joint Advanced Distributed Simulation Joint Task Force*, accessed on August 26, 2001 from <http://www.jads.abq.com/html/ads/keckppr1.htm>.

- [KFC98] Kleijnen, J., Feelders, A., Cheng, R., "Bootstrapping and Validation of Metamodels in Simulation," *Proceedings of the 1998 Winter Simulation Conference*, pp. 701-706.
- [KG00] Kulak, D., Guiney, E., *Use Cases: Requirements In Context*, Addison-Wesley, Boston, MA, 2000.
- [KGB+99] Kuhta, S., Ghoshroy, S., Baird, T., *Defense Acquisitions, DoD Efforts to Develop Laser Weapons for Theater Defense*, GAO/NSIAD-99-50, United States General Accounting Office, Washington, DC, March 1999.
- [KH02] Khan, K., Han, J., "Composing Security-Aware Software," *IEEE Software*, Vol. 19, No. 1, January-February 2002, pp. 34-41.
- [Kil02] Kilikauskas, M., *Simulation Credibility: A Risk-Based Information Oriented Approach*, Joint Accreditation Support Activity, M&S Credibility Workshop II, Reno, NV, March 4, 2002.
- [Kin01] Kin, B., "A Simple Software Agent of the Extensible Markup Language (XML) for Building Distributed Applications," Master thesis, *Progress Report: Assessment of Defense Modeling And Simulation Office (DMSO) Conceptual Models of the Mission Space*, Naval Postgraduate School, Monterey, CA, December 31, 2001.
- [Kis01] Kiselyov, O., "Subclassing Errors, OOP, and Practically Checkable Rules to Prevent Them," *Interim Progress Report, Engineering Automation for Reliable Software*, Naval Postgraduate School, Monterey, CA, September 2001, pp. 52-61.
- [KJ01] Kramer, B., Jacobsen, H., "XML-Based Integration of Interface Definition Language Extensions," *Proceedings of Monterey Workshop 2001*, Monterey, CA, June 18-22, 2001, pp. 16-27.
- [KK99] Klein, M., Kazman, R., *Attribute-Based Architectural Styles*, (CMU/SEI 99-TR-022), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, October 1999.
- [KLD02] Kang, K., Lee, J., Donohoe, P., "Feature-Oriented Product Line Engineering," *IEEE Software*, July/August 2002, pp.58-65.
- [KLM+00] Keane, J., Lutz, R., Myers, S., Collahan, J., "An Architecture for Simulations Based Acquisition," *John Hopkins APL Technical Digest*, July-September 2000, Volume 21, Number 3, pp. 348-357.

- [KM00] Kasse, T., McQuaid, P., "Factors Affecting Process Improvement Initiatives," *Crosstalk: The Journal of Defense Software Engineering*, Vol. 13, No. 8, Software Technology Support Center, Hill AFB, UT, August 2000.
- [KM01a] Ketcham, R., Muessig, P., *Software Engineering and Simulation Credibility*, Joint Accreditation Support Activity (JASA), Naval Air Warfare Center, China Lake, CA, accessed on August 20, 2001 from <http://www.nawcwpns.navy.mil/~jasa/>.
- [KM01b] Konwin, K., Miller, R., "Simulation Based Acquisition: From Motivation to Implementation," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30 2001.
- [KMM+00] Karl, R., McDaniel, B., Monnet, R., Moore, A., *Extended Air Defense Test-bed (EADTB) Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, April 5, 2000.
- [KMS02] Komi-Sirvio, S., Mantyniemi, A., Sappanen, V., "Toward a Practical Solution for Capturing Knowledge for Software Projects," *IEEE Software*, Vol. 19, No. 3, May-June 2002, pp. 60-62.
- [KN00] Kishi, T., Noda, N., "Aspect-Oriented Analysis for Product Line Architecture," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 135-146.
- [KNR+01] Kemp, L., Nidiffer, K., Rose, L., "Knowledge Management: Insights from the Trenches," *IEEE Software*, Vol. 18, No. 6, November-December 2001, pp. 66-68.
- [Koo99] Koopman, B., *Search and Screening: General Principles with Historical Applications*, reprinted from 1946 edition, Military Operations Research Society, Inc., Alexandria, VA, 1999.
- [KOV01] Kazman, R., O'Brien, L., Verhoef, C., *Architectural Reconstruction Guidelines*, (CMU/SEI-2001-TR-026), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 2001.
- [Koz02] Kozaryn, L., "Virtual Battle Space Blends Reality with Simulation," *Armed Forces Press Service*, accessed on August 23, 2002 from: <http://www.jfcom.mil/newslink/storyarchive/2002/no090902a.htm>.
- [KP97] Kassling, D., Pyles, R., "Logistics Modeling: New Challenges and New Opportunities," *Military Modeling for Decision Making*, Third Edition, Edited

by Hughes, W., The Military Operations Research Society, Inc., Alexandria, VA, 1997, pp. 225-246.

- [Kri88] Krings, J., *Modeling and Simulation in Operational Test and Evaluation*, Office of the Secretary of Defense memorandum, Washington, DC, July 25, 1988.
- [Kri89] Krings, J., *Policy Guidance for the Application of Modeling and Simulation in Support of Operational Test and Evaluation*, Office of the Secretary of Defense memorandum, Washington, DC, January 23, 1989.
- [Kru95] Kruchten, P., "The 4+1 View Model of Architecture," *IEEE Software*, Vol. 12, No. 6, 1995.
- [KS97] Keller, T., Schneidewind, N., Successful Application of Software Reliability Engineering for the NASA Space Shuttle," *Software Reliability Engineering Case Studies*, International Symposium on Software Reliability Engineering, November 3, Albuquerque, NM, November 4, 1997, pp. 71-82.
- [KS02] Kessler, D., Schuette, M., *Verification, Validation, and Accreditation (VV&A) of EADTB for the PAC-3 IOT&E*, Quality Research, Huntsville, AL, October 2002.
- [KSZ03] Kennon, J., Sterling, S., Zuckerstein, K., *Contracting for Information Technology Services*, GAO-03-384R, United States General Accounting Office, Washington, DC, February 14, 2003.
- [KZ96] Krut, R., Zalman, N., *Domain Analysis Workshop Report for the Automated Prompt Response System Domain*, (CMU/SEI-96-TR-001), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 1996.
- [LA92] Lane, N., Alluisi, E., *Fidelity and Validity in Distributed Interactive Simulation: Questions and Answers*, IDA Document D-1066, Institute for Defense Analysis, Alexandria, VA, November 1992.
- [Lad93] Ladkin, P., *Report on the Accident to Airbus A320-211 Aircraft in Warsaw on 14 September 1993*, accessed on March 17, 2002 from <http://sunnyday.mit.edu/accidents/>.
- [Lad95] Ladkin, P., *AA965 Cali Accident Report, December 20, 1995*, accessed on March 17, 2002 from <http://sunnyday.mit.edu/accidents/>.
- [Lat97] Lattanze, A., "A Component-Based Construction Framework for DoD Software Systems Development," *Crosstalk: The Journal of Defense Soft-*

ware Engineering, Vol. 10, No. 11, Software Technology Support Center, Hill AFB, UT, November 1997.

- [Lav97] Lavine, M., *Army Simulation Support Plan Facilitates Simulation Based Acquisition*, International Test and Evaluation Association (ITEA) Workshop, December 8, 1997.
- [Law02] Law, A., *Simulation Modeling for System Design and Analysis*, presentation to the Missile Defense Agency Model and Simulation Working Group, September 26, 2002.
- [LB88] Luqi, Berzins, V., "Rapidly Prototyping Real-Time Systems," *IEEE Software*, September 1998, pp. 25-36.
- [LB95] Luqi, Berzins, V., "Software Architectures in Computer-Aided Prototyping," *Proceedings of Monterey Workshop 1995*, Monterey, CA, September 12-14 1995, pp. 44-57.
- [LB98] Luqi, Broy, M., "Software Engineering to our Planning Horizon," *Proceedings of the 1998 Monterey Workshop on Engineering Automation for Computer Systems*, Naval Postgraduate School, Monterey, CA, April 1999, pp. 1-2.
- [LB00] Lange, D., Berzins, V., "A Formal Method of System and Software Engineering Experience," *Proceedings of the 2000 Monterey Workshop on Modeling Software System Structures in a Fast Moving Scenario*, Santa Margherita Ligure, Italy, June 13-16, 2000, pp.198-207.
- [LBG+96] Luqi, Berzins, V., Goguen, J., "Software Component Search," *Journal of Systems Integration*, Vol. 6, 1996, pp. 93-134.
- [LBG+01] Luqi, Berzins, V., Ge, J., "DCAPS - Architecture for Distributed Computer Aided Prototyping System," *12th International Workshop on Rapid System Prototyping*, IEEE Computer Society, 2001, pp. 103-109.
- [LBL+97] Leonard, T., Berzins, V., Luqi, "Gathering Requirements from Remote Users," *Proceedings of the Ninth International Conference on Tools with Artificial Intelligence*, Newport Beach, CA, November 3-8, 1997, pp. 462-471.
- [LBS01] Luqi, Berzins, V., Shing, M., "The Use of Computer Aided Prototyping for Re-Engineering Legacy Software," *Proceedings of Monterey Workshop 2001*, Monterey, CA, June 18-22, 2001, pp. 169-180.

- [LBV96] Lucas, T., Bankes, S., Vye, P., *Improving the Analytic Contribution of Advanced Warfighting Experiments*, RAND Arroyo Center, Santa Monica, CA, 1996.
- [LC97] Lewis, R., Coe, G., "A Comparison Between The CMMS And The Conceptual Model Of The Federation," *Proceedings of the 1997 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-12, 1997.
- [LDG+02] Lutz, B., Dugone, T., Gustavson, P., *Product Nomination Proposal: HLA Federation Development and Execution Process (FEDEP)*, Version 1.0, Simulation Interoperability Standards Organization (SISO), Orlando, FL, April 8, 2002.
- [LE01] Lyttle, B., Ehrhardt, T., "Interconnectivity via a Consolidated Type Hierarchy and XML," Master thesis, *Progress Report: Assessment of Defense Modeling And Simulation Office (DMSO) Conceptual Models of the Mission Space*, Naval Postgraduate School, Monterey, CA, December 31, 2001.
- [LEH+00] Levin, B., Edwards, L., Hand, D., *Missile Defense, Schedule for Navy Theater Wide Program Should Be Revised to Reduce Risk*, GAO/NSIAD-00-121, United States General Accounting Office, Washington, DC, May 2000.
- [Lev95] Leveson, N., *Medical Devices: The Therac-25*, Software Engineering Research Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, accessed on March 17, 2002, from <http://sunnyday.mit.edu/accidents/>.
- [Lev00] Leveson, N., *Systemic Factors in Software-Related Spacecraft Accidents*, Software Engineering Research Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, accessed on March 17, 2002, from <http://sunnyday.mit.edu/accidents/>.
- [Lev01a] Leveson, N., *Systemic Factors in Software-Related Spacecraft Accidents*, AIAA 2001-473, Software Engineering Research Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, accessed on March 17, 2002, from: <http://sunnyday.mit.edu/accidents/>.
- [Lev01b] Leveson, N., *The Role of Software in Recent Aerospace Accidents* Software Engineering Research Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, accessed on March 17, 2002, from: <http://sunnyday.mit.edu/accidents/>.
- [Lev01c] Leveson, N., *Evaluating Accident Models using Recent Aerospace Accidents*, Software Engineering Research Laboratory, Department of Aeronau-

tics and Astronautics, Massachusetts Institute of Technology, accessed on March 17, 2002, from: <http://sunnyday.mit.edu/accidents/>.

- [Lev02] Levy, L., *The Role of Confidence in T&E of Strategic Defense Systems*, The Johns Hopkins University, Applied Physics Laboratory, Laurel, MD, 2002.
- [Lew94] Lewis, R., "A Rapid Planning Tool for Verification, Validation, and Accreditation (VV&A) of Distributed Interactive Simulation," *Proceedings of Summer Computer Conference*, San Diego, CA, July, 1994, pp.696-701.
- [Lew98] Lewis, R., "A Practical Model and Metrics for Verification, Validation, and Accreditation (VV&A) Programs," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [Lew02a] Lewis, B., "Architecture Based Model Driven Software and System Development for Avionics and Safety Critical Systems," *Monterey 2002, Radical Innovations of Software and Systems Engineering in the Future Workshop Proceedings*, Venice, Italy, October 7-11, 2002, pp. 273-284.
- [Lew02b] Lewis, R., *JDEP Meeting Summary*, email from Brig.Gen. Rick Lewis, USAF, JTAMDO, Washington, DC, January 17, 2002, 12:36 PM.
- [LFT95] Lawless, P., Flowe, R., Thordahl, J., "A Correlation Study of the CMM and Software Development Performance," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, September 1995, pp. 21-25.
- [LFW00] Lacetera, J., Frazier, G., Wertman, C., *C4ISR Systems-to-Simulations Data and Interface Lexicon*, Ver. 2000.09.30, 2000.
- [LG93] Luqi, Goguen, J., "Some Suggestions for Using Formal Methods In Software Development," *Proceedings of Monterey Workshop 1993*, Monterey, CA, October 13-15, 1993, pp. 7-12.
- [LG96] Luqi, Guo, J., *Toward Automated Retrieval for a Software Component Repository*, Dept. of Computer Science, Naval Postgraduate School, Monterey, CA, 1996.
- [LG97] Luqi, Goguen, J., "Formal Methods: Promises and Problems," *IEEE Software*, January 1997, pp. 73-85.
- [LGB94] Luqi, Goguen, J., Berzins, V., "Formal Support For Software Evolution," *Proceedings of Monterey Workshop 1994*, Monterey, CA, September 7-9, 1994, pp. 10-21.

- [LHC+02] Levin, B., Haynes, B., Chaplain, C., *Missile Defense: Review of Results and Limitations of an Early National Missile Defense Flight Test*, GAO-02-124, United States General Accounting Office, Washington, DC, February, 2002.
- [Li97] Li, A., *Ballistic Missile Defense, Improvements Needed In THAAD Acquisition Planning*, GAO/NSIAD-97-188, United States General Accounting Office, Washington, DC, September 1997.
- [Li99] Li, A., *Missile Defense, THAAD Restructure Addresses Problems But Limits Early Capability*, GAO/NSIAD-99-142, United States General Accounting Office, Washington, DC, June 1999.
- [Li02] Li, A., *Survey of NASA's Lessons Learned Process*, GAO-01-1015R United States General Accounting Office, Washington, DC, September 5, 2001.
- [Lib01] Libicki, M., *The Role of Technical Standards in Today's Society and in the Future*, Statement before the Subcommittee on Technology, U.S. House of Representatives, Washington, DC, September 13, 2000.
- [Lie97] Lieberman, A., "Models in National Policy Analysis," *Military Modeling for Decision Making*, Third Edition, The Military Operations Research Society, Inc., Alexandria, VA, 1997, pp. 270-305.
- [Lie02] Liebowitz, J., "A Look at NASA Goddard Space Flight Center's Knowledge Management Initiatives," *IEEE Software*, Vol. 19, No. 3, May-June 2002, pp. 40-49.
- [LK88] Luqi, Ketabchi, M., "A Computer-Aided Prototyping System," *IEEE Software*, March 1998, pp. 66-72.
- [LK00] Law, A., Kelton, W., *Simulation Modeling and Analysis*, Third Edition, McGraw-Hill Companies, Boston, MA, 2000.
- [LKB01] Lacy, L., King, D., Burnett, K., "Equipment Characteristics and Performance Data Interchange for Simulations using XML," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [LKK+00] Lee, K., Kang, K., Koh, E., "Domain-Oriented Engineering of Elevator Control Software," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 3-22.

- [LLC+02] Lessmann, K., Lucas, J., Conrad, G., "Foundation Initiative 2010: Update on the Development and Testing of the Test and Training Enabling Architecture," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [LLD+99] Latham, D., Lynn, L., Dugan, R., *Final Report of the Defense Science Summer Board Task Force on 21st Century Defense Technology Strategies*, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, November 1, 1999.
- [LLF+96] Lions, J., Lbeck, L., Fauquembergue, J., *ARIANE 5 Flight 501 Failure, Report by the Inquiry Board*, accessed on March 17, 2002 from <http://sunnyday.mit.edu/accidents/Ariane5accisentreport.html>.
- [LLP+02] Larsen, K., Lorenzo, M., Roose, K., "Network Challenges and Observations Noted During an HLA Distributed Simulation Exercise," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [LLT+99] Lee, D., Langston, M., Thomas, R., *Federal Enterprise Architecture Framework*, Ver.1.1, The Chief Information Officers Council, Washington, DC, September 1999.
- [LMR+00] Lorenzo, M., Morse, K., Riggs, B., Rizik, P., "Sensor Simulation Scalability Using Composable Component Federates," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [LN00] Luqi, Nogueira, J., "A Risk Assessment Model For Evolutionary Software Projects," *Proceedings of the 2000 Monterey Workshop on Modeling Software System Structures in a Fast Moving Scenario*, Santa Margherita Ligure, Italy, June 13-16, 2000, pp. 218-215.
- [Log02] Logsdon, J., *Aggregate Level Simulation Protocol*, PM WARSIM fact sheet, accessed on February 7, 2002 from <http://www.stricom.army.mil/PRODUCTS/ALSP/>.
- [Lor02] Lorenzo, M., *Composability in Current and Future Sensor Simulations*, USA Communications and Electronics Command presentation, DMSO Workshop on Composable M&S, July 9, 2002.
- [LPA+02] Lacy, L., Puhlmann, M., Aviles, G., "Providing Authoritative Equipment Characteristics and Performance Data to Simulations using Data Interchange Format Standards," *Proceedings of the 2002 Spring Simulation Conference*,

Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.

- [LQZ02] Luqi, Qiao, Y, Zhang, L., "Computational Model for High-Confidence Embedded System Development," *Monterey 2002, Radical Innovations of Software and Systems Engineering in the Future Workshop Proceedings*, Venice, Italy, October 7-11, 2002, pp. 285-303.
- [LR98] Loper, M., Rosenbaum, D., "The Federation Verification Tool," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [LRH+01] Lacy, L., Randolph., Harris, B., "Developing a Consensus Perspective on Conceptual Models for Simulation Systems," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30 2001.
- [LS01] Luqi, Sheehan, J., *Assessment of Defense Modeling And Simulation Office (DMSO) Conceptual Models of the Mission Space*, NPS-SW-01-0001, Software Engineering Group, Naval Postgraduate School, Monterey, CA, December 31, 2000.
- [LSF+01] Libicki, M., Schneider, Frelinger, D., *Scaffolding the New Web: Standards and Standards Policy for the Digital Economy*, MR-1215-OSTP, RAND, Santa Monica, CA, 2001.
- [Luc00] Lucas, T., *The Stochastic Versus Deterministic Argument for Combat Simulations: Tales of When the Averages Won't Do*, Operations Research Department, Naval Postgraduate School, Monterey, CA, 2000.
- [Luq89] Luqi, "Software Evolution Through Rapid Prototyping, *IEEE Computer*, May 1989, pp. 13-25.
- [Luq90] Luqi, "A Graph Model for Software Evolution," *IEEE Transactions on Software Engineering*, Vol. 16, No. 8, August 1990, pp. 917-927.
- [Luq92] Luqi, "Computer-Aided Prototyping for a Command-and-Control System Using CAPS," *IEEE Software*, January 1992, pp. 56-67.
- [Luq94] Luqi, "Monterey Workshop '94: Software Evolution," *Proceedings of 1994 Monterey Workshop*, Monterey, CA, September 7-1, pp. 1-4.
- [Luq95] Luqi, Monterey Workshop '95: Specification-based Software Architecture," *Proceedings of Monterey Workshop 1995*, Monterey, CA, September 12-14, 1995, pp. 1-4.

- [Luq98] Luqi, "Engineering Automation for Computer Based Systems" *Proceedings of The 1998 Monterey Workshop on Engineering Automation for Computer Systems*, Naval Postgraduate School, Monterey, CA, April 1999, pp. 3-8.
- [Luq01] Luqi, *Interim Progress Report, Engineering Automation for Reliable Software*, Naval Postgraduate School, Monterey, CA, September 2001.
- [Lut00] Lutz, R., "Migrating the HLA Object Model Template to an IEEE Standard," *John Hopkins APL Technical Digest*, July-September 2000, Volume 21, Number 3, pp. 337-347.
- [LWC+01] Latham, D., Welch, L., Corder, J., *Defense Science Board Task Force on High Energy Laser Weapon System Applications*, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, June 2001.
- [LWF+98] Latham, D., Welch, L., Fowler, B., *Report of the Defense Science Board 1988 Summer Study Task Force On Joint Operations Superiority in the 21st Century. Volume I: Integrating Capabilities Underwriting Joint Vision 2010 and Beyond*, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, October 1998.
- [LWK+02] Lachman, B., Wong, A., Knopman, D., *Lessons for the Global Spatial Data Infrastructure: Internal Case Study Analysis*, RAND Science and Technology Policy Institute, Santa Monica, CA, 2002.
- [LX99] Liang, Q., Xiangsui, W., *Unrestricted Warfare*, PLA Literature and Arts Publishing House, Beijing, China, February, 1999.
- [LZB+96] Luqi, Zhang, D., Berzins, V., "Software Component Search," *Journal of Systems Integration*, Vol. 6, 1996, pp. 93-134.
- [Mac92] Macfadzean, R., *Surface-Based Air Defense System Analysis*, Artech House, Boston, MA, 1992.
- [MAK01] McCarthy, J., Arthur, S., Kaminski, P., *Transformation Study Report, Executive Summary, Transforming Military Operational Capabilities*, Transformation Study Group, Washington, DC, April 27, 2001.
- [Mai97] Mains, S., "Adapting Doctrine to Knowledge-Based Warfare," *Military Review*, March-April 1997.

- [Man95] Mankins, J., *Technology Readiness Levels, A White Paper*, Advanced Concepts Office, Office of Space Access and Technology, NASA, April 6, 1995.
- [Man01] Mansir, J., *Joint Warfare System (JWARS) Operational Test and Evaluation Strategy*, Army Test and Evaluation Command (AEC-ITED briefing slides, Washington, DC October 24, 2001.
- [Man02] Mann, C., "Why Software is So Bad," *Technology Review*, July-August 2002, pp. 33-38.
- [Mar83a] Marino, L., "Models," *Encyclopedia of Computer Science and Engineering*, 2d Edition, Ralston, Anthony, Reilly, E., ed., Van Nostrand Reinhold Company, Inc., New York, NY, 1983, pp. 990-993.
- [Mar83b] Marsaglia, G., "Stochastic Process," *Encyclopedia of Computer Science and Engineering*, 2d Edition, Ralston, Anthony, Reilly, E., ed., Van Nostrand Reinhold Company, Inc., New York, NY, 1983, p. 1409.
- [Mar97] Martin, D., *Joint Simulation System Transition Plan*, Version 1.0, Advanced Warfighting Division, Fort Monroe, VA, August 27, 1997.
- [Mar99] Marchand, G., "A Never Ending Story The Need For Verification and Validation Throughout the Life of a Test," *Proceedings of the 1999 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 12-17, 1999.
- [Mas01] Mason, D., "Probability Density Functions in Program Analysis," Crnkovic, I., Schmidt, H., Stafford, J., eds., *Component Certification and System Prediction*, Proceedings 4th ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, May 14-15, 2001.
- [Mat00] Matthew, W., "IT Execs: Education is Weak Link," *Federal Computer Week*, June 12, 2000, pp.60-61.
- [Mat02] Mathias, K., "Composable Simulation Research at the Air Force Institute of Technology," *DMSO Workshop on Composable M&S*, Department of Defense, Office of the Director of Defense Research and Engineering, July 9, 2002
- [MB02] Mellor, S., Balcer, M., *Executable UML: A Foundation for Model-Driven Architecture*, Addison-Wesley, Boston, MA, 2002.

- [MBG+02] Mackin, M., Berrier, P., Greeley, P., "Contract Management, Intragency Contract Program Fees Need More Oversight," GAO-02-734, United States General Accounting Office, Washington, DC, July 2002.
- [MBH99] Martin, E., Bryan, B., Hoagland, D., *Tools For Including Realistic Simulations of Operator Performance in DoD Constructive Simulations*, AIAA-99-4027, American Institute of Aeronautics and Astronautics, June 1, 1999.
- [MBS+99] Melloul, L., Beringer, D., Sample, N., "CPAM, A Protocol for Software Composition," *Advanced Information Systems Engineering (CAISE 11)*, Heidelberg, GE, June 16-18, 1999.
- [MBZ99] Ming, Y., Bing, Z., Zicai, W., "Verification, Validation, and Accreditation for Modeling and Simulation: Present and Future," *Proceedings of the 1999 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 12-17, 1999.
- [MC01a] Moneymaker, P., Carter, L., "Managing the Invisible Aspects of High Performance Teams, *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, May 2001.
- [MC01b] Mohegheghi, P., Conradi, R., "Experiences with Certification of Reusable Components in the GSN Project in Ericsson, Norway," Crnkovic, I., Schmidt, H., Stafford, J., eds., *Component Certification and System Prediction*, Proceedings 4th ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, May 14-15, 2001.
- [MC02] Joint Forces Command, *Millennium Challenge 02*, United States Joint Forces Command, accessed on August 23, 2002 from <http://www.jfcom.mil/about/experiments/mc02.htm>.
- [MCC+98] Morrow, W., Christie, J., Cooper, R., *Report of the Defense Science Board Task Force on Defense Science and Technology Base for the 21st Century*, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, June 30, 1998.
- [McC00] McClure, D., *Information Security: Software Change Controls at the Department of Defense*, GAO/AIMD-00-188R, United States General Accounting Office, Washington, DC, June 30, 2000.
- [McD98] McDonald, B., "A Proposed Process For Defining Required Fidelity of Simulations," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13, 1998.

- [MCF+02] McClure, D., Collier, B., de Farrari, J., *Challenges to Effective Adoption of the Extensible Markup Language*, GAO-02-327, United States General Accounting Office, Washington, DC, April 2002.
- [McG97] McGibbon, T., *Modern Empirical Cost and Schedule Estimation Tools*, DoD Data & Analysis Center for Software, Rome, NY, August 20, 1997.
- [McG01] McGibbon, T., *A Business Case for Software Process Improvement Revised*, DoD Data & Analysis Center for Software, Rome, NY, September 30, 1999.
- [MCL+99] Myjak, M., Clark, D., Lake, T., "RTI Interoperability Study Group Final Report," *Proceedings of the 1999 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 1999.
- [McQ97] McQueen, S., *TMD System Exerciser Program Management Plan*, Ballistic Missile Defense Organization, Washington, DC, December 10, 1997.
- [MCR+00] McClure, D., Christian, J., Rehberger, J., *Information Technology Investment Management, A Framework for Assessing and Improving Process Maturity*, Exposure Draft, GAO/AIMD-10.1-23, United States General Accounting Office, Washington, DC, May 2000.
- [MCU+95] McCue, B., Callaham, M., Unger, D., *Distributed Interactive Simulation of Combat*, Office of the Technology Assessment, U.S. Government Printing Office, Washington, DC, September 1995.
- [MD02] Milks, W., Dilliard, G., "Development of a Product Line Architecture for Army Live Training," *Proceedings of the 2002 Fall Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-13, 2002.
- [MDA02a] Missile Defense Agency, *Mission-Related Scientific and Technical Data/Information Management*, MDA Directive 3200.01, Missile Defense Agency, Washington, DC, June 10, 2002.
- [MDA02b] Missile Defense Agency, *MDA Glossary*, Version 4.0, External Affairs, Missile Defense Agency, Washington, DC, accessed on December 15, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/bmdolink.html>.
- [MDA02b] Missile Defense Agency, *MDA Glossary*, Version 4.0, External Affairs, Missile Defense Agency, Washington, DC, accessed on December 15, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/bmdolink.html>.

- [MDA02c] Missile Defense Agency, *The Threat*, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/threat.html>.
- [MDA02d] Missile Defense Agency, *Ballistic Missile Defense Basics*, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/basics.html>.
- [MDA02e] Missile Defense Agency, *The Ballistic Missile Defense System*, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/system.html>.
- [MDA02f] Missile Defense Agency, *Boost Defense Segment*, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/boost.html>.
- [MDA02g] Missile Defense Agency, *Midcourse Defense Segment*, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/midcourse.html>.
- [MDA02h] Missile Defense Agency, *Sensors*, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/sensor.html>.
- [MDA02i] Missile Defense Agency, *Ballistic Missile Defense Approach*, Fact Sheet, October 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02j] Missile Defense Agency, *Ballistic Missile Defense Challenge*, Fact Sheet, October 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02k] Missile Defense Agency, *Ballistic Missile Defense Fiscal Year 2003 Budget*, Fact Sheet, October 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02l] Missile Defense Agency, *System Integration and Engineering*, Fact Sheet, March 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.

- [MDA02m] Missile Defense Agency, *Test and Evaluation*, Fact Sheet, March 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02n] Missile Defense Agency, *Boost Phase Missile Defense*, Fact Sheet, March 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02o] Missile Defense Agency, *Airborne Laser*, Fact Sheet, March 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02p] Missile Defense Agency, *Kinetic Energy*, Fact Sheet, March 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02q] Missile Defense Agency, *Ground Based Midcourse*, Fact Sheet, October 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002 from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02r] Missile Defense Agency, *Sea-Based Midcourse*, Fact Sheet, October 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02s] Missile Defense Agency, *Targets and Countermeasures*, Fact Sheet, October 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02t] Missile Defense Agency, *Terminal Phase Missile Defense*, Fact Sheet, March 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02u] Missile Defense Agency, *THAAD*, Fact Sheet, October 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02v] Missile Defense Agency, *Patriot Advanced Capability-3*, Fact Sheet, March 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02w] Missile Defense Agency, *Technology Segment*, Fact Sheet, October 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.

- [MDA02x] Missile Defense Agency, *Space Tracking and Surveillance System (STSS)*, Fact Sheet, March 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02y] Missile Defense Agency, *Terminal Phase Missile Defense*, Fact Sheet, March 2002, Missile Defense Agency, Washington, DC, accessed on December 23, 2002, from <http://acq.osd.mil/bmdo/bmdolink/html/factsheet.html>.
- [MDA02z] Missile Defense Agency, *Capability Assessment Plan*, Version 4.5, Missile Defense Agency, Washington, DC, August 26, 2002, pp. A-6.
- [MDA03a] Missile Defense Agency, *Terminal Defense Segment*, Missile Defense Agency, Washington, DC, accessed on January 4, 2003, from <http://acq.osd.mil/bmdo/bmdolink/html/terminal.html>.
- [MDA03b] Department of Defense, *Missile Defense Operations Announcement*, News Release No. 642-02, Department of Defense, Washington, DC, December 17, 2002, accessed on 4 January 2003 from http://www.defenselink.mil/news/Dec2002/b12172002_bt642-02.html.
- [MDB00a] McEver, J., Davis, P., Bigelow, J., "Implementing Multiresolution Models and Families of Models: From Entity-Level Simulation to Desktop Stochastic Models and 'Repro' Models," *Exploratory Analysis and a Case History of Multiresolution, Multiperspective Modeling*, RAND National Defense Research Institute reprint, Santa Monica, CA, 2001.
- [MDB00b] McEver, J., Davis, P., Bigelow, J., *EXHALT: An Interdiction Model for Exploring Halt Capabilities in a Large Scenario Space*, RAND National Defense Research Institute, Santa Monica, CA, 2001.
- [MDW+01] Mathis, M., Dutchyshyn, H., Wilson, J., *Single Integrated Air Picture (SIAP) Attributes*, Technical Report 2001-001, Single Integrated Air Picture (SIAP) System Engineering Task Force, Arlington, VA, June 2001.
- [Mec64] Meckling, W., "Strategies for Development," *Analysis for Military Decisions*, R-387-PR, reprinted by the Military Operations Research Society, The RAND Corporation, 1964, pp. 227-236.
- [MEH01] Maier, M., Emery, D., Hilliard, R., "Software Architecture: Introducing IEEE Standard 1471," *IEEE Computer*, April 2001, pp. 107-109.
- [Men02] Mens, T., "A State-of-the-Art Survey on Software Merging," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 5, May 2002, pp. 449-462.

- [Met92] Metzger, J., "Chapter III – Verification," *Simulation Validation Workshop Proceedings (SIMVAL II)*, Military Operations Research Society, Alexandria, VA, 31 March – 2 April 1992, pp. III-1 – III-11.
- [Met00] Metz, M., "Comparing the Joint Warfare System (JWARS) Conceptual Model to a Conceptual Model Standard," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [MET02] Morisio, M., Ezran, M., Tully, C., "Success and Failure Factors in Software Reuse," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 4, April 2002, pp. 340-357.
- [Mey98] Meyer, R., "You Say Resolution, I Say Fidelity, You Say Accuracy, I Say Detail,...Let's NOT Call the Whole Thing OFF!," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13, 1998.
- [Mey01] Meyer, R., "Joint Modeling and Simulation System (JMASS) What it does...and What it doesn't!" *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [MG96] Monroe, R., Garlan, D., "Style-Based Reuse for Software Architectures," *Proceedings of the 1996 International Conference on Software Reuse*, 1996.
- [MG00] Money, A., Gansler, J., *Commercial Item Acquisition: Considerations and Lessons Learned*, Office of the Secretary of Defense, Washington, DC, July 14, 2000.
- [MG98] MacKinnon, S., Gibson, K., *Warsim 2000 Crosswalk with the OMSC Object Model Standard*, National Simulation Center, February 26, 1998.
- [MH98] Margopoulos, W., Haggerty, A., "End-To-End Systems Lethality and Ground Effects Simulation," *American Institute of Aeronautics and Astronautics*, 1998.
- [MHL+02] Morton, R., Haugh, B., Loaiza-Lemos, F., "A Comparison of the Joint Common Database Data Model and the Warsim Object Model," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, 2002.

- [Mic02a] Michael, J., *Challenges & Ways Forward: Building Safety into BMDS*, presentation by Dr. J. Bret Michaels, Department of Computer Science, Naval Postgraduate School, Monterey, CA, March 5, 2002.
- [Mic02b] Michael, J., *Safety and BMDS: Background Material on Safety-Critical Functions*, presentation by Dr. J. Bret Michaels, Department of Computer Science, Naval Postgraduate School, Monterey, CA, March 5, 2002.
- Mic03] Michael, J discussions between author and Dr. J. Bret Michaels, Department of Computer Science, Naval Postgraduate School, Monterey, CA, June 2003.
- [Mih72] Mihram, G., "Some Practical Aspects of the Verification and Validation of Simulation Models," *Operations Research Quarterly*, Volume 23, 1972, pp. 17-29.
- [Mil01] Miller, R., "Are We Doing Enough V&V to Support Accreditation of Acquisition Support M&S? (DRAFT White Paper), Under Secretary of Defense Acquisition, Technology and Logistics, May 8, 2001.
- [Mil02a] Milicev, D., "Domain Mapping Using Extended UML Object Diagrams," *IEEE Software*, Vol. 19, No. 2, March-April 2002, pp. 90-97.
- [Mil02b] Milicev, D., "Automatic Model Transformations Using Extended UML Object Diagrams in Modeling Environments," *IEEE Transactions on Software Engineering*, Vol. 28, No. 4, April 2002, pp. 413-431.
- [MJL01] Murrah, M., Johnson, C., Luqi, "Enhancements and Extensions of Formal Models for Risk Assessment in Software Projects," *Progress Report: Assessment of Defense Modeling And Simulation Office (DMSO) Conceptual Models of the Mission Space*, Naval Postgraduate School, Monterey, CA, December 31, 2001.
- [MK98] Morse, P., Kimball, G., *Methods of Operation Research*, First Edition revised, Military Operations Research Society, Alexandria, VA, 1998.
- [ML97] Muessig, P., Laack, D., "Cost Effective VV&A: Five Prerequisites," Joint Accreditation Support Activity (JASA), Naval Air Warfare Center, China Lake, CA accessed on June 2, 2001 from:
<http://www.nawcwpns.navy.mil/~jasa/library/vva/papers/CostEffective/cstef.html>.
- [ML01] McIntyre, R, Lockhart, R., *Joint C4ISR Outcome-Based Integrated Architecture Assessment (JCOBIAA), Joint Feasibility Study Report*, Under Secretary of Defense Acquisition, Technology and Logistics, September 2001.

- [MLJ01] Murrah, M., Luqi, Johnson, C., "Enhancements & Extensions of Formal Models for Risk Assessment in Software Projects," *Proceedings of Monterey Workshop 2001*, Monterey, CA, June 18-22, 2001, pp. 191-197.
- [MLU+00] Miller, C., Lee, M., Uzdinski, J., Shelby, J., Lyle P., *Qualification: An Incremental Approach to Modeling and Simulation Verification, Validation, and Accreditation*, Navy Theater Wide Program, Ballistic Missile Defense Office, Office of the Secretary of Defense, Washington, DC.
- [MLW+96] McClure, D., Latta, D., Wright, A., *Information Technology Investment: Agencies Can Improve Performance, Reduce Costs, and Minimize Risks*, GAO/AIMD-96-64, United States General Accounting Office, Washington, DC, September 24, 1996.
- [MLW01] Muessig, P., Laack, D., Wroblewski, J., *An Integrated Approach to Evaluating Simulation Credibility*, Joint Accreditation Support Activity (JASA), Naval Air Warfare Center, China Lake, CA, accessed on August 20, 2001 from <http://www.nawcwpns.navy.mil/~jasa/>.
- [MM98] Morrison, J., McKay, M., "A Statistical Framework for Simulation Model Design and Validation," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [MM01] Malveau, R., Mowbray, T., *Software Architect Bootcamp*, Prentice Hall, Upper Saddle River, NJ, 2001.
- [MNJ+02] McGregor, J., Northrop, L., Jarrad, S., "Initiating Software Product Lines," *IEEE Software*, July-August 2002, pp 24-27.
- [Mon00a] Money, A., *Use of the Ada Computer Programming Language*, Assistant Secretary of Defense memorandum, Washington, DC, January 24, 2000.
- [Mon00b] Money, A., *Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP)*, Assistant Secretary of Defense, Washington, DC, July 2000.
- [Moo65] Moore, G., "Cramming More Components onto Integrated Circuits," *Electronics*, Vol. 38, No. 8, April 19, 1965.
- [Moo99] Moore, J., "An Integrated Collection of Software Engineering Standards," *IEEE Software*, 16, November-December 1999, pp. 51-57.

- [Moo01] Moore, M., "Software Reuse: Silver Bullet," *IEEE Software*, September-October 2001, pp. 86.
- [Moo02] Moore, M., "Reverse Engineering and Reengineering: After Y2K," *IEEE Software*, Vol. 19, No. 1, January/February 2002, pp. 8-10.
- [Mor00a] Morisawa, Y., "A Computing Model of Product Lines for Distributed Processing Systems, its Product Sets, and its Applications," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 371-394.
- [Mor00b] Morrow, W., *Defense Science Board Report on the Adequacy of the DoD Science and Technology Program*, Office of the Undersecretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, June 1, 2000.
- [Mor01] Morosco, J., "Threat Modeling and Simulation In Support of Test and Evaluation," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [MOR02] Military Operation Research Society, *M&S, VV&A and T&E: A Report by the Ad Hoc Committee on M&S and VV&A in T&E*, Military Operation Research Society, Alexandria, VA, 2002.
- [Mos00] Moss, T., *DoD M&S Master Plan Baseline Assessment*, Modeling and Simulation Center Information Analysis Center presentation, Defense Modeling and Simulation Office (DMSO), November 6, 2000.
- [Mos02] Mosemann, L., "Did We Lose Our Religion?" *Crosstalk: The Journal of Defense Software Engineering*, Vol. 15, No 8, Software Technology Support Center, Hill AFB, UT, August 2002, pp. 22-25.
- [MR01] Mathiasen, G., Reeder, S., *Phase 2 Air Force SSR Development for Extended Air Defense Testbed (EADTB), AWACS and CRC System SSRs*, Version 1.1, Theater Aerospace Command and Control Simulation Facility (TACCSF), Kirtland AFB, NM, June 2001.
- [MR02] Maier, M., Rechtin, E., *The Art of Systems Architecting*, 2nd ed., CRC Press, New York, NY, 2002.
- [MRW77] McCall, J., Richards, P., Walters, G., "Factors in Software Quality," Vols. 1-3, AD/A-049-014/015/055, *National Technology Information Service*, Springfield, VA, 1977.

- [MS02] Mitchell, B., Spriesterbach, T., *Confidence-Based Performance Assessments for the BMDO Family of Systems*, The Johns Hopkins University, Applied Physics Laboratory, Laurel, MD, 2002.
- [MSB+00] Myers, J., Smith, P., Bender, F., *Hercules Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, May 25, 2000.
- [MSI+99] Matsumura, J., Steeb, R., Isensee, E., *Joint Operations in the 21st Century*, Arroyo Center National Defense Research Institute, RAND, Santa Monica, CA, 1999.
- [MSO01] Myrtveit, I., Stensrød, E., Olsson, U., "Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods," *IEEE Transactions on Software Engineering*, Vol. 27, Issue 11, November 2001, pp. 999-1013.
- [MT98] McGlynn, L., Timian, D., "Army Model and Simulations - A Solid Foundation for V&V and Accreditation," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [MT00] Medvidovic, N., Taylor, R., "A Classification and Comparison Framework for Software Architecture Description Languages," *IEEE Transactions on Software Engineering*, Vol. 26, No. 1, 2000, pp. 70-93.
- [MT01] Musse, S., Thalman, D., "Hierarchical Model for Real Time Simulation of Virtual Human Crowds," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 7, No. 2, April-June 2001, pp. 152-164.
- [MTC96] Mead, N., Tobin, L., Couturiar, S., *Best Training Practices Within the Software Engineering Industry*, (CMU/SEI-96-TR-034), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, November 1996.
- [MTW96] Medvidovic, N., Taylor, R., Whitehead, E., *Formal Modeling of Software Architectures at Multiple Levels of Abstraction*, Department of Information and Computer Science, University of California, Irvine, CA, 1996.
- [Mue97a] Muessig, P., *A 'SMART' Approach to VV&A*, Joint Accreditation Support Activity (JASA), Naval Air Warfare Center, China Lake, CA, accessed on June 7, 2001, from <http://www.nawcwpns.navy.mil/~jasa/library/vva/papers/SmartApproach/>.

- [Mue97b] Muessig, P., "Cost vs. Credibility: How Much V&V is Enough?," Joint Accreditation Support Activity (JASA), Naval Air Warfare Center, China Lake, CA, accessed on June 2, 2001, from <http://www.nawcwpns.navy.mil/~jasa/library/vva/papers/CosrVCred/cost>,
- [Mus02] Mussington, D., *Concepts for Enhancing Critical Infrastructure Protection: Relating Y2K to CIP Research and Development*, RAND Science and Technology Policy Institute, Santa Monica, CA, 2002.
- [MV96] Marciniak, J., Vienneau, R., *Software Engineering Baselines*, Data and Analysis Center for Software, Rome Laboratory, Rome, NY, July 18, 1996.
- [MW02] Mitra, P., Wiederhold, G., *An Ontology-Composition Algebra*, draft paper, accessed on March 28, 2003 from <http://www-db.stanford.edu/pub/gio/paperlist.html>.
- [MWC00] Might, R., Willis, R., Crain, F., "Developing Requirements for Analytical Models and Simulations," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [MWD+00] Moss, T., Wrigley, J., Dickinson, R., *Warfighter M&S Needs Assessment of the Unified Commands and Selected Supporting Commands (Final Report)*, Modeling and Simulation Information and Analysis Center, Defense Modeling and Simulation Office (DMSO) Washington, DC, November 17, 2000.
- [Mye82a] Myers, M., *How Much Does the Federal Government Spend on Economic Modeling Activities?*, PAD-82-26, United States General Accounting Office, Washington, DC, February 16, 1982.
- [Mye82b] Myers, M., *Survey to Identify Models Used By Executive Agencies in the Policymaking Process*, GAO/ PAD-82-46, United States General Accounting Office, Washington, DC, September, 1982.
- [Mye99] Myers, M., "An Investment-Based Approach for Managing Software Intensive Systems," *Acquisition Review Quarterly*, Winter 1999, pp. 61-69.
- [NAB+02] Nichols, R., Anderson, J., Baker, J., *Remotely Sensed Data From Space: Distribution, Pricing, and Applications*, U.S. Congress, Office of Technology Assessment, U.S. Government Printing Office, Washington, DC, May-July 1992.
- [NAS97] NASA, *Formal Methods Specification and Analysis Guidebook for the Verification of Software and Computer Systems, Volume II: A Practitioner's*

Companion [NASA-GB-001-97], 1997, accessed on August 9, 2001 from http://eis.jpl.nasa.gov/quality/Formal_Methods/document/NASAgb2.pdf.

- [Nay02] Naylor, S., "Fixed War Games?" *Army Times*, August 26, 2002, p. 8.
- [Nel01] Nelson, P., "A Requirements Specification of Modifications to the Functional Description of the Mission Space Resource Center," Master thesis, *Progress Report: Assessment of Defense Modeling And Simulation Office (DMSO) Conceptual Models of the Mission Space*, Naval Postgraduate School, Monterey, CA, December 31, 2001.
- [NF67] Naylor, T., Finger, J., "Verification of Computer Simulation Models," *Management Science*, Vol. 14, Issue 2, pp. B92-B101, 1967.
- [Ngu95] Nguyen, D., *An Architectural Model for Software Component Search*, Ph.D. dissertation, Naval Postgraduate School, Monterey, CA, 1995.
- [NJL00] Nogueira, J, Jones, C., Luqi, "Surfing the Edge of Chaos: Applications to Software Engineering," *IEEE Workshop on Rapid Prototyping RSP 2000*, Paris, France, June 2000.
- [NLB00a] Nogueira, J, Luqi, Berzins, V., "Risk Assessment in Software Requirement Engineering," *IDTP 2000, Dallas, TX*, June 2000.
- [NLB00b] Nogueira, J, Luqi, Bhattacharya, S., "Risk Assessment Model for Software Prototyping Projects," *IEEE Workshop on Rapid Prototyping RSP 2000*, Paris, France, June 2000.
- [NLB+00] Nogueira, J, Luqi, Berzins, V., "A Formal Risk Assessment Model for Software Evolution," *SEKE 2000, Chicago, IL*, July 2000.
- [NLR99] Nada, N., Luqi, Rine, D., *A Knowledge-Based Tutoring System for Software Reuse Practices*, Naval Postgraduate School, Monterey, CA, 1999.
- [NM98] Nayani, N., Mollaghasemi, M., "Validation and Verification of the Simulation Model of a Photolithography Process in Semiconductor Manufacturing," *Proceedings of the 1998 Winter Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13, 1998.
- [NMY+98] Norton, W., McGlynn, L., Youngblood, S., *White Paper Report of the Verification, Validation & Certification (VV&C) Tiger Team*, VV&A Technical Working Group, Washington, DC, 4 June 1998.

- [NPS02] Naval Postgraduate School, *Software Engineering Doctor of Philosophy Program Handbook*, Naval Postgraduate School, Monterey, CA, April 2002.
- [Nog00] Nogueira, J., *A Formal Model for Risk Assessment in Software Projects*, Ph.D. dissertation, Naval Postgraduate School, Monterey, CA, 2000.
- [Nor03] Northrop, L., "Software Engineering Practice Areas," *A Framework for Software Product Line Practice, Version 4.1*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, accessed on May 16, 2003, from <http://www.sei.cmu.edu/plp/framework.html>.
- [NPS02] U.S. Naval Postgraduate School, *Software Engineering Doctor of Philosophy Program Handbook*, Monterey, CA, March, 2002.
- [NRL95] Naval Research Laboratory, *Environmental Effects for Distributed Interactive Simulations (E2DIS) Program Requirements Document (Draft)* Ver.0.10, Modeling and Simulation Office, Alexandria, VA, April 20, 1995.
- [NS01] Nusser, G., Schimkat, R., "Rapid Application Development of Middleware Components by Using XML," 12th *International Workshop on Rapid System Prototyping*, IEEE Computer Society, 2001, pp. 116-121.
- [Nur02] Nurenberg, G., "Simulating War," *Tech Live*, accessed on August 23, 2002 from <http://www.techtv.com/news/culture/story/0,24195,3395742,00.html>.
- [OAH+97] Odeen, P., Armitage, R., Hearney, R., *Transforming Defense National Security in the 21st Century*, Report of the National Defense Panel, Arlington, VA, December, 1997.
- [OBB+03] O'Rourke, R., Best, R., Bolkcom, C., *Iraq War: Defense Program Implications for Congress*, Congressional Research Service, Penny Hill Press, Washington, DC, June 4, 2003.
- [Obe02] Obering H., *Joint Project Optic Windmill VI Final Report*, Missile Defense Agency, Washington, DC, November 2002.
- [Obr99] O'Bryon, J., "Meet 'MASTER' –Modeling & Simulation Test and Evaluation Reform," *PM*, March-April 1999, pp. 8-14.
- [Obr01a] O'Brien L., *Dali: A Software Architecture Reconstruction Workbench*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 2001.

- [Obr01b] O'Brien L., *Architectural Reconstruction to Support a Product Line Effort: Case Study*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, July 2001.
- [OCD+02] Oppenheimer, J., Chaplain, C., Dinkelacker, D., *NASA: Better Mechanisms Needed for Sharing Lessons Learned*, GAO-02-195, United States General Accounting Office, Washington, DC, January 2002.
- [Oes02] Oestereich, B., *Developing Software with UML, Object-Oriented Analysis and Design in Practice*, 2d edition, Addison-Wesley, London, England, 2002.
- [OF02] O'Connor, M., Faier, A., *Real-time Platform Reference Federation Object Model (RPR FOM)*, Ver. 2.0, Draft 14, unapproved draft, Simulation Interoperability Standards Organization (SISO), Orlando, FL, July 29, 2002.
- [OHR99] Orso, A., Harrold, M., Rosenblum, D., "Component Metadata for Software Engineering Tasks," *Proceedings of EDO 2000*, Springer, LCNS Vol. 1999
- [Oll83] Olle, T., "Database Management" *Encyclopedia of Computer Science and Engineering*, 2d Edition, Ralston, Anthony, Reilly, E., ed., Van Nostrand Reinhold Company, Inc., New York, NY, 1983, pp. 443-447.
- [Ols95] Olsem, M., *Reengineering Technology Report*, Volume 1, Software Technology Support Center, Hill AFB, UT, October 1995.
- [OMB02] Office of Management and Budget, *Federal Enterprise Architecture Business Reference Model*, Office of Management and Budget, Washington, DC July 24, 2002.
- [OMG99] Object Management Group, *OMG Unified Modeling Language Specification Version 1.3*, Framingham, MA, June 1999.
- [OPV+02] O'Donahue, R., Pozda, A., Voigt, R., *Strategic Plan for Transforming DoD Training*, Office of the Under Secretary of Defense for Personnel and Readiness, Washington, DC, March 1, 2002.
- [OS93] Olsem, M., Sittenauer, C., *Reengineering Technology Report*, Volume 1, Software Technology Support Center, Hill AFB, UT, August 1993.
- [OS00] O'Connell, E., Saiedian, H., "Can You Trust Software Capability Evaluations?" *Computer*, February 2000, pp. 28-35.

- [OSA+01] Okraski, H., Shiflet, J., Amico, G., *Modeling and Simulation Professional Certification Commission (DRAFT)*, Modeling and Simulation Professional Certification Commission, November 16, 2001.
- [OSV02] O'Brien L., Stoermer, C., Verhoef, C., *Software Architecture Reconstruction: Practice Needs and Current Approaches*, (CMU/SEI-2002-TR-024), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 2002.
- [Pac97] Pace, D., "Fidelity Considerations for RDE Distributed Simulation," *Proceedings of the 1997 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-12, 1997.
- [Pac98a] Pace, D., "Synopsis of Fidelity Ideas and Issues," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13, 1998.
- [Pac98b] Pace, D., "Dimensions and Attributes of Simulation Fidelity," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [Pac98c] Pace, D., "Impact of Simulation Description on Conceptual Validation," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [Pac98d] Pace, D., "Verification, Validation, and Accreditation," *Applied Modeling and Simulation: An Integrated Approach to Development and Operation*, The McGraw-Hill Companies, Inc., New York, NY, 1998, pp. 369-409.
- [Pac99a] Pace, D., "Development and Documentation of a Simulation Conceptual Model," *Proceedings of the 1999 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 12-17, 1999.
- [Pac99b] Pace, D., "Use of Subject Matter Experts (SMEs) in Simulation Evaluation," *Proceedings of the 1999 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 12-17, 1999.
- [Pac00a] Pace, D., "Ideas About Simulation Conceptual Model Development," *John Hopkins APL Technical Digest*, July-September 2000, Volume 21, Number 3, pp. 327-335.

- [Pac00b] Pace, D., "Simulation Conceptual Model Development," *Proceedings of the 2000 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL. March 26-31, 2000.
- [Pac00c] Pace, D., "Simulation Conceptual Model Development Issues and Implications for Reuse of Simulation Components," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [Pac01a] Pace, D., "Simulation Conceptual Model Role In Determining Compatibility of Candidate Simulations for A HLA Federation," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [Pac01b] Pace, D., "Issues Related to Quantifying Simulation Validation," *Proceedings of the 2001 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 9-14, 2001.
- [Pac01c] Pace, D., *A White Paper on Class and Application Specific Accreditation* John Hopkins Applied Physics Laboratory, Laurel, MD, March 21, 2001.
- [Pac02a] Pace, D., "Capability of Explicit Conceptual Model on M&S Credibility and Reuse," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [Pac02b] Pace, D., "Perspectives on Approaches to Simulation Verification and Validation," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [Pac02c] Pace, D., "Issues Related to Quantifying Simulation Validation," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [Pac02d] Pace, D., *Draft TOR for Simulation Conceptual Model Study Group*, email from Dr. Pace to SISO VV&A Forum, September 11, 2002.
- [PAD78] Comptroller General of the United States, *Models and Their Roles in GAO*, PAD-78-84, United States General Accounting Office, Washington, DC, October 1978.
- [PAD79a] Comptroller General of the United States, *Guidelines for Model Evaluation*, PAD-79-17, United States General Accounting Office, Washington, DC, January 1, 1979.

- [PAD79b] Comptroller General of the United States, *Uses of National Economic Models By Federal Agencies*, PAD-79-24, United States General Accounting Office, Washington, DC, May 16, 1979.
- [PAD80] Comptroller General of the United States, *Models, Data and War: A Critique of the Foundation for Defense Analysis*, PAD-80-21, United States General Accounting Office, Washington, DC, March 12, 1980.
- [Pai97] Paige, E., *Information Technology Management (ITM): Supporting National Defense*, Office of the Assistant Secretary of Defense (Command, Control, Communications, and Intelligence), Washington, DC, March 1997.
- [Par72] Parnas, D., "On the Criteria to Be Used in Decomposing Systems into Modules," *Communications of the ACM*, Vol. 15, No. 12, December 1972, pp.1053-1058.
- [Par76] Parnas, D., "On the Design and Development of Program Families," *IEEE Transactions on Software Engineering*, Vol. SE-2, No. 1, March 1976, pp.1-9.
- [Par79] Parnas, D., "Designing Software for Ease of Extensions and Contractions," *IEEE Transactions on Software Engineering*, Vol. SE-5, No. 2, March 1979, pp.128-137.
- [Par94] Parker, T., "Acquisition Task Force on Modeling and Simulation," *Final Report of the Acquisition Task Force on Modeling and Simulation*, June 17, 1994.
- [Par99] Parnas, D., "Software Engineering Programs Are Not Computer Science Programs," *IEEE Software*, November/December 1999, pp. 19-30.
- [Par01] Parnas, D., *Software Fundamentals, Collected Papers by David L. Parnas*, Hoffman, D., Weiss, D., Eds., Addison-Wesley, Boston, MA, 2001.
- [Pat99] Patel, R., *Core Ballistic Missile Defense Test Range Capability*, "Version 1.0, Ballistic Missile Defense Organization Test Ranges and Instrumentation WIPT, Washington, DC, August 20, 1999.
- [PAT01] PATRIOT Program Office, *Lower Tier Simulation Support Plan (PATRIOT)*, PATRIOT Program Office, Huntsville, AL, January 3, 2001.
- [Pau99] Paul, R., *Assurance-Based Testing and Its Application to Y2K Testing*, Office of the Assistant Secretary of Defense, Year 2000 (Y2K) Office, Washington, DC, 1999.

- [Pay96] Payne, R., *Modeling Space in the Air Force Command Exercises (ACES)*, Master thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, March 1996.
- [PB00] Parsons, G., Barnett, T., *Extended Air Defense Testbed (EADTB) Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, January 14, 2000.
- [PBA+94] Parker, T., Bond, M., Anderson, M., *Final Report of the Acquisition Task Force on Modeling and Simulation*, Office of the Director of Defense Research and Engineering, Washington, DC, 10 June 1994.
- [PBG+94] Piowar, T., Burgeson, J., Grantham, D., *A Survey of Requirements For the Effects of the Natural Environment in Military Models and Simulations*, 13th Workshop on Standards for the Interoperability of Distributed Interactive Simulations, September 1995.
- [Pea99] Pearman, G., "A Method to Assess the Fidelity of a DIS-Compliant or HLA-Compliant Legacy Model," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13 1998.
- [Pet02] Petty, M., "Two Aspects of Composability: Lexicon and Theory," *DMSO Workshop on Composable M&S*, July 9, 2002.
- [Pfl98] Pfleeger, S., *Understanding and Improving Technology Transfer in Software Engineering: A DACS State-of-the-Art-Report*, University of Maryland, College Park, MD, November 14, 1998.
- [PFL+00] Pace, S., Frelinger, D., Lachman, B., *The Earth Below: Purchasing Science Data and the Role of Public-Private Partnerships*, DB-316-NASA/OSTP, RAND Science and Technology Policy Institute, March 2000.
- [PGF96] Park, R., Goethert, W., Florac, W., *Goal-Driven Software Measurement – A Guidebook*, (CMU/SEI-96-HB-002), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, August 1996.
- [PGW00] Paulk, M., Goldenson, D., White, D., *The 1999 Survey of High Maturity Organizations*, (CMU/SEI-2000-SR-002), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, February 2000.
- [PH99] Parker, G., Hobbs, I., *CIO University Federal Executive Competencies Learning Objectives*, Ver.1.0, General Services Administration Office of Government-Wide Policy, August 1999.

- [PHP+96] Patenaude, A., Helmuth, D., Potter, P., *Study on the Effectiveness of Modeling and Simulation in the Weapon System Acquisition Process Final Report*, Under Secretary of Defense Acquisition, Technology, and Logistics (USD (AT&L)), October 1996.
- [PHW99] Plakosh, D., Hissam, S., Wallnau, K., *Into the Black Box: A Case Study in Obtaining Visibility into Commercial Software*, (CMU/SEI-99-TN-010), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, March 1999.
- [PKL01] Pickard, L., Kitchenham, B., Linkman, S., "Using Simulated Data Sets to Compare Data Analysis Techniques Used for Software Cost Modeling," *IEEE Proceedings*, Vol.148, No.6. December 2001.
- [PKM+98] Patton, C., Knouse, K., Morrison, F., *Results Act: DoD's Annual Performance Plan for Fiscal Year 1999*, GAO/NSIAD-98-188R, United States General Accounting Office, Washington, DC, June 5, 1998.
- [PLN02] Peters, S., LaVine, N., Napravnik, L., "Composable Behaviors in an Entity Based Simulation, *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [PLL+02] Powell, E., Lessmann, K., Lucas, J., "The Test and Training Enabling Architecture (TENA) 2002 Overview," *Proceedings of the 2002 Fall Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-13, 2002.
- [PLP01] Pretschner, A., Lotzbeyer, H., Philipps, J., "Model Based Testing in Evolutionary Software Development," 12th *International Workshop on Rapid System Prototyping*, IEEE Computer Society, 2001, pp. 155-161.
- [PMR94] Piplani, L., Mercer, J., Roop, R., *Systems Acquisition Manager's Guide for the Use of Models and Simulations*, Defense Systems Management College Press, Fort Belvoir, VA, September 1994.
- [Pol98] Polak, W., "Formal Methods in Practice," *Proceedings of the Monterey Workshop on Engineering Automation for Computer Systems*, Naval Postgraduate School, Monterey, CA, April 1999, pp. 86-92.
- [Pol99] Pollak, B., "Standards for Modeling and Simulation," [News@SEI](#), Vol. 2 No 2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Spring 1999, pp. 6-7.

- [POP00] Pons, C., Olsina, L., Prieto, M., "A Formal Mechanism for Assessing Polymorphism in Object-Oriented Systems," *Proceedings of the First Asia-Pacific Conference on Quality Software*, Hong Kong, China, 2000, pp. 53-62.
- [Por96] Portmann, H., *Study on the Application of Modeling and Simulation to the Acquisition of Major Weapon Systems*, American Defense Preparedness Association (ADPA), September 26, 1996.
- [Pos00] Posner, P., *Budget Issues: July 2000 Update of the GAO's Long-Term Fiscal Simulations*, GAO/AIMD-00-272R, United States General Accounting Office, Washington, DC, July 26, 2000.
- [PP00] Pasetti, A., Pree, W., "Two Novel Concepts for Systematic Product Line Development," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 249-270.
- [PP02] Peters, D., Parnas, D., "Requirements-Based Monitors for Real-Time Systems," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 2, February 2002, pp. 146-157.
- [Pra01] Pradeep, K., "XML as a Data Exchange Medium in Real-Time Systems," Berzins, V. ed., *XML Technology Assessment Progress Report (7/17/2000 – 12/31/2000)*, NPS-SW-01-002, Naval Postgraduate School, Monterey, CA, January 2001, pp. 444-446.
- [Pre97] Pressman, R., *Software Engineering: A Practitioner's Approach*, McGraw-Hill, New York, NY, 1997.
- [PRM97] Pettitt, B., Rhinesmith, F., Milks, W., "The Functional Description of the Battlespace," *Proceedings of the 1997 Spring Simulation Conference* Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 1997.
- [Pro00] Pronk, B., "An Interface-Based Platform Approach," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 331-352.
- [Pru03] Pruett, J., *Holistic Framework for Establishing Interoperability of Heterogeneous Software Development Tools*, dissertation, Naval Postgraduate School, Monterey, CA, June 2003.

- [PSA79] United States General Accounting Office, *Review of GSA's Method of Supply Model*, United States General Accounting Office, Washington, DC, June 1979.
- [PSR+01] Popov, P., Strigini, L., Riddle, S., "Protective Wrapping of OTS Components," Crnkovic, I., Schmidt, H., Stafford, J., eds., *Component Certification and System Prediction*, Proceedings 4th ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, May 14-15, 2001.
- [Pug01] Pugh, R., "Methods for Determining Object Correspondence during System Integration," Master thesis, *Progress Report: Assessment of Defense Modeling And Simulation Office (DMSO) Conceptual Models of the Mission Space*, Naval Postgraduate School, Monterey, CA, December 31, 2001.
- [Put92] Putnam, L., *Linking the QSM Productivity Index with the SEI Maturity Level and Projecting Transitions to Various Maturity Levels*, Quantitative Software Management, Inc., McLean, VA, 1992.
- [Put93] Putnam, L., *The Economic Value of Moving up the SEI Scale*, Quantitative Software Management, Inc., McLean, VA, 1993.
- [PW92] Perry, D., Wolf, A., "Foundations for the Study of Software Architecture," *Software Engineering Notes*, Vol. 17, No. 2, pp. 4052.
- [PW93] Peng, W., Wallace, D., *Software Error Analysis*, NIST Special Publication 500-209, National Institute of Standards and Technology, Gaithersburg, MD, March 1993.
- [PWC+95] Paulk, M., Weber, C., Curtis, B., Chrissis, M., *The Capability Maturity Model: Guidelines for Improving the Software Process*, Addison-Wesley, Reading, MA, 1995.
- [Qua64] Quade, E., *Analysis for Military Decisions*, R-387-PR, reprinted by the Military Operations Research Society, The RAND Corporation, 1964.
- [Rad02] Raduege, H., *Joint Distributed Engineering Plant (JDEP) Capstone Document*, Defense Information Systems Agency, Department of Defense, August 2002.
- [Rag01] Ragsdale, C., *Spreadsheet Modeling and Decision Analysis*, 3rd Edition, South-Western College Publishing, Cincinnati, OH, 2001.
- [Rak01] Rakitin, S., *Software Verification and Validation for Practitioners and Managers*, 2d Edition, Artech House, Norwood, MA, 2001.

- [Ray01a] Raytheon/TRW, *Extended Air Defense Testbed (EADTB) Support Maintenance and Enhancements Program, Software User's Manual*, Vol. I, Huntsville, AL, December 1, 2001.
- [Ray01b] Raytheon/TRW, *Extended Air Defense Testbed (EADTB) Support Maintenance and Enhancements Program, Software User's Manual, Task Reference Manual*, Vol. III, Huntsville, AL, December 1, 2001.
- [Ray01c] Raytheon, *Extended Air Defense Testbed (EADTB) Kill Chain Capability*, Ballistic Missile Defense Organization, October 22, 2001.
- [Ray02a] Raytheon/TRW, *Extended Air Defense Testbed (EADTB) Support Maintenance and Enhancements Program, Software User's Manual, Methodology Manual*, Vol. II, Huntsville, AL, January 14, 2002.
- [RBB+82] Rea, M., Baird, A., Batchelder, F., *Missile System Simulation at the Advanced Simulation Center*, U.S. Army Missile Command, Redstone Arsenal, AL, January 25, 1982.
- [RBB+00] Reddy, R., Berger, I., Benhamou, E., *Resolving the Digital Divide: Information Access, and Opportunity*, The President's Information Technology Advisory Committee, Arlington, VA, February 2000.
- [RC99] Rus, I., Collofello, J., "A Decision Support System for Software Reliability Engineering Strategy Selection, *Proceedings of COMPSAC*, Scottsdale, AZ, October 1999.
- [RC01] Rus, I., Collofello, J., "Integrating Process Simulation and Reliability Models," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, January 2001, pp.15-18.
- [RD00] Roberts, J., Dobbs, V., *A Prototype C4I Federation Object Model (FOM)*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 22, 2000, accessed on April 1, 2002, from http://www.sisostds.org/doclib/obtain_doc.cfm?record_id=REF_1001407.
- [RDB+00] Reddy, R., Benhamou, E., Cerf, V., *Report to the President: Developing Open Source Software to Advance High End Computing*, President's Information Technology Advisory Committee, Panel on Open Source Software for High End Computing, October 2000.
- [RDK+99] Rothenberger, M., Dooley, K., Kulkarni, U., *Characteristics of Systematic Software Reuse Methodologies: The Linkage between Methodology and Adoption Success*, unpublished paper, 1999.

- [Rei92] Reiss, R., "Hierarchy of Models for Conventional Airpower Analysis," *Proceedings of Conference on Variable-Resolution Modeling*, RAND National Defense Research Institute, Washington, DC, 5-6 May 1992, pp. 1-43.
- [Rei93a] Reifer, D., Ed., *Software Management*, 4th Edition, IEEE Computer Society Press, Los Alamedas, CA, 1993.
- [Rei93b] Reifer, D., Ed., "Managing the 3 P's: The Key to Success in Software Management," *Software Management*, 4th Edition, IEEE Computer Society Press, Los Alamedas, CA, 1993, pp. 198-208.
- [RF01] Ray, W., Farrar, A., "Object Model Driven Code Generation for the Enterprise," 12th *International Workshop on Rapid System Prototyping*, IEEE Computer Society, 2001, pp. 84-89.
- [RFH+00] Rodriquez, L, Francis, P., Hunter, J., *Best Practices: A More Constructive Test Approach is Key to Better Weapon System Outcomes*, GAO/NSIAD-00-199, United States General Accounting Office, Washington, DC, July 2000.
- [RFL+98] Rodriquez, L, Francis, P., Lusby, G., *Best Practices: Successful Application to Weapon Acquisitions Requires Changes in DoD's Environment*, GAO/NSIAD-98-56, United States General Accounting Office, Washington, DC, July 2000.
- [RGH00] Roza, M., Gross, D., Harmon, S., "Report Out of the Fidelity Experimentation Implementation Study Group," *Proceedings of the 2000 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 26-31 2000.
- [RGJ98] Roza, M., van Gool, P., Jense, H., "A Fidelity Management Process Overlay onto the FEDEP Model," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13, 1998.
- [RHS99] Ressler, R., Hieb, M., Sudnikovich, "M&S/C4ISR Conceptual Reference Model," *Proceedings of the 1999 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 1999.
- [Ric02] Richey, "Modeling and Simulation CMMI: A Conceptual View," *Crosstalk: The Journal of Defense Software Engineering*, Software Technology Support Center, Hill AFB, UT, March 2002, pp. 29-30.

- [Rie01] Riehle, R., *Understanding Software Engineering*, Naval Postgraduate School, Monterey, CA, June 30, 2001.
- [RIO01] Reynolds, R., Iskenderian, H., Ouzts, S., "The Ocean, Atmosphere and Space Environmental Services (OASES) System," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [RIO02] Reynolds, R., Iskenderian, H., Ouzts, S., "Using Multiple Representations and Resolutions to Compose METOC Environments," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [Rit92] Richie, A., Ed., *Simulation Validation Workshop Proceedings (SIMVAL II)*, Military Operations Research Society, Alexandria, VA, 31 March – 2 April 1992.
- [RJ02] Ramasubramanian, S., Jagadeesan, G., "Knowledge Management at Infosys," *IEEE Software*, Vol. 19, No. 3, May-June 2002, pp. 53-55.
- [RJI02] Rubin, H., Johnson, M., Ivantosh, S., "The U.S. Software Industry," *IEEE Software*, Vol. 19, No. 1, January/February 2002, pp. 95-97.
- [RK02] Randolph, W., King, D., "Scenario Generation: XML to the Rescue," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [RL95] Ramesh, Luqi, "An Intelligent Assistant for Requirements Validation for Embedded Systems," *Journal of Systems Integration*, Vol. 5, No. 2, 1995, pp. 157-177.
- [RL02] Rus, I., Lindvall, M., Knowledge Management in Software Engineering," *IEEE Software*, Vol. 19, No. 3, May-June 2002, pp. 26-38.
- [RMJ+02] Richbourg, R., Miller, D., Janett, A., "The Environmental Data Coding Specification: The Standard for Specification of Environmental Objects and Properties," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [RN99] Rine, D., Nada, N., *Software Reuse Reference Model: Development and Validation*, accessed on December 16, 1999 from:
<http://www.gmu.edu/depts/survey/>.

- [RNJ99] Rine, D., Nada, N., Jaber, K., *Using Adapters to Reduce Interaction Complexity in Reusable Component-Based Software Development*, George Mason University, Fairfax, VA, 1999.
- [Roa98] Roache, P., *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, NM, 1998.
- [Rob98] Robertson, D., "Pitfalls of Formality in Early System Design, *Proceedings of The 1998 Monterey Workshop on Engineering Automation for Computer Systems*, Naval Postgraduate School, Monterey, CA, April, 1999, pp. 135-141.
- [Rod00] Rodrigues, L., *Defense Acquisitions: Status of Strategic C4 System Modernization and Plans to Integrate Additional Mission Capabilities*, GAO/NSIAD-00-212R, United States General Accounting Office, Washington, DC, August 25, 2000.
- [ROG98] Rao, L., Owen, L., Goldsman, D., "Development and Application of a Validation Framework for Traffic Simulation Models," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13, 1998.
- [Ros01] Rosenberg, L., "Verification and Validation Implementation at NASA," *Crosstalk*, Software Technology Support Center, Hill AFB, UT, May 2001.
- [Rot83] Roth, P., "Simulation," *Encyclopedia of Computer Science and Engineering*, 2d Edition, Ralston, Anthony, Reilly, E., ed., Van Nostrand Reinhold Company, Inc., New York, NY, 1983, pp. 1327-1341.
- [RPG00] Department of Defense, *Department of Defense Verification, Validation and Accreditation (VV&A) Recommended Practices Guide (DRAFT)*, Office of the Director of Defense Research and Engineering, Defense Modeling and Simulation Office, Washington, DC, May 15, 2000.
- [RPK02] Redeski, A., Parr, S., Keith-Magee, R., "Component-Based Development Extensions to HLA," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [RR83] Ralston, Anthony, Reilly, E., ed., *Encyclopedia of Computer Science and Engineering*, 2d. Edition, Van Nostrand Reinhold Company, Inc., New York, NY, 1983.
- [RSC00] Rational Software Corporation, *System Variants*, Rational Software Corporation, Cupertino, CA, 2000.

- [RSF+00] Rodrigues, L., Schinasi, K., Francis, P., *Defense Acquisition: Employing Best Practices Can Shape Better Weapon Systems Decisions*, GAO/NSIAD-00-137, United States General Accounting Office, Washington, DC, April 26, 2000.
- [RSH90] Rothenberg, J., Shapiro, N., Hefley, C., *A “ Propagative” Approach to Sensitivity Analysis*, RAND Note N-3192-DARPA, July 1990.
- [RSV+93] Reed, D., Spencer, R., Vincent, D., *Acquisition and Utilization of Modeling and Simulation Assets at Contractor Facilities*, 93-107, Department of Defense, Office of the Inspector General, Washington, DC, June 11, 1993.
- [RTI02] Research Triangle Institute, *The Economic Impacts of Inadequate Infrastructure for Software Testing Final Report*, RTI Health, Social, and Economics Research, Research Triangle Park, NC, May 2002.
- [RTC94] Report to Congress, *DoD Software Reuse Initiative*, accessed on May 18, 2000, from: <http://dii-sw.ncr.disa.mil/ReuseIC/pol-history/report-to-congress.html>.
- [RTM+99a] Rybicki, J., Tharp, D., McPherson, G., *BMDO Modeling and Simulation (M&S) Comprehensive View: Executive Summary Major Defense Acquisition Program (MDAP) and BMDO Program-Level Summaries*, Ballistic Missile Defense Organization, Washington, DC, May 7, 1999.
- [RTM+99b] Rybicki, J., Tharp, D., McPherson, G., *BMDO Modeling and Simulation (M&S) Comprehensive View: Major Defense Acquisition Program (MDAP) and BMDO Program-Level Summaries*, Ballistic Missile Defense Organization, Washington, DC, May 7, 1999.
- [RTM+99c] Rybicki, J., Tharp, D., McPherson, G., *BMDO Modeling and Simulation (M&S) Comprehensive View: Annex A: A Study of the TMD FoS Lethality Intercept / Post-Intercept Assessment Portion of the Weapon System Employment Cycle*, Ballistic Missile Defense Organization, Washington, DC, May 7, 1999.
- [RTM+99d] Rybicki, J., Tharp, D., McPherson, G., *BMDO Modeling and Simulation (M&S) Comprehensive View: Annex B: A Study of the BMDO MDAP Ballistic Missile Threat Modeling Process*, Ballistic Missile Defense Organization, Washington, DC, May 7, 1999.
- [RTM+99e] Rybicki, J., Tharp, D., McPherson, G., *BMDO Modeling and Simulation (M&S) Comprehensive View: Annex A: A Study of the Planned Use of M&S*

in the Development and Test of the TMD Family of Systems, Ballistic Missile Defense Organization, Washington, DC, May 7, 1999.

- [RTM+99f] Rybicki, J., Tharp, D., McPherson, G., *BMDO Modeling and Simulation (M&S) Comprehensive View: Impact Brief*, Ballistic Missile Defense Organization, Washington, DC, July 13, 1999.
- [RTM+99g] Rybicki, J., Tharp, D., McPherson, G., *BMDO Modeling and Simulation (M&S) Comprehensive View: Overview Briefing to Dr. Sanders*, Ballistic Missile Defense Organization, Washington, DC, August 31, 1999.
- [Rub97] Rubin, *Worldwide Benchmark Report 1997 Final Report*, Rubin Systems, Inc., New York, NY, March 13, 1998.
- [Rum01] Secretary of Defense, *Quadrennial Defense Review Report*, Department of Defense, Washington, DC, September 30, 2001.
- [Rum02a] Secretary of Defense, *Nuclear Posture Review Report*, Department of Defense, Washington, DC, January 9, 2002.
- [Rum02b] Secretary of Defense, *Annual Report to the President and the Congress*, Department of Defense, Washington, DC, 2002.
- [Rum02c] Secretary of Defense, *Missile Defense Program Guidance*, Department of Defense, Washington, DC, January 2, 2002.
- [Rus98] Rus, I., *Modeling the Impact on Cost and Schedule of Software Quality Engineering Practices*, Ph.D. Dissertation, Arizona State University, 1998.
- [RVJ+99] Roza, M., Voogd, J., Jense, H., Gool, P., "Fidelity Requirements Specifications Specification: A Process Oriented View," *Proceedings of the 1999 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 12-17, 1999.
- [Sad91] Sadowski, R., "Avoiding the Problems and Pitfalls in Simulation," *Proceedings of the 1991 Winter Simulation Conference*, pp. 48-55.
- [SAF00] Department of the Air Force, SAF/AQIC, *Modeling and Simulation Support Plan Template*, SAF/AQIC, Washington, DC, March 29, 2000.
- [SAG+00] Swank, W., Alfieri, P., Giley, C., *Acquisition Trend Metrics in the Department of Defense*, TR 1-00, Defense Acquisition University Press, Fort Belvoir, VA, October 2000.

- [SAG02] Solution Architects Working Group, *Components-Based Architecture (Guidance and Recommendations) White Paper*, Working Draft Version 1.3, Federal Enterprise Architecture, Program Management Office, Office of Management and Budget, Washington, DC October 24, 2002.
- [Sal02] Sall, K., *XML Family of Specifications: A Practical Guide*, Addison-Wesley, Boston, MA, 2002.
- [San97a] Sanders, P., "Accreditation: An Ingredient for Decision Making Confidence," *Military Modeling for Decision Making*, Third Edition, The Military Operations Research Society, Inc., Alexandria, VA, 1997, pp. 351-360.
- [San97b] Sanders, P., "Simulation Based Acquisition, An Effective, Affordable Mechanism for Fielding Complex Technologies," *Program Manager*, Vol. 26, No. 140, Defense Systems Management College Press, Sep-Oct 1997, pp. 72-76.
- [San97c] Sanders, P., "Simulation Based Acquisition," *Program Manager*, September-October 1997, pp. 72-77.
- [San98a] Sanders, P., "Knowledge Sharing - Global Information Networks," *Tenth Annual Software Technology Conference*, Office of the Under Secretary of Defense (Acquisition and Technology), April 20, 1998.
- [San98b] Sanders, P., "Simulation Based Acquisition: The Revolution Is Coming!" *Army RDA Magazine*, May-June 1999.
- [San99] Sanders, P., "Improving Software Engineering Practice," *Crosstalk: The Journal of Defense Software Engineering*, Vol. 12, No 1, Software Technology Support Center, Hill AFB, UT, January 1999, pp. 4-7.
- [San00] Sanders, P., "Warfare Analysis: Relevant, Reliable, and Right on Time," *John Hopkins APL Technical Digest*, July-September 2000, Volume 21, Number 3, pp. 313-319.
- [SAP98] Smith, R., Andress, R., Parsons, G., "EADTB Experience with Run-time Performance Improvement via Time Warp and Implications for Existing and Developing HLA-Compliant Military Simulations," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18, 1998.
- [Sar91] Sargent, R., "Simulation Model Verification and Validation," *Proceedings of the 1991 Winter Simulation Conference*, edited by Nelson, B., Kelton, W., and Clark, G., Phoenix, AZ, pp. 37-47.

- [Sar92] Sargent, R., " Verification and Validation of Simulation Models," *Proceedings of the 1992 Winter Simulation Conference*, edited by Swain, J., Goldsman, D., Crain, C., and Wilson, J., Arlington, VA, pp. 104-114.
- [Sar98] Sargent, R., "Verification and Validation of Simulation Models," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 9-13, 1998, pp. 121-130.
- [SB99] Schalle, R., *National Air Intelligence Center (NAIC) DXDI Interview Notes*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, December 8, 1999.
- [SB00] Stytz, M., Banks, S., "Using XML to Support Command and Control for an Integrated Air Defense System," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [SB01] Stytz, M., Banks, S., "Enhancing the Design and Documentation of High Level Architecture Simulations Using the Unified Modeling Language," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [SB02a] Stytz, M., Banks, S., "The Dynamic Threat Environment Architecture," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [SB02b] Stytz, M., Banks, S., "XML in Distributed Simulation for Data Management and Architecture Description," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [SBA99] Simulation-Based Acquisition Steering Group (SBA ISG), *Simulation Based Acquisition Functional Description, Version 1.1*, Executive Committee on Modeling and Simulation (EXCIMS), Washington, DC February 24, 1999.
- [SBA00] Simulation-Based Acquisition, "Simulation-Based Acquisition (SBA): A DoD and Service Perspective", *ITEA Modeling and Simulation Workshop*. Office of the Under Secretary of Defense for Acquisition, Technology and Logistics, Washington, DC, December 6, 2000.
- [SBA01a] Simulation-Based Acquisition, "Enabler Classes for Advanced Acquisition Environments," *Acquisition Functional Area Council*, Washington, DC, January 2001.

- [SBA01b] Simulation-Based Acquisition, "Acquisition in the 21st Century," *Simulation-Based Acquisition Conference Summary*, presented to the DoD Acquisition Functional Area Council, Washington, DC, July 11, 2001.
- [SBB+98] Stelling, J., Bernstein, R., Binford, J., *A Road Map for Simulation Based Acquisition*, The Joint Simulation Based Acquisition Task Force, Washington, DC, December 8, 1998.
- [SBB+99a] Stelling, J., Bernstein, R., Binford, J., *Strawman Implementation Plan for Simulation Based Acquisition*, The Joint Simulation Based Acquisition Task Force, Washington, DC, June 14, 1999.
- [SBB+99b] Stelling, J., Bernstein, R., Binford, J., *A Road Map for Simulation Based Acquisition, Revised Executive Summary*, The Joint Simulation Based Acquisition Task Force, Washington, DC, August 3, 1999.
- [SBD+95] Stillman, R., Beers, R., Dolak, M., *Report to the Congress: Information SuperHighway An Overview of Technology Challenges*, GAO/AIMD-95-23, United States General Accounting Office, Washington, DC, January 23, 1995.
- [SBH+01] Solomon, J., Baird, T., Horiuchi, R., *Defense Acquisitions, Space-Based Infrared System-low at Risk of Missing Initial Deployment Date*, GAO-01-6, United States General Accounting Office, Washington, DC, February 2001.
- [SBJ+02] Saxton, J., Bennett, R., Juster, K., *Security in The Information Age: New Challenges, New Strategies*, U.S. Congress, Joint Economic Committee, Washington, DC, May 2002.
- [SBL98] Stiharu-Alexe, I., Belanger, J., Lam, L., "Requirements for Interoperability with Dynamic Models of Different Fidelity," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 14-18 1998.
- [SBM02] Saiedian, H., Bagert, D., Mead, N., "Software Engineering Programs: Dispelling the Myths and Misconceptions," *IEEE Software*, Vol. 19, No. 5, September/October 2002, pp.35-41.
- [SC96] Shaw, M., Clements, P., *A Field Guide to Boxology: Preliminary Classification of Architectural Styles for Software Systems*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 1996.

- [SC02] Sheppard, Cartwright, M., "Predicting with Sparse Data," *IEEE Transactions on Software Engineering*, Vol. 27, Issue 11, November 2001, pp. 986-998.
- [SCC+88] Stever, H., Clem, R., Currie, M., *SDI: Technology, Survivability, and Software*, U.S. Congress, Office of Technology Assessment, U.S. Government Printing Office, Washington, DC, May 1988.
- [Sch91] Scherkenbach, W., *Deming's Road To Continual Improvement*, SPC Press, Knoxville, KY, 1991.
- [Sch92] Scheckeler, G., "A Hierarchy of Models at Different Resolution Levels," *Proceedings of Conference on Variable-Resolution Modeling*, RAND National Defense Research Institute, Washington, DC, May 5-6, 1992, pp. 315-230.
- [Sch96] Schneidewind, N., "Software Reliability Engineering for Client-Server Systems," *Proceedings of The Seventh International Symposium on Software Reliability*, White Plains, New York, October 30, 1996 - November 2, 1996, pp. 226-335.
- [Sch97a] Schneidewind, N., "Successful Application of Software Reliability Engineering for the NASA Space Shuttle," *International Symposium on Software Reliability Engineering*, November 3, Albuquerque, New Mexico, November 4, 1997, pp.71-82.
- [Sch97b] Schneidewind, N., "Reliability Modeling for Safety Critical Software," *IEEE Transactions on Reliability*, Volume 46, No. 1, March 1997, pp. 88-98.
- [Sch99a] Schneidewind, N., *Introduction to Software Reliability with Space Shuttle Example*, Naval Postgraduate School, Monterey, CA, 1999.
- [Sch99b] Schneidewind, N., "Measuring and Evaluating Maintenance Process Using Reliability, Risk and Test Metrics," *IEEE Transactions on Software Engineering*, Vol. 25, No. 6, November/December 1999, pp. 768-781.
- [Sch00a] Schmid, K., "Scoping Software Product Lines," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 513-532.
- [Sch00b] Schneidewind, N., "Software Quality Control and Prediction Model for Maintenance," *Annals of Software Engineering*, Baltzer Science Publishers, Volume 9, May 2000, pp. 79-101.

- [Sch00c] Schneidewind, N., *The Ruthless Pursuit of the Truth about COTS*, Naval Postgraduate School, Monterey, CA, 2000.
- [Sch01a] Scheinberg, P., *Highway Infrastructure: FHWA's Model for Estimating Needs has Been Modified for State-Local Planning*, GAO-01-299, United States General Accounting Office, Washington, DC, February, 2001.
- [Sch01b] Schneidewind, N., *Investigation of the Risk to Software Reliability and Maintainability of Requirements Changes*, Naval Postgraduate School, Monterey, CA, 2001.
- [Sch01c] Schneidewind, N., *Data Analysis of Software Requirements Risk*, Naval Postgraduate School, Monterey, CA, 2001.
- [Sch01d] Schneidewind, "Software Requirements Risk and Responsibility," *Proceedings of the Monterey Workshop 2001*, Monterey, CA, June 18-22, 2001.
- [Sch02a] Schneidewind, N., "Body of Knowledge for Software Quality Measurement," *IEEE Transaction on Software Engineering*, Vol. 28, No. 2, Jan 2002, pp. 77-83.
- [Sch02b] Schrage, M., "Military Overkill Defeats Virtual War," *The Washington Post*, September 22, 2002, p. B5.
- [SED98a] U.S. Army Simulation, Training, And Instrumentation Command, *SEDRIS and The Synthetic Environment Domain*, Defense Modeling and Simulation Office, Alexandria, VA, March 28, 1998.
- [SED98b] U.S. Army Simulation, Training, And Instrumentation Command, *Synthetic Environment Data Representation and Interchange Specification Overview*, Defense Modeling and Simulation Office, Alexandria, VA, March 28, 1998.
- [SED98c] U.S. Army Simulation, Training, And Instrumentation Command, *SEDRIS Common Glossary*, Defense Modeling and Simulation Office, Alexandria, VA, June 26, 1998.
- [SEH+97] Schulz, T., Edwards, L., Hand, D., *Ballistic Missile Defense, Improvements Needed in Navy Area Acquisition Planning*, GAO/NSIAD-98-34, United States General Accounting Office, Washington, DC, November 1997.
- [SEI95] Software Engineering Institute, *Example Barriers*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1995.

- [SEL97] Schulz, T., Edwards, L., Lipscomb, S., *Ballistic Missile Defense: Prototype THAAD System*, NSIAD-97-137R, United States General Accounting Office, Washington, DC, March 27, 1997.
- [SEM97] Software Engineering Measurement and Analysis Team, *Process Maturity Profile of the Software Community 1996 Year End Update*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 1997.
- [SEM00] Software Engineering Measurement and Analysis Team, *Process Maturity Profile of the Software Community 1999 Year End Update*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, March 2000.
- [SFJ+98] Schuster, C., Furr, R., Jones, M., *Joint Military Operations: Weaknesses in DoD's Process for Certifying C4I Systems' Interoperability*, GAO/NSIAD-98-73, United States General Accounting Office, Washington, DC, March 1998.
- [SFL+97] Schow, G., Freeman, L., Lashley, T., Maier, H., Swauger, J., "Derivation and Publication of Simulation Fidelity Metrics," *Proceedings of the 1997 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-12, 1997.
- [SFM97] Storey, M., Fracchia, D., Muller, H., "Cognitive Design Elements to Support the Construction of a Mental Model During Software Visualization," *Proceedings of the 5th Workshop on Program Comprehension*, Dearborn, MI, May 28-30, 1997, pp. 7-28.
- [SG93] Shaw, M., Garlan, D., "An Introduction to Software Architecture," in *Advances in Software Engineering and Knowledge Engineering*, Ambriola, V., and Tortora, G., Eds., World Scientific Publishing Company, River Edge, NJ, 1993.
- [SG94] Standish Group, *The CHAOS Report (1994)*, The Standish Group International, accessed on June 4, 2002 from http://www.Standishgroup.com/sample_research/chaos_1994_1.php.
- [SG96a] Solomond, J., Grable, D., "Software Support: Critical to the Army's Future," *Army RD&A*, March-April 1996, pp. 13-16.
- [SG96b] Shaw, M., Gaines, B., "Requirements Engineering," *Software Engineering Journal*, Vol. 11, No. 3, May 1996, pp. 149-165.
- [SG96c] Shaw, M., Clements, P., *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

- [SG99] Standish Group, *CHAOS: A Recipe for Success*, The Standish Group International, 1999.
- [SH96] Sullivan, G., Harper, M., *Hope Is Not A Method*, Random House, New York, NY, 1996.
- [SH00] Sheehan, J., Haddix, F., *Conceptual Models of the Mission Space: Data Representation and Interface Specification (CMMS-DRIS)*, Defense Modeling and Simulation Office presentation, 2000.
- [SH01] Smits, R., Harris, B., "Data Integration for Simulation Support," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [Sha75] Shannon, R. *System Simulation: The Art and Science*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [Sha93] Shaw, M., "Prospects for an Engineering Discipline of Software," *Software Management*, 4th Edition, IEEE Computer Society Press, Los Alamedas, CA, 1993, pp. 486-495.
- [Sha97] Shalikashvili, J. *Shape, Respond, Prepare Now: A National Military Strategy for a New Era*, The Joint Chiefs of Staff, Washington, DC, 1997.
- [Sha00a] Sharp, D., "Component-Based Product Line Development of Avionics Software," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp.353-370.
- [Sha00b] Shaw, M., "Software Engineering Education: A Roadmap," *The Future of Software Engineering*, Finkelstein, A., Ed., ACM Press, New York, NY, pp. 371-380.
- [Sha01] Shaw, M., "The Coming-of-Age of Software Architecture Research," *Proceedings of the 23rd International Conference on Software Engineering*, Toronto, Canada, IEEE Computer Society, pp. 656-664.
- [SHB00] Seiger, T., Holm, G., Bergsten, U., "Aggregation/Disaggregation Modeling in HLA-based Multi Resolution Simulations," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [She82] Sheley, W., *Deficiencies Identified Within an Urban Warfare Modeling Program at the TRADOC Systems Analysis Activity*, MASAD-82-46, United States General Accounting Office, Washington, DC, August 20, 1982.

- [She97] Sheard, S., *The Frameworks Quagmire, A Brief Look*, Software Productivity Consortium, Herndon, VA, 1997.
- [She00a] Shelton, H., *Joint Vision 2020*, Director for Strategic Plans and Policy, J5; Strategy Division, US Government Printing Office, Washington DC, June 2000.
- [She00b] Sheehan, J., *Conceptual Models of the Mission Space: Communicating Warfighter Requirements to System Engineers*, Defense Modeling and Simulation Office presentation, 2000.
- [SHM02] Surendran, K., Hays, H., Macfarlane, A., "Simulating a Software Engineering Apprenticeship," *IEEE Software*, Vol. 19, No. 5, September/October 2002, pp. 48-56.
- [Sie01] Siegal, J., *Developing in OMG'S Model-Driven Architecture*, Object Management Group White Paper, Revision 2.6, November 2001.
- [Sil92] Silverman, B., "The Software Engineering Paradox: Unexploited Cost Savings and Productivity Improvements," *Proceedings of Conference on Variable-Resolution Modeling*, RAND National Defense Research Institute, Washington, DC, 5-6, May 1992, pp. 456-465.
- [SJW+02] Scharch, S., Jin, B., Wright, D., "Maintainability of the Linux Kernel," *IEEE Proceedings on Software*, Vol. 149, No. 1, February 2002, pp. 18-23.
- [SK02] Sheppard, M., Kadoda, G., "Comparing Software Prediction Techniques Using Simulation," *IEEE Transactions on Software Engineering*, Vol. 27, Issue 1, November 2001, pp.1014-1022.
- [SKO+01] Sekine, S., Kanou, M., Ogato, M., Higashide, A., "Advanced Techniques for MRM," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [SL94] Sogame, H., Ladkin, P., *Aircraft Accident Investigation Report 96-5*, April 26, 1994, accessed on March 17, 2002, from: <http://sunnyday.mit.edu/accidents/>.
- [SLB+98] Shing, M., Luqi, Berzins, V., "The Story of Reengineering of 350,000 Lines of FORTRAN Code," *Proceedings of the Monterey Workshop on Engineering Automation for Computer Systems*, Monterey, CA , April, 1999, pp. 151-155.

- [SLB+99] Shing, M., Luqi, Berzins, V., *Architectural Re-engineering of Janus using Object Modeling and Rapid Prototyping*, Naval Postgraduate School, Monterey, CA, 1999.
- [SLB01] Shing, M., Luqi, Berzins, V., "The Use of Computer-Aided Prototyping for Reengineering Legacy Software," *Proceedings of Monterey Workshop 2001*, Monterey, CA, June 18-22, 2001, pp. 169-180.
- [SLM91] Steigerwald, R., Luqi, McDowell, J., "CASE Tools for Reusable Software Component Storage and Retrieval in Rapid Prototyping," *Information and Software Technology*, England, Vol. 38, no. 9, November 1991, pp. 698-706.
- [SM00] Schulmer, G., Mackenzie, G., *Verification & Validation of Modern Software-Intensive Systems*, Prentice Hall, Inc., Upper Saddle River, NJ, 2000.
- [SMB+99] Stephenson, A., Mulville, D., Bauer, G., *Mars Climate Orbiter Mishap Investigation Board Phase I Report*, November 10, 1999, accessed on March 17, 2002 from: <http://sunnyday.mit.edu/accidents/>.
- [SMP98] Smith, R., Andress, R., Parsons, G., "EADTB Experience with Run-Time Performance Improvement via Time Warp and Implications for Existing and Developing HLA-Compliant Military Simulations," *Proceedings of the 1998 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 1998.
- [SNF+99] Sakamoto, N., Nicol, J., Floyd, D., *Best Practices Applicable to DoD Developmental Test and Evaluation*, Directorate of Test, Systems Engineering and Evaluation, Office of the Secretary of Defense, Washington, DC June 1, 1999.
- [SNH95] Soni, D., Nord, R., Hofmeister, C., "Software Architecture in Industrial Applications," *Proceedings of the 17th International Conference on Software Engineering*, 1995, pp. 196-207.
- [Som95] Somerville, I., *Software Engineering*, Fifth Edition, Addison-Wesley Publishers Ltd., Harlow, England, 1995.
- [Sow88] Sowa, J., *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA, 1988.
- [Spa00] SPARTA, Inc., *Commander's Analysis And Planning Simulation (CAPS) Methodology Manual*, SPARTA, INC., Arlington, VA, 22209, 2000.

- [Spa01] SPARTA, Inc., *Commander's Analysis And Planning Simulation (CAPS) Database Guide*, Version 6.0 SPARTA, INC., Arlington, VA, 22209, December 2001.
- [SPC+98] Sheehan, J., Prosser, T., Conley, H., "Conceptual Models of the Mission Space (CMMS): Basic Concepts, Advanced Techniques, and Pragmatic Examples," *Proceedings of the 1998 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL. March 9-13, 1998.
- [SPG03] Seacord, R., Plakosh, D., Grace, L., *Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices*, Addison-Wesley, Boston, MA, 2003.
- [SPM95] Software Program Manager's Network, *Software Risk Management and Metrics*, Airlie Software Council, Airlie, VA, 1995.
- [SPM99] Software Program Manager's Network, *The Guidebook of Software Acquisition Questions*, Airlie Software Council, Airlie, VA, 1999.
- [SR02] Sametinger, J., Riebisch, M., "Evolution Support by Homogeneously Documenting Patterns, Aspects and Traces," *Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02)*, IEEE Computer Society, 2002.
- [SS92] Seglie, E., Sanders, P., "Accreditation," *Simulation Validation Workshop Proceedings (SIMVAL II)*, Military Operations Research Society, Alexandria, VA, 31 March – 2 April 1992, pp. V-1-V-12.
- [SSJ+00] Smith, L., Stanko, J., Johnson, S., *Guidelines for Successful Acquisition and Management of Software-Intensive Systems: Weapon Systems Command and Control Systems, Management Information Systems*, Version 3.0, Department of the Air Force Software Technology Support Center, May 2000.
- [SSP+00] Stuart, D., Sull, W., Pruitt, S., "The SSEP Toolset for Product Line Development," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 413-436.
- [SST+00] Smyte, W., Slaughter, S., Tincher, D., Blalock, T., Feagan, B., Crooks, R., Foshee, J., Allison, G., *MSIC Notes - Combined with MSIC Revisions Included*, compiled by Berg, S., Battlespace Environments & Signatures Toolkit Project, Ballistic Missile Defense Organization, Office of the Secretary of Defense, Washington, DC, 2000.

- [ST00] Schach, S., Tomer, A., "Development/Maintenance/Reuse: Software Evolution in Product Lines," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 437-450.
- [ST02] Sinclair, M., Tolk, A., "Building up a Common Data Infrastructure," *NATO Studies, Analysis and Simulation (SAS) Symposium on Analyses of the Military Effectiveness of Future C2 Concepts and Systems*, SAS-039, The Hague, April 2002.
- [Sta74] Staats, E., *Review of the 1974 Project Independence Evaluation System*, OPA-76-20, United States General Accounting Office, Washington, DC, April 1976.
- [Sta96] Stark, G., *Return on Investment from a Software Measurement Program*, The MITRE Corporation, Colorado Springs, CO, 1996.
- [Sta97] Starr, S., "Modeling and Simulation to Support the Acquisition Process," *Military Modeling for Decision Making*, Third Edition, Edited by Hughes, W., The Military Operations Research Society, Inc., Alexandria, VA, 1997, pp. 248-270.
- [STB+00] Sell, R., Towers, J., Ballard, G., Graham, J., McMahon, P., "Extended Air Defense Testbed (EADTB) Migrates Toward High Level Architecture (HLA) Compatibility," *Proceedings of the 2000 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 26-31, 2000.
- [Ste02a] Steven, J., "Putting Software Terminology to the Test," *IEEE Software Engineering*, May/June 2002, pp. 88-89.
- [Ste02b] Stenbit, J., *Information Assurance Scholarship Program Academic year 2002-2003: Training Opportunity for Civilian and Military Personnel*, Department of Defense, Washington, DC, September 16, 2002.
- [Ste03a] Stenbit, J., *DoD Net-Centric Data Management Strategy: Metadata Registration*, Department of Defense, Washington, DC, April 3, 2003.
- [Ste03b] Stenbit, J., *DoD Net-Centric Strategy*, Department of Defense, Washington, DC, May 9, 2003.
- [STG+02] Schank, J., Thie, H., Graf, C., *Finding the Right Balance: Simulator and Live Training for Navy Units*, MR-1441-NAVY, RAND Corporation, Santa Monica, CA, October 2002.

- [STR96] STRICOM, *Standard for Distributed Interactive Simulation—Fidelity Description Requirements (DRAFT)*, Institute for Simulation and Training, USA STRICOM, Orlando, FL, February 1996.
- [STS93a] Software Technology Support Center, *Software Management Guide*, Software Technology Support Center, Hill AFB, UT, October 1993.
- [STS93b] Software Technology Support Center, *Software Management Guide, Volume II*, Software Technology Support Center, Hill AFB, UT, October 1993.
- [STS94a] Software Technology Support Center, *Software Configuration Management Technology Report*, Software Technology Support Center, Hill AFB, UT, September 1994.
- [STS94b] Software Technology Support Center, *Project Management Technology Report*, Software Technology Support Center, Hill AFB, UT, December 1994.
- [STS99] Software Technology Support Center, *A Gentle Introduction To Software Engineering*, Software Technology Support Center, Hill AFB, UT, September 1999.
- [STS00] Software Technology Support Center, *Software Quality Assurance*, Software Technology Support Center, Hill AFB, UT, September 2000.
- [Sul02] Sullivan, K., “Enabling Proprietary Information Protection in Simulation Based Acquisition Environments,” *Proceedings of the 2002 Fall Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-13, 2002.
- [Sut99] Sutton, P., *Interoperability Strategy: Quo Vadis?*, Space and Naval Warfare Systems Command, San Diego, CA, 1999.
- [SW96] Sessler, J., Wells, R., Department of Defense, *Campaign Modeling & Simulation for Missile Defense Assessments*, Ballistic Missile Defense Organization, Washington, DC, 1996.
- [SW01] Stafford, J., Wallnau, K., “Is Third Party Certification Necessary,” Crnkovic, I., Schmidt, H., Stafford, J., eds., *Component Certification and System Prediction*, Proceedings 4th ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, May 14-15 2001.
- [SWR+99] Seacord, R., Wallnau, K., Robert, J., *Custom vs. Off-The-Shelf Architecture*, (CMU/SEI-99-TN-006), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, May 1999.

- [TA01] Tran, T., Allen, J., "Interoperability and Security Support for Heterogeneous COTS/GOTS/Legacy Component-Based Architecture," Berzins, V. ed., *XML Technology Assessment Progress Report (7/17/2000 – 12/31/2000)*, NPS-SW-01-002, Naval Postgraduate School, Monterey, CA, January 2001, pp.263-439.
- [TB97] Teewu, W., van den Berg, H., *On the Quality of Conceptual Models*, accessed on July 17, 2001, from <http://osm7.cs.byu.edu/er97/workshop4/tvdb.html>.
- [TBE02a] Teledyne Brown Engineering, *User's Reference Manual Extended Air Defense Simulation (EADSIM)*, Version 10.0, Huntsville, AL, May 2002.
- [TBE02b] Teledyne Brown Engineering, *Methodology Manual Extended Air Defense Simulation (EADSIM)*, Version 10.0, Huntsville, AL, May 2002.
- [TCD+00] Toole, M., Cummings, T., Doar, N., *National Missile Defense Modeling and Simulation Plan*, National Missile Defense Joint Program Office, Washington, DC December 15, 2000.
- [TCO00] Toft, P., Coleman, D., Ohta, J., "A Cooperative Model for Cross Divisional Product Development," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 111-132.
- [TEM00] BMDO/TEM *Theater Missile Defense System Exerciser (TMDSE) Operational Concept Document*, Ballistic Missile Defense Organization, Washington, DC, September 19, 2000.
- [TEM03] MDA/TEM, *Missile Defense Agency Cost Analysis Requirement Description (CARD)*, (Draft), Missile Defense Agency, Washington, DC, December 2002.
- [TG99] Troup, B., Gallagher, B., "Using Contractor Capability Evaluations to Reduce Software Development Risk," *Crosstalk: The Journal of Defense Software Engineering*, Vol. 12, No 8, Software Technology Support Center, Hill AFB, UT, August 1999, pp. 4-8.
- [TH02] Thompson, J., Hardy, C., "Use and Evaluation of the SWEBOK by Postgraduate Students," *Proceedings of the 15th Conference on Software Engineering and Training (CSEET'02)*, 2002.
- [TH03] Tolk, A., Hieb, M., "Building & Integrating M&S Components into C4ISR Systems for Supporting Future Military Operations," *2003 Western Multi*

Conference on Computer and Simulation (WMC'03), Orlando, FL, January 2003.

- [THA02] THAAD Program Office, *THAAD Simulation Support Plan (TSSP)*, Rev. B, THAAD Program Office, Huntsville, AL, July 2002.
- [Tho97a] Thomas. C., "Verification Revisited," *Military Modeling for Decision Making*, Third Edition, Edited by Hughes, W., The Military Operations Research Society, Inc., Alexandria, VA, 1997, pp. 332-349.
- [Tho97b] Thomen, D., "The Need for Conceptual Models of the System Space," *Proceedings of the 1997 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-12, 1997.
- [TIL+95] Target Interaction, Lethality, and Vulnerability Sub-panel (TILV), Joint Directors of Laboratories Technology, *DoD FY95 Master Plan for TILV Science and Technology (S&T) Programs, Volume I, Classical Ballistic Threats*, 4 May 1995 (Revised).
- [Til98] Tilley, S., *A Reverse-Engineering Environment Framework* (CMU/SEI-98-TR-005), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, April 1998.
- [TJ00] Tsai, J., Jaun, E., "Compositional Approach for Modeling and Verification of Component-Based Software Systems," *Proceedings of the 2000 Monterey Workshop on Modeling Software System Structures in a Fast Moving Scenario*, Santa Margherita Ligure, Italy, June 13-16, 2000, pp.267-274.
- [TK02] Tahvildari, L., Kontogiannis, K., "On the Role of Design Patterns in Quality-Driven Re-engineering," *Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02)*, IEEE Computer Society, 2002.
- [TLW+00] Timmian, D., Lacetera, J., Wertman, C., "Report out of the C4I Study Group," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 2000.
- [TM97] Tabacchi, L., Meyers, K., *Defense Information Infrastructure Master Plan: Overview*, Ver.6, Defense Information Systems Agency, June 27, 1997.
- [TNM01] Tan, G., Ng, W., Moradi, F., "Aggregation/Disaggregation in HLA Multi-Resolution Simulation," *IEEE*, National University of Singapore, Singapore, 2001.

- [Tol99] Tolk, A., "Requirements for Simulation Systems When Being Used as Decision Support Systems," *Proceedings of the 1999 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 12-17, 1999.
- [Tol01] Tolk, A., "Bridging the Data Gap - Recommendations for Short, Medium and Long Term Solutions," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [Tol02] Tolk, A., "Avoiding another Green Elephant-A Proposal for the Next Generation HLA based on the Model Driven Architecture," *Proceedings of the 2002 Fall Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-13, 2002.
- [Tol03] Tolk, A., "A Common Framework for Military M&S and C4I Systems," *Proceedings of the 2003 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, April 2003.
- [TOM97] Modeling and Simulation Networks Directorate (TOM), *Model Capabilities Assessment, Extended Air Defense Simulation (EADSIM)*, Version 6.0, Ballistic Missile Defense Organization, Washington, DC, December 31, 1997.
- [TOM99] Modeling and Simulation Networks Directorate (TOM), *Model Capabilities Assessment, Wargame 2000 (WG2K)*, Version 6.0, Ballistic Missile Defense Organization, Washington, DC, January 20, 1999.
- [Too94] Toole, M., *Theater Missile Defense System Exerciser (TMDSE) Build Requirements*, Ballistic Missile Defense Organization, Washington, DC, October 3, 1994.
- [TP96] Tufarolo, J., Page, E., "Evolving the VV&A Process for the ALSP Joint Training Confederation," *Proceedings of the 1996 Winter Simulation Conference*, Coronado, CA, December 8-11, 1996, pp. 952-958.
- [TP00] Thiel, S., Peruzzi, F., "Starting a Product Line Approach for an Envisioned Market," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 495-512.
- [Tra93] Trachsel, W., *Object Interaction in a Parallel Object-Oriented Discrete Event Simulation*, Master thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, December 15, 1993.

- [TRG01] Department of Defense, *Test Resources Management Guide* (Draft), Ballistic Missile Defense Organization (BMDO/TER), Washington, DC, November 29, 2000.
- [TRM01a] Department of Defense, *Technical Reference Model*, Version 2.0, Defense Information Systems Agency Center for Standards, Washington, DC, April 9, 2001.
- [TRM01b] Department of Defense, *Technical Reference Model User Guide*, Version 1.0, Defense Information Systems Agency Center for Standards, Washington, DC, April 10, 2001.
- [TRW00a] TRW Systems, *Joint National Test Facility Verification and Validation Methodology Guide* (Draft) *Volume I*, Joint National Test Facility, Schriever AFB, CO, May 1, 2000.
- [TRW00b] TRW Systems, *Wargame 2000 System/Sub-System Design Description*, Joint National Test Facility, Schriever AFB, CO, August 31, 2000.
- [TRW02a] TRW Systems, *Wargame 2000 Vision Document*, Joint National Test Integration Center, Schriever AFB, CO, January 11, 2002.
- [TRW02b] TRW Systems, *Conceptual Model for Wargame 2000 System/Subsystem Design Description (SSDD)* Appendix B, Joint National Test Integration Center, Schriever AFB, CO, July 31, 2002.
- [TS92] Thomas, B., Schmidt, W., "Building a Knowledge Base for the Knowledge Worker System," *USACERL Interim Report FF-92/02*, August 1992, accessed on March 15, 2000 from <http://www.cecer.army.mil/kws/bevkb.htm>.
- [Tuc02] Tucker, A., "On the Balance between Theory and Practice," *IEEE Software*, Vol. 19, No.5, September-October 2002, pp.94-96.
- [Tzu71] Tzu, S., *The Art of War*, Translated by Samuel B. Griffith, Oxford University Press, New York, NY. 1971.
- [UHC02] Umphress, D., Hendrix, T., Cross, J., "Software Process in the Classroom: The Capstone Project Experience," *IEEE Software*, Vol. 19, No. 5, September/October 2002, pp. 78-85.
- [UML99] Object Management Group, *OMG Unified Modeling Language Specification*, Ver.1.3, Object Management Group, Framingham, MA, June 1999.

- [UMS+01a] Ugone, M., Santoni, C., Mitchell, S., *Management Issues At The Joint Simulation System Program Office*, D-2001-089, United States General Accounting Office, Washington, DC, March 30, 2001.
- [UMS+01b] Ugone, M., Santoni, C., Mitchell, S., *Management Of The Joint Simulation System*, D-2002-005, United States General Accounting Office, Washington, DC, October 12, 2001.
- [VED98] VEDA, Incorporated, *Authoritative Data Source Designation Project, FY98 Initial Report*, Defense Modeling And Simulation Office, Alexandria, VA, March 31, 1998.
- [Ves02] Vessels, P., "Service-Based Architecture for the Network-Centric Battlefield," *DMSO Workshop on Composable M&S*, July 9, 2002.
- [VFH+01] Vito, V., Ferren, B., Howard, B., *Report of the Defense Science Board Task Force on Managed Information Dissemination*, Office of the Undersecretary of Defense For Acquisition, Technology, and Logistics, Washington, DC, October, 2001.
- [VFS+99] Van Buren, J, Friedrich, S., Swenson, R., *Disciplined Software Development: A Review of the Personal Software Process (PSP) and Team Software Process (TSP)*, Software Technology Support Center, Hill AFB, UT, April 30, 1999.
- [VG97] Vann, E., Gravitz, R., "Verification, Validation, and Accreditation (VV&A) Issues of the National Missile Defense (NMD) Integrated System Test Capability) ISTC) Transition to a Distributed Test System," *Proceedings of the 1997Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-12, 1997.
- [Voa99] Voas, J., "Certifying Software for High-Assurance Environments," *IEEE Software*, July-August 1999.
- [Voa02] Voas, J., "Composing Software Component 'ilities'," *IEEE Software*, Vol. 18, No. 4, July-August 2001, pp. 16-17.
- [VT01] Vecellio, G., Thomas, W., *Issues in the Assurance of Component-Based Software*, accessed on May 31, 2002 from http://www.mitre.org/pubs/edge_perspectives/march_01/vecellio.html.
- [VW00] Van Zyl, J., Walker, A.J., "Strategic Product Development," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 85-110.

- [WAA+98] Welch, L., Adolph, C., Albright, P., *Report of the Panel on Reducing Risk in Ballistic Missile Flight Test Programs*, Department of Defense, Director, Operational Test and Evaluation, Washington, DC, February 27, 1998.
- [WAA+99] Welch, L., Adolph, C., Albright, P., *National Missile Defense Review*, Department of Defense, Washington, DC, February 27, 1998.
- [WAD+02] Williams, R., Apo, D., De La Pena, R., *M&S Fidelity TWG Status*, Missile Defense Agency, Washington, DC, November 21, 2002.
- [Wai94] Waite, W., *Partition of 'Truth' and 'Perceived' Data for Representation of BMC3I in DIS Entities, Ensembles and Exercises*, 94-173, Aegis Research Corporation, Huntsville, AL, September 1, 1994.
- [Wai97] Waite, W., "Object-Oriented Representation Schemas for Conceptual Models of the Mission Space (CMMS)," *Proceedings of the 1997 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 3-7, 1997.
- [Wai99] Waite, W., *Extended Air Defense Test Bed (EADTB) High Level Architecture (HLA Migration Plan, Simulation Object Model (SOM) Development Activity)*, Aegis Research Corporation, Huntsville, AL, August 31, 1999.
- [Wai02a] Waite, W., *HLA Federation Design/ Development and Federation Implementation Process Model*, Aegis Research Corporation, email to the author from bwaite@aegistg.com, March 24, 2002.
- [Wai02b] Waite, W., "Using Simulation-The Past as Prologue...Probably," *ITEA Journal*, March-April 2002, pp. 6-8.
- [Wal94] Wallmuller, E., *Software Quality Assurance – A Practical Approach*, Prentice – Hall, 1994.
- [Wap00] Wappler, T., "Remember the Basics," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 73-84.
- [Was01] Washington, D., *Theater Missile Defense System Exerciser, Build 3.2 Capabilities and Limitations*, Ballistic Missile Defense Organization, Department of Defense, Washington, DC, September, 2001.
- [WBF+00] Wheeler, D., Brenner, A., Frickanisce, M., *Ballistic Missile Defense Positioning, Navigation, and Timing Standard*, Ballistic Missile Defense Organization, Washington, DC, July 20, 2000.

- [WBL+01] Woodman, M., Benediktsson, O., Lefever, B., "Issues of CBS Quality and Process Quality," Crnkovic, I., Schmidt, H., Stafford, J., eds., *Component Certification and System Prediction*, Proceedings 4th ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, May 14-15, 2001.
- [WBS+97] Weiderman, N., Bergey, J., Smith, S., *Approaches to Legacy System Evolution*, (CMU/SEI-97-TR-014), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, December 1997.
- [Web95] Webster, *Riverside Webster's II New College Dictionary*, Houghton Mifflin Company, Boston, MA, 1995.
- [Wel00] Welch, L., *National Missile Defense Independent Review Team Executive Summary*, National Missile Defense Independent Review Team, Washington, DC, June 13, 2000.
- [Wel01] Weller, E., "Practical Applications of Statistical Process Control," *IEEE Software*, Vol. 18, No. 3, May/June 2001, pp. 48-55.
- [Wel02] Welch, S., "From Information Delivery to Knowledge Management," *The 2002 Annual Users Meeting and Training Conference*, Ft. Belvoir, VA, April 15-18, 2002.
- [WF93] Wallace, D., Fujii, R., "Software Verification and Validation: An Overview," *Software Management*, 4th Edition, IEEE Computer Society Press, Los Alamedas, CA, 1993, pp. 417-424.
- [WFP+01] Waag, G., Feinberg, J., Painchaud, L., Heist, R., *Modeling and Simulation for Information Assurance, State-of-the-Art Report*, jointly published by the Information Assurance Technology Analysis Center and the Modeling and Simulation Information Analysis Center, Washington, DC, May 11, 2001.
- [WG99] Welch, D., Greaney, K., "Y2K Defect Propagation Risk Assessment using Achilles," *Crosstalk: The Journal of Defense Software Engineering*, Vol. 12, No 11, Software Technology Support Center, Hill AFB, UT, November 1999, pp. 22 (Web Edition).
- [WGB+03] Welch, L., Graham W., Bryant, D., *Defense Science Board on Missile Defense, Phase III – Modeling and Simulation*, V4.2, (Draft), Defense Science Board, Washington, DC, August 2003.
- [WGC02] Wooff, D., Goldstein, M., Coolen, F., "Bayesian Graphical Models for Software Testing," *IEEE Transactions on Software Engineering*, Vol. 28, Issue 5, May 2002, pp. 510-525.

- [WGS84] National Imagery and Mapping Agency, *Department of Defense World Geodetic System-1984*, NIMA TR8350.2, Third edition, Amendment 1, National Imagery and Mapping Agency, Washington, DC, January 3, 2000.
- [WH94] Wieland, F., Heckathorn, H., *An Object-Oriented Technology for Integrating Environmental Effects and Distributed Interactive Simulations*, proceedings of the Object-Oriented Simulation Conference, 1994 SCS Western Multiconference for Computer Simulation, Tempe, Arizona, January 24, 1994.
- [WH02] Wang, F., Hsiung, P., "Efficient and User-Friendly Verification," *IEEE Transactions on Computers*, Vol. 51, Issue 1, January 2002, pp. 61-83.
- [WHA+00] Wong, C., Horn, K., Axelband, E., "Maintaining the Government's Ability to Buy Smart," *Acquisition Review Quarterly*, Summer 2000, Vol. 7, No.3, Defense Systems Management College Press, Ft. Belvoir, VA, 2000, pp. 259-274.
- [Whi99] White, M., "Verification, Validation and Accreditation of Legacy Software - Can We Get There From Here?" *Proceedings of the 1999 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 12-17, 1999.
- [Whi00] Whittaker, J., "What is Software Testing? And Why Is It So Hard?" *IEEE Software*, January-February 2002, pp. 70-79.
- [WHL+01] Wartik, S., Haugh, B., Loaiza, F., "Building in Interoperability: A Comparison of C41 Data Models and Simulation Object Models," *Proceedings of the 2001 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 25-30, 2001.
- [WHL+02] Wartik, S., Haugh, B., Loaiza, F., "Methodological Issues in Alignment of the Land C2 Information Exchange Data Model and the WARSIM Object Model," *Proceedings of the 2002 Fall Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8-13, 2002.
- [WHM+79] Ware, W., Haverty, J., Mosier, R., *Security Controls for Computer Systems: Report of Defense Science Board on Computer Security*, the RAND Corporation, Santa Monica, CA, October, 1979.
- [Wie90] Wiederhold, G., "Future Architectures for Information Processing Systems," *Proceedings of Parbase-90, International Conference on Databases, Paral-*

el Architectures, and their Implementation, IEEE Computer Society Press, March 1990, pp. 160-176.

- [Wie93] Wiederhold, G., "Intelligent Integration of Information," *ACM-SIGMOD 93*, Washington, DC, May 1993, pp. 434-437.
- [Wie96a] Wiederhold, G., Ed., *Intelligent Integration of Information*, Kluwer Academic Publishers, Boston, MA, 1996.
- [Wie96b] Wiegers, K., *Creating a Software Engineering Culture*, Dorset House Publishing, New York, NY, 1996.
- [Wie97a] Wiederhold, G., "The Conceptual Basis for Mediation Services," *IEEE Expert*, Vol. 12, Issue 5, Sep/Oct 1997, pp. 38-47.
- [Wie97b] Wiegers, K., *Software Metrics: Ten Traps to Avoid*, Process Impact, accessed on February 13, 2002 from <http://www.processimpact.com/articles>.
- [Wie99] Wiederhold, G., "Mediation to Deal With Heterogeneous Data Sources," *Interoperating Geographic Information Systems*, Springer LNCS, Zurich, March 1999, pp. 1-16.
- [Wie00a] Wiederhold, G., "Precision in Processing Data from Heterogeneous Sources," *Advances in Databases, Proceedings 17th British National Conference on Databases*, Lings, B., Jefferys, K., eds., Exeter, UK, July 2000, pp. 1-18.
- [Wie00b] Wiederhold, G., "Information Systems that Really Support Decision-making," *Journal of Intelligent Information Systems*, Vol. 14, Kluwer Academic Publishers, March 2000 pp. 85-94.
- [Wie01] Wiederhold, G., *Collaboration Requirements, A Point of Failure in protecting Information*, accessed on March 28, 2003 from <http://www-db.stanford.edu/pub/gio/2001/westpoint-ieee3.htm>.
- [Wie02] Wiederhold, G., *Information Systems that also Project into the Future*, accessed on March 28, 2003 from <http://www-db.stanford.edu/people/gio.html>.
- [Wie03] Wiederhold, G., "The Impossibility of Global Consistency," position paper for the NLM/NSF workshop on *A Research Challenge for Biological Data Management*, accessed on March 28, 2003 from: <http://www-db.stanford.edu/pub/gio/2003/ConsistencyChallenge5Feb2003.htm>.

- [Wil96] Willis, C., "Analysis of Inheritance and Multiple Inheritance," *Software Engineering Journal*, Vol. 11, No. 4, July 1996, pp. 215-214.
- [Wil00] Williams, R., *Inertial, Earth-Centered Earth-Fixed, and Local Coordinate Systems*, technical paper, Ballistic Missile Defense Agency, Washington, DC, April 23, 2000.
- [Wil01] Wile, D., "Ensuring General-purpose and Domain-specific Properties Using Architectural Styles," Crnkovic, I., Schmidt, H., Stafford, J., eds., *Component Certification and System Prediction*, Proceedings 4th ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, May 14-15, 2001.
- [Wil02] Williams, R., *It's Not Brain Surgery-It's Just Rocket Science*, presentation to M&S Directorate, Missile Defense Agency, Washington, DC, 2002.
- [Wil03] Williams, R., *Environmental Standards for MDA Core Models and Simulations*, memorandum to Director, M&S, Missile Defense Agency, Washington, DC, February 4, 2003.
- [Wit96] Withey, J., *Investment Analysis of Software Assets for Product Lines* (CMU/SEI-96-TR-010), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, November 1996.
- [WJ98] Wiederhold, G., Jiang, R., *Access to Simulations in Order to Extend Information Systems*, accessed on March 28, 2003 from <http://www-db.stanford.edu/pub/gio/paperlist.html>.
- [WJC02] Wei, C., Hu, P., Chen, H., "Design and Evaluation of Knowledge Management System," *IEEE Software*, Vol. 19, No. 3, May-June 2002, pp. 56-59.
- [WL99] Weiss, D., Lai, C., *Software Product Line Engineering: A Family-Based Software Development Process*, Addison-Wesley, Reading, MA, 1999.
- [WLL+01] Weiss, K., Leveson, N., Lundquist, K., *An Analysis of Causation in Aerospace Accidents*, Software Engineering Research Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, accessed on March 17, 2002. from <http://sunnyday.mit.edu/accidents/>.
- [WMC02] Westfechtel, B., Munch, B., Conradi, R., "A Layered Architecture for Uniform Version Management," *IEEE Transactions on Software Engineering*, Vol. 27, Issue 12, December 2001, pp. 1111-1133.

- [WNS+97] Weiderman, N., Northrop, L., Smith, *Implications of Distributed Object Technology for Reengineering*, (CMU/SEI-97-TR-005), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 1997.
- [Wof95] Woffinden, S., *Solving the Information Interoperability Puzzle*, HQDA, SAIS-AI, Washington, DC, May 26, 1995.
- [Wol01] Wolfowitz, P., *Command and Control (C2) Legacy Interoperability Strategy and Milestone Action Plan*, Deputy Secretary of Defense memorandum, Washington, DC, October 12, 2001.
- [Wol02a] Wolfowitz, P., *Cancellation of DoD 5000 Defense Acquisition Policy Documents*, Deputy Secretary of Defense memorandum, Washington, DC, October 30, 2002.
- [Wol02b] Wolfowitz, P., *Defense Acquisition*, Deputy Secretary of Defense memorandum, Washington, DC, October 30, 2002.
- [Woo98] Woods, W., *Why Does Army Resource Management Software Never Work?*, paper provided by author, 1998.
- [Woo00a] Wood, W., "Government Product Lines," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 183-193.
- [Woo00b] Woodward, J., *Information Assurance Through Defense in Depth*, The Joint Staff, Washington, DC, February 2000.
- [WPL02] Wroblewski, J., Petty, M., Laack, D., *Does A Good Development Process = A Suitable Model?*, Computer Sciences Corp presentation, 2002.
- [WR94] Wohlin, C., Runeson, P., "Certification of Software Components," *IEEE Transactions on Software Engineering*, Vol. 20, Issue 6, June 1994, pp. 494-499.
- [Wri01a] Wright, S., *Covalidation of Dissimilarly Structured Models*, Ph.D. Dissertation, Air Force Institute of Technology, Wright-Patterson AFB, OH, March 2001.
- [Wri01b] Wrigley, J., *MSIAC Survey of Acquisition M&S Needs*, DMSO Modeling and Simulation Information Analysis Center presentation, October 2, 2001.
- [WS92] Williams, M., Sikora, J., "Chapter I - Overview," *Simulation Validation Workshop Proceedings (SIMVAL II)*, Military Operations Research Society, Alexandria, VA, 31 March – 2 April 1992, pp. I-1-I-6.

- [WS94] Waite, W., Sargent, M., "*Thinker*" *Entity Construct for Military Engagement Simulations*, Aegis Research Corporation, Huntsville, AL, 1994.
- [WS99] Williams, J., Saluto, M., *Re-engineering and Prototyping Legacy Software Systems - Janus Version 6.X*, Master thesis, Naval Postgraduate School, Monterey, CA, March 1999.
- [WS00] Weiner, J., Shrider, K., "Experiencing Federating EADTB with GBRSim to Model Distributed Handover Operations," *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [WS02] Williams, M., Smith, R., "The Development of an Intelligence and Electronic FOM to Bridge Constructive, Virtual, and Live Simulations," *Proceedings of the 2002 Spring Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, March 2002.
- [WSM+96] Worley, R., Simpson, H., Moses, F., Aylward, M., Bailey, M., Fish, D., *Utility of Modeling and Simulation in the Department of Defense: Initial Data Collection*, Institute for Defense Analysis, Alexandria, VA, May 1996.
- [WSW91] Weatherly, R., Seidel, D., Weissman, J., "Aggregate Level Simulation Protocol," *Summer Simulation Conference*, Baltimore, MD, July 1991.
- [WW90] Walker, M., Wiederhold, G., "Acquisition and Validation of Knowledge from Data," *Intelligent Systems*, Ras, Z., Zemankova, M., Eds., Ellis Horwood, May 1990, pp. 415-428.
- [WWF+98] Wang, J., Wiederhold, G., Firschein, O., "Content-based Image Indexing and Searching Using Daubechies' Wavelets," *International Journal of Digital Libraries*, Vol.1, No. 4, 1998, pp. 311-328.
- [WWG93] Weatherly, R., Wilson, A., Griffin, S., "ALSP-Theory, Experience, and Future Directions," *Winter Simulation Conference*, Los Angeles, CA, December 1993.
- [WWN97] Wallnau, K., Weideman, N., Northrop, L., *Distributed Object Technology With CORBA and JAVA: Key Concepts and Implications*, (CMU/SEI-97-TR-004), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, June 1997.

- [XWG02] XML Working Group, *Draft Federal XML Developer's Guide*, U.S. Federal CIO Council Architecture and Infrastructure Committee, Vers. 0.2, Washington, DC, April 2002,
- [YAB+00] Young, T., Arnold, J., Brackey, T., *Mars Program Independent Assessment Team Summary Report*, Mars Program Independent Assessment Team, March 14, 2000, Massachusetts Institute of Technology, accessed on March 17, 2002 from <http://sunnyday.mit.edu/accidents>.
- [YBG+01] Young, P., Berzins, V., Ge, J., "Use of Object Oriented Model For Interoperability in Wrapper-Based Translator for Resolving Representational Differences between Heterogeneous Systems," *Proceedings of Monterey Workshop 2001*, Monterey, CA, June 18-22, 2001, pp. 258-265.
- [YD01] Yurcik, W., Doss, D., "Achieving Fault-Tolerant Software with Rejuvenation and Reconfiguration," *IEEE Software*, Vol. 18, No. 4, July-August 2001, pp. 48-52.
- [YK02] Yeh, D., Wen-Yuan, K., "Reverse Engineering Aggregation Relationship Based on Propagation of Operations," *IEEE Software*, Vol. 19, No. 2, March-April 2002, pp. 90-97.
- [YMK+00] Yacoub, S., Mili., Kaveri, C., "A Hierarchy of COTS Certification Criteria," *Software Product Lines, Experience and Research Directions*, Donohoe, P., ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 397-412.
- [YPE+00] Youngblood, S., Pace, D., Eirich, P., Gregg, D., Coolahan, J., "Simulation, Verification, Validation, and Accreditation," *John Hopkins APL Technical Digest*, July-September 2000, Volume 21, Number 3, pp. 359-367.
- [YPH01] Youngblood, S., Pace, D., Harmon, S., *Federation Credibility Challenges*, Defense Modeling and Simulation Office, Alexandria, VA, July 2001.
- [You89] Yourdon, E., *Modern Structured Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [You93] Yourdon, E., *Decline and Fall of the American Programmer*, Prentice Hall, Englewood Cliffs, NJ, 1993.
- [You97] Youngren, M., "Joint Modeling and Analysis," *Military Modeling for Decision Making*, Third Edition, Edited by Hughes, W., The Military Operations Research Society, Inc., Alexandria, VA, 1997, pp. 193-205.
- [You01] Young, P., "Integration of Heterogeneous Software Systems through Computer-Aided Resolution of Data Representation Differences," Berzins, V.

ed., *XML Technology Assessment Progress Report (7/17/2000 – 12/31/2000)*, NPS-SW-01-002, Naval Postgraduate School, Monterey, CA, January 2001, pp.137-181.

- [You02a] Youngblood, S., *Verification, Validation, & Accreditation Technical Working Group Annual Review*, Defense Modeling and Simulation Office presentation, February 21, 2002.
- [You02b] Youngblood, S., *DMSO VV&A Status Briefing*, Defense Modeling and Simulation Office presentation to the Missile Defense Agency, June 4, 2002.
- [You02c] Young, P., *Heterogeneous Software System Interoperability Through Computer-Aided Resolution of Modeling Differences*, Ph.D. dissertation, Naval Postgraduate School, Monterey, CA, June 2002.
- [YPE00] Youngblood, S., Pace, D., Eirich, P., "Simulation Verification, Validation, and Accreditation," *John Hopkins APL Technical Digest*, July-September 2000, Volume 21, Number 3, pp. 359-367.
- [YS00] Yanni, P., Stanek, C., "Multi-Domain Correlation Engines in the JDN JCTN Environment" *Proceedings of the 2000 Fall Simulation Conference*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 17-22, 2000.
- [ZB02] Zellers, K., Barnes, G., "HLA Implementation of a Multi-Language Legacy Missile Simulation," *Proceedings of the 2002 Fall Interoperability Workshop*, Simulation Interoperability Standards Organization (SISO), Orlando, FL, September 8 13, 2002.
- [ZCE+01] Zimmerman, P., Coulup, M., Eagleberger, S., *JDN Thinker Requirements Document for the SSR Development Program*, Version 2.0.3, Ballistic Missile Defense Agency, MDA/TE, July 13, 2001.
- [Zei91] Ziegler, B., "Object-oriented Modeling and Discrete-Event Simulation," *Proceedings of Conference on Variable-Resolution Modeling*, RAND, National Defense Research Institute, Washington, DC, 5-6 May 1992, pp. 72-95a.
- [Zei92a] Ziegler, B., "Modular Hierarchal Model Representations," *Proceedings of Conference on Variable-Resolution Modeling*, RAND National Defense Research Institute, Washington, DC, 5-6 May 1992, pp. 52-71.
- [Zei92b] Ziegler, B., "Systems Methodology for Structuring Families of Models at Multiple Levels of Resolution," *Proceedings of Conference on Variable-*

Resolution Modeling, RAND National Defense Research Institute, Washington, DC, 5-6 May 1992, pp. 377-408.

- [ZHF98] Zuckerstein, K., Haynes, B., Foster, D., *Decision Nears on Medium Extended Air Defense System*, GAO-NSIAD-98-145 United States General Accounting Office, Washington, DC, June 1998.
- [ZHS02] Zimmerman, P., Heusmann, J., Sciaretta, A., *Executing the DoD Modeling and Simulation Strategy – Making Simulation Interoperability and Reuse a Reality*, DMSO, Alexandria, VA, 2002.
- [Zit97] Zittel, R., *Simulation, IPPD, & Systems Engineering: To the Next Level*, U.S. Department of Defense Acquisition University, Ft. Belvoir, VA, 1997.
- [Zit01] Zittel, R., "The Reality of Simulation-Based Acquisition - And an Example of U.S. Military Implementation," *Acquisition Review Quarterly*, Vol. 8, No. 2, Defense Acquisition University Press, Spring-Summer 2001, pp. 121-131.
- [Zob01] Zobair, H., "Common Data Attributes," Berzins, V. ed., *XML Technology Assessment Progress Report (7/17/2000 – 12/31/2000)*, NPS-SW-01-002, Naval Postgraduate School, Monterey, CA, January 2001, pp. 447-470.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, VA
2. Dudley Knox Library
Naval Postgraduate School
Monterey, CA
3. BMD Information Resource Center (BIRC)
Missile Defense Agency
Washington, DC
4. Software Engineering Automation Center
Naval Postgraduate School
Monterey, CA
5. United States Army War College Library
Carlisle Barracks
Carlisle, PA
6. Professor Luqi
Software Engineering Automation Center
Naval Postgraduate School
Monterey, CA
7. Professor V. Berzins
Software Engineering Automation Center
Naval Postgraduate School
Monterey, CA
8. Professor James B. Michael
Software Engineering Automation Center
Naval Postgraduate School
Monterey, CA
9. Dr. Patricia A. Sanders
Missile Defense Agency (MDA/DB)
Washington, DC
10. Dr. John A. Hamilton
Auburn University
Auburn, AL

11. Professor Mantak Shing
Software Engineering Automation Center
Naval Postgraduate School
Monterey, CA
12. Mr. Richard Riehle
Software Engineering Automation Center
Naval Postgraduate School
Monterey, CA
13. Mr. Dale B. Caffall
Missile Defense Agency (MDA/BC)
Washington, DC
14. Major (P) Thomas Cook
Missile Defense Agency (MDA/SES)
Washington, DC
15. Director, Models and Simulations
Missile Defense Agency (MDA/SES)
Washington, DC
16. COL Kevin J. Greaney
Missile Defense Agency (MDA/SES)
Washington, DC