REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
11-06-2010	Final Report	15-Apr-2005 - 31-Dec-2009

4. TITLE AND SUBTITLE
Documentation Driven Software Development

5a. CONTRACT NUMBER

5b. GRANT NUMBER
ARO MI-PR--

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Luqi

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Naval Postgraduate School (NPS) 14,973.00
Office of Sponsored Programs
Naval Postgraduate School
Monterey, CA 93943 -5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
45614-CS.42

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
Our objective is to develop an integrated, systematic, documentation centric approach to software development, known as Documentation Driven Software Development (DDD). The research issues for DDD are creation and application of three key documenting technologies that will drive the development process and a Document Management System (DMS) that will support them. These technologies address (1) representations for active documents; (2) representations for repositories; (3) methods for analysis, transformation, and presentation of this

15. SUBJECT TERMS
Documentation, Transformation, Multiple Views, Interface, Disambiguation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Lucia Luqi
UU	UU	UU	UU		19b. TELEPHONE NUMBER: 831-656-2735

Report Title

Documentation Driven Software Development

ABSTRACT

Our objective is to develop an integrated, systematic, documentation centric approach to software development, known as Documentation Driven Software Development (DDD). The research issues for DDD are creation and application of three key documenting technologies that will drive the development process and a Document Management System (DMS) that will support them. These technologies address (1) representations for active documents; (2) representations for repositories; (3) methods for analysis, transformation, and presentation of this information.

In addition, we explored new possibilities for computer-aided interfaces that help humans with routine tasks. In doing so we applied Cognitive Science and machine learning methods to design user interfaces that can learn and assist users. We also expanded our work in the area of integration of ontologies from heterogeneous sources. Specifically, we studied Knowledge System Integration Ontology (KSIO) that aligns data and information systems with current situational context for the efficient knowledge collection, integration and transfer. The role of ontology is to organize and structure knowledge (e.g. by standardized terminology) so that semantic queries and associations become more efficient. We assessed the degree to which natural language processing can be usefully applied to the analysis of requirement changes and their impact on system structure and implementation.

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

1. Luqi, L. Zhang, V. Berzins, Y. Qiao, "Documentation Driven Development for Complex Real-Time Systems", IEEE Transactions on Software Engineering 30, 12, p. 936-952.
2. Y. Qiao, H. Wang, Luqi, V. Berzins, "An Admission Control Method for Dynamic Software Reconfiguration in Complex Embedded Systems", International Journal of Computers and Their Applications, Vol. 13, No. 1, March, 2006, pp. 28-38.
3. G. Jacoby, R. Marchany, Davis IV, "Using Battery Constraints Within Mobile Hosts To Improve Network Security," IEEE Security & Privacy Magazine, Summer 2006.
4. G. Jacoby, N. Davis IV, "Battery-Based Intrusion Detection: A Focus on Power for Security Assurance," 2005 Journal of Space and Aeronautical Engineering, 2005.
5. G. Jacoby, Luqi, "Intranet Model and Metrics", Communications of ACM, Volume 50, Issue 2, 2007. pp. 43 – 50.

Number of Papers published in peer-reviewed journals: 5.00

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals: 0.00

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

Peer-Reviewed Conference Proceeding publications (other than abstracts):

1. Y. Qiao, V. Berzins, Luqi, “FCD: A Framework for Compositional Development in Open Embedded Systems”, International Conference on Information Technology, Las Vegas, Nevada, April 2005.
2. Luqi, V. Berzins, William Roof, “Nautical Predictive Routing Protocol (NPRP) for the Dynamic Ad-Hoc Nautical Network (DANN)”, Monterey Workshop 2005: realization of reliable systems on top of unreliable networked platforms, Laguna Beach, California, September, 2005.
3. B. Lewis, “The SAE Architecture Analysis & Design Language (AADL) A Standard for Engineering Performance Critical Systems”, Proc. Embedded Real Time Software Congress, Toulouse France, Jan 25-27, 2006
4. D. Castle, A. Darensburg, B. Griffin, T. Hickman, S. Warders, G. Jacoby, "Gibraltar: A Mobile Host-Based Intrusion Protection System," National Conference on Undergraduate Research, April 2006.
5. Y. Wei, M. Rodríguez, C. Smidts, "How Time-Related Failures Affect the Software System”, in Proc. of The 8th International Conference on Probabilistic Safety Assessment and Management (PSAM 8), New Orleans, Louisiana (USA), May 14-19, 2006.
6. Luqi, “Transforming Documents to Evolve High-Confidence Systems”, Proceedings of Workshop on Advances in Computer Science and Engineering, Berkeley, CA, May 6, 2006, pp. 71-72.
7. V. Berzins, Luqi, “Achieving Dependable Flexibility via Quantifiable System Architectures”, Proceedings of Workshop on Advances in Computer Science and Engineering, Berkeley, CA, May 6, 2006, pp. 53-54.
8. Luqi, L. Zhang, “Documentation Driven Evolution of Complex Systems”, Proceedings of Workshop on Advances in Computer Science and Engineering, May 2006, pp.141-170.
9. T. Buennemeyer, G. Jacoby, R. Marchany, J. Tront, "Battery-Sensing Intrusion Protection System," Proceedings of the 7th IEEE SMC 2006 Information Assurance Workshop, June 21-23, 2006.
10. D. Lange, “PAL Boot Camp: Acquiring, Training, and Deploying Systems with Learning Technology,” In Proceedings of CCRTS 2006: the Command and Control Research and Technology Symposium, San Diego, CA, June 20–22, 2006.
11. G. Jacoby, T. Hickman, S. Warders, B. Griffin, A. Darensburg, D. Castle, “Mobile Intrusion Protection,” Proceedings from The 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing, June 26-29, 2006.
12. B. Huang, M. Rodríguez, J. Bernstein, C. Smidts, “Software Reliability Estimation of Microprocessor Transient Faults”, in Proc. of The 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit (AIAA 2006), Sacramento, CA (USA), July 9-12, 2006.
13. M. Rodriguez, Luqi, V. Ivanchenko, V. Berzins, “Reliability and Flexibility Properties of Models for Design and Run-time Analysis”. In Proceedings of 2006 Monterey Workshop, October 16-18, 2006, Paris, France.
14. Luqi, D. Lange, “Schema Changes and Historical Information in Conceptual Models in Support of Adaptive Systems”, First International Workshop on Active Conceptual Modeling of Learning, Tucson, AZ, 8 November, 2006.
15. Luqi, V. Berzins, W. Roof, “Nautical Predictive Routing Protocol (NPRP) for the Dynamic Ad-Hoc Nautical Network (DANN)”, Springer LNCS 4322, 2007, pp. 106-120.
16. D. Lange, M. Carlin, V. Ivanchenko, V. Berzins, Luqi, “PAL Boot Camp: Preparing Cognitive Assistants for Deployment”. Command and Control Research and Technology Symposium, 2007.
17. V. Berzins, M. Rodríguez, M. Wessman, “Putting Teeth into Open Architectures: Infrastructure for Reducing the Need for Retesting”, Proceedings of the Fourth Annual Research Symposium – Acquisition Research: Creating Synergy for Informed Change, Monterey, CA, May 16-17, 2007, pp.285-312.
18. Luqi, V. Ivanchenko, ”Advanced Interface for Examining and Upgrading Complex Systems”. The World Congress in Computer Science, Computer Engineering, & Applied Computing. Las Vegas, Nevada, USA. June 25-28, 2007.
19. V. Berzins, "Which Unchanged Components to Retest after a Technology Upgrade”, Proceedings of the Fourth Annual Research Symposium – Acquisition Research: Creating Synergy for Informed Change, Monterey, CA, May 14-15, 2008, pp.142-153.
20. C. Georgiou, P.M. Musial, A.A. Shvartsman, and E. Sonderegger, “An Abstract Channel Specification and an Algorithm Implementing It Using Java Sockets.” To appear in Proc. of 7'th IEEE International Symposium on Network Computing and Applications (IEEE NCA), 2008.
21. Luqi, F. Kordon, "Innovations for Requirements Analysis: From Stakeholders to Formal Designs", Proc. Monterey Workshop 2007, Monterey CA, Sep. 2007.
22. V. Berzins, C. Martell, Luqi, V. Ivanchenko, " Innovations on Natural Language Document Processing for Requirements Engineering ", Proc. Monterey Workshop 2007, Monterey CA, Sep. 2007.
23. V. Berzins, Luqi, P. Musial, “Formal Reasoning about Software Object Translations”, Proceedings of the 2008 Monterey Workshop, Budapest, 23-27 Sep. 2008, pp. 65-78.
24. V. Berzins, C. Martell, Luqi, P. Adams, “Innovations in Natural Language Document Processing for Requirements Engineering”, Springer LNCS 5320, 2008, pp. 125-146.
25. Luqi, D. Lange, “Schema Changes and Historical Information in Conceptual Models in Support of Adaptive Systems”, Springer LNCS 4512, 2008, pp. 112-121, ISBN 978-3-540-77502-7.
26. V. Berzins, P. Dailey, “How to Check If It Is Safe Not to Retest a Component”, In Proceedings of the Sixth Annual Research Symposium – Acquisition Research: Defense Acquisition in Transition, Monterey, CA, May 12-14, 2009, pp. 189-200.
27. J. Rivera, Luqi, V. Berzins, “Effective Programmatic Software Safety Strategy for US Navy Gun System Acquisition Programs”, In Proceedings of the Sixth Annual Research Symposium – Acquisition Research: Defense Acquisition in Transition, Monterey, CA, May 12-14, 2009, pp. 159-164.

28. Luqi, F. Kordon, Preface, Proceedings of the Monterey Workshop: Modeling, Development and Verification of Adaptive Systems, 31 March – 2 April 2010, Microsoft Research, Redmond, WA, pp. 2-3.
29. V. Berzins, "How to Certify Software Architectures for Reliable Reconfiguration", Proceedings of the Monterey Workshop: Modeling, Development and Verification of Adaptive Systems, 31 March – 2 April 2010, Microsoft Research, Redmond, WA, pp. 128-129.
30. V. Berzins and P. Dailey, "Improved Software Testing for Open Architecture", Proceedings of the Seventh Annual Research Symposium – Acquisition Research : Creating Synergy for Informed Change, Monterey, CA, May 11-13, 2010.
31. V. Berzins, Luqi, P. Musial, "Formal Reasoning about Software Object Translations", Springer LNCS 6028, 2010, pp. 43-58, ISBN 978 -3-642-12565-2.

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts): 31

(d) Manuscripts

Number of Manuscripts: 0.00

Patents Submitted

Patents Awarded

Graduate Students

NAME	PERCENT_SUPPORTED
Andrew Chen	
Roberto Sandoval	
Doug Anunciado	
Robert Halle	
John Evans	
Paul Dailey	
Joey Rivera	
Bruce Lewis	
Doug Lange	
William Smuda	
William Roof	
Randy Maule	
Ron Chen	
Conrad Whittaker	
Christopher Mushenski	
Matt Lisowski	
FTE Equivalent:	
Total Number:	**16**

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
Cuixia Ma	1.00
Manuel Rodriguez	1.00
Volodymyr Ivanchenko	1.00
Peter Musial	1.00
FTE Equivalent:	**4.00**
Total Number:	**4**

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Luqi	0.25	No
Valdis Berzins	0.25	No
Craig Martell	0.10	No
Grant Jacoby	0.05	No
Peter Musial	0.10	No
FTE Equivalent:	**0.75**	
Total Number:	**5**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	0.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:	0.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:	0.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):	0.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

NAME
Total Number:

Names of personnel receiving PHDs

NAME
William Roof
Doug Lange
William Smuda
Andrew Chen
Paul Dailey
Total Number: 5

Names of other research staff

NAME	PERCENT SUPPORTED	
Allen Ganaden	0.25	No
FTE Equivalent:	**0.25**	
Total Number:	**1**	

Sub Contractors (DD882)

Inventions (DD882)

Documentation Driven Software Development

ARO Final Report

45614CI

Luqi
Computer Science Department
Naval Postgraduate School
Monterey, CA 93943

June 2010

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
06-11-2010	ARO Final Report	04/01/2005-05/31/2010

4. TITLE AND SUBTITLE
Documentation Driven Software Development

5a. CONTRACT NUMBER
45614CI

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Luqi

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Department of Computer Science
1411 Cunningham Road
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-CS-10-008

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. ARMY RESEARCH OFFICE
P.O. BOX 12211
RESEARCH TRIANGLE PARK, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

14. ABSTRACT

The objective of this project is to develop an integrated, systematic, documentation centric approach to software development, known as Documentation Driven Software Development (DDD) approach. The main research issues for DDD are creation and application of three key documenting technologies that will drive the development process and a Document Management System (DMS) that will support them. These technologies address representations for active documents; representations for repositories; and methods for analysis, transformation, and presentation of this information.

15. SUBJECT TERMS
Documentation, Transformation, Multiple Views, Interface, Disambiguation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Luqi
Unclassified	Unclassified	Unclassified	UL		**19b. TELEPHONE NUMBER** *(include area code)* 831-656-2735

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

REPORT DOCUMENTATION PAGE (SF298)

(Continuation Sheet)

Please see the attached pages.

(1) LIST OF PAPERS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS REPORTING PERIOD

Published Works:

Peer-Reviewed Journal Publications

1. Luqi, L. Zhang, V. Berzins, Y. Qiao, "Documentation Driven Development for Complex Real-Time Systems", IEEE Transactions on Software Engineering 30, 12, p. 936-952.
2. Y. Qiao, H. Wang, Luqi, V. Berzins, "An Admission Control Method for Dynamic Software Reconfiguration in Complex Embedded Systems", International Journal of Computers and Their Applications, Vol. 13, No. 1, March, 2006, pp. 28-38.
3. G. Jacoby, R. Marchany, Davis IV, "Using Battery Constraints Within Mobile Hosts To Improve Network Security," IEEE Security & Privacy Magazine, Summer 2006.
4. G. Jacoby, N. Davis IV, "Battery-Based Intrusion Detection: A Focus on Power for Security Assurance," 2005 Journal of Space and Aeronautical Engineering, 2005.
5. G. Jacoby, Luqi, "Intranet Model and Metrics", Communications of ACM, Volume 50, Issue 2, 2007. pp. 43 – 50.

Peer-Reviewed Conference Proceeding Publications

1. Y. Qiao, V. Berzins, Luqi, "FCD: A Framework for Compositional Development in Open Embedded Systems", International Conference on Information Technology, Las Vegas, Nevada, April 2005.
2. Luqi, V. Berzins, William Roof, "Nautical Predictive Routing Protocol (NPRP) for the Dynamic Ad-Hoc Nautical Network (DANN)", Monterey Workshop 2005: realization of reliable systems on top of unreliable networked platforms, Laguna Beach, California, September, 2005.
3. B. Lewis, "The SAE Architecture Analysis & Design Language (AADL) A Standard for Engineering Performance Critical Systems", Proc. Embedded Real Time Software Congress, Toulouse France, Jan 25-27, 2006

4. D. Castle, A. Darensburg, B. Griffin, T. Hickman, S. Warders, G. Jacoby, "Gibraltar: A Mobile Host-Based Intrusion Protection System," National Conference on Undergraduate Research, April 2006.

5. Y. Wei, M. Rodríguez, C. Smidts, "How Time-Related Failures Affect the Software System", in Proc. of The 8th International Conference on Probabilistic Safety Assessment and Management (PSAM 8), New Orleans, Louisiana (USA), May 14-19, 2006.

6. Luqi, "Transforming Documents to Evolve High-Confidence Systems", Proceedings of Workshop on Advances in Computer Science and Engineering, Berkeley, CA, May 6, 2006, pp. 71-72.

7. V. Berzins, Luqi, "Achieving Dependable Flexibility via Quantifiable System Architectures", Proceedings of Workshop on Advances in Computer Science and Engineering, Berkeley, CA, May 6, 2006, pp. 53-54.

8. Luqi, L. Zhang, "Documentation Driven Evolution of Complex Systems", Proceedings of Workshop on Advances in Computer Science and Engineering, May 2006, pp.141-170.

9. T. Buennemeyer, G. Jacoby, R. Marchany, J. Tront, "Battery-Sensing Intrusion Protection System," Proceedings of the 7th IEEE SMC 2006 Information Assurance Workshop, June 21-23, 2006.

10. D. Lange, "PAL Boot Camp: Acquiring, Training, and Deploying Systems with Learning Technology," In Proceedings of CCRTS 2006: the Command and Control Research and Technology Symposium, San Diego, CA, June 20–22, 2006.

11. G. Jacoby, T. Hickman, S. Warders, B. Griffin, A. Darensburg, D. Castle, "Mobile Intrusion Protection," Proceedings from The 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing, June 26-29, 2006.

12. B. Huang, M. Rodríguez, J. Bernstein, C. Smidts, "Software Reliability Estimation of Microprocessor Transient Faults", in Proc. of The 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit (AIAA 2006), Sacramento, CA (USA), July 9-12, 2006.

13. M. Rodriguez, Luqi, V. Ivanchenko, V. Berzins, "Reliability and Flexibility Properties of Models for Design and Run-time Analysis". In Proceedings of 2006 Monterey Workshop, October 16-18, 2006, Paris, France.

14. Luqi, D. Lange, "Schema Changes and Historical Information in Conceptual Models in Support of Adaptive Systems", First International Workshop on Active Conceptual Modeling of Learning, Tucson, AZ, 8 November, 2006.

15. Luqi, V. Berzins, W. Roof, "Nautical Predictive Routing Protocol (NPRP) for the Dynamic Ad-Hoc Nautical Network (DANN)", Springer LNCS 4322, 2007, pp. 106-120.

16. D. Lange, M. Carlin, V. Ivanchenko, V. Berzins, Luqi, "PAL Boot Camp: Preparing Cognitive Assistants for Deployment". Command and Control Research and Technology Symposium, 2007.

17. V. Berzins, M. Rodríguez, M. Wessman, “Putting Teeth into Open Architectures: Infrastructure for Reducing the Need for Retesting”, Proceedings of the Fourth Annual Research Symposium – Acquisition Research: Creating Synergy for Informed Change, Monterey, CA, May 16-17, 2007, pp.285-312.
18. Luqi, V. Ivanchenko, ”Advanced Interface for Examining and Upgrading Complex Systems”. The World Congress in Computer Science, Computer Engineering, & Applied Computing. Las Vegas, Nevada, USA. June 25-28, 2007.
19. V. Berzins, "Which Unchanged Components to Retest after a Technology Upgrade”, Proceedings of the Fourth Annual Research Symposium – Acquisition Research: Creating Synergy for Informed Change, Monterey, CA, May 14-15, 2008, pp.142-153.
20. C. Georgiou, P.M. Musial, A.A. Shvartsman, and E. Sonderegger, “An Abstract Channel Specification and an Algorithm Implementing It Using Java Sockets.” To appear in Proc. of 7'th IEEE International Symposium on Network Computing and Applications (IEEE NCA), 2008.
21. Luqi, F. Kordon, "Innovations for Requirements Analysis: From Stakeholders to Formal Designs", Proc. Monterey Workshop 2007, Monterey CA, Sep. 2007.
22. V. Berzins, C. Martell, Luqi, V. Ivanchenko, " Innovations on Natural Language Document Processing for Requirements Engineering ", Proc. Monterey Workshop 2007, Monterey CA, Sep. 2007.
23. V. Berzins, Luqi, P. Musial, “Formal Reasoning about Software Object Translations”, Proceedings of the 2008 Monterey Workshop, Budapest, 23-27 Sep. 2008, pp. 65-78.
24. V. Berzins, C. Martell, Luqi, P. Adams, “Innovations in Natural Language Document Processing for Requirements Engineering”, Springer LNCS 5320, 2008, pp. 125-146.
25. Luqi, D. Lange, “Schema Changes and Historical Information in Conceptual Models in Support of Adaptive Systems”, Springer LNCS 4512, 2008, pp. 112-121, ISBN 978-3-540-77502-7.
26. V. Berzins, P. Dailey, “How to Check If It Is Safe Not to Retest a Component”, In Proceedings of the Sixth Annual Research Symposium – Acquisition Research: Defense Acquisition in Transition, Monterey, CA, May 12-14, 2009, pp. 189-200.
27. J. Rivera, Luqi, V. Berzins, “Effective Programmatic Software Safety Strategy for US Navy Gun System Acquisition Programs”, In Proceedings of the Sixth Annual Research Symposium – Acquisition Research: Defense Acquisition in Transition, Monterey, CA, May 12-14, 2009, pp. 159-164.
28. Luqi, F. Kordon, Preface, Proceedings of the Monterey Workshop: Modeling, Development and Verification of Adaptive Systems, 31 March – 2 April 2010, Microsoft Research, Redmond, WA, pp. 2-3.
29. V. Berzins, “How to Certify Software Architectures for Reliable Reconfiguration”, Proceedings of the Monterey Workshop: Modeling, Development and Verification

of Adaptive Systems, 31 March – 2 April 2010, Microsoft Research, Redmond, WA, pp. 128-129.

30. V. Berzins and P. Dailey, "Improved Software Testing for Open Architecture", Proceedings of the Seventh Annual Research Symposium – Acquisition Research : Creating Synergy for Informed Change, Monterey, CA, May 11-13, 2010.
31. V. Berzins, Luqi, P. Musial, "Formal Reasoning about Software Object Translations", Springer LNCS 6028, 2010, pp. 43-58, ISBN 978-3-642-12565-2.

Book Chapters

1. Luqi, "Rapid Prototyping", Encyclopedia of Computer Science and Engineering, John Wiley & Sons, January 2009, pp. 2343-2348
2. Luqi, L. Zhang, V. Berzins, "Software Component Repositories", Encyclopedia of Computer Science and Engineering, Wiley, January 2009, pp. 2559-2563.

Technical Reports

1. Luqi, V. Berzins, "Dependable Software Architecture Based On Quantifiable Compositional Model", NPS TR NPS-CS-08-003, 2008.
2. Luqi, C. Martell, "Innovations for Requirements Engineering", NPS TR NPS-CS-08-001, 2008.
3. Luqi, P. Dailey, "Profile-Based Automated Testing Process for Open Architecture Track-Processing Software", Technical Report #NPS-CS-10-005, Mar. 2010.
4. Luqi, V. Berzins, J. Rivera, "Requirements Framework for the Software Systems Safety Review Panel (SSSTRP)", Technical Report # NPS-PM-09-145, Sep. 2009. Also appeared as Technical Report # NPS-GSBPP-10-003, Sep. 2009.
5. Luqi, V. Berzins, P. Dailey, "Driving Automated Open-Architecture Testing: An Operational Profile Model-Development Strategy", Technical Report #NPS-PM-09-146, Sep. 2009.

PhD Theses

1. R. Sandoval. "Security Software Development and Integration Testing for Advanced Concept Technology Demonstrations (ACTD) Programs Based on a Flexible Testbed". PhD Thesis – in progress. Complete draft submitted to the dissertation committee.
2. D. Anunciado. "A Systematic Method for Interfacing Real-Time Systems and Non-Real Time Systems in an Enterprise Environment", PhD Thesis – in progress. Complete draft submitted to the dissertation committee.
3. R. Halle. "Risk Assessment Methodology of Projects based on Orientation, Alignment, and Synchronization of Software Components during System Development, Delivery, and Integration (A Total Software-Based System of Systems Risk Assessment)", NPS PhD Thesis – in progress.
4. J.R. Evans. "Semi-automatic methods to aid requirements development", NPS PhD Thesis – in progress.

5. J. Rivera. “Weapon System Explosive Safety Review Board (WSESRB) Risk Mitigation Strategy for COTS Software Integration in Naval Weapons Systems,” NPS PhD Thesis – in progress.

PhD’s Awarded

1. William Roof, “Nautical Predictive Routing Protocol (NPRP) for the Dynamic Ad-Hoc Naval Network (DANN)”, Ph.D. Dissertation, NPS, June 2006
2. D. Lange. “Boot Camp for Cognitive Systems: A Model for Preparing Systems with Machine Learning for Deployment”. PhD Dissertation. NPS, March 2007.
3. W. Smuda. “Rapid Prototyping of Robotic Systems”. PhD Dissertation, NPS, June 2007
4. A. Chen. “Intellectual risk management methodology for quantitative risk assessment”, Ph.D. Dissertation, NPS, June, 2010.
5. P. Dailey. “Acquiring Operational Profile Models to Drive Automated Testing of Open Architecture Weapon and Combat System Software”, Ph.D. Dissertation, NPS, June, 2010.

(2) STUDENT/ SUPPORTED PERSONNEL MATERIAL FOR THIS REPORTING PERIOD:

(a) Graduate Students: **16**

A. Chen, R. Sandoval, D. Anunciado, R. Halle, J. Evans, P. Dailey, J. Rivera, B. Lewis, D. Lange, W. Smuda, W. Roof, R. Maule, R. Chen, C. Whittaker, C. Mushenski, M. Lisowski,

(b) Post Doctorates: **4**

P. Musial, M. Rodriguez, V. Ivanchenko, C. Ma

(c) Faculty: **5**

Luqi, V. Berzins, G. Jacoby, P. Musial, C. Martell

(d) Undergraduate Students: **0**

Note: NPS does not have any undergraduates.

I. Number who graduated during this period: **0**
II. Number who graduated during this period with a degree in science, mathematics, engineering, or technology fields: **0**
III. Number who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields: **0**
IV. Number who achieved a 3.5 GPA to 4.0 (4.0 max scale): **0**
V. Number funded by a DoD funded Center of Excellence grant for Education, Research and Engineering: **0**
VI. Number who intend to work for the Department of Defense: **0**
VII. Number who will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields: **0**

(3) TECHNOLOGY TRANSFER

The results of this project are applicable to problems of current interest to the DoD, including how to reliably share components across platforms and how to make systems more agile. Our research has established the scientific basis for dealing with these problems, and our collaborators have been using our ideas to address specific issues related to these problems.

For example, the design of the Army FCS system did achieve a high degree of component reuse across the various vehicles in their product line using concepts developed in our research, and similar efforts are continuing in current ground vehicle architectures. Ongoing work is exploring the use of principles developed in our research to support the systematic choice of data bus protocols for ground vehicles, and we are exploring applications of our ideas to quality assurance for SOSCOE in the context of multi-platform configurations. This research is complementary to our effort in the ARO research work described in this report.

Our work has also led to a recent collaboration with the Navy and joint organizations such as DTRA for achieving dependability in systems with open architectures. These collaborations focus on applying the results of the fundamental research to develop practical DoD test and evaluation procedures. The main objectives are (i) to identify current weaknesses in the development of DoD/DoN systems according to open systems principles, (ii) to develop and adapt new methods for overcoming current weaknesses in quality assurance for flexible architectures and (iii) to improve quality assurance for probabilistic systems dealing with noisy and incomplete data. Some of the expected contributions of this work encompass the development of methods for reducing and limiting the scope of testing, and methods for assuring open systems' dependability.

(4) SCIENTIFIC PROGRESS AND ACCOMPLISHMENTS

Abstract

The objective of this project is to develop an integrated, systematic, documentation centric approach to software development, known as Documentation Driven Software Development (DDD). The main research issues for DDD are creation and application of three key documenting technologies that will drive the development process and a Document Management System (DMS) that will support them. These technologies address (1) representations for active documents; (2) representations for repositories; (3) methods for analysis, transformation, and presentation of this information.

In addition, we explored new possibilities for computer-aided interfaces that help humans with routine tasks. In doing so we applied Cognitive Science and machine learning methods to design user interfaces that can learn and assist users. We also expanded our work in the area of integration of ontologies from heterogeneous sources. Specifically, we studied Knowledge System Integration Ontology (KSIO) that aligns data and information systems with current situational context for the efficient knowledge collection, integration and transfer. The role of ontology is to organize and structure knowledge (e.g. by standardized terminology) so that semantic queries and associations become more efficient. We assessed the degree to which natural language processing can be usefully applied to the analysis of requirement changes and their impact on system structure and implementation.

In the context of transformation of information between different levels, ("transition drivers"), we explored methods for automatically mapping changes to real-time requirements into architectures and real-time schedules. This study included the case where new services with real-time constraints were added to the requirements.

In order to guarantee high reliability of a complex system and reduce the risk early in a life cycle we applied risk analysis methods on the level of software architecture. This enables identifying unreliable features of architecture early in the software lifecycle, thus reducing the cost of development.

We explored reasoning support for system evolution related to replacement of reusable components, specifically with respect to the input and output transformations sometimes needed to adapt a new reusable component to an existing system architecture.

Active documentation ideas have also been applied to support improved testing for flexible systems and improved safety assessment for weapons systems.

CONTENTS

(1) LIST OF PAPERS SUBMITTED OR PUBLISHED UNDER ARO SPONSORSHIP DURING THIS REPORTING PERIOD II
(2) STUDENT/ SUPPORTED PERSONNEL MATERIAL FOR THIS REPORTING PERIOD: VII
(3) TECHNOLOGY TRANSFER VIII
(4) SCIENTIFIC PROGRESS AND ACCOMPLISHMENTS IX
CONTENTS X
FIGURES XII

CHAPTER I 1

INTRODUCTION 1
Documentation Driven Development 1
Scientific Progress and Accomplishments During the Reporting Period: 2
Unfinished Research 5
Document Outline 5

CHAPTER II 6

ADMISSION CONTROL 6

CHAPTER III 7

EFFECTIVE TESTING FOR FLEXIBLE SYSTEMS 7
Impact 9

CHAPTER IV 10

DEPENDABLE SOFTWARE ARCHITECTURE BASED ON QUANTIFIABLE COMPOSITIONAL MODEL 10
Impact 11

CHAPTER V 12

FORMAL REASONING ABOUT SOFTWARE OBJECT TRANSLATIONS 12
Impact 12

CHAPTER VI 13

INNOVATIONS IN NATURAL LANGUAGE DOCUMENT PROCESSING FOR REQUIREMENTS ENGINEERING 13
Impact 14

CHAPTER VII 16

AN ABSTRACT CHANNEL SPECIFICATION AND AN ALGORITHM 16
IMPLEMENTING IT USING JAVA SOCKETS 16
Impact 16

CHAPTER VIII **18**
INNOVATIONS FOR REQUIREMENTS ENGINEERING 18
Goal of Monterey Workshop 2007 ..... *18*
Impact *19*
CHAPTER IX **20**
REMAINING CITATIONS 20
Activity summary *21*
Impact *23*
CHAPTER X **24**
STUDENT DISSERTATIONS 24
A. Chen *24*
R. Sandoval *25*
D.H. Anunciado *25*
J. Evans *26*
R. Halle *27*
P. Dailey *31*
J. Rivera *32*
BIBLIOGRAPHY **34**

FIGURES

Figure 1 – Perspective-Based Architectural Approach for Dependable Systems of Systems 21
Figure 2 – QCM for compositional interactions, quantitative assessment 21
Figure 3 -- Risk Reporting Matrix 29

Chapter I

INTRODUCTION

Software systems in the civilian and military domains are increasing in complexity and have an ever increasing impact on human safety, financial resources, and national security. Complexity of these systems requires incremental development by design and gradual enhancement of subsystems which are composed together to yield the complete complex system of systems.

Complex software systems share any subset of the following properties: long development time, global deployment strategies, mission critical requirements, significant resource demands, timing constraints, high quality and reliability standards, ease of reconfigurability, and interoperability with other systems.

The key challenges encountered during design of complex systems include: how to generate high quality and high confidence software, how to support system evolution and accommodate changing requirements without compromising quality, how to enable support for a variety of stakeholders, and how to improve the efficiency and productivity of the development process. The feature that ensures successful development, implementation, deployment, and sustainability is precise documentation. To remain current, this documentation has to actively contribute value to system developers and provide tangible support for the processes they carry out.

Documentation Driven Development

The proposed DDD framework is a software engineering methodology that provides assistance for all software life cycle processes, most notably, requirements gathering, quality assurance, design, system evolution and re-engineering, and project management [28, 163, 165, 166]. Each of the software life cycle stages involves communication between stakeholders and the development teams. These two groups share the same objective, but their expertise is in different and sometimes mutually unfamiliar domains. DDD provides mechanisms that allow project information to be effectively communicated between all involved parties, hence providing a bridge between domain of the stakeholders and the domain of developers (which is software design and implementation). Finally, the developers and stakeholders will utilize software and hardware tools during each of the software life cycles. The challenge here is to ensure proper transformation of project requirements, which may be specified informally, into the formal and mathematical format that is required by the utilized tools. Again, the DDD framework provides mechanisms that help to do just that.

Documentation is the backbone of the DDD framework. The novel approach is that all aspects of the project information are considered as documentation, and documentation is considered in terms of its functions, such as answering questions about the system to be developed, rather than as a passive printed text. This means that documentation is not limited to system design specifications, manuals, and similar traditional documents. In our context it also includes code, simulation results, query and

transformation capabilities, etc. With this definition, the documentation in our approach can provide more effective support for the entire development process.

Our research addresses the issues of meaning extraction from informally specified system requirements, documentation verification as the requirements change, abstraction and translation of documentation for use by various entities (eg. stakeholders, developers, tools), and project analysis where the likelihood of failure is assessed based on current development and rate of requirements change. More specifically, the following are the goals of the DDD framework that our research addresses directly:

1) Provide architecture and methods for computer aided software documentation management to serve as a basis for all software life cycles processes, most notably, requirements gathering, quality assurance, design system evolution and re-engineering, and project management.
2) Support for communication among stakeholders.
3) Support for communication with software tools.

This report covers contributions to each of the three goals stated above.

Scientific Progress and Accomplishments During the Reporting Period:

Following is a list of direct contributions to the DDD project accomplished by our team to date:

1) We developed an admission control method for dynamic software reconfiguration in high confidence software architectures [17]. Our method for dynamic software reconfiguration in Dependable System of Systems performs a dynamic scheduling analysis based on an integrated dynamic scheduling algorithm for heterogeneous embedded systems. The quantifiable compositional model supports the new admission control method. This provides one step toward the systematic construction of reliable software architectures for mission critical systems that are changing, particularly with respect to timing constraints extracted from updated requirements.
2) We developed a software risk management methodology based on quantitative metrics and expert systems, to alleviate the harm or loss in a software project. The study developed a revolutionary software risk management method that integrates quantitative metrics with domain risk knowledge to support risk assessment and facilitate management decision making processes throughout a software development lifecycle. Our formal approach permits repeatability, predictability and usability in a software risk management program. This covers software acquisition, development and deployment phases. It also aims at integrating already collected metrics or automating the collection of metrics, so that risk management activity is transparent to the software project management. This kind of live and constantly updating information about project risk levels is an example of the type of active documentation our research is seeking to enable [2].
3) We developed a formal model for reasoning and verification of translations used in the compositions of software objects[149]. Our framework is abstracted to accommodate other translations. For instance, the DDD documentation repository relies on templates to extract information from the repository and transform it into the forms accepted by software tools. Our framework supports construction and verification of such templates, but it is not limited to this task only [23].

4) We developed a model and architecture for reliable wireless networking. Novelty of this architecture is the use of information about physical locations of mobile nodes, properties of the physical environment (geography and weather), and plans for operations to predict, anticipate, and prevent communication interruptions due to relative motion of obstacles and node motion that will exceed radio range restrictions for individual links. This is an example of a case where DDD concepts enable systems to adapt to changing requirements without the need for reprogramming. Applications include air and surface communications over land and sea. Civilian applications include wireless Internet service for the Washington State Ferry System [11, 21, 36].
5) We carried out an in depth assessment of natural language processing technologies with respect to requirements engineering. Our study outlines the basic issues in requirements engineering and how they relate to interactions between a natural language processing front-end and system-development processes. We suggest some improvements to natural language processing that may be possible in the context of requirements engineering and present an assessment of what should be done to improve likelihood of practical impact in this direction to better support reactions to requirement changes. The motivation is to provide automated support for the transformation from the natural language used for communication with stakeholders and the more formal notations used by software development environments. Since requirements for long-lived systems are constantly changing, this gap must be bridged repeatedly, making unaided manual processes unattractive unless they can be made incremental. The main current difficulty is the error rates characteristic of current natural language capabilities, which are low enough to be useful in some contexts but not low enough to rely on without integration with other processes for detecting and removing residual errors [150].
6) We studied ways to reduce testing effort and costs associated with technology-advancement upgrades to systems with open architectures. This situation is common in Army, Navy, and DoD contexts such as ground vehicle, submarine, aircraft carrier, and airframe systems, and accounts for a substantial fraction of the testing effort. We developed methods for determining when testing of unmodified components can be reduced or avoided, and outlined some methods for choosing test cases efficiently to focus retesting where it is needed, given information about past testing of the same component. This provides another example of active documentation: information about past testing of a system is modeled as a probability distribution (operational profile) that characterizes the frequency of system inputs to be expected from the environment, together with the number of samples from this distribution that have been used as test data and the number of test cases that have produced acceptable results for each software version. This representation is active because it can be used to determine and automatically execute the test cases needed to establish system reliability in a new deployment environment, characterized by a new probability distribution and a new reliability level goal. Changes to the environment of a system can affect its reliability, even if the behavior of the system remains unchanged. The new capabilities added by a technology upgrade can interact with previously existing capabilities, changing

the frequency of their usage as well as the range of input values and, hence, changing their effect on overall system reliability [151].

7) We analyzed the data requirements for architecture-based safety assessment of weapons systems. Currently, the System Software Safety Technical Review Panel (SSSTRP) is tasked with reviewing the software safety processes and practices of software-intensive Gun System acquisition programs from the early stages of the acquisition process. As these systems grow in complexity and as Open Architecture (OA) is implemented, the acquisition and demonstration of safe software is becoming a more challenging task— often resulting in unexpected safety risks, schedule delays, and cost overruns. This research is developing an approach to mitigate common risks in this domain from the Program Management level. This approach focuses on analyzing historical weapon system SSSTRP data to identify trends that could lead to a strategy to increase software safety as well as reduce unexpected findings at the SSSTRP. This research effort is still in the early stages, but data are being collected, and progress is being made [152, 161].
8) The central theme of DDD is improving methods for using knowledge about the past to support evolution of designs and systems. We examined how knowledge schemas are modified as a result of unexpected or surprising events. Conceptual changes and historical information have not been emphasized in traditional approaches to conceptual modeling such as the entity-relationship approach. Effective representations for such changes are needed to support robust machine learning and computer-aided organizational learning. However, these aspects have been modeled and studied in other contexts, such as software maintenance, version control, software transformations, etc. We reviewed some relevant previous results, showed how they have been used to simplify conceptual models to help people make sense out of complex changing situations, and suggested some connections to conceptual models of machine learning. Areas where further research is required to support conceptual models for adaptive systems were also identified [20, 153].
9) The theme for the 2007 Monterey Workshop, an annual event organized by our group, was devoted to requirements engineering. The purpose of this workshop was to bring together the community of experts to share and discuss new innovations in the area of requirements and natural language processing. By encouraging interactions among these talented researchers, new results were proposed which contribute to the goals of DDD [167].
10) The theme for the 2008 Monterey Workshop, an annual event organized by our group, was devoted to Foundations of Computer Software and Techniques for Development. The purpose of this workshop was to bring together the community of experts to share and discuss new innovations in the areas of specification, certification, software product lines and architectures.
11) The theme of the 2010 Monterey Workshop is Modeling, Development, and Verification of Adaptive Systems. The workshop addressed a variety of topics related to DDD, focusing on novel approaches to engineer robust software.

Unfinished Research

Despite significant advancement made thus far, there is more work that needs to be done. Here are the areas related to DDD that still require further research.

1) Requirements engineering from natural language representation still needs improvement.
2) Information flow within the DDD framework needs development of tools that manage information and enhance communication between stakeholders, developers, and software tools. More research is needed on tools that:
    a. Transform data among different representations as needed to support integration of development processes and tools.
    b. Materialize external representations of documents suitable for particular stakeholders or tools.
    c. Find appropriate subsets or projections of the documents suitable for particular purposes.
    d. Extract computed attributes of project documents, such as expected completion date of the project.
3) Current project health modeling provides a good measurement of progress and provides valuable feedback to the stakeholders; however our modeling techniques require further improvements to support better predictions.

Document Outline

The following chapters provide a description of the theoretical and practical advances resulting from work related to this project and for the specified period. Chapter II describes our work on admission control, which utilizes an active form of documentation related to real-time requirements to map this information into lower levels of design concerned with system architecture and real-time scheduling. Chapter III presents our results in the area of selective testing that leads to reduced project costs and to improved software quality. In Chapter IV we discuss the dependable software architecture that is based on quantifiable compositional model. Chapter V covers the problem of object composition where the interfaces do not necessarily match, but there is enough information to enable the desired functionality. Chapter VI covers the topic of natural language processing technology and its application to processing requirements documents and its use in requirements engineering. In Chapter VII we present our approach to translating high-level abstraction of a lossy-asynchronous communication channel to a low-level Java implementation. Chapter VIII covers our synergy activities where we propagate, share, and discuss ideas to further improve progress related to this project -- specifically the Monterey Workshop series. In Chapter IX we present a summary of all technical reports that are not covered specifically in this report, but have been covered at length in other technical reports. These reports cover development related to the Documentation Driven Software Development framework. Finally, we conclude with Chapter X, where we present relevant work performed by the supervised doctoral students.

Chapter II

ADMISSION CONTROL

Details of the following research results appear in the following documents:

Y. Qiao, H. Wang, Luqi, V. Berzins, "An Admission Control Method for Dynamic Software Reconfiguration in Complex Embedded Systems", International Journal of Computers and Their Applications, Vol. 13, No. 1, March, 2006, pp. 28-38.

We developed an admission control method for dynamic software reconfiguration in high confidence software architectures. Much of the previous work on software reconfiguration in complex embedded systems concentrates on providing methods or frameworks to support the adjustment of system configurations and depends on the assumption that the requested reconfiguration is safe. However, this assumption is not always true in practice. Dynamic software reconfiguration may induce the failure of whole Systems of Embedded Systems (SoES) since configuration changes may have negative impacts on satisfaction of some key properties such as timing constraints. Previous methods ignore this potential risk and the arbitrary acceptance of reconfiguration may damage high confidence in the whole system.

Our contribution was to develop an admission control method for dynamic software reconfiguration in SoES that performs a dynamic scheduling analysis based on an integrated dynamic scheduling algorithm for heterogeneous embedded systems. Only new component systems whose addition will not result in unschedulability of SoES are allowed to enter the system. If the scheduling analysis shows the addition of new component systems violates schedulability of the whole system, those new component systems are put into a training process, which finds more suitable parameters for the new component systems according to the suggestions given by the scheduling analysis. This admission control method improves the confidence of SoES by preventing a class of failures that could be caused by dynamic software reconfiguration.

This is an example of the proposed technique for transition drivers that automatically convert system knowledge from one level to another, in this case from requirements to architecture. We also worked out an example of decision fusion rules for design constraints. This is part of the decision support to be provided by the DMS, and further validates our hypothesis that the attributed object graph model provides a good foundation for the automated influence support needed for DDD.

Chapter III

EFFECTIVE TESTING FOR FLEXIBLE SYSTEMS

Details of the following research results appear in the following documents:

V. Berzins, "Which Unchanged Components to Retest after a Technology Upgrade", Proceedings of the Fourth Annual Research Symposium – Acquisition Research: Creating Synergy for Informed Change, Monterey, CA, May 14-15, 2008, pp.142-153.

V. Berzins, P. Dailey, "How to Check If It Is Safe Not to Retest a Component", In Proceedings of the Sixth Annual Research Symposium – Acquisition Research: Defense Acquisition in Transition, Monterey, CA, May 12-14, 2009, pp. 189-200.

Luqi, P. Dailey, "Profile-Based Automated Testing Process for Open Architecture Track-Processing Software", Technical Report #NPS-CS-10-005, Mar. 2010.

P. Dailey. "Acquiring Operational Profile Models to Drive Automated Testing of Open Architecture Weapon and Combat System Software", Ph.D. Dissertation, NPS, June, 2010.

One of the goals of the DDD framework is reducing the cost and duration of software testing following regular software maintenance, any technology upgrades, or changes in system requirements. The DoD's open architecture framework is intended to promote reuse and reduce costs. This paper focuses on exploiting and extending open architecture principles to reduce testing effort and costs in cases in which the requirements and code for a subsystem have not been changed, but the code is running on new hardware and/or new operating systems due to a technology-advancement upgrade. This situation is common in DoD contexts such as FCS, submarine, aircraft carrier, and airframe systems, and accounts for a substantial fraction of the testing effort. Unmodified software components need to be retested after a technology upgrade in some, but not necessarily in all cases. We studied conditions under which testing of unmodified components can be avoided after a technology upgrade, outlined an approach for identifying situations in which retesting can be safely reduced, and indicated how to focus retesting in cases in which it cannot be avoided.

The DoD is implementing the open architecture framework for developing joint interoperable systems that adapt and exploit open system design principles and architectures. Research being performed at the Naval Postgraduate School is pursuing a complementary effort to identify weaknesses and gaps in the current state of knowledge with respect to the development and testing of DoD systems according to such open systems principles, and to develop or adapt new methods for overcoming those weaknesses. The purpose of this effort is to provide sound engineering approaches to better realize the potential benefits of modular architectures and to provide concrete

means that support economical acquisition and effective sustainment of such systems. This research focuses primarily on improving test and evaluation of systems with open architectures, since this aspect can greatly benefit from improvements. Specific goals of this research are to enable the following: (i) reduction of unnecessary testing on every system change, (ii) identification of what specific testing and checking procedures need to be repeated after changes, (iii) limiting the scope of retesting when the latter is necessary, and (iv) enabling a single analysis to provide assurance that all possible configurations that can be generated in a model-driven architecture will satisfy given dependability requirements. A roadmap and technical approach for reaching the fourth goal are outlined in Berzins, Rodriquez and Wessman (2007). The roadmap provides a long-term plan for eventually eliminating the need for regression testing after each reconfiguration and eventually enabling a "plug-and-fight" capability. This plan depends on the design and certification of a common architecture for a family of systems that span a parameterized range of expected requirements, based on detailed standards for the components and connections. In this approach, the architecture is certified to meet its requirements, components are tested against standards and requirement parameters, and reconfiguration is achieved by swapping plug-compatible components with different requirement parameters.

Our current work focuses on the shorter-term problem of safely reducing testing for software components whose code has not been changed, without waiting for the results of long-term research and without relying on architecture-level certification [147, 148, 151, 157, 158, 159, 160, 162].

The motivating context for the work reported here was to increase the effectiveness of quality assurance for technology upgrades. The first step was to investigate conditions under which it is safe to reduce testing for software components whose code has not been changed so that a larger fraction of the available time and effort could be focused on testing the new functionality introduced by the upgrade. This focus was adopted after the author interviewed representatives from four of the organizations actually involved in developing such technology upgrades. These interviews indicated (with unanimous support) that those organizations' highest current priorities are reducing testing for unmodified software components after a technology upgrade and adapting automated testing methods into production use. The initial research, therefore, explored practical methods for checking conditions under which it is safe to reduce or eliminate retesting for unchanged components, and sought solutions that leverage automated testing in the contexts in which it is easiest and most effective to do so.

Technology upgrades typically often involve migration to the best hardware and operating system version available at the time, where "best" implies a balanced tradeoff between high performance and reliable operation. Typically, only a small fraction of the application code has been changed. However, current certification practices require all of the code to be retested prior to deployment, whether it has been modified or not. Retesting of an unchanged module can be avoided only if we can establish that it has not been adversely impacted by the change. The rest of this paper explores ways to determine that, and the conditions under which such a determination is possible. The proposed approach is to use program slicing, augmented with statistically significant maintenance testing to deal with situations where code or hardware has changed but the required subsystem behavior remains the same.

Further research is recommended to substantiate the practical applicability of the ideas outlined above. Experimental evaluation of the slicing method for identifying modules that do not have to be retested should be performed, together with the focused automated testing methods needed to fully realize the potential savings of the approach. Measurement and analysis of the operational profiles of reusable components can be used to support analysis of changes in the operating environment that may require focused retesting of components whose behavior has not changed. Operational profiles are probability distributions that serve as mathematical representations of the operating environment and are needed to support statistically significant testing that can reduce the testing effort, as described above. These distributions can be measured by instrumenting components and collecting statistics as they run, either in exercises or during actual missions, and can be used to drive statistically based automated testing that can quantitatively assess the reliability of systems to confidence levels derived from the degree of risk tolerance of military commanders.

These distributions can also be used to drive automated testing, and to determine the amount and the type of testing needed to reuse a subsystem in a different deployment environment (see Berzins and Dailey, 2009). Studies on practical methods for estimating these distributions from measurements and historical data are ongoing (Dailey PhD).

Impact

The DoD and Army in particular are moving towards flexible systems that can be reconfigured by replacing subsystems with plug-compatible components that have different characteristics. Current test and evaluation procedures, particularly with respect to software, require each new configuration to be retested before it is released for use in the field. If we wish to enable reconfiguration in the field ("plug-and-fight"), this approach requires pre-testing all possible configurations, which has cost exponential in the number of independently replaceable subsystems. Our research seeks to reduce this non-affordable cost to linear (or at worst a low-order polynomial) by enabling reliability to be achieved via standards-based testing augmented with symbolic certification of standards with respect to architectures and family-wide requirements that are to be met by all possible system configurations. Although it is not easy to convince contractors to automate their testing if they are not familiar with this approach, the economic incentives to do so are getting more compelling. This practical problem is particularly evident in the current situation—in which domain experts are often doing the project management and coding with little knowledge of or experience with recent advances in the techniques and tools used in software engineering. The increasing popularity of agile methods, which depend heavily on semi-automated testing, should help change this perception. Pilot projects demonstrating the effectiveness of the suggested approach are recommended to provide concrete data about costs and benefits, thereby alleviating concerns about project risks due to technology innovations.

Chapter IV

DEPENDABLE SOFTWARE ARCHITECTURE BASED ON QUANTIFIABLE COMPOSITIONAL MODEL

Details of this research appear in:

Luqi, V. Berzins. "Dependable Software Architecture Based On Quantifiable Compositional Model", NPS TR NPS-CS-08-003, 2008.

Luqi, L. Zhang, V. Berzins, Y. Qiao, "Documentation Driven Development for Complex Real-Time Systems", IEEE Transactions on Software Engineering 30, 12, p. 936-952.

Y. Qiao, V. Berzins, Luqi, "FCD: A Framework for Compositional Development in Open Embedded Systems", International Conference on Information Technology, Las Vegas, Nevada, April 2005.

The following research fits in the DDD framework and is geared toward translating system requirements into a quantifiable architecture that ensures dependable implementations. Specifically, we propose a set of techniques to create new architecting methods that enable quantifiable architectural synthesis for Dependable System of Systems (DSoS). With the aim of improving software flexibility and ensuring dependability of the resultant systems, quantifiable architecture is the abstraction stratum that bridges the great gap between software requirements and system implementation. An effective and practical Quantifiable Compositional Model (QCM) is studied [27, 164]. The QCM defines compositional patterns with which we associate quantifiable constraints. This forms a formal foundation for establishing and binding precise metrics to computational activities and compositional interconnections so that quantitative assessment can be automatically done at the architectural level. Based on this foundation, a quantitative assessment can be performed at the architectural level. The proposed research enables the development of quantifiable metrics related to the software architecture. This research significantly contributes to the improvement of the dependability of systems of systems.

We seek to support flexible configuration of organizational structure, quantifiable assessment, rapid development of DSOS, and involvement of various stakeholders throughout the software lifecycle processes. Based on our previous research, the proposed new methods and formulated models enable incorporation of three perspective-based models (representing development stages) and two automatic transitions of explicit architecting and componential derivation, together with quantifiable assessment (design inspection) into an automated and user-friendly developmental environment.

The proposed research enables effective, practical architecting methods from a requirement-acquiring level to an implementation-fulfilling level to support the development and evolution of DSOS. The basic idea of this approach is to strengthen

system composition in combination with explicit architecting and quantifiable constraints attached to the architectural artifacts. This kind of composition improves dependability of the intended systems by strengthening the link between software architecture descriptions and the quality assurance processes they should support, and increases software productivity. This research is rooted in multiple perspectives reflecting various stakeholders concerns, incorporating requirements prototyping, quantifiable architecting and assessing, and implementing derivation techniques into an architectural synthesis approach. Specifically, the proposed models and methods promise automatic transitions with inherent semantic consistencies at various points of the software development process and strengthen the connection between requirements and quality assurance processes.

Impact

The result of this research improves system dependability and promotes affordable flexibility for system evolution and maintenance in the future. Successful systems are in a constant state of change. The proposed approach improves on the current DSOS weakness of assuring dependability during system evolution. It reduces the level of re-certification efforts required after each requirement change that remains within the envelope of the invariants that should be defined by the quantifiable constraints bound to the architecture. These invariants are intended to apply to all instances of the family of systems that is spanned by a given dependable architecture. This research has a potential of broader applicability, for example to the area of Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR architecture) to support acquisition of systems that meet the needs of the military. Today, military organizations must respond to a variety of situations by quickly assembling and organizing coalitions from various components. The use of DSOS architectures can address issues involving requirements uncertainty and rapid changes in technology, as well as a widening of spectrum of supportable missions and operations.

Chapter V

FORMAL REASONING ABOUT SOFTWARE OBJECT TRANSLATIONS

Details of this result appear in:

V. Berzins, Luqi, P. Musial, "Formal Reasoning about Software Object Translations", Proceedings of the 2008 Monterey Workshop, Budapest, 23-27 Sep. 2008, pp. 65-78.

This work provides a formalization of the translation between outputs of one software component to the inputs of another along with a verification mechanism based on constraint logic programming (CLP). This problem is important in the software reuse domain, and has applicability in other areas of software engineering such as transformation of information from one phase of the development process to another. The key challenges are to ensure that a viable translation exists and that it enables functionality of the software component designated as an input entity as well as supports the range of values produced by the output software component. Our model allows formalization of the translation problem in the form of constraints and relations between the outputs and the inputs of two subsystems to be composed to meet a larger purpose. CLP tools are used to verify existence and validate a proposed translation. Since CLP tools can be computationally expensive we identify characteristics of translation problems where our technique is rendered practical.

Complex systems of systems are a composition of individual systems that may or may not have been designed independently. There are very strong economic incentives to reuse (parts of) legacy systems where possible. This means that simple interface matching may not be sufficient to make desired composition possible. A sophisticated translation of outputs of the source system to the inputs of the target system may be necessary. Furthermore, the translation may involve application of logical and relational operators to the outputs. In order to ensure that the composition of two systems and the translator results in a functional system of systems, the translation must be verified. Specifically, it is important to verify that the translation enables functionality of the target software component while supporting the outputs of output software component.

Impact

DDD provides methods for engineering systems that are a composition of individual subsystems, i.e. system of systems. It is of vital importance to ensure that systems are compatible and that the necessary translations are verified against the desired functionality. Modeling system composition via translation and verifying integrity of that composition through use of a CLP model enables us to reduce potential cost associated with experimentation and on-line testing of the system. Also, reasoning at the formal level about correctness of the system design prior to the implementation increases our confidence in the final product.

Chapter VI

INNOVATIONS IN NATURAL LANGUAGE DOCUMENT PROCESSING FOR REQUIREMENTS ENGINEERING

This research result appears in:

V. Berzins, C. Martell, Luqi, P. Adams. "Innovations in Natural Language Document Processing for Requirements Engineering", Springer LNCS, 2008.

Development of any system starts with the specification of system requirements. The scope and difficulty of this task are uniquely challenging for software projects because software is a much more abstract and multi-purpose entity than other projects, such as suspension bridges. This work evaluates the potential contributions of natural language processing to requirements engineering. We present a selective history of the relationship between requirements engineering (RE) and natural-language processing (NLP), and provide a brief summary of relevant recent trends in NLP. We outline the basic issues in RE and how they relate to interactions between a NLP front end and system-development processes. We suggest some improvements to NLP that may be possible in the context of RE and conclude with an assessment of what should be done to improve the likelihood of practical impact in this direction.

A major challenge in requirements engineering is dealing with changes, especially in the context of systems of systems with correspondingly complex stakeholder communities and critical systems with stringent dependability requirements. Documentation driven development (DDD) is a recently developed approach for addressing these issues that seeks to simultaneously improve agility and dependability via computer assistance centered on a variety of documents [28-30]. The approach is based on a new view of documents as computationally active knowledge bases that support computer aid for many software engineering tasks from requirements engineering to system evolution, which is quite different from the traditional view of documents as passive pieces of paper. Value added comes from automatically materializing views of the documents suitable for supporting different stakeholders and different automated processes, as well as transformations that connect different levels of abstraction and representation. The sheer size and complexity of enterprise-wide systems makes such automation support a necessary condition for reliability rather than a convenience. The bodies of documents that encompass the requirements of such systems are encyclopedic in size and scope, and consequently impossible for a single person to understand in detail. Assuring absence of contradictions or other non-local quality properties on such scales is practically impossible for unaided humans.

At the level of requirements engineering, the central problems are related to bridging the gap between stakeholders, who communicate in natural language, and software tools, which depend on a variety of formal representations. A prominent problem is resolving ambiguity, which is typical of natural language and to a somewhat lesser degree the popular informal design notations such as UML. Other significant

requirements engineering problems include finding implied but unstated requirements, detecting conflicts between needs of different stakeholders, and resolving such conflicts. Communication gets increasingly difficult as systems scale up. Stakeholders are typically comprised of diverse groups, each of which has its own specialized domain knowledge, jargon, and unique tacit understanding of the problem. Bridging the gaps becomes key to success as complexity increases because each group typically has only a partial understanding of the issues, constraints, possible solutions, and cost implications [28, 29].

Progress on increasing flexibility without damaging reliability depends on computer aid, in an end-to-end process that includes requirements engineering. This leads to a need for natural language processing that can help bridge the gap between natural stakeholder communication and unambiguous requirements models such as those embodied in the DDD view of documents. Ever present changes in requirements imply that this gap must be bridged repeatedly.

In the 1970s the automatic programming group at MIT headed by Prof. Bill Martin sought to create an end-to-end system that went from user requirement documents to running code for business information systems. The project made progress at the top and bottom levels of this process, but the two ends were never integrated together.

The capabilities of natural language processing (NLP) software and our understanding of requirements engineering (RE) have improved substantially over the past 30 years. This paper re-examines how the current state of NLP can contribute to requirements engineering, how close is it to making a practical impact in this context, and what needs to be improved to enable widespread adoption. We examine the connection between a hypothetical NLP front end and requirements engineering processes that would follow, and identify some of the differences between generic NLP and domain-specific NLP embedded in a requirements engineering process.

Challenges of NLP for Requirements Engineering

Requirements engineering is a critical part of the system development process because requirement errors cost roughly 100 times less to correct during requirement engineering than after system delivery [86]. This imposes extreme constraints on the accuracy of NLP that we might use to derive system requirements. However, NLP accuracies are currently in 90%-92% range, at best. Therefore NLP must be augmented with other methods for removing residual errors, and accuracy must be greatly improved if it is to be seriously used for RE.

Why All is Not Yet Lost

NLP in the context of RE should be more tractable than generic NLP, because it has the usual advantages of a domain-specific approach: the scope is narrower, more is known about the context, and specialized methods may apply. In particular, much more is known about the intentions of the speaker and the context, such as typical goals and surrounding tasks.

Impact

It appears that NLP is getting close to the point where it can contribute to requirements engineering, but it cannot do so in a vacuum. The results must be checked and reviewed,

and existing methods must be improved by using more aspects of the context of the process to improve accuracy.

Even approximate NLP could facilitate text analysis and reduce workload by prioritizing documents, using context for effective search, making summaries, and classifying texts or their fragments even if accuracy of the process is insufficient to support requirements engineering based solely on the raw output of the NLP. The difference from fully automated processing is that NLP methods will typically give users several options and it will be their responsibility to select the right one. Thus currently the most safe and effective use of NLP is to integrate its methods with human processing as it is conceptualized in Human System Integration (HSI) framework. The value added would be that, the automated processing could identify some weaknesses that unaided humans might miss [31,32].

The issues that will determine whether or not NLP enters widespread use in requirements engineering are economic: it must cost less and produce more accurate results than corresponding manual processes that rely on human experts to interpret and model the raw statements from the stakeholders. This is a challenging goal that reaches beyond the traditional bounds of NLP to include social, organizational and psychological issues. Although current accuracy of NLP does not appear to support completely automated processing, the repetitive nature of the bridging from NL to more formal requirement models holds out the hope that an appreciable fraction of the currently necessary human guidance might be captured on the first iteration and not need to be repeated completely on subsequent iterations.

Chapter VII

AN ABSTRACT CHANNEL SPECIFICATION AND AN ALGORITHM IMPLEMENTING IT USING JAVA SOCKETS

Details of this research result appear in:

C. Georgiou, P.M. Musial, A.A. Shvartsman, and E. Sonderegger. An Abstract Channel Specification and an Algorithm Implementing It Using Java Sockets. To appear in Proc. of 7'th IEEE International Symposium on Network Computing and Applications (IEEE NCA), 2008.

Developing systems from high level specifications presents numerous challenges during the implementation process as some of the subtle low-level physical issues may be overlooked and swept under assumptions that may seem to be obvious but are never the less invalid. In this work we examine the assumption that abstract lossy-assynchronous channels can be implemented and correctly composed within the rest of the software system. More specifically, abstract models and specifications can be used in the design of distributed applications to formally reason about their safety properties. However, the benefits of using formal methods are often negated by an imperfect ad hoc process of mapping the functionality of an abstract specification to detailed algorithms designed to be executed on target distributed platforms. The challenge of formally specifying communication channels and correctly implementing them as algorithms that use realistic distributed system services is the focus of this paper. This work provides an original formal specification of an abstract asynchronous communication channel with support for dynamic creation and tear down of communication links between participating network nodes, and its implementation as an algorithm using Java sockets. The specification and the algorithm are expressed using the Input/Output Automata formalism, and it is proved that the algorithm correctly implements the specification, viz. that any externally observable behavior (trace) of the algorithm has a corresponding behavior of the specification. The approach presented here can be used to implement algorithms for dynamic systems, where communicating nodes may join, leave, and experience arbitrary delays. The result is also of direct benefit to automated code generation, such as that implemented within the Input/Output Automata Toolkit at MIT.

Impact

The work presented in this result is the first formal presentation of an abstract asynchronous communication channel with graceful comings and goings. Many algorithm implementations rely on such abstract channels without providing a proof of correctness for the composition of the source algorithm and the communication channel.

Therefore, our solution can be used to claim that such implementations are, in fact, correct.

The importance of proving the correctness of the composite automaton cannot be overemphasized. It was only by going through the proof process that several subtle errors in the design of the component automata were discovered and corrected.

We intend to use this work in formally reasoning about the correctness of dynamic distributed data-sharing applications. In addition, our proposed solution can be used in automated code generation for dynamic networked applications. Future extensions to the model will include support for bidirectional communication over socket pairs, multiple connections between pairs of nodes, and timing considerations.

This work contributes to the DDD goal of automation support for transitions between ideas at an abstract level and realizations of similar ideas at a more concrete level. The DDD vision seeks a complete path from the informally stated needs of the stakeholders that are the starting point of all real projects and the formal and precise quality assurance processes that are possible at the lowest levels. All aspects of this path need to achieve high confidence, because a chain is only as strong as its weakest link, and it can function as a whole chain only if all the links are properly connected.

Chapter VIII

INNOVATIONS FOR REQUIREMENTS ENGINEERING

Please note that Monterey Workshops held in 2007, 2008 and 2010 are funded in part by the same sponsor and address topics related to DDD [149, 156, 157].

The Monterey Workshop series seeks to improve software practice via the application of engineering theory and to encourage development of engineering theory that is well suited for this purpose. The 2007 workshop focused on requirements, particularly the process of transforming vague and uncoordinated needs of individual stakeholders into consistent and well-defined requirements that are suitable for supporting automated and computer-aided methods for engineering subtasks in the subsequent development process, which is consistent with the theme of DDD framework. Innovations are effective technology transfers of sound inventions. The workshop case study was targeted at identification and assessment of sound inventions of technology that can be used to support innovations in requirement engineering. For example, we wanted to gain a better understanding about how to deal with natural language as the vehicle from which we derive system/software requirements, how to use intelligent agents as entities to facilitate semi-automatic requirements-documentation analysis, and how to build automatic systems to aid in requirements/specifications elicitation. The overall aim was to exchange ideas for continued research in the intersection of these two areas and to reduce the gap between theory and practice.

Goal of Monterey Workshop 2007

Errors or failures of software-based systems are due to a variety of causes, e.g. misunderstanding of the real world, erroneous conceptualization, or problems in representing concepts via the specification or modeling notations. Precise specification is a key success factor as are communication and the deliberation about whether the specification is right and whether it has been properly implemented. Not all stakeholders are familiar with the formal models and notations employed. Some important requirements might be difficult to quantify and/or express using formal languages, such as the desire that a system should be user-friendly or easily maintainable. Rather than ignoring requirements issues that do not fit current requirements engineering tools, extended technologies for requirements analysis that can address remaining gaps should therefore be considered.

The majority of requirements are given by stakeholders in natural language, either written or orally expressed. Other requirements might also be visually expressed in terms of figures, diagrams, images or even gestures. Artificial-intelligence approaches might be used to develop prototypes which can then be re-engineered using more conventional requirements technologies and safety assurance techniques. For example, we might employ large amounts of semantic and statistical data, knowledge bases and reasoning

systems to infer as much contextual information as possible from the (vague) textual or visual requirements. Then, some extra questions could be raised to system/software stakeholders to point out some fuzzy (or missing) requirements to be refined or some conflicting requirements to be reconciled.

Accurate automatic analysis of natural language expressions has not yet been fully achieved and interdisciplinary methodologies and tools are needed to successfully go from natural language to accurate formal specifications. Conformance of a system implementation to its requirements necessitates dynamic and efficient communication and iteration among system stakeholders. It is in supporting this process, and not in supplanting it, that innovative approaches to requirements analysis need to find their proper role. We want to gain a better understanding about how to deal with natural language as the vehicle from which we derive system/software requirements, how to use intelligent agents as entities to facilitate semiautomatic requirements-documentation analysis, and how to build automatic systems to aid in requirements/specifications elicitation. The overall aim is to exchange ideas for continued research in the intersection of these two areas and to reduce the gap between theory and practice.

Impact

The overarching goals of the annual Monterey Workshops (since 1992) are to create a shared community-wide articulation of the system/software engineering enablement challenge, reach consensus on the set of intellectual problems to be solved, and create a common vision of how the solutions to these problems will fit together in a comprehensive engineering environment.

The Monterey Workshop has been able to bring the brightest minds in Software Engineering together with the purpose of increasing the practical impact of formal methods for software development so that these potential benefits can be realized in actual practice. In the workshop, attendees and organizers work to clarify what are good formal methods, what are their feasible capabilities, and what are their limits. Overall, the workshop strives to reduce the gap between theory and practice. This has been a slow and difficult process because theoreticians and practitioners do not normally talk to each other, and did not at the beginning of the workshops. This gap has been gradually reduced. In particular, researchers have focused on problems that are relevant to the practitioners, and have helped demonstrate how recent theory can be applied to solve current problems in software development practice.

These workshops have helped focus the attention of the community on many productive directions – DDD being one of these directions. For example, since the 1995 workshop identified specification-based architectures as a key means to achieve system flexibility and reuse, there has been a large increase in activity in these areas. A great deal of research has produced architecture description languages and associated analysis methods, there have been commercial advances on ”plug and play” hardware and software, adoption of service-based architectures in electronic commerce, and a move toward open architectures in government and defense systems. Currently the practical impact of software architecture is no longer in doubt.

Chapter IX

REMAINING CITATIONS

The remaining citations have been discussed at length in the following documents:

Luqi, V. Berzins. "Dependable Software Architecture Based On Quantifiable Compositional Model", NPS TR NPS-CS-08-003, 2008.

In order to improve readability of this report and to make an efficient use of time, we will cover all of the remaining citations that are listed in Chapter I under a single section, since these were discussed at length in the above noted report.

These reports cover the proposed set of techniques to create new architecting methods that enable quantifiable architectural synthesis for Dependable System of Systems (DSoS). With the aim of improving software flexibility and ensuring dependability of the resultant systems, quantifiable architecture is the abstraction stratum that bridges the great gap between software requirements and system implementation. An effective and practical Quantifiable Compositional Model (QCM) was studied. The QCM defines compositional patterns with which we associate quantifiable constraints. This forms a formal foundation for establishing and binding precise metrics to computational activities and compositional interconnections so that quantitative assessment can be automatically done at the architectural level. Together with associated formal foundations, the proposed research enables the development of quantifiable metrics related to the software architecture. An effective and practical QCM was studied (Figure 1).

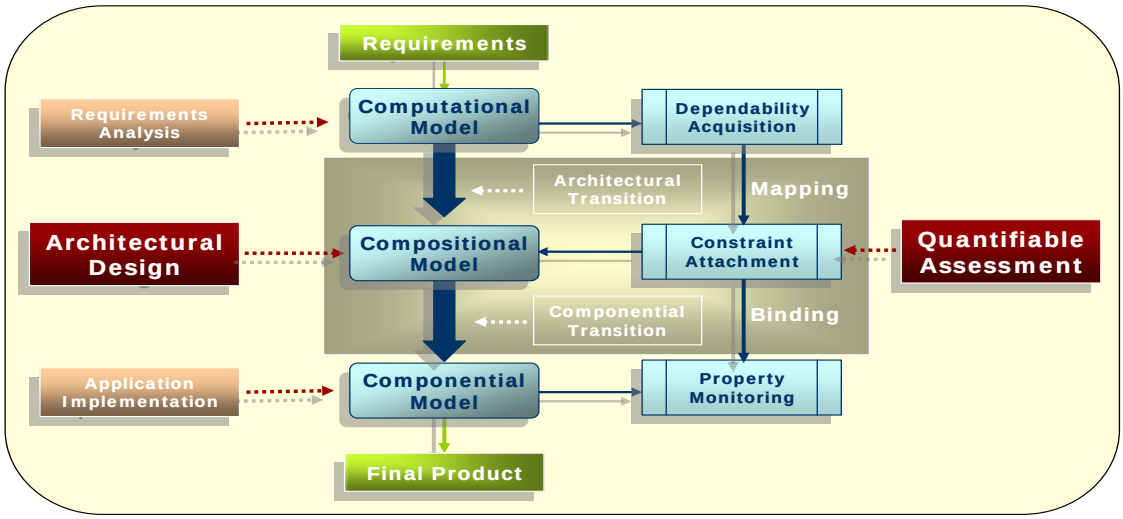

Figure 1 – Perspective-Based Architectural Approach for Dependable Systems of Systems

This research significantly contributes to the improvement of the dependability of systems of systems.

We seek to support flexible configuration of organizational structure, quantifiable assessment, rapid development of DSOS, and the involvement of various stakeholders throughout the software lifecycle processes. Based on our previous research the proposed new methods and formulated models enable incorporation of three perspective-based models (representing development stages) and two automatic transitions of explicit architecting and componential derivation, together with quantifiable assessment (design inspection) into an automated and user-friendly developmental environment.

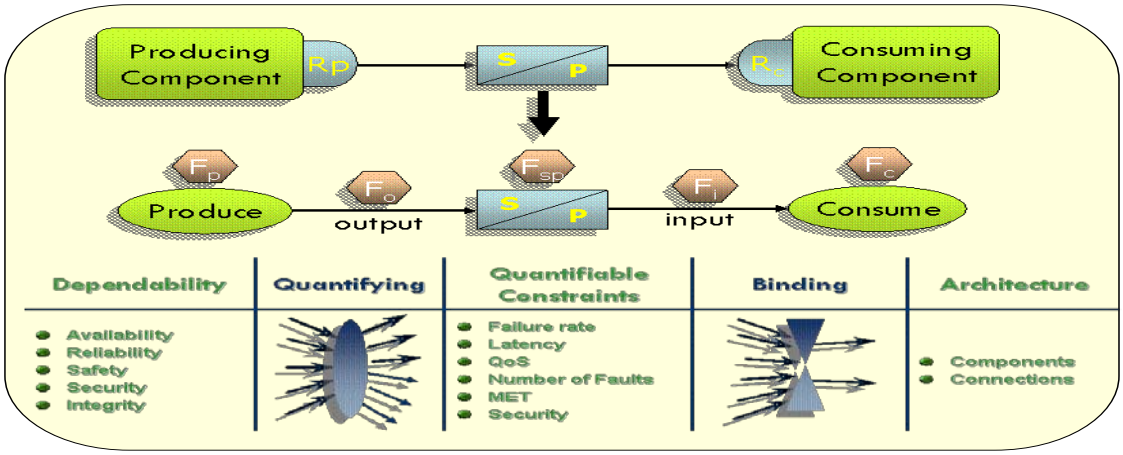

Figure 2 – QCM for compositional interactions, quantitative assessment

The proposed research enables effective, practical architecting methods from the requirement-acquiring level to the implementation-fulfilling level to support the development and evolution of DSOS. The basic idea of this approach is to strengthen system composition in combination with explicit architecting and quantifiable constraints attached to the architectural artifacts (Figure 2). This composition improves dependability of the intended systems and increases software productivity. This research is rooted in multiple perspectives reflecting various stakeholders' concerns, incorporating requirements prototyping, quantifiable architecting and assessing, and implementation derivation techniques into an architectural synthesis approach. Specifically, the proposed models and methods promise automatic transitions with inherent semantic consistencies at various points of the software development process.

The result of this research should improve system dependability and promote affordable flexibility for system evolution and maintenance in the future. Successful systems are in a constant state of change. The proposed approach improves on the current DSOS weakness of assuring dependability during system evolution. It reduces the level of re-certification efforts required after each requirement change that remains within the envelope of these invariants.

Activity summary

The QCM model described in our proposal was extended and applied to support the new admission control method. This provides one step towards the systematic construction of reliable software architectures for mission critical systems, particularly with respect to timing constraints. Such methods are useful for both DoD and civilian systems that must provide reliable service while adapting to requirements change.

We also developed a model and architecture for reliable wireless networking. The novel aspects of this architecture are the use of information about physical locations of nodes, properties of the physical environment (geography and weather), and plans for operations to predict, anticipate, and prevent communication interruptions due to obstacles and exceeding radio range restrictions for individual links. The architecture orchestrates predictive re-routing to prevent loss of communication sessions and suggests navigational corrections that enable more effective communications. The mathematics includes a three dimensional motion and obstacle model, and can accommodate nodes entering and leaving the system. Applications include air and surface communications over land and sea. Civilian applications include wireless internet service for the Washington State Ferry System.

Another example of our work on architectural features to support reliability is a mechanism for intrusion detection based on an analysis of power consumption. This approach also depends on interfaces to subsystems that are physical rather than computational.

We developed a model of Intranet portals and a set of metrics for measuring their effectiveness [16]. Our work is relevant to large scale architectures. DoD is transforming its systems into Net-Centric orientation systems that emphasize data over services and flexible data sharing (publish/pull information) over periodic fixed broadcasting patterns (push information). Quantifiable architectures need ways to measure the effectiveness of information provided in this manner which has the same structure as an intranet portal. The same techniques are applicable to commercial intranet portals.

We developed a software risk management methodology based on quantitative metrics and expert systems to alleviate the harm or loss in a software project. The study developed a revolutionary software risk management method that integrates quantitative metrics with domain risk knowledge to derive risk assessment and facilitate management decision making processes throughout software development lifecycle. Formal models are developed which are driven by quantitative metrics. The formal models permit repeatability, predictability and usability in a software risk management program. This covers software acquisition, development and deployment phases. It also aims at integrating already collected metrics or automating the collection of metrics so that risk management activity is transparent to the software project management.

A method has been developed to address the ultimate goals of interoperability and automatic agent generation. The method implements a class of protocol agents based on decisions made via interactions with network systems. The method includes a decision support protocol language, an infrastructure for the automatic generation and execution of protocol agents, and a template-based procedure for protocol agent generation. Our work led to interoperable compositional models architectural roles, styles and protocols,

interoperation techniques via wrappers, and formalization of patterns (agent types, agent interfaces, semantics, and hierarchical composition).

Impact

This research has a potential of broader applicability, for example to the area of Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR architecture) to support acquisition of systems that meet the needs of the military. Today, military organizations must respond to a variety of situations by quickly assembling and organizing coalitions from various components. The use of DSOS architectures can address issues involving requirements uncertainty and rapid changes in technology, as well as a widening of spectrum of mission and operations.

Chapter X

STUDENT DISSERTATIONS

A number of PhD students are actively contributing to this research project under the supervision of Prof. Luqi. Their efforts are covered under the general DDD framework as part of their dissertation work. In this chapter, we discuss the main ideas behind each of the dissertations and provide an update on the current state of the progress.

A. Chen

Intellectual Risk Management Methodology for Quantitative Risk Assessment

The goal of this research is to develop a software risk management method that integrates quantitative metrics with domain risk knowledge to derive risk assessment and facilitate management decision making processes throughout software development life cycle. The decision aid from the method helps to produce a high level software risk management plan that encompasses risk assessment and risk control. Risk assessment includes risk identification and risk prioritization. Risk control includes risk resolution and risk monitoring. Risk resolution can possibly include termination of the project due to irresolvable risks. The method also supports simulation using strategic plan for risk control to predict and estimate risk outcomes as a predictor or trend analysis.

A complex real-time system is generally composed of individual real-time systems that were developed by different organizations with different tools and run on different platforms. The development of complex real-time systems is more challenging than to the development of individual real-time systems. In general, complex real-time systems are usually deployed for long periods of time, are used globally, and have mission critical requirements. Complex systems must rapidly accommodate frequent changes in requirements, mission, environment, and technology. Risk assessment is essential to the successful development of a software system, especially for evolutionary development of complex systems.

The flexibility to adapt complex systems to changing requirements (agility) in the DDD approach comes from an efficient documentation management system (DMS) and a process measurement system (PMS) that can bridge gaps between different disciplines and reduce the requirements for participants' specific knowledge in system evolution. Risk assessment is the main purpose of PMS, which monitors the frequent changes in system requirements and assesses the effort and success probability of the project with a measurement model based on a set of quantitative metrics. The metrics can be automatically collected in the requirements phase and stored and organized in DMS.

Based on the PMS framework, this dissertation presents an evolutionary software risk management methodology that integrates quantitative metrics with domain risk knowledge to derive risk assessments and support risk control. This method aims to

facilitate management decision making processes throughout a software development life cycle. Risk assessment includes activities of risk identification, risk estimation and risk prioritization. Risk control includes risk resolution and risk monitoring. Risk resolution can possibly include termination of the project due to irresolvable risks. The decision aid from this method is intended to produce a high level software risk management plan that encompasses risk assessment and risk control. The method also supports simulation using strategic plans for risk control to predict and estimate risk outcomes as a predictor or for trend analysis.

We identify four major indicators for accurately measuring investment risk during the software cycle that are essential to the evolutionary development of complex real-time systems: requirements volatility, organizational efficiency, product complexity, and technological maturity. We provide a set of quantitative metrics to measure the indicators. The metrics can be collected and derived early in the software lifecycle. Rapid prototyping techniques, information theory and other innovative theories and technologies are used in development of these metrics to accommodate characteristics of complex real-time software systems. Based on these metrics, a formal model can be developed to help the project manager control risk in a complex software project. By integrating the risk assessment method into the DDD framework, complex software systems that can quickly respond to changes in requirements and guarantee real-time performance with high confident constraints can be developed.

R. Sandoval

Security Software Development and Integration Testing for Advanced Concept Technology Demonstrations (ACTD) Programs Based on a Flexible Testbed

This dissertation explores the advantages of having a dedicated computer network testbed for vulnerability assessments of ACTD's that is flexible enough for varying system architectures and can be quickly reconfigured for repeated testing and analysis. This dedicated testbed capability enables the IWRT to stress the system without being restricted by security policies of operational installations and also to find and evaluate fixes and countermeasures for any discovered vulnerabilities. Requirements, scheduling, assessment execution, and reporting was all accomplished in under two weeks and with minimal expenditures and added resources. Similar efforts prior to the establishment of the NPS testbed would take from 90 to 120 days, and not allow for retesting and rapid reporting of significant results that would aid in system development, improving the ACTD's security posture.

D.H. Anunciado

A Systematic Method For Interfacing Real-Time Systems And Non-Real-Time Systems In An Enterprise Environment

The subject of this dissertation is rapid system composition for the military systems that need to collaborate in order to solve a common goal. Specifically, integration of software and hardware systems into a complex system of systems is a common problem

encountered in both military and civilian domains. Since the individual systems comprising the final system of systems may have incompatible interfaces and implement incompatible communication protocols, the composition process is expensive and at the end results in a system that is brittle, inflexible, and difficult to test, verify, and correct. In this dissertation we investigate the problem of system integration in the context of complex endeavors commonly encountered in the military domain. By a complex endeavor we mean an activity that is characterized by participation of a large number of disparate entities that includes various military units, civilian authorities, multinational and international organizations, non-governmental organizations, and private companies and volunteer organizations. There is usually very little overlap between each of the aforementioned organizations which implies that the software and hardware systems, communication protocols, and architectural organization used by each may be dissimilar. Therefore successful completion of the shared goals requires that possibly numerous software and hardware system must collaborate together mimicking a complex system of systems. The main challenge is to ensure that integration is performed quickly, efficiently, and results in a reliable configuration.

In this work we provide a detailed classification of systems that are commonly used by the military in its operations. Systems are classified according to their potential of interfacing with a given enterprise, where we list the relevant technical challenges that must be solved in anticipation of integration with other systems. By focusing on just those systems that can interface with the given enterprise, time and resources are efficiently used on interfacing additional systems to solve a problem that the current configuration of systems in the enterprise cannot solve. We provide an algorithm for interfacing one or more computer systems in a verifiable configuration and provide a model to evaluate if a new configuration of an enterprise indeed contributes to successfully solving a problem that the previous configuration could not solve.

J. Evans

Semi-Automatic Methods To Aid Requirements Development

The goal of this research is to develop semi-automatic methods for a diverse set of stakeholders to collaborate on requirements development using recent advances in semantic web technology, natural language processing, and information retrieval. Another goal is to further develop the document driven development framework. The contribution this work will provide is a novel integration of software and language engineering techniques to exploit a body of documents, or corpus, as an aid in requirements engineering.

The information needed to build large complicated systems, and system of systems is typically not in one place, not in one format, not from one knowledge domain, not linked, not easily searchable, not always documented, and not easily traceable. If the information was in one place, was in one format, was understandable by non-domain experts, was fully linked and searchable, was documented, and traceable then the chances of synthesizing correct, complete requirements would be greatly increased.

Building requirements is just not a technical problem but a social one, especially in the context of systems of systems with diverse stakeholder communities. Getting

diverse groups to cooperate and share information is a challenge on several levels. On the basic level of communicating, each group typically has its own vocabulary or jargon, its own domain of knowledge, and its own tacit understanding of the problem. On the level of technical knowledge needed to solve the problem, each group by itself typically does not have a full understanding of the issues, costs, constraints, and means to solve the problem.

The Government Accountability Office examined 62 DoD programs investing over $950 billion dollars and found that less than 16 percent had the necessary knowledge at program start, referred to as knowledge point 1 in the report. Knowledge point 1 is when a match is made between the customer's requirements and the product developer's available resources in terms of knowledge, time, money, and capacity. Not having enough knowledge of the customer's requirements before proceeding to development inevitably led to cost overruns, schedule delays, and reduced capabilities. Another significant problem is systems interoperability. In a report the Government Accountability Office outlined why it is so difficult for the DoD to build interoperable systems. To overcome this problem, the DoD has instituted and has continually updated a document driven process called the Joint Capability Integrated Development System (JCIDS).

The JCIDS process starts with the creation of documents describing capability gaps, called Initial Capability Documents, or Joint Capability Documents. These are high level requirements documents that are then refined into system specific solutions called Capability Development Documents, which are written by the organization needing the system. These documents refer to other documents such as the Joint Task Lists, Service Task Lists, Joint Doctrine, and other documents that spell out tactics, techniques, and procedures. This body of documents is a huge corpus of institutional knowledge that is encyclopedic in size and scope, and virtually impossible for any one person to comprehend.

Viewing this corpus as a computationally active knowledge base that can be exploited using the novel methods, to be outlined here, as well as methods developed in the related disciplines of computational linguistics and natural language processing and applied in new ways to requirements engineering, this problem of too much information can be turned into a solution.

R. Halle

Risk Assessment Methodology Of Projects Based On Orientation, Alignment, And Synchronization Of Software Components During System Development, Delivery, And Integration

Risk analysis of the project as it evolves though the software life-cycle phases is a key objective of the DDD framework. Risk is a measure of potential future problems that could impact the meeting of system development and deployment objectives in terms of overall impact on program/system cost, delay delivery of products, or prevent the meeting of system performance parameters. In short, risk is a problem or issue that may or may not occur in the future. The greater the risk describes the greater potential of that

risk becoming a problem and/or the greater impact of the problem will have on the program should it occur.

To mitigate effects of risk, risk management is applied during all life-cycle phases of the project, which is a continuous process put in place to predict and measure the potential of these risks occurring and the corresponding means to resolve these risks should they transition into actual problems. It is an organized and continuous process to measure and track these unknowns. Key in risk management is the ability to identify these risks early and conduct the analyses of those risks and to put into place corrective actions that will be used to mitigate these risks. These risks are continuously monitored and reassessed to minimize their occurrence and/or minimize their impact should they occur.

As defined the Risk Management Guide for DoD Acquisition the components of risk management is: *risk identification*, *risk analysis*, *risk mitigation planning*, *risk mitigation plan implementation*, and *risk tracking*.

First step in risk management is risk identification. This is where all potential problems (or issues) are identified by examining historical data, past performance, current ongoing program data resulting from program performance, test data, and negative trends that indicate the likelihood of that problem occurring in the future. This step in risk management requires the program manager to examine all this data and assess whether the identified risks indeed pose a significant risk to the project. There is some level of subjectivity used in this prediction that cannot be avoided since this risk identification is focusing on predicting a problem that hasn't and potentially won't occur. In fact if a risk can be predicted to occur in the future with 100% accuracy, then it is not a risk it's actually a problem that will need to be dealt with. As a result subjective analyses must be included as part of overall risk identification process. The program manager can minimize the level of subjectivity by assembling as much data as possible to support the risk identification.

The next step in risk management is determining the likelihood and impact of this risk. This step is called risk analysis and is the primary feedback to the program manager on assessing the risk. In the Department of Defense this risk assessment is generally portrayed on a risk report matrix shown in Figure 3.

The Risk Reporting Matrix shows the likelihood and consequence of a risk and the corresponding green, yellow, and red risk rating. Green is considered low risk, yellow is considered moderate risk, and red is considered high risk. The program manager takes these risk assessments and initiates risk mitigation of those risks.

Risk mitigation is the next step in the risk management process. The intent of risk mitigation is effort to address these risks with different mitigation means. First is avoiding the risk all together by eliminating causes and/or consequences of the risk. By eliminating the causes of the risk, the risk cannot occur. By eliminating the consequence of the risk, the program manager eliminates the ability of the risk to impact the program if it does occur. If the program manager cannot eliminate the consequence, he/she could put in the means to control the problem when it does occur. This may be a more economic alternative than trying to eliminate the cause altogether. The next method to mitigate the risk is to try to transfer the risk to another program where it will have lesser impact. The final means to mitigate the risk is to accept the risk and continue on with the program with the hope it won't occur. Risk mitigation can be accomplished by pursuing

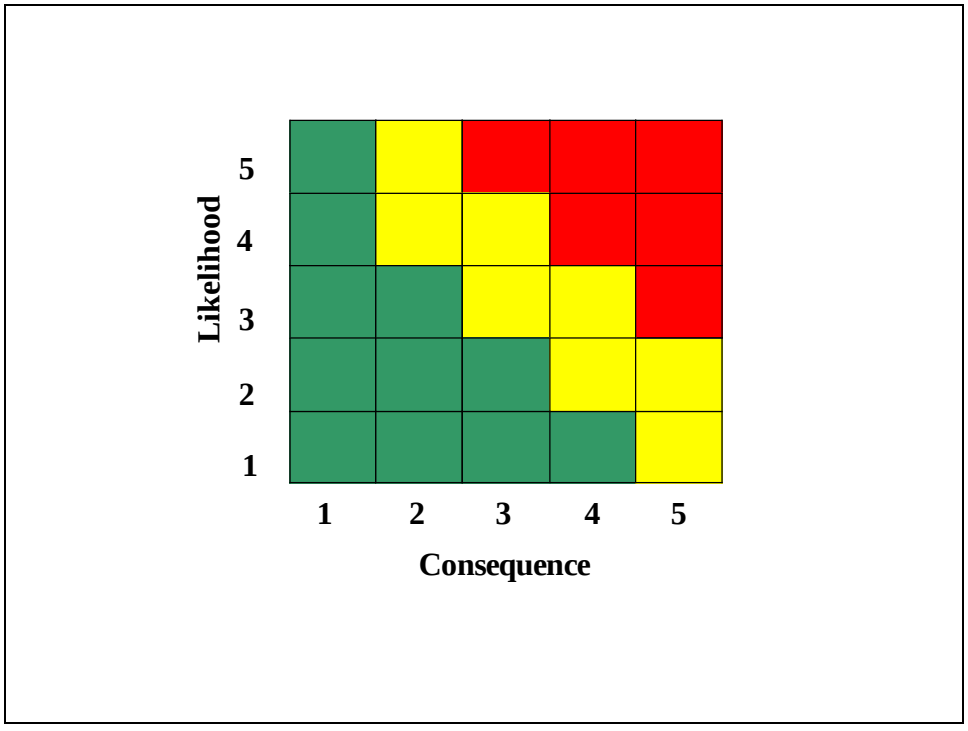

Figure 3 -- Risk Reporting Matrix

any of these mitigation methods or combination of these methods. Once the program manager determines the mitigation method(s) it is documented on the risk mitigation plan.

This Risk Mitigation Plan is the execution plan and the program manager's tool to program the resources necessary to enact the mitigation, should they be required. It defines the program tasks, team establishment, budget, contract changes, schedule changes, etc. that have to occur to support the monitoring of these risks and the mitigation method(s) should the risk occur. This plan is the means by which the program manager communicates these risks with upper management.

The final step in the risk management process is risk tracking where the risks are monitored to see if their likelihood of occurring and consequence if they do occur has changed. This feedback drives the risk management process that is triggered when changes happen in the likelihood of the risk occurring or in the consequences of the risk. Should this happen, then modified risks must be identified, reassessed, mitigation methods revised, and plans modified to account for these changes. Risk tracking is incorporated in the program manager's management of the program. The program manager uses numerous indicators to track risk including specific program metrics, program/technical reports, earned value reports, watch lists, schedule performance reports, and critical risk process reports.

The objective of a well-managed risk management program is to ensure the program stays on budget, schedule, and delivers the desired performance. The failure to manage risk can bring about the opposite result. With the lack of a risk management program the program manager will find himself/herself constantly having to react to problems as they occur and constantly reprogramming cost, schedule, and performance to solve these problems. What's worse is that depending on the consequence of the problems, they could generate additional problems resulting in the program spiraling out of control yielding significant cost, schedule, and performance impacts that could result in program cancellation.

Current software based risk management practices put additional burden on a program manager when managing risk and conducting risk assessment of system of systems. To account for this the program manager must employ subjective analyses to assess the risks of software-based system of systems. Below are just two examples of ways the program manager could conduct a risk assessment of a software-based system of systems.

The first approach would be to establish a panel of experts who would track individual software system risks and then turn those risks over to the program manager who would still be required to develop some level of subjective assessment based upon the advice of this panel of experts. This approach would force the program manager to bear a higher administrative burden by maintaining this panel of experts and still have to rely on subjective assessment of risk. It is this subjectivity that this research effort is seeking to minimize.

Another approach for the program manager would be to manage the program with a lesser knowledge of health of total software risks and potentially be unprepared for future problems should and when they occur. Basically this means ignoring these risks and waiting for the problems and reacting to them as they occur. The problem with this approach is that in software-based system of systems a problem arising in any individual software systems can and probably will cause perturbations in the other related software

system development efforts. This approach results in costly fixes and total system of systems program adjustments that would impact overall program cost, schedule, and performance. Only using a true software-based system of systems risk assessment method can the program manager know the health of the program and plan for those risks at the program level, thus minimizing cost, schedule, and performance impacts across the program and in each of the software-base systems making up the total system of systems.

It is expected at the completion of this research a method, or methods, to execute risk assessments of a software-based system of systems will be available for use in ongoing software-based system of systems programs. This research will show how the proposed methods better support execution of program management risk assessment throughout the software lifecycle model/processes. It will also demonstrate how these risk assessment methods can be used to synchronize the software system of systems. These methods will be demonstrated using mathematical models to support quantitative risk assessment evaluation. Finally, to assist the program manager in portraying this software-based system of systems risk assessment, improved and intuitive graphical representations of software-base system of system risk assessments will be developed.

P. Dailey

Acquiring Operational Profile Models to Drive Automated Testing of Open Architecture Weapon and Combat System Software

The main objective of this research effort is to derive a process for developing operational profile models to be used by weapon and combat system software developers. The largest challenge within that objective is figuring out how to use historical and/or real-time data sources to derive the PDFs that represent the inputs, which make up the model. One of the main technical issues that come with using such data for profile model development is choosing a realistic granularity that will result in adequate levels of coverage and confidence in the model's accuracy. Either discrete or continuous PDFs, from some family of distributions, combined with a small number of parameters that can be estimated from the data, will be used to make up the modeled inputs for the system under test. The available source data is finite and usually does not provide unlimited resolution, requiring some degree of approximation for the construction of the PDFs. Along with the approximations, some degree of statistical uncertainty in the accuracy of the model exists and should be understood. The broader challenge is determining what methodology for such calculation should be used as well as linking the results to confidence levels in the accuracy of the model. Understanding the tradeoffs associated with model fidelity, available data and development time are important and they will vary for each application of this approach.

An operational profile model is an *a priori* model that provides inputs to an Open Architecture (OA) software system module under test which are generated by sampling from probability density functions (PDFs) which characterize and represent the inputs and interfaces from the actual environment to the software module. For this research, the operational profile model's purpose is to drive automated testing of OA software, aimed

at probabilistic reliability assessments, supported by statistical confidence levels. Ideally, the automated software testing process utilizes the operational profile model for the generation of inputs to the software under test, but the model can also aid in the automated analysis of the software's outputs. This study further focuses on determining how to most effectively develop and implement such models within the weapon and combat system software domain.

There are four main goals of this research:

- Create an overall methodology for developing an operational profile model to drive automatic OA software testing.
- Determine how to efficiently use an operational profile model in the weapon and combat system software automated testing domain.
- Determine how to best calculate the reliability of the software component being tested using the operational profile model.
- Determine how to practically derive and calculate statistical levels of confidence in the accuracy of operational profile model.

J. Rivera

Weapon System Explosive Safety Review Board (WSESRB) Risk Mitigation Strategy for COTS Software Integration in Naval Weapons Systems

Research on the US Navy's Software Systems Safety Review Panel (SSSTRP) Requirements Framework resulted in discovery of the primary causes for the high level of vendor failure rates during the SSSTRP process. Research showed that the lack of structure associated with the vendor-provided Technical Review Package (TRP) led to inconsistent documentation and standards when trying to evaluate the vendor's software safety risk. The application of the NASA Software Safety Standard to Naval Weapons Systems development processes resulted in a new Software Evaluation Framework for the SSSTRP, specifically for evaluating safety of Commercial-Off-The-Shelf Software (COTS).

The application of COTS solutions in safety critical application environments poses a problem as commercial programs are not commonly designed to a high standard for safety-critical applications. The NASA Software Safety Standard is one of the most robust software safety assessment standards that currently exists, and thus provides a promising path to assessment of COTS software components for DoD requirements. This report identifies the portions of the NASA Software Safety Standard that are relevant to the assessment of COTS software and addresses a guideline of how these standards can be applied to weapons systems development. This discussion includes an analysis of the standard itself, justification of the need for safety-critical applications within weapons systems development as well as a brief discussion of the program, and identification and application of the appropriate portions of the standard to Naval weapons systems

development (including identification of checklists and other features that must be integrated into the system).

This research is seeking practical answers to the questions of what kind of documentation will most effectively support safety assessment of weapons software, what decision processes it must support, and how that can most effectively be done.

Bibliography

1 Doug Anunciado, A systematic method for interfacing real-time and non-real-time systems in an enterprise environment, PhD Dissertation, Naval Postgraduate School, in progress.

2 Andrew Chen, Intellectual risk management method for risk assessment support based on quantitative metrics, PhD Dissertation, Naval Postgraduate School, June, 2010.

3 Bruce Lewis, Integrated architectural modeling and cost modeling, PhD Dissertation, Naval Postgraduate School, in progress.

4 Christopher Mushenski, PM-Crusader Mobility systems, PhD Dissertation, Naval Postgraduate School, in progress.

5 Robert Sandoval, A predictive analysis of the implementation and effects of role base access control in the IEEE 802.11b wireless network, PhD Dissertation, Naval Postgraduate School, in progress.

6 Doug Lange, Lightweight inference for the generation of protocol agents supporting flexible system integration, PhD Dissertation, Naval Postgraduate School, 2007.

7 Randy Maule, Enterprise Software Architecture with Secure Ontology for Cross-Domain Knowledge Management, PhD Dissertation, Naval Postgraduate School, in progress.

8 Conrad Wittaker, Requirements Documentation Driven System Integrator, Master's Thesis, Naval Postgraduate School, in progress.

9 John Evans, Capability package assessment (CPA) test script development, PhD Dissertation, Naval Postgraduate School, in progress.

10 William Smuda, Rapid Prototyping of Robotic Systems, PhD Dissertation, Naval Postgraduate School, 2007.

11 W. Roof, "Nautical Predictive Routing Protocol (NPRP) for the Dynamic Ad-Hoc Naval Network (DANN)", NPS, June 2006.

12 Murrah M. R., PhD dissertation, Enhancements and Extensions of Formal Models for Risk Assessment in Software Projects, 2002.

13 Saboe, M., S., "A Software Technology Transition Entropy Based Engineering Model", PhD Dissertation (advisor: Luqi), Naval Postgraduate School, March, 2002.

14 J. P. Dupont, Complexity measure for the prototype system description language (PSDL). Master's Thesis (advisor: Luqi), Naval Postgraduate School, March, 2002.

15 Nogueira, J. C., *A Formal Model for Risk Assessment in Software Projects*, Ph.D. Dissertation (Advisor: Luqi), Naval Postgraduate School, September, 2000.

16 G. Jacoby, Luqi, "Int*ra*net Model and Metrics", Communications of ACM, February 2007.

17 Y. Qiao, H. Wang, Luqi, V. Berzins, “An Admission Control Method for Dynamic Software Reconfiguration in Complex Embedded Systems”, *International Journal of Computers and Their Applications,* Vol. 13, No. 1, March, 2006, pp. 28-38.

18 Luqi, Berzins, V., Yeh, R., *A Prototyping Language for Real-Time Software, IEEE Transactions on Software Engineering*, 1988, Vol. 14, No. 10, pp 1409-1423.

19 Luqi & J. Goguen, Formal Methods: Promises and Problems, *IEEE Software,* 14(1), 1997, 73-85.

20 Luqi, D. Lange, “Schema Changes and Historical Information in Conceptual Models in Support of Adaptive Systems”, *First International Workshop on Active Conceptual Modeling of Learning*, Tucson, AZ, 8 November, 2006.

21 Luqi, V. Berzins, William Roof, “Nautical Predictive Routing Protocol (NPRP) for the Dynamic Ad-Hoc Nautical Network (DANN)”, Springer, August, 2006.

22 Luqi, “Schema Changes and Historical Information in Conceptual Models of Learning”, position paper, *Active Conceptual Modeling of Learning Workshop*, SSC San Diego, CA, 10-12 May, 2006.

23 Luqi, “Transforming Documents to Evolve High-Confidence Systems”, *Proceedings of Workshop on Advances in Computer Science and Engineering*, Berkeley, CA, May 6, 2006, pp. 71-72.

24 Grant A. Jacoby, Thadeus Hickman, Stuart P. Warders, Barak Griffin, Aaron Darensburg and Daniel E. Castle, “Mobile Intrusion Protection,” *Proc. from The 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing*, June 26-29, 2006.

25 Timothy K. Buennemeyer, Grant A. Jacoby, Randolph C. Marchany, and Joseph G. Tront, "Battery-Sensing Intrusion Protection System," Proc. of the 7th IEEE SMC 2006 Information Assurance Workshop, June21-23, 2006.

26 Lange, Douglas. “PAL Boot Camp: Acquiring, Training, and Deploying Systems with Learning Technology,” Proc. of CCRTS 2006: the Command and Control Research and Technology Symposium, June 20–22, 2006, San Diego, CA.

27 V. Berzins, Luqi, “Achieving Dependable Flexibility via Quantifiable System Architectures”, Proc. of Workshop on Advances in Computer Science and Engineering, Berkeley, CA, May 6, 2006, pp. 53-54.

28 Luqi, L. Zhang, “Documentation Driven Evolution of Complex Systems”, Proc. of Workshop on Advances in Computer Science and Engineering, May 2006, pp.141-170.

29 A. Stone, P. Sawyer, “Identifying Tacit Knowledge-Based Requirements”, IEEE Proceedings, Vol. 156, No. 6, 2006, pp. 211-218.

30 D. Kelley, “A Software Chasm: Software Engineering and Scientific Computing”, IEEE Software, Vol. 24, No. 6, Nov. 2007, pp. 118-120.

31 Plummer S. USAF. (2000). “Memorandum: Awareness of Human-Systems Integration (HSI) in Air Force Acquisitions”.

32 Blasch, E. Assembling a distributed fused information-based human-computer cognitive decision making tool, Aerospace and Electronic Systems Magazine, IEEE, pp. 11-17, Volume: 15, Issue: 5, May 2000.

33 Grant A. Jacoby, Randy Marchany and Nathaniel J. Davis IV, “Using Battery Constraints Within Mobile Hosts To Improve Network Security,” IEEE Security & Privacy Magazine, Summer 2006.

34 Daniel E. Castle, Aaron Darensburg, Barak Griffin, Thadeus Hickman, Stuart P. Warders, and Grant A. Jacoby, "Gibraltar: A Mobile Host-Based Intrusion Protection System," National Conference on Undergraduate Research, April 2006.

35 B. Lewis. “The SAE Architecture Analysis & Design Language (AADL) A Standard for Engineering Performance Critical Systems”, Proc. Embedded Real Time Software Congress, Toulouse France, Jan 25-27, 2006.

36 Luqi, V. Berzins, William Roof, “Nautical Predictive Routing Protocol (NPRP) for the Dynamic Ad-Hoc Nautical Network (DANN)”, Monterey Workshop 2005: realization of reliable systems on top of unreliable networked platforms, Univ. of California, Irvine, USA Laguna Beach, September 22-24, 2005, pp. 1-9.

37 G. Jacoby, Luqi, “Critical Business Requirements Model and Metrics for Intranet ROI”, Journal of Electronic Commerce Research, Vol. 6, No. 1, 2005, pp. 1-30.

38 G. Jacoby, Luqi, “Intranet Portal Model and Metrics: A Strategic Management Perspective”, IT Professional, Vol. 7, No. 1, 2005, pp. 37-44, ISSN 1520-9202.

39 Grant A. Jacoby and Nathaniel J. Davis IV, “Battery-Based Intrusion Detection: A Focus on Power for Security Assurance,” 2005 Journal of Space and Aeronautical Engineering, 2005.

40 Luqi, Zhang, Lynn, Quantitative Metrics of Risk in Evolutionary Software Development, 2004.

41 Luqi, Zhang, L., Berzins, V., *Quantitive Metrics for Risk Assessment in Software Projects*, Proceedings of IASTED International Conference on Software Engineering and Applications (SEA 2003), Marina del Rey Ca., USA, November 3-5, 2003, pp. 76-81.

42 Luqi, L. Zhang, V. Berzins, “Software Component Repositories”, chapter in Wiley Encyclopedia of Computer Science and Engineering, accepted 2006.

43 Kazman R. “Handbook of Software Engineering and Knowledge Engineering,” December 2001, ftp://cs.pitt.edu/chang/handbook/15.pdf, 09 April 2007.

44 SEI Software Architecture, http://www.sei.cmu.edu/ata/ata_init.html, 09 April 2007.

45 2004 Indian Ocean Earthquake, Wikipedia, http://en.wikipedia.org/wiki/2004_Indian_Ocean_earthquake, Accessed on 12 November 2006.

46 Hurricane Katrina, Wikipedia, http://en.wikipedia.org/wiki/Hurricane_Katrina, Accessed on 12 November 2006.

47 Wong, L., Lange, D., Sebastyn, J., and Roof, W., “Command World”, Proceedings of the 2006 CCRTS, DOD Command and Control Research Program, San Diego, California, June 2006.

48 Cohen, P., and Pool, M., “The CALO 2005 Experiment Data Analysis”, http://calo.sri.com, Accessed 15 January 2007.

49 CHIPS, “The SSC San Diego Concept of the Composeable FORCEnet, CHIPS, Summer 2004. Available at

http://www.findarticles.com/p/articles/mi_m0OBA/is_3_22/ai_n6338487, Accessed on 15 January 2007.

50 G. Xie and Z. Dang. An automata-theoretic approach for model-checking systems with unspecified components. In FATES'04, LNCS. Springer, to appear.

51 Rozanski N., Woods E., "Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives," Addison Wesley Professional, 2005.

52 D. Giannakopoulou, C. S. Pasareanu, and J. M. Cobleigh. Assume-guarantee verification of source code with design-level assumptions. In ICSE'04, pages 211–220. IEEE Press, 2004.

53 Richards, Chet, *Certain to Win: The Strategy of John Boyd, Applied to Business*, Xlibris Corporation, 2004.

54 Xiaoqing, Liu, Kane, Gautam, Bambroo, Monu, "An Intelligent Early Warning System for Software Quality Improvement and Project Management", Proceeding of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTA'03), 2003.

55 D. Peled. Model checking and testing combined. In ICALP'03, volume 2719 of LNCS, pages 47–63. Springer, 2003.

56 F. Xie and J. C. Browne. Verified systems by composition from verified components. In FSE'03, pages 277–286. ACM Press, 2003.

57 A. Bertolino and A. Polini. A framework for component deployment testing. In ICSE'03, pages 221–231. IEEE Computer Society, 2003, http://www1.isti.cnr.it/~polini/downloads/icse03/icse2003.pdf.

58 H. Li, S. Krishnamurthi, and K. Fisler. Verifying cross-cutting features as open systems. ACM SIGSOFT Software Engineering Notes, 27(6):89–98, 2002.

59 J. Whaley, M. C. Martin, and M. S. Lam. Automatic extraction of object-oriented component interfaces. In ISSTA'02, pages 218–228. ACM Press, 2002, http://suif.stanford.edu/papers/whaley02.ps.

60 Reliasoft (2002). Weibull++ Ver 5.0. Reliasoft. From http://www.reliasoft.com.

61 V. Basili, , P. Costa, M. Lindvall, M. Mendonca, C. Seaman, R. Tesoriero, and M. Zelkowitz, *An Experience Management System for a Software Engineering Research Organization*. In Proceedings of the 26th Annual NASA Goddard Software Engineering Workshop, December 2001.

62 A. Orso, M. J. Harrold, and D. Rosenblum. Component metadata for software engineering tasks. Volume 1999 of LNCS, pages 129–144, 2001, http://www.cc.gatech.edu/aristotle/Publications/Papers/edo00.pdf.

63 P. E. Black, V. Okun, and Y. Yesha. Mutation operators for specifications. In ASE'00, pages 81–. IEEE Computer Society, 2000.

64 Levitt, R. *VDT Computational Emulation Models of Organizations: State of the Art and the Practice,* Center for Integrated Facility Engineering, Stanford University, 2000.

65 Murdock, J., "Semi-Formal Functional Software Modeling with TMK", Technical Report GIT-CC-00-05, Georgia Institute of Technology, 11 February 2000.

66 Boehm, B., et al. Software Cost Estimation with COCOMO II, Prentice Hall, 2000.

67 QSM (2000). *Software Lifecycle Management (SLIM) Training Version 2.02. Quantitative Software Management®, Inc. McLean*, Virginia, USA, 2000.

68 J. Voas. Developing a usage-based software certification process. IEEE Computer, 33(8):32–37, August 2000.

69 F. Salles, M. Rodriguez, J.-C. Fabre and J. Arlat, "MetaKernels and Fault Containment Wrappers", in Proc. of the 29th Fault-Tolerant Computing Symposium (FTCS'99), Madison, WI (USA), pp. 22-29, 1999.

70 Levitt, R., *The Vite Project Handbook: A User's Guide to Modelling and Analyzing Project Work Processes and Organizations*, Vité ©, 1999.

71 D'Souzaand D., Wills A, "Objects Components and Frameworks with UML," Addison Wesley, 1999.

72 T. A. Henzinger, S. Qadeer, and S. K. Rajamani. You assume, we guarantee: Methodology and case studies. In CAV'98, volume 1427 of Lecture Notes in Computer Science, pages 440–451. Springer, 1998, http://mtc.epfl.ch/~tah/Publications/you_assume_we_guarantee.pdf.

73 E. Hall, Managing Risk. *Methods for Software Systems Development.* Addison Wesley, 1997.

74 O. Kupferman and M. Vardi. Module checking revisited. In CAV'97, volume 1254 of Lecture Notes in Computer Science, pages 36–47. Springer, 1997.

75 SEI, *Software Risk Management*. Technical Report, Software Engineering Institute, CMU/SEI-96-TR-012, June 1996.

76 D. Karolak, *Software Engineering Management.* IEEE Computer Society Press, 1996.

77 Simos M., Creps D., Klinger C., Levine L., and Allemang D., "Organization Domain Modeling (ODM) Guidebook," Version 2.0. Informal Technical Report for STARS, STARS-VC-A025/001/00, 14 June 1996.

78 Devore, J. (1995). *Probability and Statistics for Engineering and the Sciences*. Duxbury.

79 Lyu, M. (1995). *Software Reliability Engineering*. IEEE Computer Society Press.

80 Jones, C., *Assessment and Control of Software Risks*, Yourdon Press Prentice Hall, 1994.

81 Johnson, N., Kotz, S., & Balakrishnan, N. (1994). *Continuous Univariate Distributions*. Vol. 1. Wiley & Sons.

82 Kirk P., Robert J. Walker, Julie & Firth, Robert. "Software Development Risk Management: An SEI Appraisal," Software Engineering Institute Technical Review '92 (CMU/SEI-92-REV). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1992.

83 Putnam, L., & Myers, W., *Measures for Excellence. Reliable Software on Time Within Budget*, Yourdon Press, 1992.

84 M. van Genuchten, *Why is Software Late? An Empirical Study of the Reasons for Delay in Software Development.* IEEE Transactions on Software Engineering. June, 1991.

85 Boehm, B., et al. Software Cost Estimation with COCOMO II, Prentice Hall, 2000.

86 Boehm, B., *Software Engineering Economics*, Prentice Hall, 1981.

87 R. Burton, and B. Obel, *Strategic Organizational Diagnosis and Design. Developing Theory for Application*. Kluwer Academic Publishers. 1998.

88 J. Voas. Certifying off-the-shelf software components. IEEE Computer, 31(6):53–59, June 1998, http://ieeexplore.ieee.org/iel4/2/15014/00683008.pdf?arnumber=683008.

89 H. Zuse, Software Complexity – Measures and Methods, Document W021, *Metricating A-KINDRA BT Project 610287*, Bache, R., South Bank, 1987.

90 A. Albrecht, and J. Gaffney, *Software Function Source Lines of Code and Development Effort prediction.* IEEE Transactions on Software Engineering, SE-9, 1983.

91 L. Lamport. Specifying concurrent program modules. ACM Transactions on Programming Languages and Systems (TOPLAS), 5(2):190–222, 1983.

92 A. Albrecht, *Measuring Application Development Productivity*. Proceedings IBM. October 1979.

93 G. Myers, *Software Reliability*. John Wiley & Sons. 1976.

94 Navy Modeling and Simulation Office, "Joint Semi-Automated Forces", http://nmso.navy.mil/index.cfm?RID=MNS_N_1010133.

95 Schmidt D., "Model-driven Engineering," IEEE Computer, February 2006, pp 25-31.

96 "Joint Architecture for Unmanned Systems," http://www.jauswg.org, 09 April 2007.

97 NATO Working Group, STANAG 4586 "Standard Interface of the Unmanned Control System (UCS) for NATO UAV interoperability.", http://www.cdlsystems.com/stanag.html, 10 May 2007.

98 "IEEE standard for distributed interactive simulation – applicationprotocols", http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel1/3700/10849/00499701.pdf?arnumber=499701, 10 May 2007.

99 McGregor, D, Brutzman, D, Blais, C, Arnold, A, Falash, M, Pollak, E, "DIS-XML: Moving to Open Data Exchange Standards", Proceedings of the Simulation Interoperability Standards Organization (SISO) Spring 2006 Simulation Interoperability Workshop, Huntsvill, AL, April 2006.

100 Davis, D, "Design Implimentation and Testing of a Common Data Model Supporting Autonomous Vehicle Compatibiliy and Interoperabiliyt", Naval Post Graduate School Dissertation, September 2006.

101 Szyperski C., "Component Technology - What, Where, and How?" icse, p. 684, 25th International Conference on Software Engineering (ICSE'03), 2003.

102 Szyperski C., "Component Software, Beyond Object Oriented Programming," Boston, MA, Addison-Wesley, 2002.

103 Czarnecki, K. and Eisenecker, U. W., "Generative Programming: Methods, Tools, and Applications." Boston: Addison-Wesley, 2000.

104 Gamma E., Helm R., Johnson R., Vlissides J., “Design Patterns,” Boston, MA, Addison-Wesley, 1995.

105 Freeman E., Freeman E., “Head First Design Patterns,” Sebastopol, CA, O’Reilly, 2004.

106 Schmidt D., Stal M., Rohnert H., Buschmann F., “ Pattern-Oriented Software Architecture, Vol 2, Patterns for Concurrent and Networked Objects,” West Sussex, England, 2000.

107 CMSC491D Design Patterns In Java, http://www.research.umbc.edu/~tarr/dp/fall00/cs491.html, 09 April 2007.

108 Generic Modeling Environment, http://www.isis.vanderbilt.edu/projects/gme, 09 April 2007.

109 Czarnecki K., Antkiewicz M., Hwan C., Kim P., Lau S., Pietroszek K., “ Model-Driven Software Product Lines,” OOPSLA ’05, San Diego, CA, October 2005

110 Eclipse, http://www.eclipse.org/, 09 April 2007.

111 Storey, V., and Dey, D., “A Methodology for Learning Across Application Domains for Database Design Systems,” IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 1 (2002), pp. 13-28.

112 Storey, V., Goldstein, R., and Ullrich, H., “Naïve Semantics to Support Automated Database Design,” IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 1 (2002), pp. 1-12.

113 Fensel, D., and Motta, E., “Structured Development of Problem Solving Methods,” IEEE Transactions on Knowledge and Data Engineering, Vol. 13, No. 6 (2001), pp. 913-932.

114 Maule, R., and Gallup, S., “TACFIRE: Enterprise Knowledge in Service-Oriented Architecture,” Proceedings of the 2006 Military Communications Conference (MILCOM 2006), October 23-25, 2006, Washington, DC.

115 Fayad, M., and Schmidt, D., “Object-Oriented Applications Frameworks,” Communications of the ACM, Vol. 40, No. 10 (1997), pp. 32-38.

116 Fayad, M., Schmidt, D., and Johnson, R., Object-Oriented Application Frameworks: Problems and Perspectives. New York: Wiley, 1997.

117 Johnson, R., “Frameworks = (components + patterns),” Communications of the ACM, Vol. 40, No. 10 (1997), pp. 39-42.

118 Johnson, R., and Foote, B., “Designing Reusable Classes," Journal of Object-Oriented Programming, Vol. 1, No. 5 (1988), pp. 22-35.

119 Posnak, E., Lavendar, R., and Vin, H., “An Adaptive Framework for Developing Multimedia Software Components. Communications of the ACM, Vol. 40, No. 10 (1997), pp. 43-47.

120 Schmid, H., “Systematic Framework Design,” Communications of the ACM, Vol. 40, No. 10 (1997), pp. 48-51.

121 Baumer, D., Gryczan, G., Knoll, R., Lilienthal, C., Riehle, D., and Zullighoven, H., “Framework Development for Large Systems,” Communications of the ACM, Vol. 40, No. 10 (1997), pp. 52-59.

122 Maule, R., “Current and Future Initiatives in Military Knowledge Management, in (D. Schwartz, ed.) Encyclopedia of Knowledge Management, Hershey, PA: Idea Group, 2006.

123 Maule, R., Gallup, S., and Schacher, G., “Experimentation Management Knowledge System: Information Systems Integration for DoD Network-Centric Operations,” Proceedings of the First International Conference on Web Information Systems and Technologies (WEBIST 2005), May 26-28, 2005, Miami, FL.

124 Campbell, R., and Islam, N., “A Technique for Documenting the Framework of an Object-Oriented System,” Computing Systems, No. 6 (1993), p. 4.

125 Schmidt, D., “Applying Design Patterns and Frameworks to Develop Object-Oriented Communications Software, in (P. Salus, Ed.) Handbook of Programming Languages, Vol. 1, New York: MacMillan Computer Publishing, 1997.

126 Brugali, D., Menga, G., and Aarsten, A., “The Framework Life Span,” Communications of the ACM, Vol. 40, No. 10 (1997), pp. 65-68.

127 Gamma, E., Helm, E., Johnson, R., and Vlissides, J., “Design Patterns: Elements of Reusable Object-Oriented Software,” Reading, MA: Addison-Wesley, 1994.

128 Pree, W., Design Patterns for Object-Oriented Software Development, Reading, MA: Addison-Wesley, 1994.

129 Denmeyer, S., Meijler, T., Nierstrasz, O., and Steyaert, P., “Design Guidelines for Tailorable Frameworks,” Communications of the ACM, Vol. 40, No. 10 (1997), pp. 60-64.

130 Codenie, W., Hondt, K., Steyaert, P., and Vercammen, A., “From Custom Applications to Domain-Specific Frameworks,” Communications of the ACM, Vol, 40, No. 10 (1997), pp. 71-77.

131 Maule, R., and Gallup, S., “Quality of Service in Next-Generation Knowledge Management,” Proceedings of the International Conference on Information Society (i-Society 2006), ISBN 0-9546628-1-4, August 7-10, 2006, Miami, FL.

132 Berners-Lee, T. “The Semantic Web,” Scientific American. Available http://scientificamerican.com/2001/0501issue/0501berners-lee.html, 2001.

133 Chandrasekaran, B., Josephson, J., and Benjamins, V., “What are Ontologies, and Why Do We Need Them,” IEEE Intelligent Systems, Vol. 14, No. 1 (1999), pp. 20-26.

134 Grimes, S., “The Semantic Web,” Intelligent Enterprise, Vol. 5, No. 6 (2002), pp. 16, 52.

135 Avigdor, G., and Mylopoulos, J., “Toward Web-Based Application Management Systems,” IEEE Transactions on Knowledge and Data Engineering, Vol. 13, No. 4 (2001), pp. 683-702.

136 Fikes, R., and Farquhar, A., “Distributed Repositories of Highly Expressive Reusable Ontologies,” IEEE Intelligent Systems, Vol. 14, No. 2 (1999), pp. 73-79.

137 Maule, R., Gallup, S., and Schacher, G., “Quality of Service Process Variables in Complex B2B Systems Integration Assessment,” Proceedings of the IEEE International Conference on E-Commerce Technology (CEC 2004), July 6-9, 2004, San Diego, CA, pp. 155-161.

138 Maule, R., “Enterprise Knowledge Security Architecture for Military Experimentation,” Proceedings of the IEEE SMC 2005 International Conference on Systems, Man and Cybernetics, October 10-12, 2005, Waikoloa, HI.

139 Maule, R., and Gallup, S., “Innovation Engine for Online Analytics and Information Assurance in Enterprise B2B Infrastructure,” Paper presented at the Fifth International Conference on Electronic Commerce, September 30-October 3, 2003, Pittsburgh, PA.

140 Fayad, M., Schmidt, D., and Johnson, R., Object-Oriented Application Frameworks: Implementation and Experience. New York: Wiley, 1997.

141 Tim Hale, Naval Space and Warfare Command (PMW-170), 2005.

142 Ludlow, Nelson D, Ph.D., CEO, Mobilisa Inc. Chief Designer, WSF Internet Project.

143 Royer, Elizabeth M and Perkins, Charles E., “Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Algorithm,” 1999.

144 Sinnott, R.W., "Virtues of the Haversine", Sky and Telescope 68 (1984).

145 Tanenbaum, Andrew S., Computer Networks, Fourth Edition, Prentice Hall, 2003, p 357.

146 Hecker, Chris; Physics Part 4, The Third Dimension, Behind the Screen, 1995, pp 1-10.

147 V. Berzins, M. Rodríguez, M. Wessman, “Putting Teeth into Open Architectures: Infrastructure for Reducing the Need for Retesting”, Proceedings of the Fourth Annual Research Symposium – Acquisition Research: Creating Synergy for Informed Change, Monterey, CA, May 16-17, 2007, pp. 285-312.

148 V. Berzins, "Which Unchanged Components to Retest after a Technology Upgrade”, Proceedings of the Fourth Annual Research Symposium – Acquisition Research: Creating Synergy for Informed Change, Monterey, CA, May 14-15, 2008, pp.142-153.

149 V. Berzins, Luqi, P. Musial, “Formal Reasoning about Software Object Translations”, Proceedings of the 2008 Monterey Workshop, Budapest, 23-27 Sep. 2008, pp. 65-78.

150 V. Berzins, C. Martell, Luqi, P. Adams, "Innovations on Natural Language Document Processing for Requirements Engineering", Springer LNCS 5320, 2008, pp. 125-146.

151 V. Berzins, P. Dailey, "How to Check If It Is Safe Not to Retest a Component”, In Proceedings of the Sixth Annual Research Symposium – Acquisition Research: Defense Acquisition in Transition, Monterey, CA, May 12-14, 2009, pp. 189-200.

152 J. Rivera, Luqi, V. Berzins, " Effective Programmatic Software Safety Strategy for US Navy Gun System Acquisition Programs”, In Proceedings of the Sixth Annual Research Symposium – Acquisition Research: Defense Acquisition in Transition, Monterey, CA, May 12-14, 2009, pp. 159-164.

153 Luqi, D. Lange, “Schema Changes and Historical Information in Conceptual Models in Support of Adaptive Systems”, Springer LNCS 4512, 2008, pp. 112-121, ISBN 978-3-540-77502-7

154 Luqi, “Rapid Prototyping”, Encyclopedia of Computer Science and Engineering, John Wiley & Sons, January 2009, pp. 2343-2348

155 Luqi, L. Zhang, V. Berzins, “Software Component Repositories”, Encyclopedia of Computer Science and Engineering, Wiley, January 2009, pp. 2559-2563.

156 Luqi, F. Kordon, Preface, Proceedings of the Monterey Workshop: Modeling, Development and Verification of Adaptive Systems, 31 March – 2 April 2010, Microsoft Research, Redmond, WA, pp. 2-3.

157 V. Berzins, “How to Certify Software Architectures for Reliable Reconfiguration”, Proceedings of the Monterey Workshop: Modeling, Development and Verification of Adaptive Systems, 31 March – 2 April 2010, Microsoft Research, Redmond, WA, pp. 128-129.

158 V. Berzins and P. Dailey, “Improved Software Testing for Open Architecture”, Proceedings of the Seventh Annual Research Symposium – Acquisition Research : Creating Synergy for Informed Change, Monterey, CA, May 11-13, 2010.

159 P. Dailey. “Acquiring Operational Profile Models to Drive Automated Testing of Open Architecture Weapon and Combat System Software”, Ph.D. Dissertation, NPS, June, 2010.

160 Luqi, P. Dailey, “Profile-Based Automated Testing Process for Open Architecture Track-Processing Software”, Technical Report #NPS-CS-10-005, Mar. 2010.

161 Luqi, V. Berzins, J. Rivera, “Requirements Framework for the Software Systems Safety Review Panel (SSSTRP)”, Technical Report # NPS-PM-09-145, Sep. 2009. Also appeared as Technical Report # NPS-GSBPP-10-003, Sep. 2009.

162 Luqi, V. Berzins, P. Dailey, “Driving Automated Open-Architecture Testing: An Operational Profile Model-Development Strategy”, Technical Report #NPS-PM-09-146, Sep. 2009.

163 Luqi, L. Zhang, V. Berzins, Y. Qiao, “Documentation Driven Development for Complex Real-Time Systems”, IEEE Transactions on Software Engineering 30, 12, p. 936-952.

164 Y. Qiao, V. Berzins, Luqi, “FCD: A Framework for Compositional Development in Open Embedded Systems”, International Conference on Information Technology, Las Vegas, Nevada, April 2005.

165 M. Rodriguez, Luqi, V. Ivanchenko, V. Berzins, “Reliability and Flexibility Properties of Models for Design and Run-time Analysis”. In Proceedings of 2006 Monterey Workshop, October 16-18, 2006, Paris, France.

166 Luqi, V. Ivanchenko, ”Advanced Interface for Examining and Upgrading Complex Systems”. The World Congress in Computer Science, Computer Engineering, & Applied Computing. Las Vegas, Nevada, USA. June 25-28, 2007.

167 Luqi, F. Kordon, "Innovations for Requirements Analysis: From Stakeholders to Formal Designs", Proc. Monterey Workshop 2007, Monterey CA, Sep. 2007.