



**US Army Corps
of Engineers®**

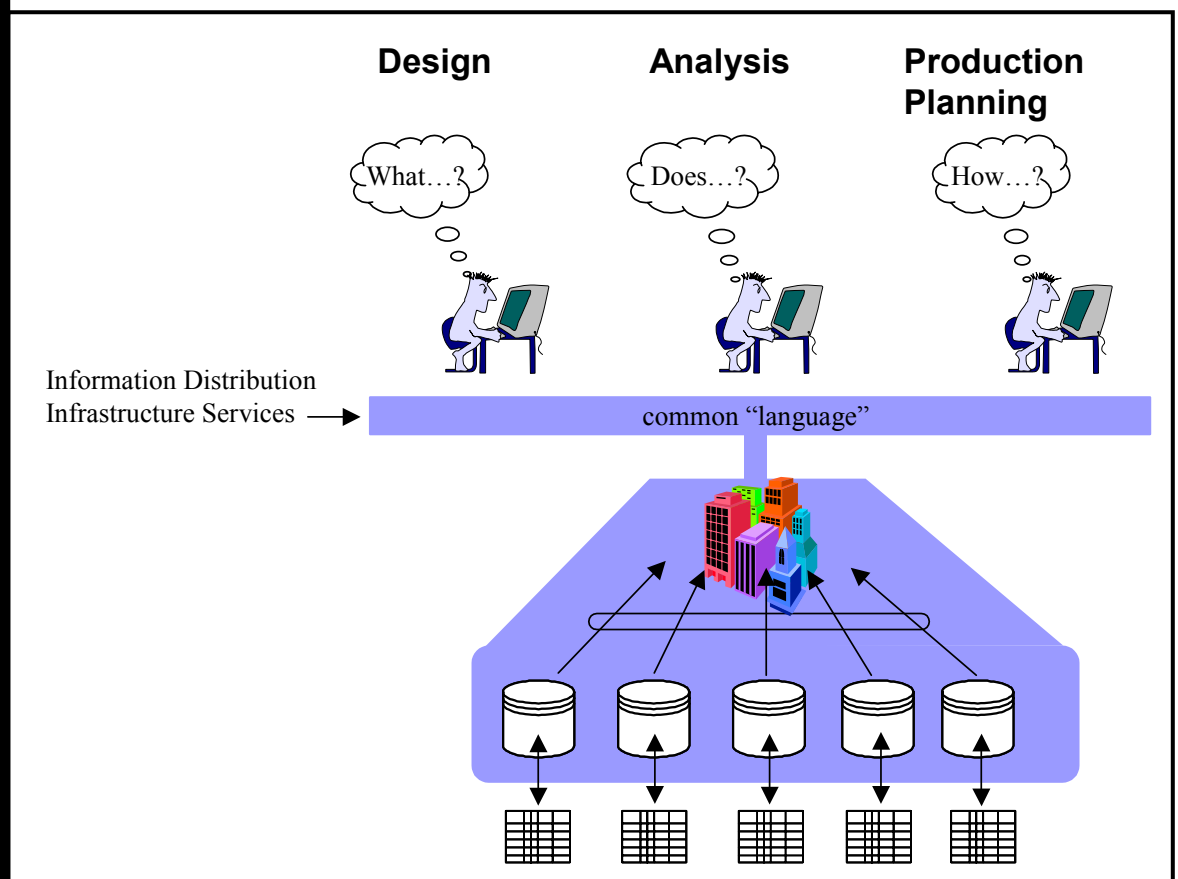
Engineer Research and
Development Center

Conflict Management in Collaborative Engineering Design

Basic Research in Fundamental Theory, Modeling Framework, and Computer Support for Collaborative Engineering Activities

Stephen C-Y. Lu, Firdaus Udwadia, James Cai,
William Burkett, Michael P. Case, Blessing Adeoye,
and Van J. Woods

January 2002



REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 01-01-2002		2. REPORT TYPE Final		3. DATES COVERED (FROM - TO) 01-06-2001 to 01-01-2002	
4. TITLE AND SUBTITLE Conflict Management in Collaborative Engineering Design: Basic Research in Fundamental Theory, Modeling Framework, and Computer Support for Collaborative Engineering Activities Unclassified				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Lu, Stephen C-Y. ; Author Udwadia, Firdaus ; Author Cai, James ; Author Burkett, William ; Author Case, Michael P ; Author				5d. PROJECT NUMBER 611102AT23	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Engineer Research and Development Center (ERDC) Construction Engineering Research Laboratory (CERL) PO Box 9005 Champaign, IL61826-9005				8. PERFORMING ORGANIZATION REPORT NUMBER ERDC/CERL TR-02-2	
9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS U.S. Army Engineer Research and Development Center (ERDC) Construction Engineering Research Laboratory (CERL) P.O. Box 9005 Champaign, IL61826-9005				10. SPONSOR/MONITOR'S ACRONYM(S) CEERD-CV-ZT	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APUBLIC RELEASE					
13. SUPPLEMENTARY NOTES Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.					
14. ABSTRACT All real-world engineering tasks involve collaborative activities among a group of human participants. The ability to understand, support, and improve collaboration is a critical factor in determining the overall cost, time, and effectiveness of modern engineering activities. Collaborative engineering tools are being introduced into the market at a rate so high that it is difficult to infuse technology in a reasoned and effective manner. Practitioners must decide which tools to adopt and to develop new and more effective processes. These decisions are made even more difficult by the fact that no body of theory exists that has been shown to describe the interaction between complex object-oriented data models, engineering processes, and human decisionmaking. The objective of this work was to develop the Theory for Collaboration in support of complex engineering system decisions in a highly distributed and heterogeneous environment. The results of this research will lead to a sound theoretical foundation that may be used to analytically and mathematically model, simulate, manage, and optimize collaborative engineering activities. Such a theory of collaboration will enable researchers to design, predict, and control various collaborative activities, systems, and environments, and to implement practical IT systems to support these important human endeavors.					
15. SUBJECT TERMS collaborative engineering; computer aided design CAD; engineering design; conflict management; computer aided engineering CAE					
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT Same as Report (SAR)		18. NUMBER OF PAGES 176	
19. NAME OF RESPONSIBLE PERSON Wolfe, William William.J.Wolfe@erdc.usace.army.mil					
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified	19b. TELEPHONE NUMBER International Area Code Area Code Telephone Number 217352-6511 DSN -		
					Standard Form 298 (Rev. 8-98) Prescribed by ANSI Std Z39.18

Executive Summary

All real-world engineering tasks involve some type of collaborative activities among a group of human participants. For example, modern facility design and installation management tasks are often decomposed into sub-tasks, and then distributed to many engineers and users. Communication, coordination, and collaboration are major concerns when managing these decomposed tasks and their distributed participants because of disciplinary, geographical, and temporal differences. The ability to understand, support, and improve collaboration is a critical factor in determining the overall cost, time, and effectiveness of modern engineering activities.

New collaborative engineering tools are currently being introduced into the market at a high rate that makes it difficult to infuse technology in a reasoned and effective manner. Practitioners must assess factors of interoperability, automation, and collaborative utility to decide which tools to adopt and to develop new and more effective processes. These decisions are made even more difficult by the fact that no body of theory exists that has been shown to describe the interaction between complex object-oriented data models, engineering processes, and human decisionmaking. Such a theory could serve as a foundation for new forms of software tools and for collaborative frameworks.

This basic research was performed jointly by the Construction Engineering Research Laboratory (CERL), U.S. Army Engineer Research and Development Center (ERDC), and the IMPACT Research Laboratory, School of Engineering, University of Southern California (USC). The overall goal of this project was to develop a *Theory for Collaboration* in support of complex engineering system decisions in a highly distributed and heterogeneous environment. The research objective was to contribute to a better understanding of human collaborative behaviors in making technical decisions, especially to an understanding of how these behaviors are influenced by social interactions, and to how modern IT systems should be designed to support these group technical activities. The results of this research will lead to a sound theoretical foundation that can potentially be used to analytically and mathematically model, simulate, manage, and optimize collaborative engineering activities.

Basic research in theories for collaboration is intrinsically multi-disciplinary, and should be grounded in some specific application domains. This research covered very broad intellectual ground, from engineering disciplines to behavior, decision, psychology, organization, and the social sciences. “Conflict management activity in collaborative engineering design” was used as an application domain to guide, test, and demonstrate basic research results. The investigations devoted significant effort to address the problem of engineering collaboration from many different viewpoints, rather than committing to a more limited solution based on conventional thinking. The aim was to contribute to a fundamental understanding of the problem of real-world collaboration, and find comprehensive answers and rigorous theories that may at first seem to be a bit ambitious, or even uncomfortable.

This research investigation began with an approach in game theory. This classical approach was abandoned after its limitations in real world collaborative engineering became apparent. Researchers instead began to search for an entirely new paradigm, starting from a theory in social science, to construct a conceptual framework to describe the reality of collaborative engineering activities. Following this, a system architecture was developed that specified the overall structure and individual components of this conceptual framework. Some mathematical techniques were used to model the key components of this framework to make it operational, functional, and implementable. Some demonstrative prototypes were built to illustrate the research approaches.

The foundation of this research program is a new paradigm, the *Socio-Technical Framework of Collaborative Engineering*, which is meant to more realistically describe real world collaboration activities. This conceptual framework forms the basis for a collaborative design system architecture with several key components, each of which involves and utilizes rigorous modeling techniques. The main ideas behind this framework and its architecture have come (generally) from many social and organizational sciences, and are based (specifically) on the *co-construction* process adapted from the theory of social construction. Whenever appropriate, the modeling techniques used for key components are drawn from theoretical studies and fundamental knowledge in the fields of logic, mathematics, decision sciences, information technologies, and organizational theories, etc. The adaptation and advancement of these fundamental techniques, in combination with the new knowledge generated from their integration, collectively represent new contributions to the basic research in this theory of collaboration.

Modeling the evolving perspectives of stakeholders during a collaboration process is the cornerstone of this Socio-Technical framework. Perspective modeling

facilitates both “*understanding people*” and “*data understanding*” during group interactions, and is consequently the core on which this theory for collaboration is built. Accordingly, this research program has two main components: one that treats collaborative design as a conflict management task to support “people understanding,” and one that builds an information-sharing infrastructure to support “data understanding” during collaboration.

It is anticipated that continued, long-term fundamental research into this new Socio-Technical Framework will help close three key knowledge gaps critical to the establishment of a theory of collaboration:

2. A new information theory that directly relates to “meaning”
3. Self-organizing, continuously evolving, intelligent, collaborative systems
4. Computer-mediated human-to-human interactions.

Such a theory of collaboration will enable researchers to design, predict, and control various collaborative activities, systems, and environments, and to implement practical IT systems to support these important human endeavors.

Foreword

This study was conducted for U.S. Army Engineer Research and Development Center (ERDC), Construction Engineering Research Laboratory (CERL) under project number, 4A162720AT23, "Basic Research in Military Construction." The technical monitor was Dr. Paul A. Howdyshell, CECER-CV-ZT.

The work was performed by the Engineering Processes Branch (CF-N), of the Facilities Division (CF), Construction Engineering Research Laboratory (CERL). The CERL Principal Investigator was Blessing F Adeoye. Michael P. Case is Chief, CF-N, and L. Michael Golish is Operations Chief, CF. Part of this work was done by The IMPACT Research Laboratory, School of Engineering, University of Southern California (USC) under Contract No. DACA-88-96-D-0003. Stephen C-Y. Lu and Firdaus Udwadia are Professors at USC, and William Burkett and James Cai were Research Assistants. The technical editor was William J. Wolfe, Information Technology Laboratory. The associated Technical Director was William D. Severinghaus. The Director of CERL is Dr. Alan W. Moore.

CERL is an element of the U.S. Army Engineer Research and Development Center (ERDC), U.S. Army Corps of Engineers. The Director of ERDC is Dr. James R. Houston and the Deputy to the Commander is A.J. Roberto, Jr.

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.
DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

Contents

Executive Summary	3
Foreword	6
List of Figures and Tables	9
1 Introduction	11
1.1 Background.....	11
1.2 Objective	12
1.3 Approach.....	12
1.4 Mode of Technology Transfer	14
2 Evolution of Computer Aided Engineering Research	15
2.1 Engineering as Collaboration	15
2.2 Individual Engineers Working with Separated Tools.....	16
2.3 Groups of Engineers Working with Integrated Product Models.....	17
2.4 Team Collaboration with Scalable Enterprise Information Resources	18
2.5 Needs for Basic Research in Collaborative Engineering Activities	21
2.6 A Brief History of These Research Investigations	31
3 A Socio-Technical Framework For Collaborative Engineering Activities	40
3.1 The Social Construction Theory	41
3.2 The Basics of the Socio-Technical Framework.....	43
3.3 The Architecture of the Socio-Technical Framework	56
3.4 An Overview of Key Components of Socio-Technical Framework	64
4 Collaborative Design as Conflict Management	70
4.1 Perspective Modeling using Dynamical Systems.....	70
4.2 Process Modeling	77
4.3 Conflict Management.....	86
4.4 The STARS Prototype System	98
5 Information Sharing for Collaborative Design.....	108
5.1 Information Sharing in Collaborative Engineering Design.....	108
5.2 A Framework for Perspective Model Integration.....	117
5.3 Interoperability Through Data Abstraction	132
5.4 Case Studies in Data Transformation through Abstraction.....	148
5.5 XML—Its Usefulness and Its Shortcomings	154

6 Summary and Conclusion	157
6.1 Current Understanding	157
6.2 Summary of Basic Research	158
6.3 Some Future Research Directions	160
Bibliography	164
Appendix A: List of Technical Publications	168
Appendix B: Facility Engineering Framework Workshop Report	169
CERL Distribution	175
Report Documentation Page	176

List of Figures and Tables

Figures

1	The overall structure of this report.....	13
2	The first generation of CAE approach	17
3	The second generation of CAE approach	19
4	The third generation of CAE approaches	20
5	A conceptual illustration of social construction process	43
6	A high-level view of traditional engineering design process	55
7	The conceptual architecture of the socio-technical framework	56
8	DCM modeling framework	58
9	Interaction between design campaign and design environment	59
10	Social interactions in Suh's axiomatic design framework.....	60
11	Supporting collaborative design	61
12	Perspective-based approach to collaborative engineering.....	63
13	Research directions in the socio-technical framework	63
14	The five key components of the two major research directions	64
15	Dynamical model of design perspectives	73
16	Concept structure built by stakeholder	75
17	Generation of perspective model by concept structure	77
18	Collaborative design process net	81
19	Design process diagram decomposition.....	85
20	Steps of methodology	90
21	Matrices used in analysis	92
22	Adjusting process to manage conflict	93
23	Overview of STARS system architecture.....	99
24	STARS concept structure interface	101
25	Perspective model dependencies.....	102
26	Joint construction of process model	104
27	Conflict detection and resolution	107
28	Multiple perspectives on same application domain	109
29	Integrated product data model.....	110

30	Integrated information resources.....	110
31	Data model and data	111
32	Relationship of data model to data in database	112
33	Relationship between perspectives and data	112
34	Translation of data	113
35	Islands of automation.....	114
36	Enterprise data model.....	114
37	Database federation	114
38	Integrated data model approaches	115
39	Forming mental model of perspective	119
40	Externalization of a perspective: perspectives model	120
41	Formalizing perspective model.....	120
42	Perspective models, data model, and data	122
43	Multiple perspectives	122
44	Multiple perspectives across stakeholders	123
45	Variations of Perspective Model Characteristics	125
46	Relationship of LOSA, scope, and size	128
47	Relationship of LOSA, scope, and size in three axes	128
48	Layered model of collaborative information infrastructure.....	130
49	Onion model of collaborative information infrastructure	130
50	Data as a directed graph	133
51	Mapping data between databases.....	137
52	Data mapping domain and range	138
53	Abstraction as integration approach	139
54	Number of required interfaces	140
55	Truth maintenance in semantic abstraction	141
56	Integrated information infrastructure.....	141
57	Structure of PDM	150
58	XML example	155

Tables

1	Socio-technical framework and traditional engineering design approaches	50
2	Terminology correspondence	136
3	Schema measures.....	146
4	JEDMICS data fields.....	150

1 Introduction

1.1 Background

All real-world engineering tasks involve some types of collaborative activities among a group of human participants. For example, modern facility design and installation management tasks are often decomposed into sub-tasks, and then distributed to many engineers and users. Communication, coordination, and collaboration are major concerns when managing these decomposed tasks and their distributed participants because of disciplinary, geographical, and temporal differences. The ability to understand, support, and improve collaboration is a critical factor in determining the overall cost, time, and effectiveness of modern engineering activities.

New collaborative engineering tools are currently being introduced into the market at a high rate that makes it difficult to infuse technology in a reasoned and effective manner. Practitioners must assess factors of interoperability, automation, and collaborative utility to decide which tools to adopt and to develop new and more effective processes. These decisions are made even more difficult by the fact that no body of theory exists that has been shown to describe the interaction between complex object-oriented data models, engineering processes, and human decisionmaking. Such a theory could serve as a foundation for new forms of software tools and for collaborative frameworks.

This basic research was performed jointly by the Construction Engineering Research Laboratory (CERL), U.S. Army Engineer Research and Development Center (ERDC), and the IMPACT Research Laboratory, School of Engineering, University of Southern California (USC). This work seeks to contribute a theoretical basis and practical support to the development of a very important subject in the engineering profession—the collaborative activities that must occur among a group of engineers when they collectively make decisions for complex systems under various technical and nontechnical influences.

1.2 Objective

The overall objective of this work was to contribute to the fundamental knowledge of collaborative engineering. The specific goal of this basic research was to establish a theoretical foundation and software framework to support collaborative design as conflict management during the facility delivery processes.

1.3 Approach

1.3.1 Overview of This Study

This study achieved its research goals by:

1. Developing a Socio-Technical Framework for the understanding and modeling of collaborative engineering
2. Developing computer modeling methods and conflict management strategies for collaborative engineering processes
3. Developing information sharing schemes and ontological mapping methods for collaborative engineering activities
4. Integrating the above results from items 1, 2, and 3 to form a foundation that supports collaborative engineering and scalable enterprise information systems.

1.3.2 Overview of This Report

Chapter 1 of this report gives relevant general information to provide proper background to the research approach and results to be presented in the rest of this report. This includes a review of the knowledge gaps and of the needs for basic research in this subject, and an historical overview of the intellectual paths taken over the past 4 years in exploring better understandings of collaborative engineering.

Chapter 2 presents the background, scope, architecture, and components of the Socio-Technical Framework, which forms the foundation of this research in collaborative engineering. This chapter includes theoretical justifications, important rationales, and implementation arguments. The role of perspective and its modeling is explained as a core of the Socio-Technical Framework. This chapter concludes with a brief overview of the two main research directions taken: Collaborative Design as Conflict Management and Information Sharing in Collaborative Design.

Chapters 3 and 4, respectively, present Technical results in Collaborative Design as Conflict Management and Information Sharing for Collaborative Design. The implementations of the prototype systems are also included in these chapters. Specifically, Chapter 4 discusses the methodology and implementation of STARS, the Socio-Technical Analysis and Research System, that resulted from these investigations in collaborative process modeling/simulation and conflict management.

Chapter 5 discusses the details of information sharing research, which enables the results from STARS to be integrated with scalable enterprise information resources.

Chapter 6 summarizes the research understandings and accomplishments to date, and suggest a list of topics for further basic research, prototype development and future system deployment. Figure 1 shows the structure of the report graphically.

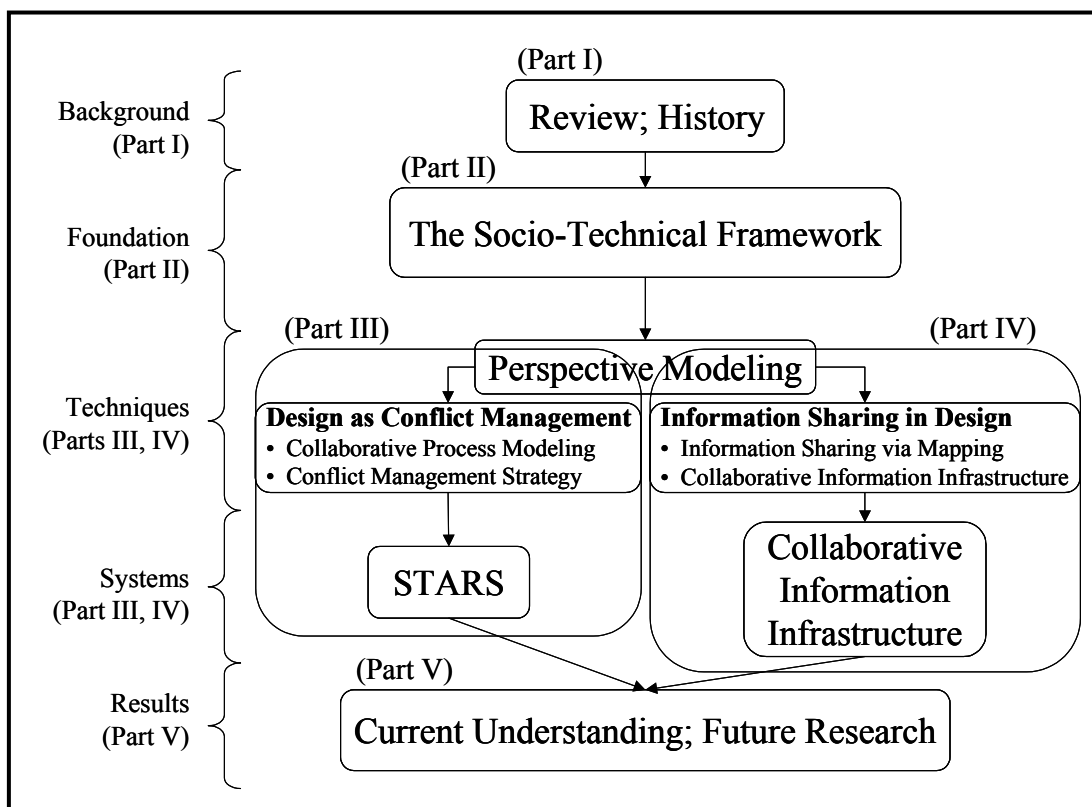


Figure 1. The overall structure of this report.

1.4 Mode of Technology Transfer

It is anticipated that the tools developed here in domain-specific collaboration environments will serve as test beds that will help advance research on enabling technologies for collaboration from a descriptive domain to a prescriptive one. This report will be made available through the World Wide Web (WWW) at URL:

<http://www.cecer.army.mil>

2 Evolution of Computer Aided Engineering Research

By definition, the subject under study is interdisciplinary in nature, and goes far beyond the usual traditional thinking adopted by the relevant research communities. As a result, it is often helpful to (at least temporarily) hold in abeyance those established viewpoints toward engineering problem solving and human interactions when attempting to understand and appreciate the new research directions presented in this report. A major change in the old thinking paradigm is required to obtain some meaningful breakthroughs in this difficult subject area. Materials presented in this part of the report prepare readers with the necessary background for this significant paradigm shift.

2.1 Engineering as Collaboration

Unlike science, which deals with the discovery of knowledge via analysis, engineering deals with the creation of artifacts via synthesis. The ability of humans to synthesize is among one of the most creative endeavors of cognition and far beyond what the best computers can do. Notwithstanding this basic limitation, digital computers and information technologies have had significant impact on the engineering profession over the past three decades. Many computer tools and methods have been developed for various engineering tasks, and fundamentally changed the ways through which engineers solve their problems and interact with each other. These changes are particularly visible over the past few years as the World Wide Web and Internet become integral parts of all professional communities.

Within the scope of Computer Aided Engineering (CAE), engineers solve problems by using computer tools to manipulate, communicate, and process data. Therefore, from the CAE perspective, any problem solving activity consists of three key components, namely *humans*, *tools*, and *databases*. As engineers must collaborate with each other to solve large-scale, real world problems in team settings, the interaction among humans, computer tools, and databases becomes an important issue that determines the overall problem-solving productivity. Managing these interactions has become increasingly difficult as engineering tasks

and teams are increasingly distributed over the geographical, temporal, and disciplinary boundaries.

As the complexity and degree of distribution of engineering tasks increase, the understanding of interactions must be expanded from an understanding of data and tools to an understanding of the level of human activity in CAE. Otherwise, degradation of the overall productivity will soon appear, as has already been observed in some domains, when more CAE efforts are introduced. Due to the basic limitations of digital computers and the present state of knowledge in information technologies, most studies to date have focused on various interfaces and/or integration approaches on the database levels.

These CAE efforts at the data level, although they offer some practical solutions for less complicated and small-scale problems, often fail to scale up to match the complexity and dynamic nature of the real world situations. Some limited efforts have been devoted to improve the interactions among computer tools; but the most critical issue of human interactions in CAE has been largely ignored so far. This chapter briefly reviews the three stages of CAE evolutions in terms of how they treat the interaction issues among databases, tools, and humans.

2.2 Individual Engineers Working with Separated Tools

The classic CAE scenario is to first divide a complex engineering task into smaller ones, and then assign them to individual engineers who use separated computer tools for their solutions stored in separated databases. For example, a product development task is generally divided into three subtasks, namely design, process planning, and manufacturing (Figure 2). Different CAE tools, such as CAD, CAPP, and CAM, for each of these subtasks produce separate databases, which are difficult to integrate to form a consistent product model.

Although some specialized interfaces can be written to link these separated databases for specific application queries, the fact that an integrated product model does not exist severely limits their usefulness. Furthermore, since these separated databases are often not part of the enterprise information system, the integration between engineering and other organizational and business concerns is practically nonexistent.

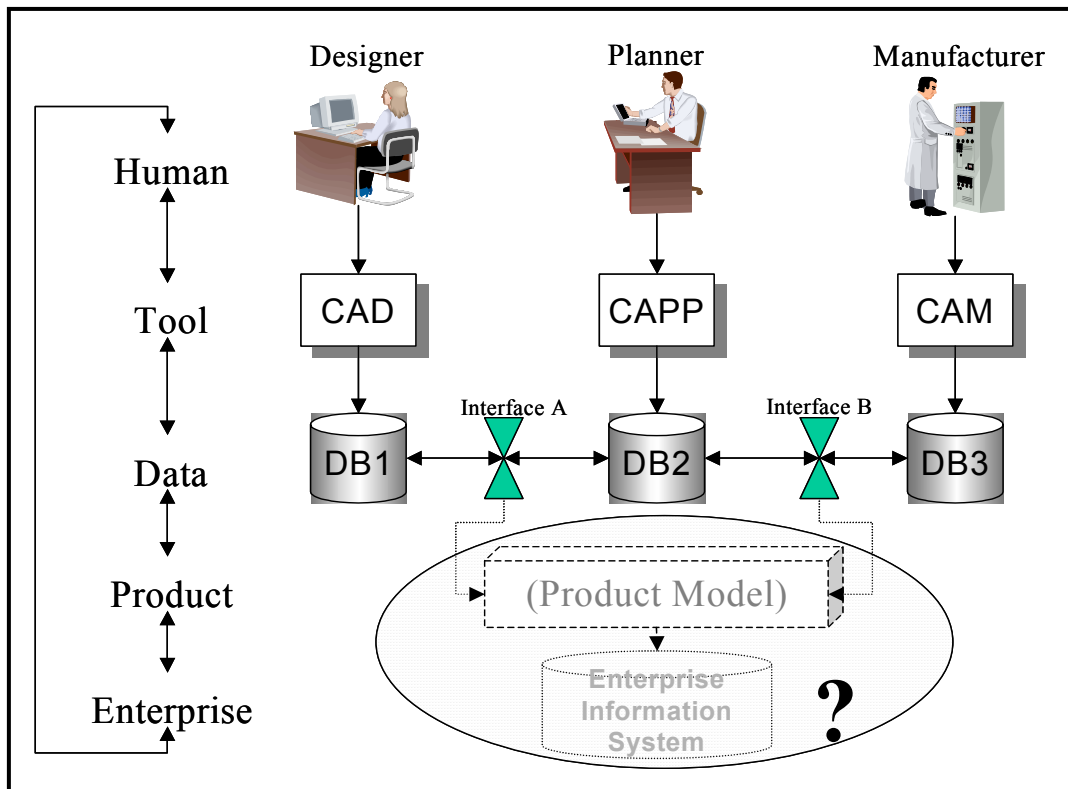


Figure 2. The first generation of CAE approach.

2.3 Groups of Engineers Working with Integrated Product Models

As a way to overcome the above difficulties, significant efforts have been devoted over the past decade in the CAE research communities to develop “Integrated Product Models” (Lu 2000). As Figure 2 shows, these efforts result in a tighter integration across separated CAE databases under a common data structure, and as a result, provide more direct interactions among those involved CAE tools. This enhanced level of interaction offers engineers the possibility to make decisions with a broader scope of information across the life-cycle concerns of a product. In a practical sense, interactions of both CAE tools and databases are managed through this integrated product model.

Although integrated product models represent a significant step forward in CAE research in terms of tool and databases interactions, they still fail to meet a few important real world requirements due to the rigid ontological definitions and data structure used in these models. Besides the difficulty of scaling the models up for large-scale projects, these models are often hard to integrate with the enterprise information systems that contain much nontechnical information that is equally important in applications. Many efforts have been devoted to address this critical problem of “scalable and integrated” enterprise information systems

in recent years. However, most of them fail to recognize the basic fact that any enterprise consists of people, who always have different and dynamically changing perspectives under various technical and social influences when working in group settings. These changing perspectives continuously affect the interaction among humans and the ways CAE tools and databases can be integrated.

2.4 Team Collaboration with Scalable Enterprise Information Resources

To fundamentally address the important issue of interactions among a group of collaborating human problem solvers within enterprises, it is clear that the current CAE approaches, which mainly focus on database and tool integration, must be extended to also include the modeling of *human interactions*. Such a significant extension, although ambitious and difficult, provides a new foundation upon which scalable and integrated enterprise information systems can be developed. Furthermore, it establishes a new theoretical underpinning for collaborative engineering research that can lead to the third generation of CAE approaches.

Modeling human interactions is a difficult task that requires considerations of human cognition and group behaviors, which go beyond the traditional paradigm of CAE. As this is an unexplored research terrain, different approaches must be explored, compared, and synthesized to advance the states of understanding. The approach to be presented in this report is an example of this new direction toward an expanded CAE research.

Computer supports to collaborative human interactions call for problem solving information to be modeled and captured at the *content*, *context*, and *purpose* levels. Traditional CAE research has mainly focused on capturing information at the content level in databases and/or product models (Figures 1 and 2). Content-level information is static and often hard to integrate in applications, because it lacks the specific contexts and purposes from which it is generated. Information at the context and purpose levels is necessary to guide the integration of content level information during collaborative activities.

Limited amounts of efforts have been devoted to capture the contexts of decisionmaking in research communities. For example, in the database communities, schemas and ontology have been active areas of research investigations. These higher level definitions can implicitly capture information related to the context and purpose of the data, and are useful in building large-scale information infrastructures. Engineering researchers have developed various approaches to capture “rationales” during design processes, and then use these

structured rationales to guide the justification, communication, and integration of design decisions, i.e., results at the content level, in the future. However, all these approaches focus on recording context and purpose information in a static manner, and hence have only limited success in supporting collaborative activities.

The approach taken in this research is based on the realization that human behaviors and decisions within team settings are based on their “perspectives of the world,” which are dynamically influenced by many technical and nontechnical, e.g., social, factors during group interactions. Therefore, the ability to understand, capture, and model these dynamic perspectives holds the key to building successful computer supports to human collaborative activities.

As will be described in details later, this research is based on a Socio-technical Framework of collaborative engineering. This new framework allows explicit modeling of the human aspects of group problem solving as a continuous co-construction process of their different *perspectives*. A perspective is used to capture the human aspects of collaboration, and is defined as a collection of *content*, *context*, and *purpose* that is unique to each problem solver, and dynamically changes according to different social interactions. Figure 3 conceptually illustrates how this new approach can advance an understanding of collaborative engineering and facilitate the development of scalable enterprise information resources.

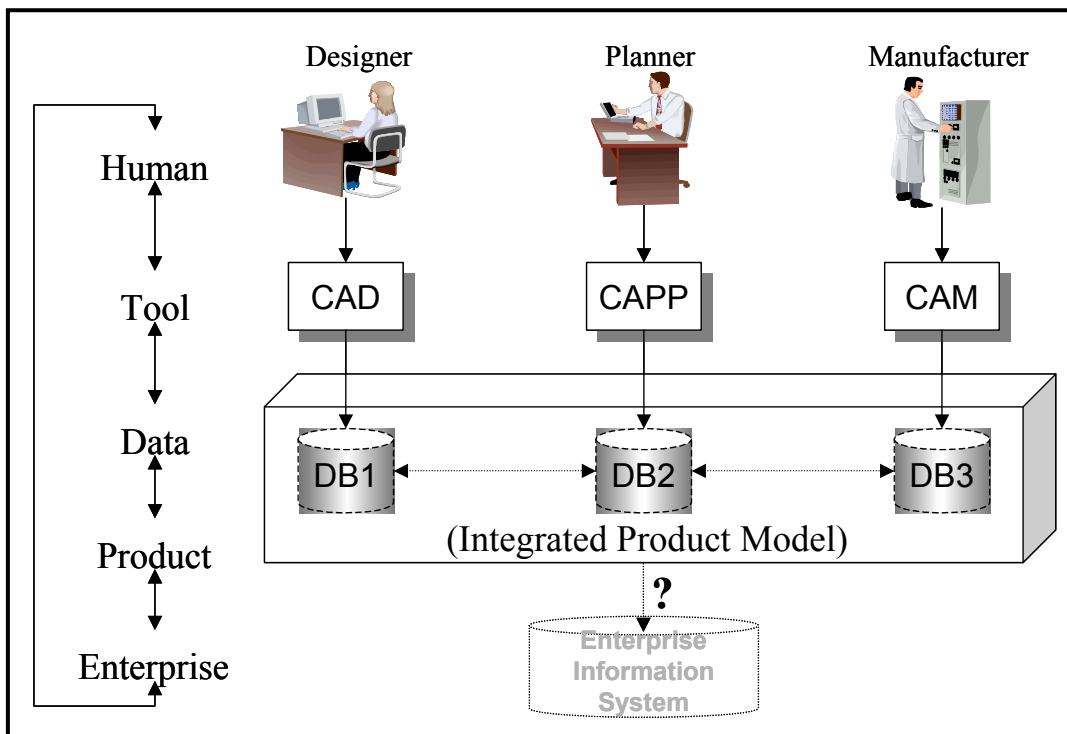


Figure 3. The second generation of CAE approach.

Compared with those previous CAE approaches that focus on capturing information content (shown in the lower-left box in Figure 4) this work models human perspectives *directly* when they evolve during group interactions as a way to capture context and purpose information (shown in the upper box in Figure 4). This perspective modeling is the foundation for a collaborative process and conflict management approaches, which result in “perspective state diagrams.” These dynamically generated perspective state diagrams cannot only facilitate collaborative processes and support conflict management, but they can also serve as guidelines for data mapping specifications. Those perspective-driven specifications can be used to guide the data mappings among individual databases, product models, and enterprise information resources, resulting in a truly scalable information infrastructure for collaborative activities.

Perspective modeling, process management, conflict strategy, data mapping, and collaborative information infrastructure are the key components of this research program under the Socio-technical Framework. More details of each of these components are described in the remaining chapters of this report.

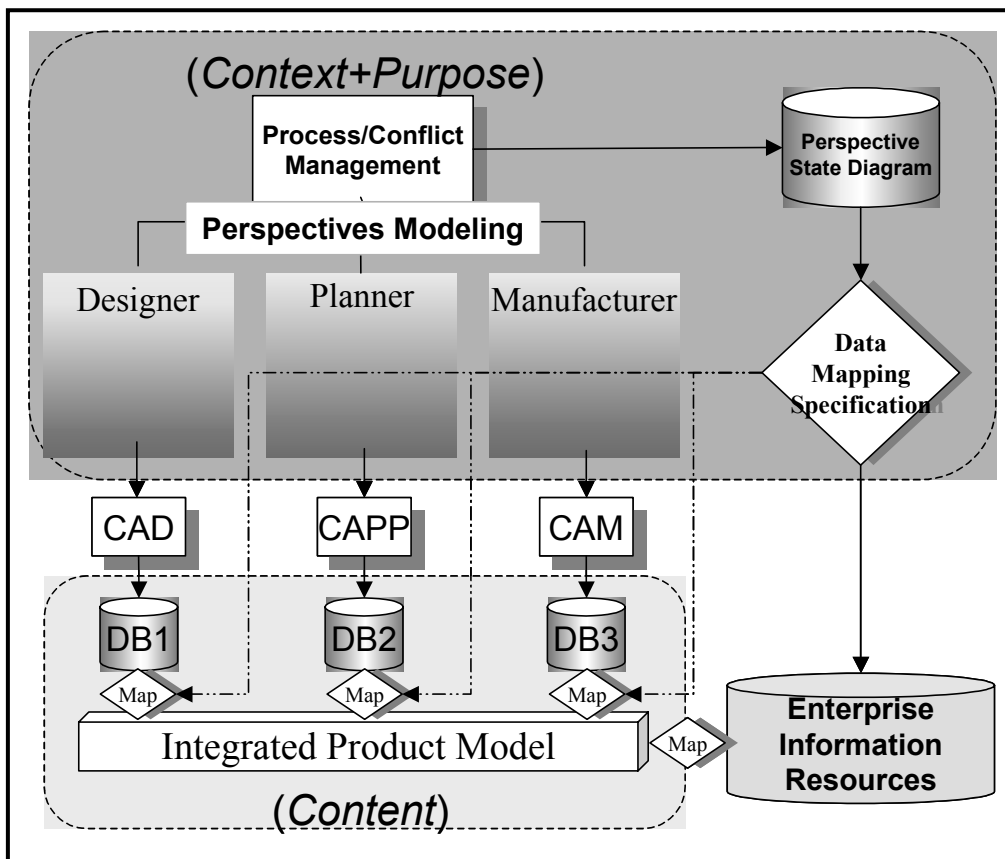


Figure 4. The third generation of CAE approaches.

2.5 Needs for Basic Research in Collaborative Engineering Activities

Collaborative engineering represents the next frontier of CAE research, and is a critical factor in determining future human productivity in this era of Internet revolution. New collaborative engineering tools are being introduced into the market at such a high rate that it is difficult for users and developers to infuse technology in a reasoned and effective manner. Practitioners must assess factors of interoperability, automation, and collaborative utility in deciding which tools to adopt and in deciding how to develop new and more effective processes. To make these decisions more difficult, there does not exist a body of basic theory that has been shown to describe the interaction between complex object-oriented data models, engineering processes, and human decisionmaking. The objective of this work is to develop a body of theory that can be used to systematically model, and to mathematically simulate and optimize collaborative engineering activity. If successful, this theory will serve as a foundation for new forms of commercial software and for collaborative frameworks, such as the Facility Engineering Framework.

The following section summarizes a few key knowledge gaps in collaborative engineering research, then lists the three specific research goals in this research program, which seeks to contribute to the closing of these knowledge gaps. Based on these research goals, some specific research questions that this research program attempts to address will be presented.

2.5.1 Where are the Key Knowledge Gaps?

The research needs of collaborative engineering cannot be met by simply continuing the current CAE efforts along thinking paths that are limited to the modeling of technical processes. Explicit modeling of humans, and how they interact with each other within changing social settings, must be included in the development of a sound theoretical foundation for collaborative engineering. There are several important knowledge gaps that exist along this new research dimension, for example:

- how to model engineering activities as human processes beyond those traditional technical viewpoints
- how to model *collaborative* engineering activities as *group* human processes that have many *social* interactions beyond those traditional technical viewpoints
- how to best understand, model, simulate, and support human processes in problem solving within a social setting

- how studies of human processes in social settings can be made systematic and rigorous enough to be grounded with some established theories from a variety of disciplines
- how these studies can be integrated with conventional CAE results and made implementable using the current and future information technologies
- how to systematically verify these implemented results with real world situations to gradually converge to a set of theories for collaborative engineering
- how to derive some practical guidelines from the above theories to guide the developments of the next generation CAE tools and methods to support collaborative engineering practice
- how to generalize these collaborative engineering theories, tools, and methods to other nontechnical application domains to form a set of useful knowledge bases for human collaboration in general.

To address these areas of needed research, this study:

1. Focuses on a particular engineering task, namely collaborative design as a vehicle to gain more insight into this complex subject
2. Takes the specific view of “collaborative design as conflict management” as a means to formulate an approach into a functional framework
3. Assumes that the research results from the above specifications of task and approach can be later expanded to form more generalized knowledge bases.

With these premises, three specific research tasks were targeted:

1. Collaborative design process modeling and conflict management strategies
2. Collaborative information sharing and integration of enterprise information systems
3. Combine the above results to support collaborative design as conflict management.

These research goals and tasks were fulfilled by devoting research efforts to the following four areas:

1. To develop a Socio-Technical Framework for the understanding and modeling of collaborative engineering
2. To develop computer modeling methods and conflict management strategies for collaborative engineering processes
3. To develop information sharing schemes and ontological mapping methods for collaborative engineering activities
4. To integrate the above results from items 1, 2, and 3 to form a foundation that supports collaborative engineering and scalable enterprise information systems.

2.5.2 Specific Research Questions This Work Seeks To Answer

The research goals of this study were phrased into a list of specific research questions that the results of this research program should answer. Note that, for the purpose of this study “collaborative design” is defined as “a *social-technical campaign* conducted by a team of *stakeholders* with the purpose of producing a shared consistent *model* that meets all the life-cycle *requirements* of a product,” where:

- *social-technical campaign* == a set of technical decisions and actions carried out by a team of stakeholders within a social environment
- *stakeholders* == an individual, or a group of individuals, who participates in a design campaign or has a material interest in the results of the campaign
- *model* == a set of statements or specifications in a given language, which represents a real-world phenomenon and can be used to illustrate, explain, understand, evaluate, record, predict, or control that phenomenon
- *requirements* == the value of a property of something that must be equaled or surpassed by the evaluated (for qualitative requirements) or measured (for quantitative requirements) value of the property on a particular instance of that thing.

With this definition, *collaborative* design and *group* design are synonyms in this research, and are used interchangeably here. Collaborative design is seen as a necessary capability to realize the virtual engineering team concept to support CERL’s collaborative engineering initiative. These research efforts were particularly focused on investigating those basic fundamental issues that are critical to the understanding, support, and realization of collaborative design.

The basic research questions and answers (Q/A’s) in collaborative design fall into the following three categories:

1. Q/A’s related to “engineering design” and “design process” as a base to study collaborative design
2. Q/A’s related to viewing and modeling collaborative design as a “conflict management” activity
3. Q/A’s related to data and information sharing in support of collaborative design.

The first group of questions contributes to the understanding of *design modeling strategies*, the second set of questions relates to *conflict management strategies*, and the third collection of questions addresses *information modeling strategies*. Together, these three aspects, namely design modeling, conflict management, and information management, constitute a possible framework upon which a sound computational model for collaborative engineering could be developed.

2.5.2.1 The Engineering Design and Design Process

Is there a design process, or are there elements of a design process, that is/are independent of the community of stakeholders in engineering design?

Traditional views treat the engineering design campaign as a set of technical tasks and activities (e.g., search, optimization, etc.). Recent studies of collaborative design have brought out the importance of treating engineering design as a process. However, this work assumes that no process can be defined and evaluated without knowing the involved stakeholders (see definition above) for that particular design campaign. Therefore, the most fundamental issues in collaborative design research, in this view, should be the investigation of stakeholders, beyond just the understanding of the process. In other words, *stakeholders* form a *process* that results in a *design*. Note that this does not mean that stakeholders cannot and should not employ a traditional design process model. Rather, it means that every collaborative campaign is ultimately unique and driven by the particular stakeholders involved

Are there elements (e.g., tasks and activities) that are common to all design campaigns regardless of who participated, when it occurred, where it occurred, and how it occurred?

This study is based on the belief that there are some intrinsic characteristics of design campaigns that are person-independent (who), time-independent (when), location-independent (where), and method-independent (how). It is important to identify and study these intrinsic elements that must be dealt with by any valid design theory and models for both individual (i.e., noncollaborative) design and group (i.e., collaborative) engineering design. Some examples are: dependencies, uncertainties, and abstractions (which jointly define the complexity of a design campaign). This will help to identify the “framework” for collaborative design.

Are there intrinsic and fundamental differences between noncollaborative (i.e., individual) and collaborative (i.e., group) design campaign, and, if there are, how can these differences be modeled?

If the answer is “no,” then one could possibly treat individual design activities as a network of black boxes, and then add various communication protocols and coordination strategies between them to support collaborative design. In other words, one can model noncollaborative design as a

purely technical activity, and then superimpose those social considerations to make it a social-technical process for collaborative design. However, this work disagrees with this viewpoint, and is based on the belief that the social-technical view must be taken from the beginning of collaborative design studies. The basic problem is how to model design in a social environment, rather than just the understanding of the effects of a social environment on any collaborative effort. Therefore, it is critical to identify and model the fundamental differences between non-collaborative and collaborative design.

Can one have a feasible computational model of the design process? At what level of “granularity” can one develop useful and practical computational models for collaborative design?

A collaborative design campaign must not be too complex, and/or too detailed, to be modeled on computers by properly managing the “modeling granularity” of a design process and its resulting conflicts. In this light the question is one of defining and measuring the complexity and abstraction of a design campaign, and assessing the *computability* of the model of a design campaign. These are all important fundamental questions to answer if the study is to search for computer simulations and/or aids in collaborative design research.

Can the modular design approach be used as a base to study, understand, and model a collaborative design campaign? What are the features of modular design that distinguish it from conventional design? How can these features be characterized?

Modularity is a way to deal with the granularity in modeling a design campaign. How does one modularize a design? Are there multiple ways to do this modularization? How to create, manage, and dispose design modules? How to resolve conflicts when multiple design modules are available? Can design modules evolve by themselves? What are the impacts of modular design on collaborative design research and applications? Modularity and granularity are part of the larger issue of *abstraction* in design modeling. These are all important questions to investigate.

How does one systematically measure the quality, or goodness, of a design, a design model, and a design campaign?

If the most fundamental issue in design research is to understand its involved stakeholders, then they should be the ones who ultimately determine the goodness of a design. But, how does one quantify this important quality measurement more systematically? By the rate with which stakeholders converge their individual, diverse opinions into a shared, cohesive model? Alternatively, can one measure design quality by the efficiency of its design process? Or by the correctness of the design decisions within their activities and tasks? The fundamental question of how to compare two designs, two design models, or two design campaigns remain unanswered as yet.

Are the salient constraints that guide the design process (e.g., the mappings) in individual-based design processes the same as the salient constraints that guide group design processes?

Though some constraints (like costs, time, etc.) may be the same, it is apparent that several constraints may arise in collaborative design that might be significantly different from those that arise in individual-based design. For example, a significant constraint in collaborative design may be the need for the same design team to function effectively on other design projects, or on design problems that may arise at later times in the life cycle of the facility that is designed. This might place constraints like: building trust between team members, building good-will, and sometimes going along with a team member's thinking even if that may not be the "best" decision from a purely technical standpoint. In fact "team building" could be thought of as a possible example of a constraint under which the collaborative design process may be required to operate. One needs to identify and categorize the types and nature of constraints in the collaborative design process, investigate the categories of conflicts these constraints may create, and determine and categorize management strategies to handle these conflicts.

2.5.2.2 Collaborative Design as Conflict Management

Why do conflicts occur in a design campaign? What are the roles of conflicts in engineering design? Can design conflicts be taken as a central view to study and model collaborative design?

Conflicts always occur in engineering design. Traditional approaches treat conflicts as abnormalities in the process, and devote resources to eliminate them as much as possible. Conflicts will occur even more in collaborative design. Can one still afford to follow these traditional approaches? Are conflicts necessarily the bad thing in a design campaign? Since conflicts occur so frequently in collaborative design, can the group take advantage of them as a normal and positive aspect of the design campaign? Can design conflicts be used to drive a design process and/or improve the design quality? Can a conflict management model be built as a base to model collaborative design? These are all important and interesting research questions to be answered.

What is a design conflict? What are the basic characteristics of a design conflict? What are the *different* types of generic design conflicts? Can one develop a general and expandable taxonomy for design conflicts?

Conflict is the situation in which viewpoints, perspectives, and/or decisions of one or more stakeholder(s) become mutually incompatible with respect to the satisfaction of some design requirements. It is important to understand what constitutes a conflict in order to manage it during the design campaign. How can a specific design conflict be described? By its contributing what, who, where, when, why, how, and importance? Are there some common features among these descriptions? Can one propose a logic structure for these descriptions to form a few generic groups so that conflict taxonomy can be developed?

What are the proper methods/strategies that one can use to manage conflicts in an engineering design campaign? Is it possible to develop a general and expandable taxonomy for various identified conflict management strategies for design?

Many methods and strategies have been developed to deal with different types of design conflicts. Each of them is particularly effective in its special situation. These conflict management strategies and the situations under which they will be effective must be clearly understood. These characteristics will enable the identification of strategies that are generic

across a set of design conflict situations. Then logic taxonomy for conflict management strategies can be developed.

Conflict management methods and strategies are greatly influenced by corporate culture (i.e., what is acceptable behavior) as opposed to organizational rules and norms (i.e., what is legitimate behavior) or engineering constraints. What are the core characteristics of corporate culture that will reduce “crucial” conflicts in the collaborative design at various stages?

Managing conflicts is never a purely technical task. The same conflict can be resolved differently at different corporations due to their different cultures. This indicates that the study of conflict management strategies must take a social-technical view. Understanding of physics and corporate culture are both important to an effective conflict management strategy. An important basic research issue is to capture those social aspects of the conflict management methods/strategies so that they can be used effectively in supporting collaborative design.

Can one develop a logic mapping between the taxonomy of design conflicts and the taxonomy of conflict management strategies? Can one adapt these logic mappings into specific design contexts?

Suppose one can develop taxonomies for generic conflict situations and resolution strategies, and then a mapping between these two types of taxonomies should be developed in order to make conflict management activities operational. One can imagine a (computer) system that can first identify the specific conflict situation by matching it against the generic conflict taxonomy, then map this situation onto its corresponding resolution strategies within the strategy taxonomy, paying attention to elements like the importance of conflict and its organizational culture. In this way, conflict management activities can be made operational on computers as an important component to support collaborative design.

2.5.2.3 Information Sharing in Collaborative Design

What are the roles of an engineering information model in collaborative design? How can one characterize engineering information models from a collaborative design point of view?

Traditionally, an engineering information model, sometime called an integrated product model, is viewed as a static data storage that contains

records of design decisions and plays a passive role during the design campaign. Can this static and passive role be adapted for collaborative design? If not, can an engineering information model be seen as a communication and coordination medium, rather than a data storage house, for stakeholders to share and discourse their viewpoints and decisions in collaborative design? What are the generic functions and characteristics of this communication/coordination medium?

How much information needs to be shared? When does it need to be shared? Why does it need to be shared? Can there be too much information shared?

The amount of information to be shared is generally determined by the stakeholders involved, and the nature of design and its requirements at hands. Are there some systematic methods to determine this information-sharing requirement before and during collaborative design? The particular design process model will further determine when information needs to be shared that the stakeholders are adapting. In studying various proposed design process models, are ways available to evaluate their resulting information sharing requirements? If too much information sharing is required, then the resulting model will be either too cumbersome or expensive to be practical.

How do information sharing requirements vary across the design life-cycle? What is the relationship between information sharing requirements of different life-cycle roles?

This study assumes that, at different stages of a design life cycle, the types, amounts, and nature of information to be shared will be different. It is important to study and correlate these different information sharing requirements with respect to different design life-cycle considerations. Such an understanding will enable the development of an information architecture that is supportive of design tasks/activities at different stages of collaborative design. It will also enable the development of a consistent shared model that will meet all the life-cycle requirements in design. Please note that the focus here is on the “information sharing requirements” rather than “information requirements.” The former is collaborative and interactive; the latter is individual and consumptive (i.e., used by the individual in the process).

Are there key concepts, central ideas, or “main things” that drive the information requirements of a stakeholder or “around which” the information requirements gravitate? Is there an “attractor” concept for a stakeholder role?

Information requirements are important not only for the development of a shared information model, but also for enabling individual stakeholders to make design decisions and “do their job.” What are the characteristics of information requirements for different stakeholders that differ from the information sharing requirements that enable them to actively participate in a collaborative design campaign? Will these requirements change as the stakeholders get more involved with the design (i.e., the difference between an experienced and inexperienced design team)?

What are the relationships between roles played by different stakeholders? Are they dynamically formed, or are there recurring roles that can be identified and stereotyped?

Information requirements for each stakeholder must capture its domain expertise, as well as the particular viewpoints and perspectives toward the design problem and the community involved with this design campaign. Stakeholders play roles within a collaborative engineering design process; special education or experience enable a stakeholder to make particular decisions or contributions to the process. Engineering design process models defined roles within which stakeholders use or apply their expertise. Are there any roles that are common to all design process models? Are these roles unique to the model? What is the variation or adaptability of these roles in a particular engineering design campaign? This study assumes that, even within well-defined roles, every stakeholder makes the role “his own” by adapting or executing the role based on his knowledge and experience.

What are types and granularity of information needed for generating computational models of the individual and collaborative design processes?

It is vital to understand the kinds of information needed and the level of granularity of the information required to have meaningful computational models of both the individual and collaborative design campaigns. Such a computational model will help to simulate the process and make them available for implementation and study.

2.6 A Brief History of These Research Investigations

Over the past 4 years, this research program can be divided into two major stages, each based on different premises and faced with different challenges. The first year began with value theory and the game theoretic approach from classical decision sciences. These conventional approaches were abandoned after their fundamental limitations in real world situations such as collaborative engineering became apparent. During the second year, efforts were directed to searching for a totally different paradigm as a new foundation for the research. (Initial ideas of the socio-technical framework were established during this time.) The third year was mainly devoted to the conceptual developments and system architecting of this new framework, detailing its modeling approach, overall structure, and key components. These results drove the fourth year's efforts, when the study implemented some initial prototypes to demonstrate some of the functionalities of this new framework. This section briefly reviews the research history in the program, and summarizes the important lessons learned.

2.6.1 The Value/Utility Theory and Multiple Objectives Decisionmaking

In searching for formal approaches to manage conflicts in collaborative design, as a way to gain a better understanding of collaborative activities, this study turned to classical decision sciences. If conflicts can result from multiple competing objectives, then classical decision sciences do provide some interesting approaches in multi-objective decisionmaking that have a rigorous theoretical and mathematical foundation. For example, the Decision-Based Design (DBD) approach proposed by Hazelrigg, is based on the von Neumann-Morgenstern (vN-M) utility theory that deals with the assessment of human values. Design is viewed as a decisionmaking process to maximize the value to humans, which is profit. The purpose of the DBD approach is to enable the assessment of a single value for every design option, for either individual or group settings, so that those design options can be rationally compared and a preferred choice taken (Hazelrigg 1996).

2.6.1.1 The Theoretical Bases and the DBD Approach

The well-known vN-M expected utility function was defined as (Mas-Colell, Winnston, and Green 1995):

Suppose that amounts of money are denoted by the continuous variable x . A monetary lottery can be described by means of a cumulative distribution function: F . That is, for any x , $F(x)$ is the probability that the re-

alized payoff is less than or equal to x . If the distribution function of a lottery has a density of $f(\bullet)$ associated with it, then:

$$F(x) = \int_{-\infty}^x f(t)dt$$

for all x . The advantage of a formalism based on distribution functions over one based on density functions, however, is that the first is completely general. Begin with a decisionmaker who has rational preferences $\succ$ defined of the option set'. The application of the expected utility theorem to outcomes defined by a continuous variable indicates that under the assumption of the theorem, there is an assignment of utility values $u(x)$ to non-negative amount of income with the property that any $F(\bullet)$ can be evaluated by a utility function $U(\bullet)$ of the form

$$U(F) = \int u(x)dF(x),$$

$U(\bullet)$ defined on lotteries, the $u(\bullet)$ defined on sure amount of money.

The vN-M utility is built upon this notion and the following six axioms:

1. All outcomes of a vN-M lottery can be ordered in terms of the decisionmakers' preferences, and that ordering is transitive.
2. Any compound lottery, that is, any lottery that has an outcome of another lottery, can be reduced to a simple lottery that has among its outcomes all the outcomes of the compound lottery with their associated probabilities of occurrence.
3. If the outcomes of a lottery, $u_1, u_2, \dots, u_r$, are ordered from most to least desired, then there exists a number p , such that one is indifferent between an outcome u_i , and $[p_i u_1, (1 - p_i) u_r]$.
4. For any lottery such as that given in axiom 3, with p_i specified, there exists an outcome $[p_i u_1, (1 - p_i) u_r]$ that can be substituted for u_i , and the preferences of the decisionmaker will remain unchanged.
5. The decisionmakers' preferences and indifferences among lotteries are transitive.
6. Given two lotteries, each with only two outcomes, and which differ only in terms of the probabilities of the outcomes, the lottery in which the probability of the more desired outcome is larger is the preferred lottery.

Based on the above axioms, the DBD approach was developed (Hazelrigg 1999). With this approach, a design begins with an option set, consisting of all the configurations. A parametric design vector represents each configuration, which is a designer's representation of the system. The design configuration is further

represented in terms of attributes to be recognized by the customers. Then the demand for a configuration can be generated, which gives the expected revenue of the particular configuration. The configuration vector and the exogenous variables also determine the expected costs consisting of all that could detract from revenues to result in net revenues or profits. Then the expected utility could be deduced from the expected revenue and the cost. By changing the design vector, the decisionmaker can optimize the design with respect to their expected utility. This approach viewed design as a utility maximization process and set the sole design goal as to make more profit.

2.6.1.2 Why the DBD Approach Does Not Work for Collaborative Design

The DBD approach offers a very clean and elegant way of modeling design decisions, and has a sound theoretical foundation in decision sciences. Furthermore, at a conceptual level, the approach should work equally well for both individual and group design scenarios. In fact, as long as a single utility function can be obtained, the design can pursue regardless of the number of designers involved. However, after careful studies of the nature of collaborative design and how individual decisions could change in the group settings, this study quickly came to the conclusion that the DBD approach, although systematic and rigorous, is not adequate to serve as a foundation for collaborative design.

Collaborative design is conducted by a group of stakeholders with different goals. There are typically two kinds of goals in collaborative design: individual and organization goals. Furthermore, customers, managers, design engineers and manufacturing engineers have different perspectives. The customers present their preferences about the product, and wish the company to provide good quality goods with the lowest price. The company owners also have an important voice in the design process, such as the desire to make more profits. The design engineers might view functionality more importantly than the safety and usability concerns. Even within the same kind of technical roles, the stakeholders may exhibit different behavior for the knowledge and personality differences. The design group as an organization also has its goals. The goals of the design group cannot be simply conceived of as a derivative or aggregation of the individual goals of the stakeholders. They are more complex because of cultural and emotional influences. In the design process, individuals influence each other and change their perspectives through social interactions. The goals, contents, and contexts of the stakeholders keep on evolving.

Although the stakeholders are supposed to make decisions to satisfy their individual and organization goals, they often face obstacles to make the right decisions—for various reasons. First, the goals of the individuals are not always well

defined. In the early stages of a new design, this happens frequently because the concepts of the product had not been clearly defined. Second, the norms, rules, and the coherent culture of the organization will often force individuals to adjust their goals to conform to the organization goals. That causes the change of the preference of the decisionmaker. Besides, the preferences of the individuals and organizations are not always easily described and compared quantitatively. The utilities of the designers and the organization are not always clear enough to be optimized. Also, the outcomes of the design are manifold. Collaborative design not only provides the form and structure of the product, but also influences the design environment. The stakeholders consider various issues related to the outcomes of the design. Therefore, it is impossible for the stakeholders to depend only on a single preference or utility function when making the decisions. In practice, stakeholders are more intent to select satisfactory options, rather than to maximize their utilities.

Therefore, in conclusion, approaches that are based upon the vN-M type of value/utility theories are too restrictive and inadequate for collaborative design because:

- The preferences of the stakeholders are not static.
- The preferences of the stakeholders are not always transitive.
- The preferences of the stakeholders are difficult to be formed and compared quantitatively.
- The sure amount of income ($u(\bullet)$ in the vN-M utility function) and the probability distribution ($F(\bullet)$) are difficult to evaluate in practice. A typical utility function is usually derived on the basis of behavior in certain given, and usually static, circumstances. In collaborative design, however, the interaction between stakeholders is dynamic and evolving and the determination of a utility function even for the design of a specific component by a group of stakeholders may be difficult to obtain, if such a unique utility function exists at all.

2.6.2 The Game Theoretic Approach and Important Lessons Learned

Another approach from decision sciences that seems to have a potential to become a foundation for collaborative engineering is the game theoretic approach that is commonly used to model group decisions under competing objectives. Most importantly, the game theoretic approach has very sound mathematical roots, which, upon casting a problem as a game, enable systematic modeling, simulation, and optimization of group decision strategies and outcomes. Therefore, significant efforts at the beginning of this research program were devoted to investigate and adapt this rigorous approach to the engineering domains. As will be explained below, this approach was found to be too limited in dealing

with the dynamic and complex nature of human decisionmakers within a social setting.

2.6.2.1 The Theoretical Bases and Mechanism Design in Game Theory

The game theoretic approaches begin with an abstraction of real-life situations into an interactive game-playing scenario. Three key ingredients of a game must be clearly identified during the abstraction process: players, strategies, and pay-offs. The abstractness allows them to be used to study a wide range of decision strategies and solutions. The basic assumptions that underlie the game theory are that decisionmakers pursue well-defined exogenous objectives (i.e., they are rational), and take into account their knowledge or expectations of other decisionmakers' behaviors (i.e., they reason strategically). A game is a description of strategic interaction that includes the constraints on the actions that the players "can" take and the players' interests, but do not specify the actions the players "do" take (Osborne and Rubenstein 1994). A solution is a systematic description of the outcomes that may emerge in a family of games.

For instance, A mixed-strategy profile σ^* is a *Nash equilibrium* if, for all players i , $u_i(\sigma_i, \sigma_{-i}^*) \geq u_i(s_i, \sigma_{-i}^*)$ for all s_i . Nash equilibrium is strict if each player has a unique best response to his rivals' strategies:

$$\text{minimize } f(X_p, X_s) = \{f_1(X_p, X_s), \dots, f_n(X_p, X_s)\}$$

There are different groups of game theory models based on the following three dimensions:

1. *Noncooperative and Cooperative Games*: In all game theory models, the basic entity is a player. A player may be interpreted as an individual or as a group of individuals making a decision. There are two types of model: those in which the sets of possible actions of individual players are primitive, and those in which the set of possible joint actions of groups of players are primitives.
2. *Strategic Games and Extensive Games (i.e., static or dynamic games)*: A strategic game is a model of a situation in which each player chooses his plan of action once and for all, and all players' decisions are made simultaneously. By contrast, it is the model of an extensive game that specifies the possible orders of events.
3. *Games with Perfect and Imperfect Information*: In the first, the participants are fully informed about each others' moves, while in the second model, they may be imperfectly informed.

The current applications of game theories to engineering design focus mainly on the noncooperative and static games with perfect information. By viewing multi-objective decisionmaking as game playing, some researchers have tried to solve engineering problems in the parametric design stages. However, most of them are to derive the Pareto-optimal sets for design variables by using weighting, min-max, or goal-programming methods (Grandhi and Bharatram 1993; Lewis and Mistree 1997; Rao and Freiheit 1991). Furthermore, these approaches ask for players and strategies as inputs, and predict payoffs as outputs in modeling engineering design problems, which is not adequate for the research goal of finding conflict management strategies for collaborative design.

After studying the game theories and those existing approaches in engineering design, this study sought to formulate its research as *a mechanism design problem* in the game theory. Designers are assigned as game players, conflict management mechanisms as game strategies, and cost and quality of design results as game payoffs. In this way, a collaborative design problem could be formulated as a mechanism design problem in game theory as:

How to design a game (i.e., design processes), so that, when it is played, the equilibrium strategy (i.e., conflict management strategy) for the game is guaranteed to satisfy certain properties (i.e., minimal cost/time, maximum quality, etc.).

Based on the above formulation, researchers conducted many modeling tasks to experiment with different collaborative design scenarios drawn from facility engineering domains (Lu 1998b). Soon the conclusion was reached that this mechanism design approach in specific, and game theory in general, were too limited in modeling the real-life collaborative engineering activities. These game theory based formulations require much simplification of real-world situations, which defeats the purposes of this study. Furthermore, many of the required inputs for these modeling approaches are very hard to quantify in the real world, leaving the modeling tasks cumbersome and nonrealistic. Limitations of the game theoretic approach are further explained below.

2.6.2.2 Why Game Theoretic Approach Does Not Work for Collaborative Design

The Game theoretic approaches pay special attention to the interactions of the utilities of the players, and try to optimize the system outcomes by reaching the game equilibrium. They take a different viewpoint of engineering design than those approaches based on the value/utility theory. In some limited cases, game theoretic approaches can work well if technical design objectives about the parameters of the product can be qualitatively treated as utility payoffs in the game. When the equilibrium points are reached during the game playing, the

design is complete and the product is defined. These approaches might find some useful applications when optimization problems are the key concerns in design (Rao and Freiheit 1991; Vincent 1983). Some researchers had long realized the limitations of Game theory in solving the conflict problems, and concluded that the conflict resolution strategies based on the game theoretic approaches could only have limited contributions (Binmore 1987). In real-life collaborative design situations, these limitations are profound.

The first problem with these approaches is the strong requirement for abstractions. Many of the established abstraction methodologies in game theory are based on the assumptions in the economics and social science domains, rather than for the more technical disciplines such as engineering. When applying the game theory to complex engineering problems, the abstractions under those traditional assumptions generate large amounts of information losses. Often, the real-life situations simply cannot satisfy these abstraction assumptions. For instance, in noncooperative games defined in classical game theory, the players always intend to increase their own payoff function, while in collaborative design, team members should consider the utility of the whole system, and hence are easier to “agree.” Many of the game theory models are based on the assumption of “imperfect information” due to the selfish nature of game players. However, the imperfect information situations in collaborative engineering often result from communication delays and the distributed nature of knowledge, rather than selfishness. Furthermore, in the coalition game, the players in a team often care more about their own gains than that of the whole team, which is not the case in real life collaborative engineering scenarios.

Secondly, due to individual designers’ limited knowledge about the overall nature and scope of the group design activities, a precise description of the game is often not available. In real-life collaborative design situations, it is common for most decisions to be highly coupled and hard to depict. When game theory is used as a modeling tool in collaborative design, most of the design tasks become a matter of building utility functions and searching for the game strategies, rather than conducting real design. In reality, designers’ ability to always conceptualize their design tasks into the game playing situations is limited. It is simply not a natural way for designers to think about their tasks and professions.

Thirdly, the options of decisionmakers, which are mapping to the strategies of game players, are not always quantifiable. For example, some key issues of design lifecycle, such as the customer interests and the functional feasibility, cannot be quantified. This is also why the majority of current game theory applications in engineering design are limited to the optimizations of design

parameters, rather than the management of design conflicts that is the essence of collaborative design. Furthermore, even if all of the preferences of decisions could be quantified, the myriad variables in large engineering systems are prohibitively beyond the existing computation ability of the human and computer.

2.6.3 The Socio-Technical Framework as a Collaborative Engineering Paradigm

This study still views collaborative design as a multi-objective decisionmaking process. Depending on the nature of the design problem and the belief of the model builder, various models and methodologies of decisionmaking exist. Traditional economic models assume that the players are totally rational and construct strict reasoning of their methodologies. The value theory and game theory are in this category. Some social scientists and psychologists focus on the real behavior of humans and build the social model of psychology. Between these two extreme viewpoints, there are Bounded rationality model and Judgment heuristic and biases model. From this point of view, collaborative design is a fundamental activity in human endeavors in general, and in engineering in particular. Thus, only to view design decisionmaking from one perspective is not adequate. The value theory and game theory approaches have their restrictions.

In this study, researchers came to realize that, in the collaborative design, decisionmakers end up *satisfying* since they do not have the ability to maximize. There are various social and technical obstacles, which prevent maximization in practice. Herbert Simon proposed the bounded rationality model to present a more realistic alternative to the economic models, something that is more applicable to a lot of the collaborative design situations, which the pure rational approaches had ignored. In his model, the decisionmakers' behavior could best be described as follows (Simon 1976; March 1978):

In choosing between alternatives, managers attempt to *satisfy*, or look for the one, which is satisfactory or "good enough." They recognize that the world they perceive is a drastically simplified model of the real world. They are content with this simplification because they believe the real world is mostly empty anyway. Because they satisfy rather than maximize, they can make their choices without first determining all possible behavior alternatives and without ascertaining that these are in fact all the alternatives. Because they treat the world as rather empty, they are able to make decisions with relatively simple rules of thumb or tricks of the trade or from force of habit. These techniques do not make impossible demands upon their capacity for thought.

However, the pure rational decisionmaking models still have their applicability. In the situations where most of the required information is available and the prescriptive rules are approximately to be conformed, the strategies and methodologies can be applied. That used to happen in the later optimization stages of design. In the real design problems, it is unfair to say which model is the dominant one at early stages. That is caused by the complicated attributes of the collaborative design process. Different stakeholder will have different rationale during different design stages. Their performances during the decisionmaking process will influence not only the final design products, but also others' performance.

3 A Socio-Technical Framework For Collaborative Engineering Activities

This chapter presents the basic framework for this research in collaborative engineering. To focus these investigations, *collaborative design* activities were targeted as a main application domain. This is because design is a fundamental activity in human endeavors in general, and in the engineering profession in particular. The scope, scale, and complexity of engineering design projects has dramatically expanded over the last 3 decades, making design projects inherently multi-disciplinary; they touch on knowledge-bases and human experience from a wide variety of diverse subject areas. The IT revolution has engendered new opportunities for handling complex design projects by enabling experts to collaborate across geographic and temporal boundaries across the globe. A question of great importance is: *What kinds of collaborative design processes will make the final product more than just the sum of the individual contributions?*

In the field of engineering design research, there are almost as many approaches as there are researchers. For example, the axiomatic design process model (Suh 1990) visualizes the design activity as a series of mappings that start at the customer domain and end up in the creation of a product, zigzagging its way across the functional domain, the design parameter domain, and the manufacturing process domain. Pahl and Beitz (1996) model the engineering design process through a structured flowchart. Others (Hazelrigg 1997; Mistree and Allen 1997) see the engineering design process as a decisionmaking process where the engineer is required to make appropriate decisions from various options using well known constructs from game theory and decision analysis under uncertainty to underpin the strategies for decisionmaking. Still others have modeled the engineering design process as a knowledge-based problem-solving process, which is intrinsically no different from any other goal-driven activity (Yoshikawa 1981).

Despite all the research, no unified framework for modeling the design process has emerged. This problem is particularly serious when dealing with collaborative design activities that have become a norm of the engineering profession. One might argue that perhaps the applicability of each of the design process models is contingent upon a particular set of prevalent circumstances; yet none of the models explicitly describe what these contingent circumstances might be.

In fact, each proponent of a different design process model usually claims his/her model to be the most useful, and in some sense the best for doing engineering design. These divergent approaches for addressing the engineering design problem motivated this study to concentrate on a more primitive phase of inquiry. This work hopes to develop a more comprehensive understanding of the design problem that will help provide new directions for design process models, and indicate the effectiveness of the models already developed. The goal here is to understand the problem more clearly, before offering yet another particular solution.

This chapter presents a new framework to describe the problem, not the solution, of collaborative engineering design. Current research approaches view engineering design as a series of isolated, technical tasks, without treating the human side of collaborative activities. In contrast, this study is based on the belief that engineering design is a human-based social activity, and must be supported as such. The interest of this work has been to develop a better understanding of the design problem as a set of unified, socio-technical tasks where human perspectives in collaboration are explicitly modeled. This expanded and fundamentally different viewpoint toward the engineering design problem is where this research departs from traditional thinking.

3.1 The Social Construction Theory

Although the “people factor” is often excluded from “hard” engineering research, many of the most difficult issues associated with collaborative engineering design are directly related to the humans involved in the process and the extent/variability of the knowledge that they bring to a design campaign. When a group of humans engage in a design campaign collaboratively, the “people-factor” enlarges into a “social-factor.” Therefore, what is needed is a basis or foundation for dealing with the “people” or “social” factors in engineering design and the technology required to support it.

Berger and Luckmann (1966) presented a treatise on the sociology of knowledge called the *Social Construction of Reality*. Social Construction Theory asserts that *meaning and institutions are a joint, negotiated, and agreed construction of those participating in an endeavor*. Each participant, called a “stakeholder,” will use their own meaning (i.e., understandings of the world, or “the local reality”) as a basis for their social interaction, communication, and learning within the institution. As the social construction process begins, participants use their local realities to construct a cohesive, institutional reality (i.e., a shared reality), through communication and negotiation. As the process progresses, interaction

among the stakeholders (i.e., the “who’s”) causes learning (i.e., modifications and updates to their local realities), which prompts modifications to both the shared social and technical processes relevant to the engineering design campaign.

It is not the intent of this study to pursue research in the social sciences, nor is it to question or delve into the details of social construction theory. Given the fact that the “*who*” is such an important element of the research thrust described above, what this research program has done is to establish a theoretical basis from this social science, to treat and examine the role of the “who” in engineering design, in conflict management, and in the design of information sharing technology to support collaborative engineering activities.

Social construction is a social science theme that asserts that meaning (i.e., reality) is a joint, negotiated, and agreed construction of those participating in an endeavor (such as engineering design). Krogstie, Lindland, and Sindre (1995) have looked upon conceptual modeling as a social construction process. They use “conceptual modeling” to refer to models constructed in the design of information systems. Data models and database schema are a variety of these models. They describe social construction as follows:

An organization will consist of individual actors that see the world in a way specific to them. The *local reality* is the way the individual perceives the world that s/he acts in... When the social actors of an organization act, they *externalize* their local reality. The most important ways the social actors of an organization externalize their internal realities are to speak and to construct languages, artifact, and institution. What they do is construct *organizational reality* that may consist of different things, for instance conceptual models and computerized information systems. Finally, *internalization* is the process of making sense out of the external actions, institutions, artifacts, etc. in the organization, and making this organizational reality part of the individual local reality.

Figure 5 gives a high-level conceptual illustration of the social construction process according to Berger and Luckmann.

If design can be considered as a model-building process, then Krogstie’s above analogy can be extended to *view collaborative engineering design as a process of social construction*. The perspective of a stakeholder can be equated with his “local reality,” and the result of the engineering design campaign can be equated with the “organizational reality.” As will be explained below, the viewpoint of viewing collaborative design as a social construction process sets the basis for a socio-technical framework for collaborative engineering.

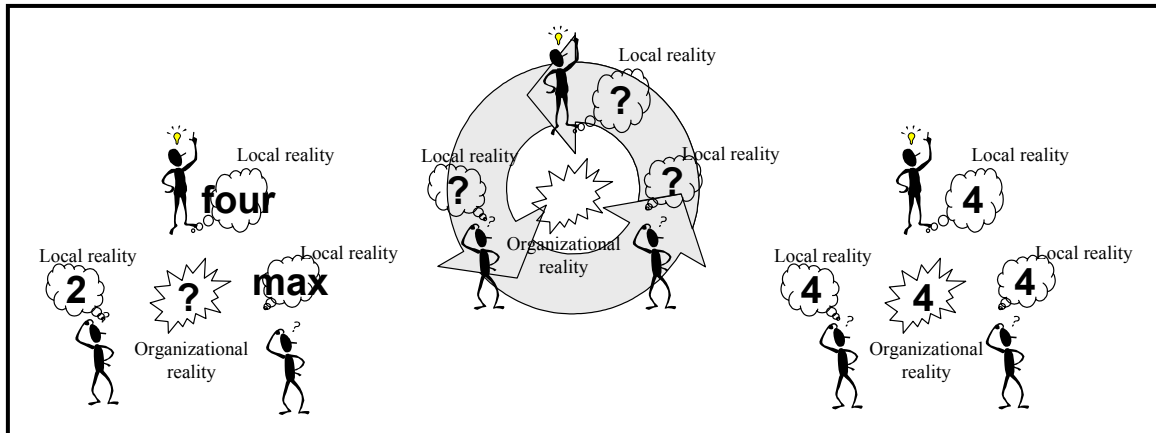


Figure 5. A conceptual illustration of social construction process.

The significance of this new view of the collaborative engineering design process is that it greatly balances the need to consider procedures, statements of requirements, and laws of nature in the engineering design process by reducing the process to language, negotiation, and the common objective of creating the product model that is an “organizational reality.” This view does not eliminate the utility or need for the discipline and methodology provided by traditional approaches such as quality function deployment (QFD) or total quality management (TQM), but rather provides a different theoretical framework or basis for their applications.

3.2 The Basics of the Socio-Technical Framework

Engineering design means different things for different people. As a discipline that engages a significant number of individuals with different technical backgrounds, the preponderant approach to understanding the design process so far has been related to concepts like optimal design, correctness, consistency, and truth. The design activity is often considered a totally rational procedure undertaken by supposed automatons devoid of incentives, motivations, emotions, and polity. This traditional viewpoint has limited applicability when a team of designers is present. Various project and data management techniques have been developed with limited successes, because they treat collaborative activities as merely workflow or data integration problems without explicitly recognizing the roles of humans involved.

This study is based on the belief that collaborative engineering design must be treated as a *human system* that accounts for the unique knowledge and individual goals of the participants in the process. Therefore, this work has developed a Socio-Technical Framework for Collaborative Engineering that views collabora-

tive design as *a technical co-construction process in which a set of stakeholders, working within a design environment, undertakes a design campaign*. To clearly explain what is meant by this, the following basic concepts are defined:

- *Stakeholder*. An individual, groups of individuals, or any entities, with an interest, or possible interests, in the outcomes of the design campaign and/or of the design environment.
- *Perspective*. A collection of information that is relevant to a purpose or a goal of a stakeholder, and which acts as a “lens” or “filter” through which a stakeholder produces and consumes information. A stakeholder has many perspectives on, or relative to, an endeavor in which he participates.
- *Design Campaign*. A set of goals, decisions, processes and actions, which lead to a design model (i.e., the specification of the final product suitable for production in the case of product design), which meets the life-cycle requirements of a product.
- *Conflicts and Conflict Management*. A *conflict* is a situation in which viewpoints/perspectives and/or decisions from the same and/or different stakeholder(s) become mutually incompatible with respect to the satisfaction of some design requirements. *Conflict management* is those techniques and methods employed to detect, resolve, and prevent the occurrence of conflicts.
- *Design Environment*. The sum of technical and nontechnical infrastructure within which a design campaign is immersed. This includes, for example, physical plant, lines of communication, command and control within the organization, corporate cultures, pertinent design codes, etc.
- *Model*. A set of statements or specifications in an agreed upon language, which represents a real-world phenomenon/artifact and can be used to illustrate, explain, understand, evaluate, record, predict or control that phenomenon/artifact.

Each stakeholder has his/her own set of perspectives, which, at the very least, must include: (a) himself/herself, (b) the organizational aspects of the environment he perceives he is operating within, and (c) his perception of the perspectives of the other stakeholders within the environment. Each of these three items above includes aspects related to technical viewpoints, managerial viewpoints, and social-interaction viewpoints.

The technical co-construction done by these stakeholders, which underlies the design activity in any design campaign, necessitates a constant negotiation and *adaptive evolution* of the design goals, the design processes, and the quality metrics. In fact, it leads to an evolution of the very *meanings* of these concepts as a convergence gradually emerges toward a consensual validation (in which all the stakeholders participate) of a perceived reality. Thus, one obtains a more *integrated view* of the design goals, the design processes to be employed, and the quality metrics suitable to measure the goals and the processes.

The framework proposed herein is an overarching one, which includes the fragmentary concepts of design process models that have hereto been proposed by many schools of thought. The next section elaborates on the key aspects of this framework and expose its various significant facets.

3.2.1 Key Aspects of the Socio-Technical Framework

Some key aspects of this framework are explained in this section to present a better view of the different viewpoints elicited in this study.

3.2.1.1 The Socio-Technical Process

The socio-technical view of engineering design suggests that it is more than just a technical process carried out by a group of people. Rather, it is a *social exchange* process through which a group of people, called the *stakeholders*, interacts on technical decisions within a particular design environment that is constantly evolving. Each stakeholder has a unique *perspective* within the engineering design campaign, and the interaction of stakeholders creates a single cohesive perspective for all. “Co-construction” involves creation of a simultaneous vision of the goals of a design campaign, the design process to be used, and the quality metrics to be utilized to measure the degree of compliance with the goals and the process under consideration.

The co-construction of a consensually validated reality is an evolutionary and collaborative negotiation process in which conflict management becomes an essential and central facet. Conflict management refers to the detection, prevention, regulation, control, and even fostering, of conflicts. Supporting conflict management as a central activity in collaborative design calls for a fundamentally different approach toward engineering design research. Unlike traditional approaches, which view engineering design as a series of *technical* tasks—be they decisionmaking, or mapping, or execution of flow-charted procedures—that are organized so as to avoid and eliminate conflicts, the present framework presents collaborative engineering design as a *socio-technical process* during which conflicts drive the interactions, promote innovations, and eventually lead to a confluent vision of the design campaign. This “socio-technical process” view toward collaborative design is the cornerstone of the proposed framework.

One of the eventual goals of a collaborative design campaign is the adaptive evolution of the *design environment* in an organization. The development of a suitable design process or an end product then becomes but one element of the overall socio-technical view of the design activity. A crucial feed-back sought from any specific design campaign is not merely how better to design similar products

in the future but how co-construction affects and may be directed at the evolution of the design environment, enabling it to more effectively meet the varied and specialized demands of tomorrow in an adaptive manner.

3.2.1.2 The Stakeholders and “Who’s” in Collaborative Design

Being a socio-technical process, the design environment and any specific collaborative design campaign is necessarily made up of stakeholders; this framework thus requires a proper understanding of how these stakeholders interact. One might imagine the stakeholders (i.e., who’s) as the evolving “atoms” making up collaborative engineering design. While others may view design as a disembodied mapping from “*whats-to-hows-to-whats*” (Suh 1990), from the viewpoint of this work, the “who” aspect in collaborative design becomes a fundamental, and perhaps the most important, consideration in a particular design campaign. Thus the modeling of a collaborative design process is not complete without an explicit treatment of the “who” within the design team(s). It is the “who’s” that cause and create the co-construction. For instance, it is the “who” that chooses the product requirements; the “who” that sets the schedule; the “who” that makes decisions that cause or eliminate conflicts during design; and, the “who” that must agree upon all design decisions for the final design results to be acceptable. In terms of the constructs of axiomatic design, it is the “who” that decides what is a “what,” it is the “who” that decides what is a “how,” it is the “who” that maps the “what” to a “how,” and it is the “who” that decides who is to be the “who” in these decisions.

To make a contribution to the design process, the stakeholder must *receive*, *consume*, and *produce* information—about the design goals, about the decisions made by other stakeholders, and about management plans and objectives, for example. From the standpoint of an individual stakeholder, engineering design is a process of continuous information consumption, sharing, and production. To meet his responsibilities in the campaign, the stakeholder requires a set or collection of information (i.e., he has certain *information requirements*), which varies continuously throughout the duration of the engineering design campaign.

It is important to note that the information produced, shared, and consumed by an engineer is filtered and processed with respect to the engineer’s *perspective*. A *perspective* provides the background and context with which an engineer understands and interprets received data and affects how he conceives meaningful information from the input. Conversely, a perspective affects how an engineer understands a problem, makes decisions, and produces information within the design campaign.

3.2.1.3 The Perspectives of Stakeholders

To model the stakeholders, the notion of *perspective* of the stakeholders involved in an engineering design campaign is introduced and defined. A perspective is the combination of:

1. A *purpose* with which the stakeholder participates in an engineering design campaign
2. A *context* within which the stakeholder participates in an engineering design campaign
3. A *content* relevant to the purpose and context, i.e., the experiences, background, education, knowledge, and insight that the stakeholder brings to bear within an engineering design campaign.

The notion of *context* and *content* roughly correspond to the linguistic concepts of *pragmatics* and *semantics* (Crabtree and Powers 1991). *Syntax* is another element of the linguistic view of language dealing with the structure and grammar of sentences; this work assumes that syntactic correctness of a communication event (i.e., act of sharing information) is a given. *Purpose* does not have a direct linguistic correspondence, but is closer to the psychological/philosophical notion of *intention* as an action-controlling mental state, i.e., having the intention to do something controls, affects, or influences future behavior of the person with the intention (Bratman 1990). Concepts similar to “perspective” in existing literature include: province of meaning (Berger and Luckman 1966), universe of discourse (van Griethuyesen 1982), and relevant universe (Parsons and Wand 1997).

A stakeholder’s perspectives—as described by purpose, context, and content—are relevant from three different viewpoints when considering his/her contribution to a collaborative design campaign: his/her technical viewpoint, his/her managerial viewpoint, and his/her social-interaction viewpoint. Furthermore, each stakeholder interacts with other stakeholders based on his/her perspective of himself/herself, of his/her organization, and of the other stakeholders involved in the campaign.

Part of the legitimization of institutions during co-construction is the emergence of roles fulfilled by actors (stakeholders) in an organization. A role is a set of responsibilities associated with a functional purpose or goal that must be met or executed by an agent in a process. In general, execution of the role contributes to the completion of the process. At the beginning of a design campaign, stakeholder roles may be stereotypically defined through multiple executions of past processes, or they may be prescribed in a process. The roles to be dynamically

identified, adapted, and validated through co-construction are visualized during the design campaign.

The stakeholder may be thought of as an entity whose inputs are competence, belief systems, and constraints, and whose outputs in the design campaign are concerns, commitments, and the triad of goals, procedures, and quality metrics. The meaning and impact of both the inputs to, and outputs from, each stakeholder is co-constructed through interaction with other stakeholders during the design campaign. This co-construction influences and is influenced by the design environment.

3.2.1.4 Social Construction Among Stakeholders

Recognizing that collaborative design is a socio-technical process, and the importance of “who” during this process, the next task is to understand how a group of “who’s” interact with each other to achieve the goals of a design campaign. The basis for the introduction of “who” into the research into collaborative engineering design is the Social Construction Theory (Berger and Luckman 1966). As explained before, this theory asserts that *meaning and institutions is a joint, negotiated and agreed construction of those participating in an endeavor*. Each participant (stakeholder) will use their own meaning (i.e., understandings of the world, or “the local reality”) as a basis for social interaction, exchange, contribution, and learning within the institution. As the social construction process begins, participants use their local realities to construct a cohesive, shared reality (i.e., an institutional reality), through communication and negotiation. As the process progresses, interaction among the “who’s” causes learning (i.e., modifications to their local realities), which in turn prompts modifications to both the shared social and technical processes relevant to the design campaign and the design environment. This highly interactive and collaborative process is defined here as “co-construction.”

The co-construction activity can be visualized as having three steps:

1. The “representation” of stakeholder perspectives
2. The “sharing” of these perspectives
3. The gradual “confluence” of these perspectives, through their *modification*, and through the *creation* of new mental constructs, brought about by stakeholder interaction.

It should be pointed out that co-construction is an activity that goes beyond the mere sharing of information and the consensual validation of its meaning. It

also goes beyond the notions of conflict management (which it includes). It deals with the *creation* of an integrated whole from the multiple perspectives of the individual stakeholders.

3.2.2 Collaborative Engineering Design as a Socio-Technical Process

If social construction theory is interpreted within the context of an engineering design campaign as defined above, the following corresponding concepts apply:

- co-construction process → a collaborative engineering design campaign
- participants → all stakeholders involved in or associated with the design campaign
- local realities → each stakeholder's personal knowledge, belief, and purpose (i.e., their *perspectives*)
- shared reality → goals, design process, quality metrics for a design campaign; team culture, behavioral norms, common understanding, organizational procedures, company policies relevant to the overall design environment
- co-construction results → the agreed design outcomes (e.g., an integrated product model), new engineering design procedures, evolution of the design environment.

The above social construction view goes beyond the systematic codification of engineering design; it recognizes that any codification, classification, or documentation of institutions becomes stale over time because stakeholders *learn* and requirements *change*. The co-construction process is continuous; it happens *all the time*. The importance of this position is that while a particular systematic design process model may serve as a basis for an engineering design campaign to start with, it is always dynamically adapted and modified by the participants during the course of the campaign.

3.2.3 Comparison of This New Framework with Traditional Viewpoints

Having introduced the various facets of the framework, the next step is to consider how and where such a framework might take this research in obtaining an improved understanding of the collaborative design process (Table 1). The above comparisons by no means imply that the proposed framework is in conflict with other traditional design models. In fact, for most cases, this framework complements and extends other traditional approaches especially at early stages of design where social interactions play more important and visible roles.

Table 1. Socio-technical framework and traditional engineering design approaches.

Problem Characteristics	Traditional Design Framework	Socio-Technical Framework
Nature of engineering design	a) Purely technical activity b) Single-person or “equivalent” to a single-person activity c) Fully rational activity	a) Socio-technical human-based activity b) Multi-person group activity c) Bounded rationality
Knowledge bases needed to perform engineering design	Applied science, traditional engineering disciplines	Interdisciplinary: linguistics, computer science, management, political science, psychology
Assumptions about stakeholders	Technically oriented individuals	Need not be technically focused
Activity underlying the design process	Mappings of: what-to-how-to-what-to-	Co-construction by stakeholders
Attitude toward design conflicts	Conflicts are to be reduced or eliminated as soon as possible in the design process	Conflicts drive the design process. design conflicts are to be resolved, sustained, negotiated, and exploited, to generate fruitful co-construction.
Design quality metrics	Quality metrics for the design process/ product are often independent of process/product	Quality metrics, design process, and design goals, are simultaneously arrived at
Data structures for computational support	Values of various design variables, frs, etc., bit flows	Perspective models, data structures for co-construction, ‘computational conversation processing’
Design process model	Static design process once it has been chosen	Adaptively evolving design process
Outcomes of design process	Designed product	Designed product and feedback to design campaign and design environment

For example, traditional approaches view design activities as mapping operations between a set of “Whats” (i.e., the performance requirements) and “Hows” (i.e., the design specifications) with no regards to “Who” are doing these mappings. The approach here complements those traditional approaches by explicitly modeling the “Who’s” during “what to how” mapping operations, and enables dynamic evolutions of perspectives as those “Who’s” interact with each other during a collaborative design process.

3.2.4 New Implications and Insights from This Socio-Technical Framework

The framework calls for a more interdisciplinary approach for understanding collaborative engineering design. To illustrate the usefulness of the framework, three inter-related areas are addressed: conflict management and co-construction, design quality, and information sharing and data structures in collaborative engineering design. This framework not only provides new insights

into these three critical aspects of collaborative engineering design, but projects future needs, assuming that it will be possible to take advantage of the information/computation revolution to better support collaborative engineering design.

3.2.4.1 Conflict Management and Co-construction

When inconsistent local realities evolve and merge into a consistent global reality during a co-construction process, conflicts of various types at different abstraction levels occur among stakeholders. These conflicts could be of a technical nature, a managerial nature, or of a social interaction nature, and their effective handling plays a central role in determining the overall effectiveness of a design team. When treating engineering design as a purely technical process, conflicts are usually regarded as being abnormal, and to be avoided as soon as possible, at all costs. In the current framework, conflicts need to be systematically and explicitly dealt with as a *resource* to drive the co-construction process, and design innovations.

As gradual confluence occurs through co-construction, the design process becomes more and more driven by technical considerations allowing mental constructs such as mappings, decision analysis and game theoretic methods to provide useful ways of then developing and co-constructing design process models.

The co-construction framework requires a categorization of the different kinds of conflicts in terms of those arising from negotiations and co-constructions related to (a) goals, (b) processes, and (c) quality metrics. The aim would be to find mappings between the different types of conflicts that arise, and conflict management strategies that have been developed in the social, political, and organizational management literatures. It should be emphasized that conflict management may involve not just the detection, prevention, and resolution/extinction of conflict, but also the encouragement, sustenance, and control of conflict in a desired manner. Of great significance is the development of tools to measure and monitor the “rate” at which conflict resolution occurs so that confluence of viewpoints in the co-construction process can be achieved in a desired and controlled manner. As stated earlier, the concept of co-construction goes beyond simply conflict management. It calls for the development of goals, processes, and metrics to *create* new ideas through the interaction of stakeholder perspectives; it may thus be argued that quality is created through co-construction. This leads to the issue of redefining design quality.

3.2.4.2 The Design Quality

Conventional measurement of design quality is based on the degree of fulfillment of technical and functional requirements of the design results (e.g., Integrated Product Model) in terms of time, speed, and costs. That is, a value of a product performance characteristic is selected and the product quality is measured against that value, thus providing an objective and fixed measure of quality.

As this view of collaborative engineering design is expanded to include the socio-technical aspects, the treatment of *design quality* needs revision as well. There are some new ways in which design quality may be viewed based on this socio-technical view of collaborative engineering.

The first is through the view of Design as Co-Construction Management (DCM) presented here. One might simply begin by using conflicts as a measure of design quality: the degree of conflict needs to be controlled so as to achieve a desired and measurable rate. The degree may depend on both the number of conflicts and their perceived impact on the design campaign. Failure to meet performance objectives is a (kind of) conflict; decisions made by stakeholders in the design campaign did not produce a cohesive and consistent design result. This approach to design quality is based on the measurement of the *amount* or degree of conflicts at the end of the design. In contrast to this simple approach, the new framework indicates that it is possible to dynamically measure the team design quality by the *rate* by which the number of conflicts changes over the course of the engineering design campaign. At a point in time, each stakeholder's perspective (i.e., local reality) will conflict with the perspectives of other stakeholders in the campaign. The sum of these conflicts first rises, and then falls, as the individual perspectives are reconciled into a cohesive, global view. A high quality design team (i.e., a team with very similar/compatible perspectives) will have the ability to quickly bring the merits from different views of participants to converge onto a consistently co-constructed design result that meets all stakeholder requirements. Because this framework does not insist on conflict reductions immediately following their occurrence, the number of conflicts in the system can grow nonmonotonically during the co-construction process. Therefore, another important measure of the design quality in this research is the ability to follow a desired trajectory of changes in the amount and rate of design conflicts as the design campaign progresses.

Another view of design quality is obtained by looking at Engineering Design as a Social Co-construction process. This view complements the above DCM view of quality in that it asserts and affirms that *quality originates in the stakeholders*. Rather than being a direct measure of quality, this view of engineering design

fosters quality by rooting the design result in knowledge from, and interaction between, the stakeholders. The framework further implies that *quality is created through co-construction* and that it is the co-construction process that causes the “whole to be greater than the sum of its parts” in any specific design campaign. Hence any computer support that enhances the co-construction process will enhance design quality. This leads to the issue of information sharing and computer support for co-construction.

3.2.4.3 Information Sharing and Computer Support for Collaborative Design

The first and most apparent impact of this new framework on information technologies is that building a predefined, rigid meaning *into* application software and data systems dooms the system to a short life span in design practice. The reason for this is that the knowledge and perspectives of stakeholders change dynamically throughout an engineering design campaign, thus the needs of the stakeholder and the capabilities of the system slowly drift apart over time. Few systems are really able to *learn*, adapt and keep pace with continuous human knowledge acquisition and processing. In the framework proposed here, information technology serves as a *vehicle for conveyance of meaning* (i.e., information) between stakeholders; and *meaning* lies entirely in the hands of, and evolves through the control of the stakeholders.

A new way of thinking about the human/computer interactions is required here. One of the key aspects of computer support from this view of collaborative engineering design is the capture and use of stakeholder perspectives. A representation of perspectives of a stakeholder is referred to as a Perspective Model.

A Perspective Model is a representation of a stakeholder perspective that is deliberately built by the stakeholder to represent his/her “local reality” or is automatically constructed by his/her use/interaction with software systems. “Embryonic” perspective models already exist in the Preferences, User Profiles, and customization of commercial software applications that allow the user to make the software appear/ behave a certain way. A quantum step forward in the customization of information technology is thus required in allowing the user to construct his own view of “what’s going on” in an engineering design campaign (i.e., his local reality), and adaptively change it during interaction with other stakeholders.

Current attempts at data management rely on methods that aim to provide consistent data (i.e., noncontradictory data), and those that update information between local databases and global databases, often through a central server. This framework requires the development of databases and data management strate-

gies that enable the dynamic co-construction of the *meaning* of the data, based on the perspectives of the users of the data. Co-construction will, in general, elicit contradictions and ambiguities in data; the development of methodologies (and computer support systems), which engender gradual confluence, and call for creation of meaning. At present they apparently do not exist.

3.2.4.4 The End Products of a Collaborative Design Campaign

Perhaps the most important difference in the viewpoint of engineering design expressed here from those found in the literature is the eventual aim of a design campaign. In this framework, collaborative design activity is *threefold*. First, it aims to design the product in a cost effective and timely manner while meeting the product requirements. Second, it explicitly provides feedback for evolution of the collaborative design process itself. Third, it provides explicit feedback for the adaptive evolution of *the design environment*. The development of a suitable design process, or a suitable end product in a specific design campaign then becomes but one element of the overall socio-technical view of the design outcome. An important feedback this study seeks from any specific design campaign is not merely how to better design *specific products* in the future, but how to evolutionarily co-construct the *design environment* so that it can meet the varied and specialized demands of tomorrow in an adaptive manner. It is this over-arching framework of technical co-construction as a view of engineering design, which encompasses those presented earlier and holds the key to furthering a better understanding of design.

Figure 6 shows a high-level view of the traditional approach to engineering design. One can refer to this model as a technical construction model where the knowledge and process domains are melded together at the instigation of a design objective to yield a specific design product. The process model that is selected and adopted is usually static; the stakeholders are required, by fiat, to adhere to it, and are not explicitly modeled.

3.2.5 Some Basic Research Questions Exposed by This New Framework

The central purpose of building a new framework for a better understanding of collaborative engineering design problem is to explore new areas of design research. For any new framework to be useful, not only must it explain known observations and expose new ones, but it must expose new questions as well.

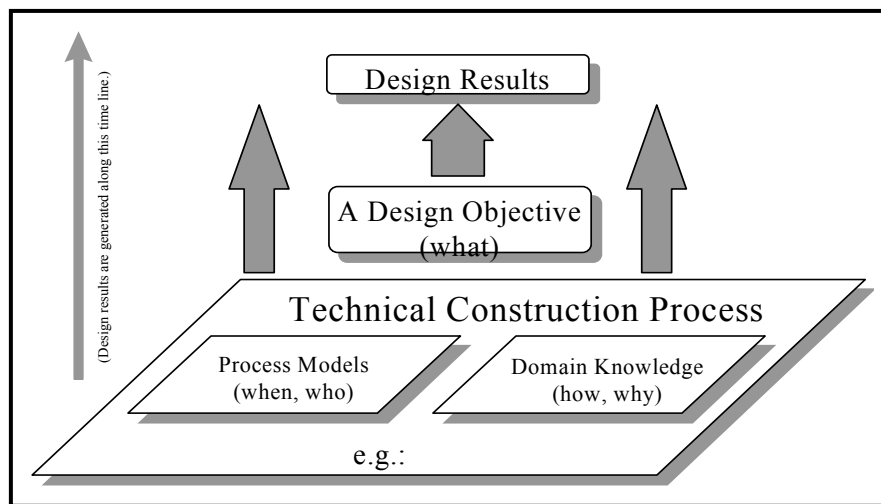


Figure 6. A high-level view of traditional engineering design process.

A few research questions that the framework points to follow.

1. Assuming that collaborative engineering design is a socio-technical process, is it possible to have a feasible computational model of the design process? At what level of granularity can one develop useful and practical computational models of collaborative design?

Clearly, the fineness of the elements of this computational model will dictate the extent to which it will be possible to simulate, and possibly predict, the outcomes of the design process. At the right level of granularity, it may perhaps be possible to perform a sort of rapid prototyping of alternative design processes to circumscribe a set that might be appropriate for a particular design campaign in a particular design environment.

2. What are the types of computational support systems needed for perspective modeling, updating, and co-construction of “shared realities”?

At the very least, this framework calls for new information technologies for the capture of stakeholder perspectives. The fields of theoretical linguistics, semantics, and philosophy would be useful here to assess the types of data structures that might be useful and that need development.

3. Are there generic categories of conflict that arise in collaborative engineering design? Are there applicable taxonomies of conflict management strategies?

A cross-pollination of knowledge from areas such as political science, management science, and international relations would be of great value here because each of these knowledge bases deals with conflict management. However, the demands of this framework may go well beyond the

types of studies done in these fields, because, besides the development of conflict resolution and negotiation strategies, this work requires strategies that can sustain and control conflict to desired extents, and at times even exacerbate it in a controlled fashion.

4. How does one *design for interaction*? One can identify collaborative design as only one example, albeit of considerable importance, which calls for new and innovative concepts related to this general topic.

The socio-technical framework laid out in this work moves ones perspective in the direction of design as a technically based “conversation” (Winograd and Flores 1986) that occurs between various stakeholders during a design campaign and the evolution of the relevant design environment. New research tools and methods will no doubt be needed to address the way one should design for such interactions.

3.3 The Architecture of the Socio-Technical Framework

The Socio-Technical framework proposed in this work is based on the recognition that collaborative engineering design is a human-based, interdisciplinary, and socio-technical activity, and must be modeled as a co-construction process. Figure 7 illustrates the key concepts of this Framework.

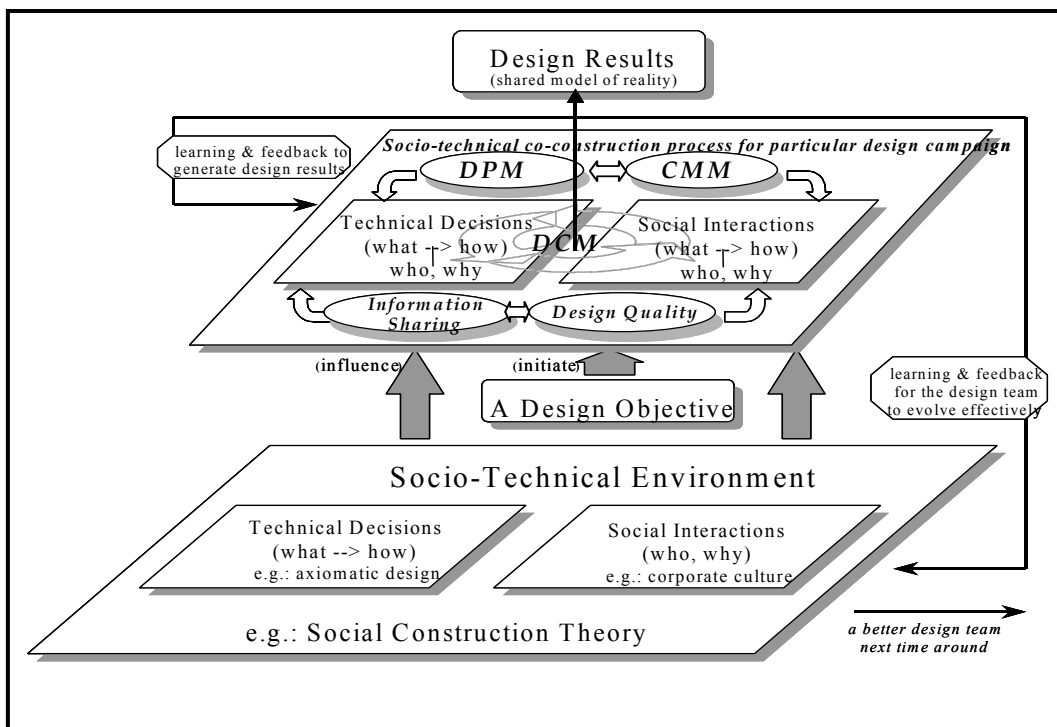


Figure 7. The conceptual architecture of the socio-technical framework.

The lower plane in Figure 7 represents the socio-technical environment, i.e., the infrastructure in which a specific design campaign is to take place. Various established design theories and methodologies govern technical decisions; while existing corporate cultures and organizational norms guide social interactions. The social construction theory is a base to link these social interactions and technical decisions within this design environment.

The upper plane shows the socio-technical co-construction during a *specific* design campaign within this environment. Initiated by an overall design objective, the design campaign evolves in the upper plane whose outputs constitute the design results (e.g., the final product model), and adaptive, evolutionary, “feed-back” modifications to the design environment (i.e., the lower plane) and the design campaign (i.e., the upper plane) itself.

The Design Process Model (DPM) and Conflict Management Model (CMM) are the two key components that integrate the technical decisions and social interactions. Information sharing and design quality are the two main concerns that govern this socio-technical integration. The co-construction that occurs in the upper plane is relevant at the product and process level, while that occurs in the lower plane may be envisioned is relevant at the system (environment) level. The end results of the design campaign are: a product model and a more efficient design process (at the upper plane), and a better design team (or design environment) the next time around (at the lower plane).

3.3.1 Design as Conflict Management (DCM)

As technical decisions are being made under the influences of social interactions, various technical and nontechnical conflicts will occur during this co-construction process. Conflict management is therefore the key “process control” that enables the product model and the design process model (DPM) to adapt, grow, and evolve to a state that is both understood and accepted by all stakeholders in a design campaign. The socio-technical view of collaborative engineering design highlights the importance of design conflicts and various conflict management strategies (CMM), and enables the study of Engineering Design as Conflict Management (DCM). Although research has produced considerable and extensive conflict management concepts and strategies from both a technical perspective and a social science perspective, there is still no systematic and effective approach to managing engineering design conflicts that are of both technical and social nature.

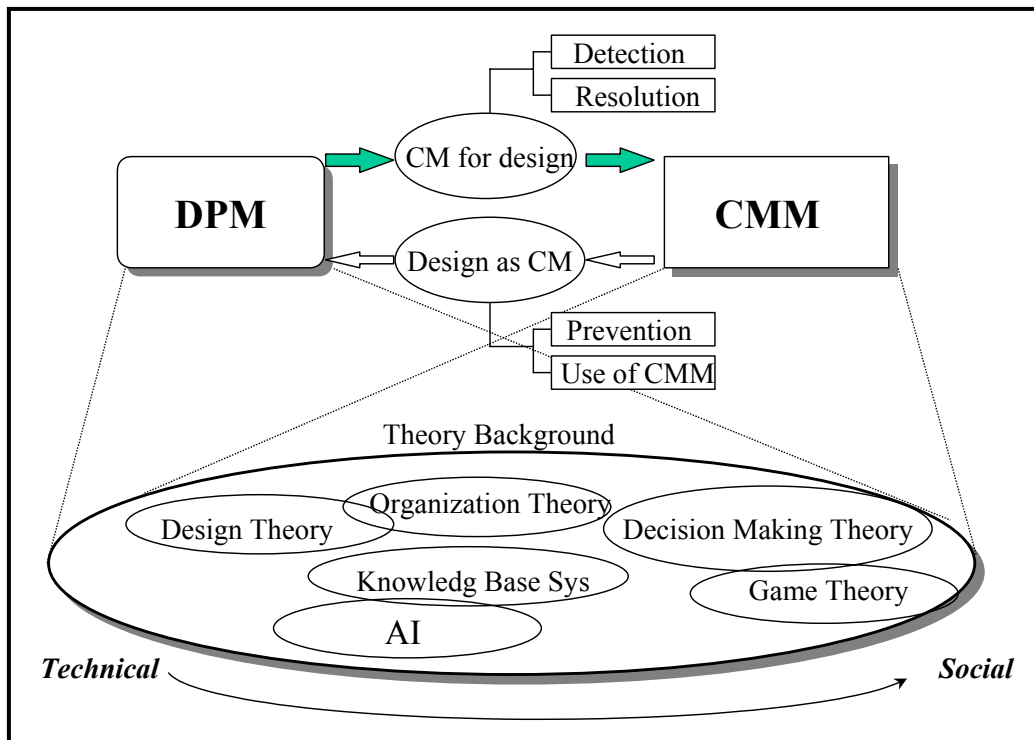


Figure 8. DCM modeling framework.

As Figure 7 shows, this approach focuses on two elements of the socio-technical framework: the design process model (DPM) and conflict management model (CMM). The framework extension, shown in Figure 8, has two layers: the DCM structure (i.e., DPM + CMM) at the higher level, and the theoretical background that supports the two DCM components at the lower level. DPM and CMM share the theory background, but content of the models reflects their different purposes.

The DPM describes the characteristics and provides procedures of collaborative engineering design activities. This work's DPM is based on an extension of the Axiomatic Design model by adding the consideration of "who" to its generic "what" to "who" mappings.

The Conflict Management Model (CMM) describes the characteristics of conflicts in the engineering design domain. It consists of a set of conflict situations according to their types and a set of conflict management strategies for each identified conflict type. In this study's CMM, design conflicts are always monitored by "Conflict Detection System" and classified into different types based on the state of design process. In the conflict type hierarchies, the state of design process is recognized, analyzed, and then matched with various nodes. Different kinds of conflict management strategies are systematically associated with the different identified conflict types. In this way, when CMM model interacts with DPM, it

can generate conflict resolution plans, provide conflict prevention strategies, and use conflicts to guide options at the divergence phase of design. Figure 8 conceptually illustrates the structure of CMM model and how it interacts and integrates with DPM.

3.3.2 The Co-Construction Process in Design Campaigns

According to the above conceptual framework, Figure 9 below shows a time sequence indicating the feedback between the co-construction process, which affects the design environment, and the co-construction process that goes on in a specific design campaign. At each vertical “time-slice” the two “layers” of the socio-technical construction are shown. The feedback between the two layers is what causes the evolution of the design environment. As time proceeds, the co-construction of goals, processes, and quality metrics for a specific design campaign gradually move from being largely in the social domain to being largely in the technical domain, and the degree of conflict gradually reduces as consensual confluence is engendered at a desired rate.

It is interesting to note that this gradual evolution from a largely social environment to a mainly technical one during a design campaign is also reflected in other traditional design theories and methodologies. For example, the House of Quality approach in TQM provides a systematic method that allows designers to express their relative “preferences” toward different design goals to capture important nontechnical factors at early stages of design.

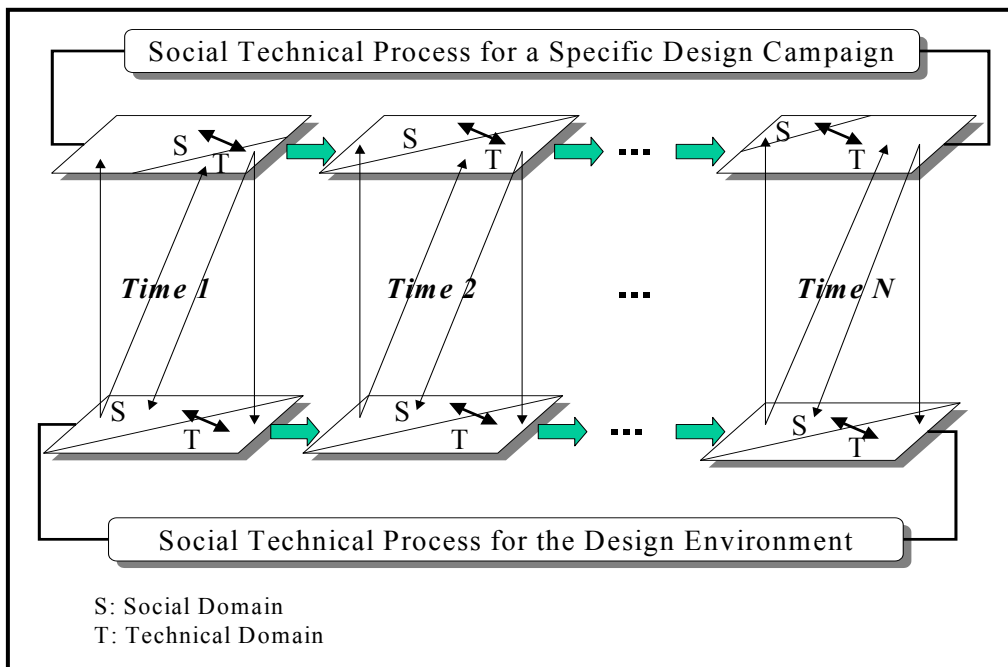


Figure 9. Interaction between design campaign and design environment.

The Decision-Based Design (DBD) approach has the design options generation and selection phases, where the former is relevant to the social interactions among designers. The Axiomatic design approach proposed by Suh suggests zig-zag mapping and decomposition operations across customer, function, physical, and process domains with multiple layers of details. It is clear that these operations between the customer and functional domains, especially at the early stages of decompositions, are highly human-dependent, and hence, social interactions play an important role. Figure 10 below conceptually illustrates this point with Suh's Axiomatic design.

3.3.3 Collaborative Design Architecture Based on the Socio-Technical Framework

The Socio-Technical Framework in Figure 7 clearly depicts the three key controllable parameters to support collaborative design. First, to control the interactions of stakeholders accessing and modification of external data, a feasible design process and a well-structured organization are needed. Second, stakeholders' perspective interaction is a critical issue to consider when modeling collaborative design. Third, conflict management strategies can be used to effectively manage the dynamic relations in collaborative design. These three factors function together and influence the overall characteristics of collaborative design.

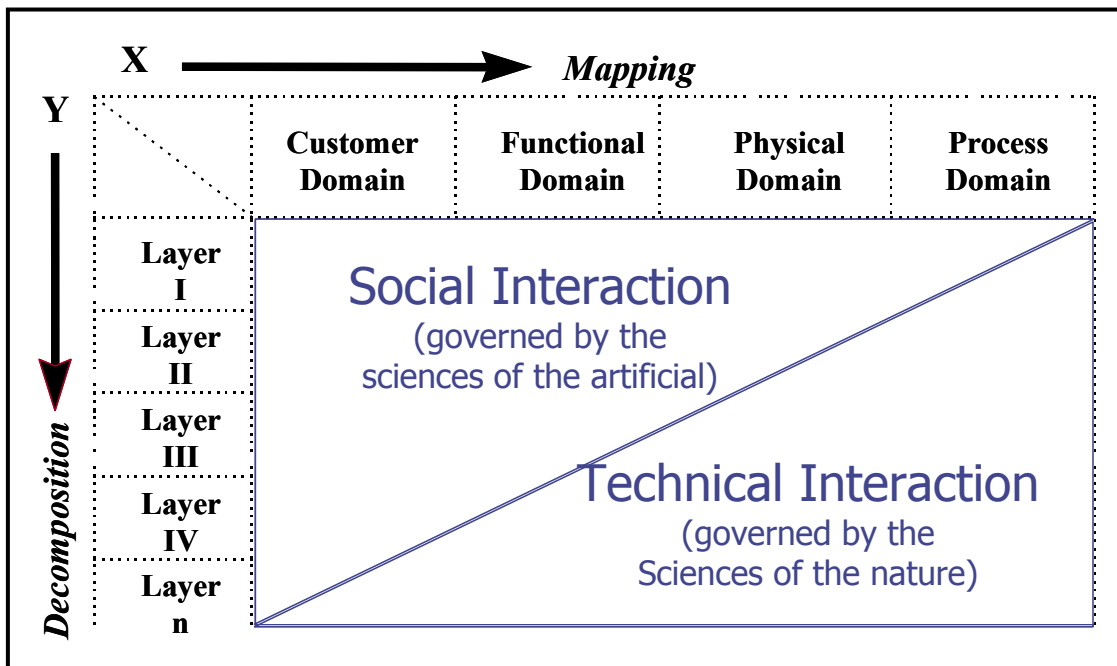


Figure 10. Social interactions in Suh's axiomatic design framework.

Figure 11 shows various elements and their relationships in collaborative design. As indicated, *technical design process*, *social interaction*, *conflict management strategies*, and *perspective model* are the critical components within the socio-technical design process.

In a design campaign, stakeholders perform both technical and social roles based on their unique perspectives. The former is conducted in the technical decision-making process while the latter is represented as social interactions. They are formed when stakeholders become part of a community undertaking a design campaign and begin to interact with other members of the community. By making technical decisions based on their technical roles, design stakeholders create, modify, and evaluate the product features. Since the involvement of social roles, which are normally influenced by the organization structure, norm, and culture, technical decisions are coupled with the social-interactions during the design co-operation. Knowledge representation is critical for designers to capture the understanding and reasoning behind technical decisions.

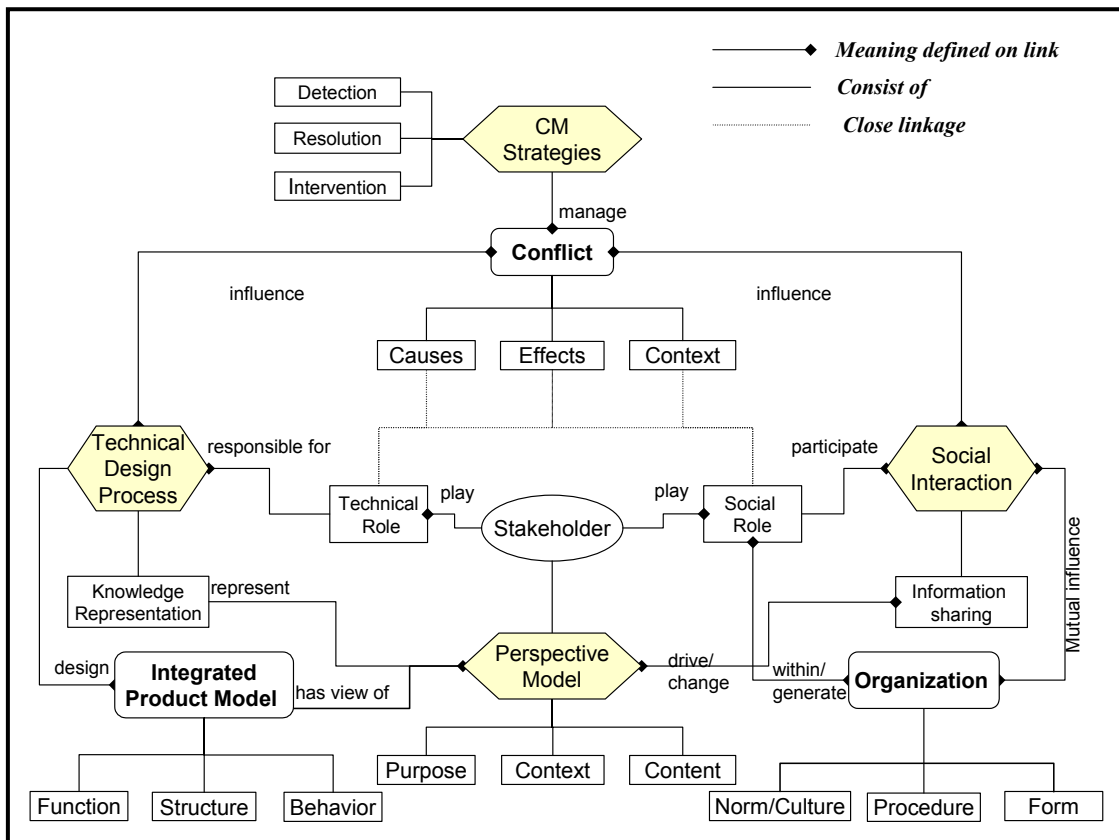


Figure 11. Supporting collaborative design.

Effective information sharing mechanisms accelerate the process of achieving shared reality. During the design interaction, various conflicts will occur due to multiple task interdependencies and perspective differences. When treating engineering design as a purely technical process, conflicts are usually regarded as being abnormal and to be avoided as soon as possible. To resolve design conflict, different approaches have been proposed by building utility functions for designers, by categorizing conflict resolution knowledge, or by capturing design rationale. However, when treating engineering design as a socio-technical process, conflicts must be systematically and explicitly dealt with as a resource to drive the social construction process and design innovations. To manage conflict near its source and root, social interaction should be considered as a controllable parameter to affect and change the design perspectives. In the early design stage, conflicts are treated as a motivation to identify the deficiencies among design team and to generate creative ideas, while at the late stage conflicts should be prevented or resolved to achieve high efficiency.

3.3.4 Perspective Modeling as the Key in the Socio-Technical Framework

The collaborative design architecture shown in Figure 11 clearly indicates that the key linkage between technical decisions, represented by integrated product models, and nontechnical decisions, represented by various organizational factors, is the Perspective Model. As explained before, perspective is the central theme of the Socio-Technical framework. The ability to model stakeholders' perspectives dynamically during collaborative activities is critical for the operation and implementation of this new framework.

"Perspective" holds a key to the establishment of a practical theory for collaborative engineering. As conceptually illustrated in Figure 12 below, a true collaboration, which can support process automation, data interoperability, and virtual teams in real life applications, requires task coordination and understanding sharing, which must occur at both the data and people levels.

In fact, according to the Socio-Technical framework, research in collaborative engineering can be seen as *a perspective-based approach to collaboration* that strives to achieve the true understandings at both people and data levels. Perspective modeling is used to improve people understanding by treating collaborative design as a conflict management problem (see Section 4.3 for details). Perspective modeling is used to realize data understanding by offering dynamic mapping mechanisms within collaborative information resources.

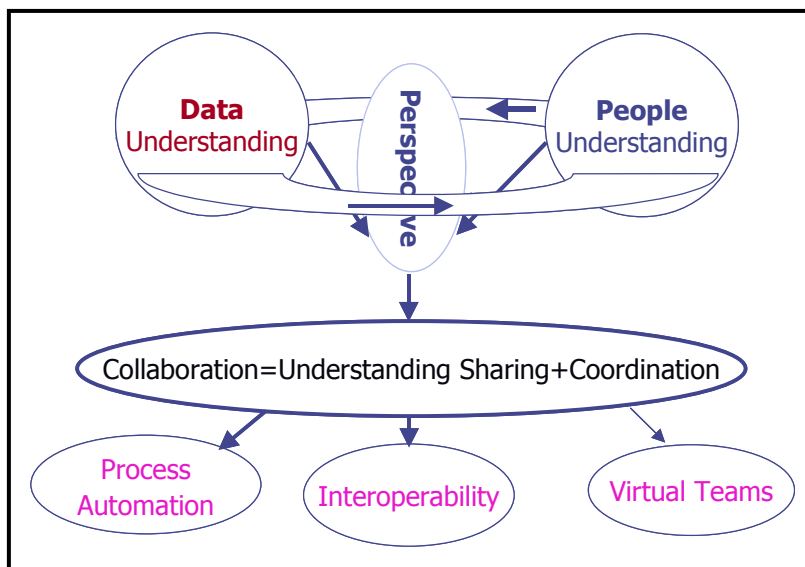


Figure 12. Perspective-based approach to collaborative engineering.

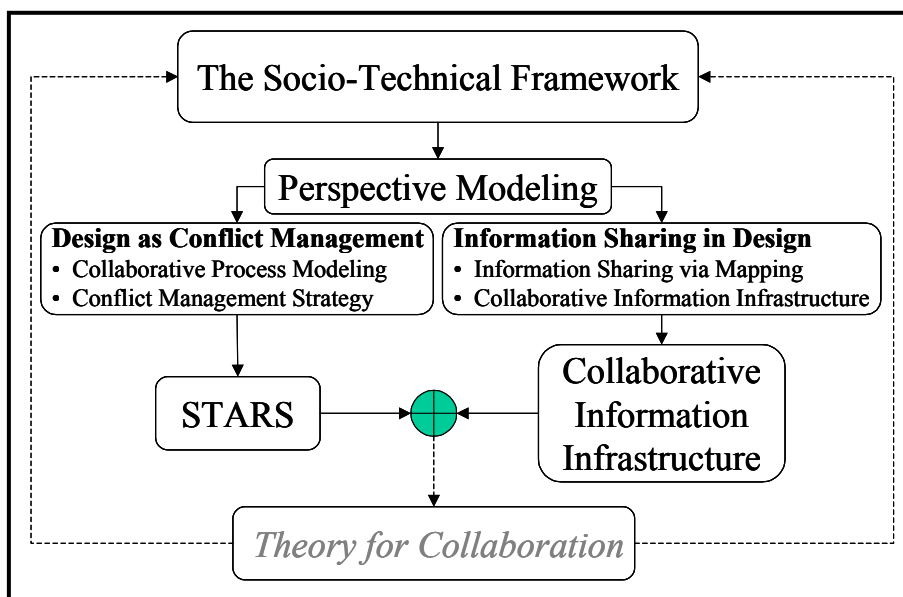


Figure 13. Research directions in the socio-technical framework.

In a sense, the perspective modeling provides a conceptual underpinning and operational link between the Socio-Technical Framework and the two major directions of this research program, namely:

- Collaborative Design as Conflict Management
- Information Sharing in Collaborative Design.

Figure 13 shows the structure and strategy of this research program, including the above two major directions, under the Socio-Technical framework. These two basic research directions, along with their prototype system implementations, can contribute to the establishment of a Theory for Collaboration in the future.

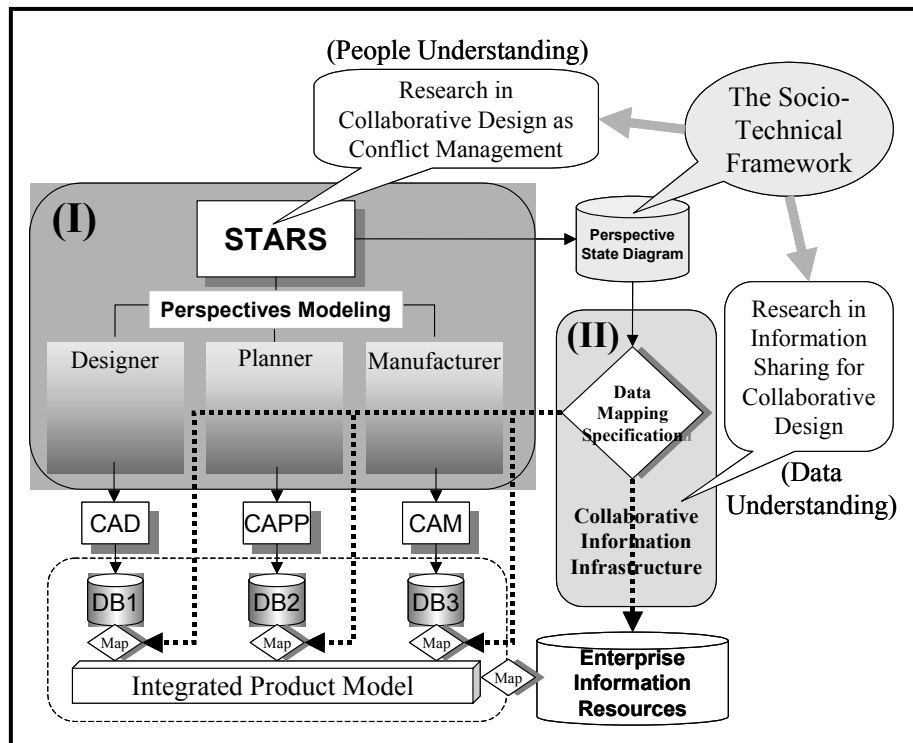


Figure 14. The five key components of the two major research directions.

Before presenting detailed technical results for this research in Design as Conflict Management and Information Sharing for Collaborative Design (in Chapters 4 and 5) of this report, the next Section briefly reviews the five key technology components that collectively represent this investigation in the Socio-Technical Framework for Collaborative Engineering.

3.4 An Overview of Key Components of Socio-Technical Framework

As explained above, this research program includes two major research directions and five key technology components:

1. Perspective Modeling
2. Collaborative Process Modeling and Simulation
3. Conflict Management Strategy
4. Information Sharing via Data Mapping
5. Collaborative Information Infrastructure.

As Figure 14 shows, perspective modeling is a common component shared by the two research directions. The resulted Perspective State Diagram is a common

resource to drive both conflict management and information sharing in collaborative design. Components (2) and (3) fall into the first research direction, which contributes to the people understanding issues. Components (4) and (5) belong to the second research direction that supports the data understanding issues.

The concept of stakeholders' perspectives perspective modeling was already introduced in Section 3.2.1.3 above. This Section reviews the remaining four key components.

3.4.1 Collaborative Processes Modeling and Simulation

The re-examination and reformulation of design process models (DPMs) in this work has been motivated by the following observations, which point out some of the common deficiencies between the DPMs that have been proposed thus far.

- As mentioned, there seems to be no unifying framework for modeling the design process. Most engineering design process models have been developed by engineers, and show a proclivity for optimal *correctness* and *consistency*. Since these concepts are hard to come by in an open-ended activity like engineering design, several design process models start from “self-evident axioms,” so that the results, which follow there from lie beyond both interrogation, and argument.
- They each treat engineering design as a single person activity, or an activity engaged in by a group that behaves as a single entity. With the rapid strides in communication technology and the complexity of modern engineering projects, collaborative design is becoming more the norm than the exception. Large design projects require diverse expertise drawn from various knowledge bases and hence necessitate the *interaction* and co-construction of different groups of designers who are experts in their specific areas. There appear to be few, if any, design process models that include this aspect of engineering design; some axiomatic design process models do not even acknowledge the existence of collaborative design.
- Conflicts are considered abnormalities in the design process. Suh's axiomatic design, for example, attempts to minimize the local effects of conflicts and its global influence in the design process by instituting the axiom of functional independence; game theoretic approaches attempt to minimize conflict by making decisions consistent with identified utility functions (or pay-off tables); and, knowledge-based design attempts to identify and weed out conflicting data so that consistency is achieved.
- Each of the design process models expects the designer to be a completely rational automaton who will follow the procedure outlined in each process model with no regard to other social and nontechnical influences.

- Though various engineering design process models have been proposed to date, it is difficult, if not impossible, to rank-order them. The question—which of these design process models (if followed to the letter) will produce the best designed product under a certain set of contingencies—appears at the present time to be unanswerable. In fact, how to design a generic quality metric to measure the quality of the design process/product has no clear answer either.

And yet, it is common knowledge that the design process is greatly influenced by the perspectives of the individual designers – perceptions of themselves, of the organization they work within, and of others who work with them. In the design of complex systems, such as facilities engineering, diverse and often contradictory knowledge-bases are involved, and group design becomes a necessity since no one individual has the knowledge (and often, even the awareness), and/or the experience to handle every aspect of the design activity. As stated before, participants in the design process differ not just in the disciplines of their expertise, but geographically and temporally as well, and distributed collaboration across social, cultural, organizational, and linguistic boundaries are often required. In short, the design activity involved in complex projects is a multi-disciplinary, multi-participant activity, each participant having only *partial* awareness/knowledge/capabilities to fulfill a partial set of the overall design needs, and yet their contributions must be consistent with the overall needs to be useful. It is only through a fruitful interaction and learning among such experts that one can expect the design of a complex project to be cost effective, timely, and well executed. Such an interaction necessitates a co-construction process where opinions gradually evolve, and finally reach confluence in a shared reality of what is to be achieved, how it is to be achieved, and how progress toward the various goals and sub-goals is to be implemented and measured.

The socio-technical co-construction done by these stakeholders, which underlies the design activity in any design campaign, necessitates a constant negotiation and *adaptive evolution* of the design goals, the design processes, and the quality metrics. In fact, it leads to an evolution of the very *meanings* of these concepts as a convergence gradually emerges toward a consensual validation (in which all the stakeholders participate) of a perceived reality. One obtains thus a more *integrated view* of the design goals, the design processes to be employed, and the quality metrics suitable to measure the goals and the processes.

3.4.2 Conflict Management

When inconsistent local realities evolve and merge into a consistent global reality during a co-construction process, conflicts of various types at different abstraction levels occur among stakeholders. These conflicts could be of a technical

nature, a managerial nature, or of a social-interaction nature, and their effective handling plays a central role in determining the overall effectiveness of a design team. When treating engineering design as a purely technical process, conflicts are usually regarded as being abnormal, and to be avoided as soon as possible, at all costs. In the current framework, conflicts need to be systematically and explicitly dealt with as a *resource* to drive the social co-construction process, and design innovations.

As gradual confluence occurs through co-construction, the design process becomes more and more driven by technical considerations allowing mental constructs such as mappings, decision analysis, and game theoretic methods to provide useful ways of then developing and co-constructing DPMs.

The co-construction framework requires a categorization of the different kinds of conflicts in terms of those arising from negotiations and co-constructions related to: (a) goals, (b) processes, and (c) quality metrics. The aim would be to find mappings between the different types of conflicts that arise, and conflict management strategies that have been developed in the social, political, and organizational management literatures. It should be emphasized that conflict management may involve not just the detection, prevention, and resolution of conflict, but also the encouragement, sustenance, and control of conflict in a desired manner. Of great significance is the development of tools to measure and monitor the “rate” at which conflict resolution occurs so that confluence of viewpoints in the socio-technical co-construction process can be achieved in a desired and controlled manner. As stated earlier, the concept of co-construction goes beyond simple conflict management. It calls for the development of goals, processes, and metrics to *create* new ideas through the interaction of stakeholder perspectives; it may thus be argued that quality is created through co-construction. This leads to the issue of redefining design quality.

3.4.3 Information Sharing

Collaborative engineering design is a social activity that depends crucially upon the ability of the participants to communicate effectively by sharing information. The purpose of the information sharing is to promote a common, shared *understanding* of the state of the design campaign to enable each participant to make decisions that best contribute to the design and business objectives of the campaign.

Effective sharing of understanding has been a long-sought goal in the design of collaborative technology and interoperable software applications. Solutions almost invariably focus on the specification of a collection of concepts (called a

schema) and a digital format for encoding the concepts for transport over digital media. Some well-known examples of this kind of solution are Electronic Data Interchange (EDI), ISO 10303 (STEP – a product data exchange standard (ISO 1994), and—most recently—the XML Vocabularies. (See <http://www.oasis-open.org> and <http://www.biztalk.org> for example of XML Vocabulary libraries.)

These solutions all share the same critical shortcoming: humans do not communicate using a semantically static, fixed vocabulary (such as that specified in a schema). Although relatively stable over long periods of time, the semantics of natural languages drift with evolving circumstances in which the language is used and with the needs of the communicating individuals. In all cases, however, the sharing of understanding—and thus exchanging knowledge—requires shared conventions for producing and interpreting utterances in a language. Information technology is no different with respect to communication of natural language semantics. Misinterpretation and semantic drift are every bit as frequent and common in the interoperation of software applications.

3.4.4 Collaborative Information Infrastructure

The first and most apparent impact of this new framework on information technologies is that building predefined, rigid meaning *into* application software and data systems dooms the system to a short life span in design practice. The reason for this is that the knowledge and perspectives of stakeholders change dynamically throughout an engineering design campaign, thus the needs of the stakeholder and the capabilities of the system slowly drift apart over time. Few systems are really able to *learn*, adapt, and keep pace with continuous human knowledge acquisition and processing. In the framework proposed in this paper, information technology serves as a *vehicle for conveyance of meaning* (i.e., information) between stakeholders and that the *meaning lies entirely in the hands of, and evolves through the control of the stakeholders*. Therefore information systems must be designed to be as *meaning-neutral* as possible.

A new way of thinking about the human/computer interface is required here. One of the key aspects of computer support for the view of collaborative engineering design is the capture and use of stakeholder perspectives. A representation of perspectives of a stakeholder might be referred to as a “Perspective Model.”

A Perspective Model is a representation of a stakeholder perspective that is deliberately built by the stakeholder to represent his “local reality” or is automatically constructed by his use/interaction with software systems. “Embryonic” perspective models already exist in the Preferences, User Profiles, and

customization of commercial software applications that allow the user to make the software appear/ behave a certain way. A quantum step forward in the customization of information technology is thus required in allowing the user to construct his own view of “what’s going on” in an engineering design campaign (i.e., his local reality), and adaptively change it during interaction with other stakeholders.

Current attempts at data management rely on methods that aim to provide consistent data (i.e., noncontradictory data), and those that update information between local databases and global databases, often through a central server. The framework described here requires the development of databases and data management strategies that enable the dynamic co-construction of the meaning of the data based on the perspectives of the users of the data. Co-construction will, in general, elicit contradictions and ambiguities in data; the development of methodologies (and computer support systems) that engender gradual confluence and creation of meaning is called for. At present these apparently do not exist.

4 Collaborative Design as Conflict Management

This chapter presents research results in a perspective-based approach to model collaborative design as conflict management within a Socio-Technical Framework. Section 4.1 explains a mathematical technique, adapted from dynamical systems, used here to model the changing perspectives of stakeholders. Sections 4.2 and 4.3 describe details of collaborative process modeling and conflict management. Section 4.4 presents implementations of a prototype system, called STARS (Socio-Technical Analysis and Research System).

4.1 Perspective Modeling using Dynamical Systems

4.1.1 *The Dynamical System Model*

The cornerstones of this Socio-Technical Framework for Collaborative Design are the concepts of stakeholders and their perspectives in a design campaign. A stakeholder's perspective is a special hybrid of domain and background knowledge; it consists of purposes, contents, and contexts. It can be visualized as different "lenses" stakeholders wear during different stages of design. The exciting finding from this research in this area is that one cannot utilize information to map from "what-to-design" to "how-to-design" in collaborative engineering design without knowing the perspective of the "who" that generates the information. Similarly, conflict resolution and control, both elements of decisionmaking, in collaborative engineering design require the explicit modeling of stakeholder perspectives.

While the need for using perspectives is essential for coordination across disciplines, across time and resources, and across organizational cultures, the development of these ideas requires a general conceptual formulation. Presented here is a dynamical modeling approach to illustrate the key concepts of the Socio-Technical framework. One of the purposes of this exploratory study is to develop an approach for modeling "understanding-sharing" among design stakeholders, and in the process refine it, so as to provide a theoretical underpinning for future work. The collaborative design campaign is viewed as a dynamic system. Such a dynamical systems approach is essential to formalize and develop several key

concepts without which one cannot develop methods to support collaborative design on a theoretically sound basis.

One of the most essential concepts introduced by this approach is stakeholders' perspective. This understanding of perspectives begins with the acknowledgement that:

- Each individual builds over her lifetime an evolving base of information that is “internal” to her
- Each individual has a perspective that evolves over time and acts like a “lens” through which she understands and collects data external to her
- The data that each individual produces, or exchanges through any medium (computers, speech, writing), is the external manifestation of her internal information, appropriately filtered through her “perspective lens.”

Consider now a collaborative design group consisting of N individuals. At any instant of time, t , each individual “ j ” can be described as having: a store of internal information, $H_{j,t}$; a perspective $P_{j,t}$; and, the external data, $D_{j,t}$. A perspective consists of two parts, the “filtering” perspective $P_{j,t}^F$ and the “learning” perspective $P_{j,t}^L$. $P_{j,t}^F$ is used by stakeholder as a filter to access and generate data. $P_{j,t}^L$ is their internal learning habit to update $P_{j,t}^F$. Assume that at some time $t=t_0$, these entities H_{j,t_0} , P_{j,t_0} , and D_{j,t_0} are partially known for each of the N individuals forming the design team.

Collaborative design can be seen as the stakeholders generating and sharing information through their perspectives. The information is represented as various formats of data. The sum total of the data external to each individual at time t shall be denoted by $E_t = \bigcup_j D_{j,t}$.

If considers an increment in time (consider, for simplicity, a discrete time dynamical system), then the following relations govern the time evolution (for $t = 0, 1, 2, \dots$) of $H_{j,t}$, $P_{j,t}$, and $D_{j,t}$ for each of the N individuals:

$$H_{j,t+1} = P_{j,t}^F(E_t) \cup H_{j,t} \quad \text{Eq. 1}$$

$$P_{j,t+1}^F = P_{j,t}^L(P_{j,t}^F, H_{j,t+1}) \quad \text{Eq. 2}$$

$$D_{j,t+1} = P_{j,t+1}^F(H_{j,t+1}) \quad \text{Eq. 3}$$

The first equation states that individual “ j ” updates her internal information store on the basis of the total external data available to the enterprise by filtering it through her perspective lens, $P_{j,t}$, and combining it with the internal in-

formation store, $H_{j,t}$ she has up until then. This equation can also be viewed as representing the “learning” process, which is known to require both background information (E_t) and mental bias ($P_{j,t}$). The second equation states that this update in internal information causes the perspective lens through which she views the world to evolve in time. This equation together with the first may be viewed as encapsulating the “thinking” process. The last equation states that the external data she generates at time (t+1) is created by the updated perspective “acting on” the updated internal information.

Note that this formulation has the advantage that it explicitly recognizes the difference between the information that is externally expressed by individual “j” and the information internal to her. It shows “perspective” as an “operator” that acts on (interprets) both the external data that is encountered and the external data that is produced. The external data that each individual produces in turn is encountered by each of the others in the enterprise system, and interpreted by each individual according to that individual’s perspective.

Since the focus here is on understanding-sharing, it is necessary to formalize the concept of understanding, $U_{j,t}$ of individual “j” at time t. One way of doing this is to think of the understanding of individual “j” at time t *with respect to the external data* $D_{k,t}$ as an operator that depends on both the perspective of individual “j” and on her internal information store, when it “acts on” the data $D_{k,t}$. Thus:

$$U_{j,t}[D_{j,t}] = U_j(P_{j,t}, H_{j,t}) [D_{j,t}] \quad \text{Eq. 4}$$

Figure 15 shows that the approach presented here now opens up several levels of incompatibility that must be addressed during collaborative design. (At this stage of theoretical development, the word “incompatibility” is used somewhat loosely to include inconsistency, irrelevancy, incompleteness, etc.) One may have at any time t :

$$D_{j,t} \text{ I } D_{k,t}, \text{ and/or, } P_{j,t} \text{ I } P_{k,t}, \text{ and/or } H_{j,t} \text{ I } H_{k,t} \quad \text{Eq. 5}$$

where the symbol I stands for “incompatible with.”

In collaborative design, these inconsistencies imply different types of conflicts. Inconsistency in external data between two individual “j” and “k” is viewed as conflict relating to the product specification level. That is *one* generic form of conflicts focused by most current conflict management approaches.

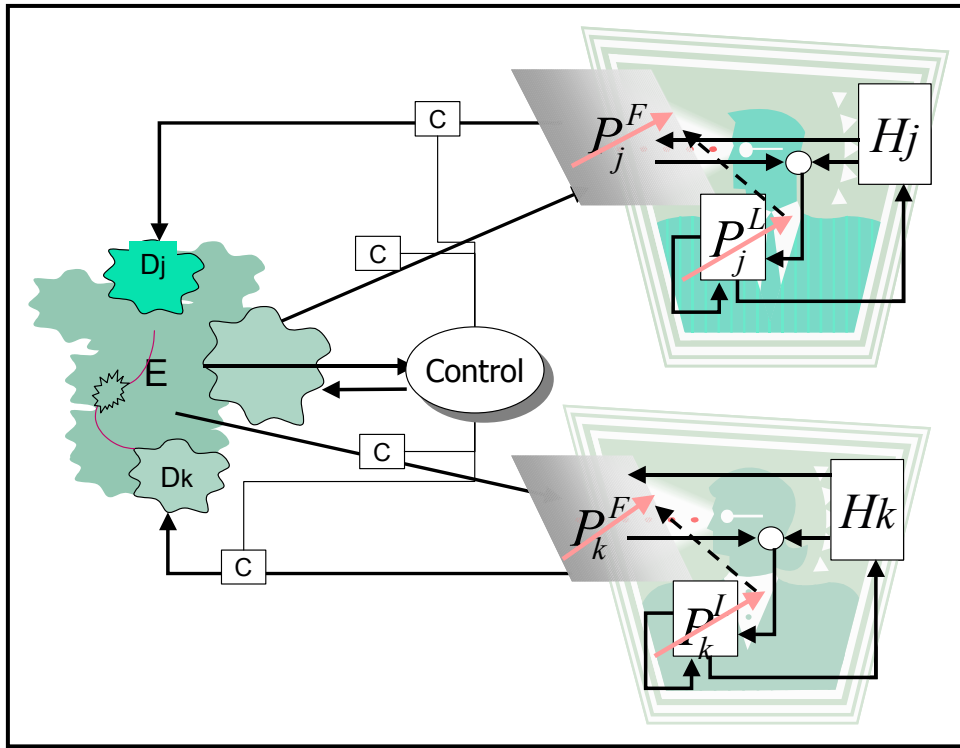


Figure 15. Dynamical model of design perspectives.

Incompatibilities in internal information may imply the knowledge conflict between different stakeholders. Since internal information is inside human minds, this kind of conflict is relatively difficult to detect and represent. Since perspective is the filter through which internal and external data are generated, the incompatibility between different perspectives is the major source of the above two sorts of conflicts.

Given this realization, relation (4) shows that methods must be able to handle design conflicts to bring about improved compatible understanding of external data. Thus, conflict management is essentially viewed as an understanding-sharing process. This framework leads to a possible formulation of the concept of “understanding-sharing” between two individuals “j” and “k.” Still needed are information systems that make:

$$U_{j,t}(D_{l,t}) \cap U_{k,t}(D_{l,t}) \text{ as large as possible for some time } t > m, D_{l,t} \in D(k, j) \quad \text{Eq. 6}$$

where $D(k, j)$ is the relevant set of data-discourse between individuals k and j , and m is a suitable time horizon that depends on the specific situation. (This concept can be further expanded when considering a set of individuals, but is not pursued here.) Understanding-sharing as described by (6) is seen to be an inherently dynamic, evolutionary process, across time.

One way of achieving (6) might be to influence the dynamical relations (1-3) through the use of a set of “controls.” In general, these controls will be time dependent and will depend on the internal information stores and the perspectives of the individuals comprising the design organization as well as the external data available at that time. Thus, instead of relations (1-3), there is, for each individual “j”, the following dynamical system (for $t = 0, 1, 2, \dots$):

$$H_{j,t+1} = P_{j,t}(E_t) \cup H_{j,t} + C_{H,j}(\cup H_{j,t}, \cup P_{j,t}, E_t) \quad \text{Eq. 7}$$

$$P_{j,t+1} = P_{j,t}(H_{j,t+1}) + C_{P,j}(\cup H_{j,t}, \cup P_{j,t}, E_t) \quad \text{Eq. 8}$$

$$D_{j,t+1} = P_{j,t+1}(H_{j,t+1}) + C_{D,j}(\cup H_{j,t}, \cup P_{j,t}, E_t) \quad \text{Eq. 9}$$

One of the purposes here has been to identify concepts and obtain an initial conceptualization of perspective evolution in collaborative design. This is useful to focus attention to areas not fully understood yet. For example, the controls C_j may also be thought of as being provided through negotiation, and provision of additional information. Negotiation at the external data ($C_{D,j}$) level might involve, as it often does in many product data management systems, simply a check for its consistency. But this formulation clearly indicates that such a simple negotiation process may not be useful because the data $D_{j,t}$ generated by individual “j” may affect the internal information store of individual “k” (see equation 7) and hence possibly her perspective (see equation 8). One can then see the possibility of causing an avalanche of other external data inconsistencies as time evolves despite having resolved the one specific data inconsistency at hand.

This formulation has the advantage that it explicitly recognizes the difference between the data that is externally expressed by individual “j” and his internal knowledge. It shows “perspective” as an “operator” that acts on (interprets) both the external data that is perceived and the external data that is produced. The external data that each individual produces in turn is encountered by each of the others in the enterprise system, and interpreted by each individual according to that individual’s perspective.

The model is a simple feedback loop and, thereby, offers an opportunity to understand and control the internalization and externalization of data. The perspective filters or affects the perceived data and adds to or changes the internal information store. It also serves as a filter on the internal data that is externalized for transmission to another. This internal information store, in turn, affects the operation of the perspective in both the subsequent internalization and externalization of data. If external data is taken as a “signal,” then the operation

of this dynamical model describes some of the behavior of the signal generator germane to the subject research.

An analogy for understanding the dynamical model is a polynomial equation. The perspective acts as the coefficients of the equation; perceived data as the values of the parameters. The result of the equation for a set of values is the internalized knowledge or the externalized data. The feedback (adaptive process illustrated in Equation 2) is a process that actively changes the coefficients.

4.1.2 Building Perspective Models

One of the essential goals of building the perspective models is to externalize the perspectives ($P_{j,t}$) so that they can be systematically represented and analyzed. This study proposes an approach to use “concept structure” as a mechanism to organize perspective models. To generate the perspective models, the stakeholders first collectively build the Concept Structure. Concept Structure is an organization of the ontologies that stakeholders propose and use in collaborative design. Stakeholders should use both top-down and bottom-up construction methods to support stakeholders to build the concept structure. It first provides some templates (e.g., “product function template,” “design organization template,” “conflict types,” etc.) for the stakeholders to clarify the concepts (Figure 16). These templates act as the content-based skeletons for organizing the external data stakeholder may share with others. For more routine design, stakeholders can extract many concept structures from previous product models and design documentation.

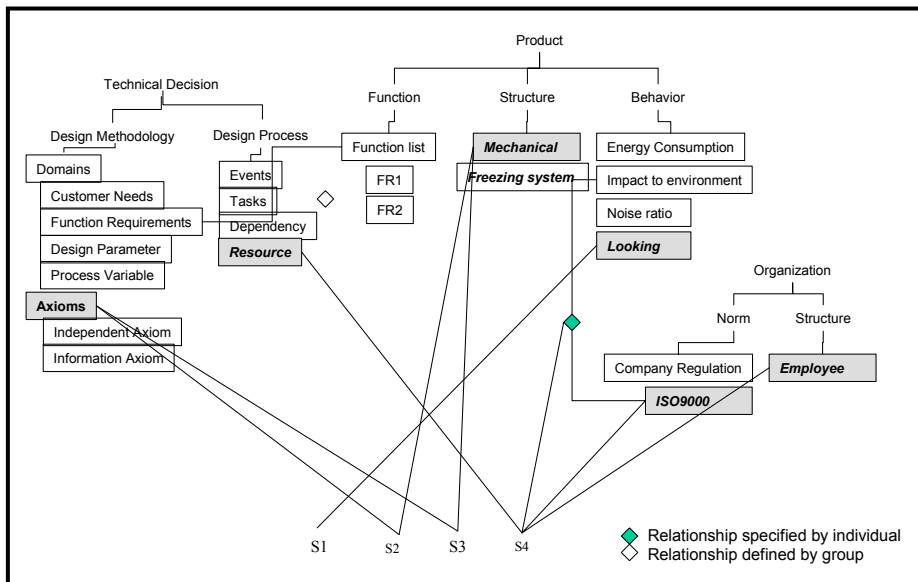


Figure 16. Concept structure built by stakeholder.

When an individual proposes a new concept, he/she should first consider whether there are similar concepts in the structure. Thus, only the novel concepts can be specified and added. When stakeholders propose new concepts in design process, the concept structure is updated and is used to systematically organize these concepts and their relationships. Individuals often are the best at generating the concepts, while the group is often best at selecting and enhancing the concepts. Therefore, the concept is classified into two types. “Shared concepts” are those that have been well-defined from previous design and have widely accepted meaning among the stakeholders (e.g., “Requirement List,” “Function Structure,” etc.). Only the particular stakeholder that possesses “private concepts” perceives those private concepts. Their names or meanings are not expressed around the group. Whether a concept is shared or not is relative to the purpose of a certain group.

If a group of people have shared purpose toward a concept, it would be better to have everyone view it. Sometimes, a concept is not shared between two groups, but may be shared within one group. After the concepts are identified, the dependencies among these concepts can be further clarified. For instance, the concept “function requirements” in technical decision will influence the “function” of product. The “structure” of product is decided by the “design parameter of the design methodology.

Given the well-organized structure of design concepts, it is possible to ask the stakeholders to build the perspective state diagrams at a certain time. A stakeholders’ perspective state diagram attempts to depict the explicit relationships among his/her concepts (include the shared concepts and individual concepts) and his/her purpose and context information.

The concepts listed in the perspective state diagram are categories of perspective contents relating to stakeholders. They are not all information of the design stakeholders’ perspectives. In fact, using concept structure in the perspective state diagram provides a structured way to systematically compare and examine the perspective differences among stakeholders.

Assume each stakeholder has already realized a purpose hierarchy (i.e., the stakeholder has specified his/her intentions within the design process). For each of the concepts, there are a set of purpose, context, and content associated with it. The operation is to ask the stakeholders to:

1. Relate this concept to their purposes. There might be more than one purpose relating to one concept. For abstract concept, the purpose could be more general. For specific concept, the purpose is more detailed.

- Specify the relationships of this concept to other concepts based on his/her context. If there are new concepts generated, add this to the PSD architecture and set it as an individual concept.

For each concept, declare his/her own knowledge/data about that concept and put them as parts of the content of that concept. Therefore, a Perspective state diagram is the picture that can depict stakeholders' perception of design concepts and his/her related purposes, context, and content. Figure 17 shows two stakeholders perspective models toward some concepts, represented in XML.

4.2 Process Modeling

Various models have been proposed to represent, analyze, and standardize the design process. The majority of approaches in this category are from the research of business operation and project management. Design is viewed as an information-driven process among design activities. Design organization is viewed as a stochastic processing network in which engineering resources are "workstations" and design tasks are "jobs" that flow among them (Adler and Mandelbaum 1995).

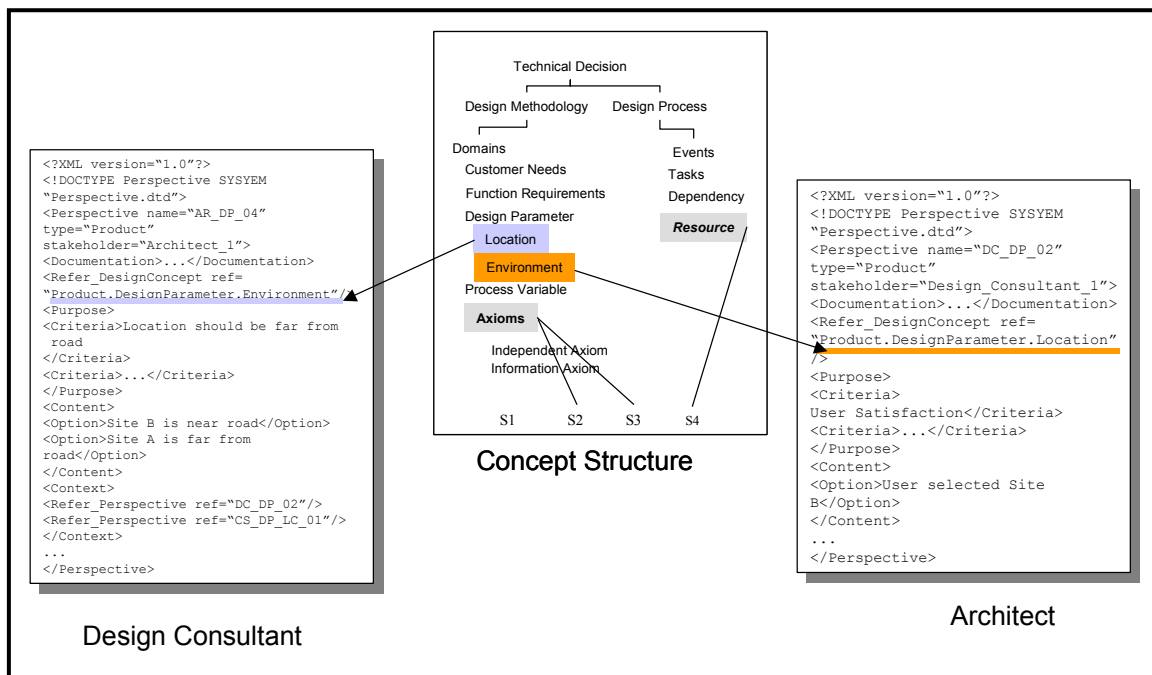


Figure 17. Generation of perspective model by concept structure.

Accordingly, a set of techniques to manipulate the design activities has been developed. Project Evaluation and Review Technique (PERT) (Wiest and Levy 1977) is widely used in engineering project management. Steward (1981) used the Design Structure Matrix (DSM) to overcome the size limitations of other process representations. Eppinger (1977) presented signal flow graphs as a flexible tool for design process modeling. Sanvido used an integrated design process model to represent the major functions and activities necessary for complex building design (Sanvido and Norton 1994). Bras and Mistree (1991) developed design process models by classifying all information into certain basic entities (e.g., phases, events, decisions, tasks, systems, and goals). The networks consisted with these entities describe design process and are represented in a form suitable for manipulation.

Although they present different ways to represent the dependencies among design activities, these tools have some limitations when they are used in collaborative process modeling. The PERT method is widely used for identifying the critical path of the process and estimating the completion time, but it does not support representation of iterations in the process. The State-Transition Diagram is popular in logic design and object-oriented modeling. One of its major disadvantages is that one has to define all of the possible states of the system beforehand. Signal Flow Graph provides a clear representation of design iterations, but it does not specify the presence of the stakeholders in the process.

The established design process models focus on the execution of activities, but do not address the interactions of the individual perspectives of the stakeholders. Most of them generate design process manipulation strategies for schedule maintenance or information management. They overlook some fundamental reasons of the conflict among the information transactions within the design activities.

A modified Petri net model is used here to represent the design activities and the coordination among stakeholders. Petri nets have the unique advantage to support process specification, representation, and evaluation at the same time (David and Alla 1992). Also, their mathematical properties help in quantitatively analyzing the behavior of the design process. Furthermore, elementary Petri nets have a simple graphical appearance, which can become a convenient and precise language for communicating among design stakeholders. However, note that collaborative design process is relatively complicated and unstructured compared with other process system (e.g., computer code [Jenson 1996], or manufacturing system).

4.2.1 Process Representation and Decision Support

To simplify the design problem, it is common to decompose it to small tasks, which are often assigned to different individuals separately. Although some design methodologies suggest that designers should increase the probability of success by maintaining the independence of sub-problems (e.g., Axiomatic Design Model), it is difficult to achieve this in collaborative design due to the various technical and social dependencies among tasks. On the other hand, individuals normally have limited capability to identify the influences of their decisions to others. Due to lack of coordination effort, the meanings about design objects might not be defined well, especially at the conceptual design stage. All of the above make the decomposition and integration of design sub-problems a rather complicated analyzing and synthesizing process. It is necessary to have a tool to simulate the design process and support stakeholders' coordination during the early design stage.

In collaborative design, the task decomposition and integration has to be achieved not only through the communication of contents, but also through communication about the creation and evolution of shared meanings. The shared meaning is always defined by the interaction of design perspectives. That reveals one of the essential aspects of collaborative design process modeling, which is to represent and manage the interactions among the individuals' perspectives. In other words, design coordination relates to not only the dependency identification among the design decisions, but also the management of changing and interaction of the design stakeholders' perspectives. In collaborative design processes, the influence of one's decisionmaking in a specific domain to others' decisionmaking in different sub-problems should be represented and evaluated. Furthermore, the design process representation model has to help design stakeholders to detect and evaluate the inter-dependencies among their design activities and to solve conflicts. Besides keeping the product data integrity, design information system should provide the "language" or "medium" for design participants to declare and depict their perspectives and aid their communication. These will definitely affect the current way of organizing design team and design process. To achieve these, it is critical to generate a design process representation model, which can facilitate the describing, tracing, and management of collaborative design interactions by referencing to design perspectives.

4.2.2 Development of A Petri-Net Type of Design Process Model

In the collaborative design process model, place and transition in the Petri net are equal to "state" and "task" respectively. A design process is represented by

an organization of states and tasks. Task is the activity stakeholders have to perform during design process. State is the status (or a group of conditions) that the stakeholders want to achieve through tasks. Both of them can have time duration. The different is that state is used to represent the measurable conditions in a process, and task is to represent the means to obtain the progress. State can be seen as the “what” in design process, while Task is similar to “how” to fulfill the “what.” The weights of the tasks can be used to represent their resource consumption. The default value of the weight is one. The arcs represent the transform directions between states and tasks during the design. The token denotes the state of each individual event. State contains token if and only if it is active (i.e., the state is happening). Thus the whole state of design process can be expressed by a marking M , which is a vector having the token numbers of each state in the design process. Since different stakeholders can conduct tasks, the “stakeholder” is introduced into the notation. Each task and state has a set of stakeholders associated. Formally, a Collaborative Design Process can be represent by a Petri net graph with the following definitions.

Definition 1: A Collaborative Design Process Net (CDPN) is a six-tuple $CDPN = (S, T, P, A, W, M)$ with a set of labels:

$S = \{s_1, s_2, \dots, s_n\}$ is a finite set of design states,

$T = \{t_1, t_2, \dots, t_q\}$ is a finite set of design tasks,

$P = \{p_1, p_2, \dots, p_m\}$ is a finite set of design stakeholders,

$A \subseteq \{(S \times T) \cup (T \times S)\}$ is a finite set of directed arc connecting states and tasks,

$W : T \rightarrow \{w_1, w_2, \dots, w_p\}$ is weight function attached to the design tasks,

$M_0 : E \rightarrow \{0, 1, 2, \dots\}$ is the initial marking.

As Figure 18 shows, a portion of building design process is represented in a graph with the above elements. To explicitly address the stakeholders in design process, each event and task has a set of stakeholders associated. P_1, P_2, P_3, P_4 denote project manager, design consultant, market surveyor, and architect respectively, which are marked on top of the events and tasks. At the beginning of design, the tokens are only contained in the beginning states (S1 and S2). After stakeholders performed the tasks, the tokens from the leftward events can be transferred to the rightward events. M_0 is defined as the initial marking of a CDPN, which is a vector containing the token number for each event. For instance, at the beginning M_0 equals $[1 \ 1 \ 0 \ 0 \ 0 \ 0]$ since only event 1 and 2 possess tokens. If M_0 equals $[0 \ 0 \ 0 \ 0 \ 0 \ 1]$, that means all of the tasks shown in the graph have been conducted since the token is only presented in the last event.

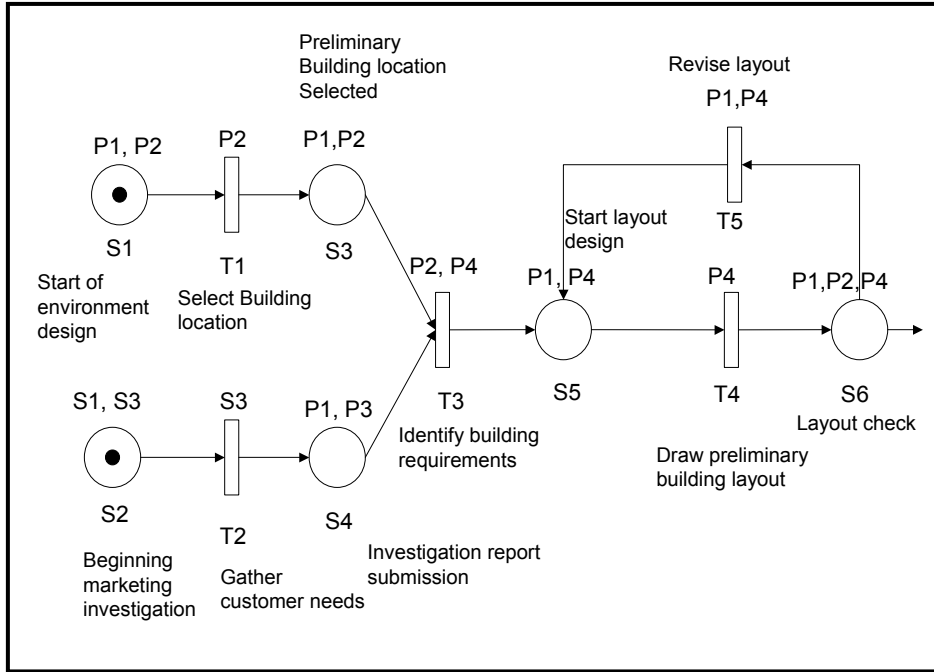


Figure 18. Collaborative design process net.

The input and output relationships between task and events are denoted as:

${}^{\circ}S(t)$ the set of input states of task t , (i.e., the set of $\{s \mid (s, t) \in A\}$),

$S^{\circ}(t)$ the set of output states of task t , (i.e., the set of $\{s \mid (t, s) \in A\}$),

${}^{\circ}T(s)$ the set of input tasks of state s , (i.e., the set of $\{t \mid (t, s) \in A\}$),

$T^{\circ}(s)$ the set of output tasks of event s , (i.e., the set of $\{t \mid (s, t) \in A\}$).

It is clear that finishing a task t consists of transforming the initial marking M_0 of the CDPN into a new marking M_{i+1} . Firing a task $t \in T$ includes two operations, which are removing a token from each $s \in {}^{\circ}S(t)$ and adding a token to each $s \in S^{\circ}(t)$ (assuming each arc has weight one). It could be formally defined as follows.

Definition 2: A task can be fired in a state M_i iff $\forall s \in {}^{\circ}S(t) : M_i(s) > 0$. The firing of a task leads to the next state M_{i+1} , which can be calculated by:

$$M_{i+1}(s) = \begin{cases} M_i(s) - 1 & \text{if } s \in {}^{\circ}t \\ M_i(s) + 1 & \text{if } s \in t^{\circ} \\ M_i(s) & \text{otherwise} \end{cases} \quad \text{Eq. 10}$$

Thus, the execution of a design process is represented by a task firing sequence $\sigma = \langle t_1, t_2, \dots \rangle$, which relates to a transformation of the marking $M_0 \rightarrow M_1 \rightarrow M_2 \rightarrow \dots$.

The process incidence matrix $U = [u_{i,j}]$ is defined to represent the relationship between tasks and events in a CDPN.

Definition 3: An incidence matrix $U = [u_{i,j}]$ is defined over all of the states $S = \{s_1, s_2, \dots, s_n\}$, and the tasks $T = \{t_1, t_2, \dots, t_q\}$ where:

$$u_{i,j} = \begin{cases} 1 & \text{if } t_j \in {}^\circ T(s_i) \\ -1 & \text{if } t_j \in T^\circ(s_i) \\ 0 & \text{otherwise} \end{cases} \quad \text{Eq. 11}$$

For example the $(n \times q)$ incidence matrix of the above graph is:

$$U = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

The relationship between state transformation and incidence matrix can be expressed in the following transformation equation:

$$\textbf{Proposition 1: } M^T = M_0^T + U \bullet V_\sigma^T \quad \text{Eq. 12}$$

In Equation 3, $V_\sigma = [v_1, v_2, \dots, v_q]$ is a counting vector for a task firing sequence σ with the following definition:

Definition 4: The counting vector of firing sequence σ is defined as $V_\sigma = [v_1, v_2, \dots, v_q]$, where v_i is the number of tasks t_i included in σ .

Given the firing sequence $\sigma = \langle T1, T2, T3, T4, T5, T4 \rangle$ in the example, its counting vector V_σ equals $[1 \ 1 \ 1 \ 2 \ 1]$. Based on Equation 3, the final marking can be calculated as follows:

$$M^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$M = [0 \ 0 \ 0 \ 0 \ 0 \ 1]$ shows that only Event 6 is active, which means the process shown in the graph might be finished.

The task dependencies are also easy to be identified from a CDPN, which is denoted by task dependence matrix $M_T = [(mt)_{ij}]$. If the one of the output states of task i is within the set of task j 's input states (i.e., task i is immediately in front of task j), this situation is called "sequential dependency." Another situation is that two tasks are sharing the same input state or output state, which is named "joint dependency." In both cases, its dependency factor is set to 1. Otherwise, it is said that there is neither sequential nor joint dependency between the two tasks. In a task dependency matrix, it is easy to identify the critical tasks (e.g., T3), which relate to many other tasks.

Definition 5: A task dependency matrix $M_T = [m_{ij}^T]$ is defined over all of the tasks $T = \{t_1, t_2, \dots, t_q\}$ where:

$$m_{i,j}^T = \begin{cases} 1 & \text{if } S^\circ(t_i) \cap S(t_j) \neq \phi \\ 1 & \text{if } (S^\circ(t_i) \cap S(t_j) \neq \phi) \vee (S^\circ(t_i) \cap S^\circ(t_j) \neq \phi) \\ 0 & \text{otherwise} \end{cases} \quad \text{Eq. 13}$$

Also, to represent the assignment of stakeholders' tasks from the CDPN, a task assignment matrix is defined as:

Definition 6: A task assignment matrix $M_{TP} = [m_{ij}^{TP}]$ is defined over the stakeholder set $P = \{s_1, s_2, \dots, s_m\}$ task set $T = \{t_1, t_2, \dots, t_q\}$ with the value:

$$m_{ij}^{TP} = \begin{cases} 1 & \text{if } t_j \in \{t \mid p_i \text{ perform } t\} \\ 0 & \text{if } t_j \notin \{t \mid p_i \text{ perform } t\} \end{cases} \quad \text{Eq. 14}$$

Definition 7: A task role matrix $R_T = [r_{ij}^T]$ is defined over the stakeholder set $P = \{p_1, p_2, \dots, p_m\}$ task set $T = \{t_1, t_2, \dots, t_q\}$ with the value

$$\text{for } j = 1, 2, 3, \dots, \sum_{i=1, 2, \dots} r_{i,j}^T = n, \quad \text{Eq. 15}$$

n is the number of non - zero items in column j of M_{PT}

The stakeholder has stronger control to a task, the role ratio is bigger.

For example, the task dependence matrix, task assignment matrix, and the role matrix of the above **CDPN** can be derived as:

$$M_T = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad M_{PT} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad R_T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1.6 & 0.8 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 1.2 & 1 \end{bmatrix}$$

4.2.3 Building Design Process Diagram

Given the above definitions, a design process diagram, which has design tasks, states, and stakeholders depicted in a structured way, can be represented as a set of matrices. These matrices are used as the mathematical basis for the simulation/representation of the process model. As a collaborative design process representation tool, design process diagram is effective in capturing and presenting the dependencies among different stakeholders' design tasks. There are five steps to build the design process diagram:

1. Specify stakeholders
2. Specify process states
3. Specify process tasks
4. Specify dependencies
5. Generate a collaborative design process diagram.

There is no strict precedence among the above steps. In fact, the generation of a design process diagram is a highly iterative process. Identification of the participants is the critical step before modeling the design process. Before drawing the diagram, the stakeholders have to investigate which person will be involved in what tasks. Then, to generate the design process diagram, the design group, which involves the relevant "who," needs to identify and specify the critical states and tasks in the design process. Design states are those situations that are perceived by the stakeholders and can be used to depict the status of the design process. Design tasks are the activities conducted by stakeholders to gener-

ate design states. The dependencies among these states and tasks are the logical relationships among them. In a design process, there exist various dependencies among states and tasks, such as resource dependency, information dependency, and decision dependency. After the dependencies are depicted and articulated, they are used to relate design states and tasks together in the process diagram. Then, the stakeholders are marked on top of every task and state in the diagram. For instance, in a design process diagram depicting a design plan, the stakeholders associated with a design task are those who are assigned to the work. The ones associated with a design state have perception of that state and will determine its occurrence.

The design states and tasks along the time axis can be further arranged so that the time sequence among the tasks becomes clear. “Token” is used to identify the status of the current design process. The task can only be fired when each of its input events has at least one token. At the beginning of design, tokens only exist at the start states. Design process diagrams can be derived at different abstraction levels. The stakeholders with expertise toward a certain design task can specify the details of the expansion of a task. Then, with the similar steps to generate design process diagram, a hierarchy of process diagrams can be built. For example, T3 and T4 can be expanded to more specific tasks and events (Figure 19). For each of the abstraction links, there is also a set of stakeholders associated, who will be in charge of that abstraction link. In this example, the stakeholders who are assigned to the task have the right to specify the detail tasks and events.

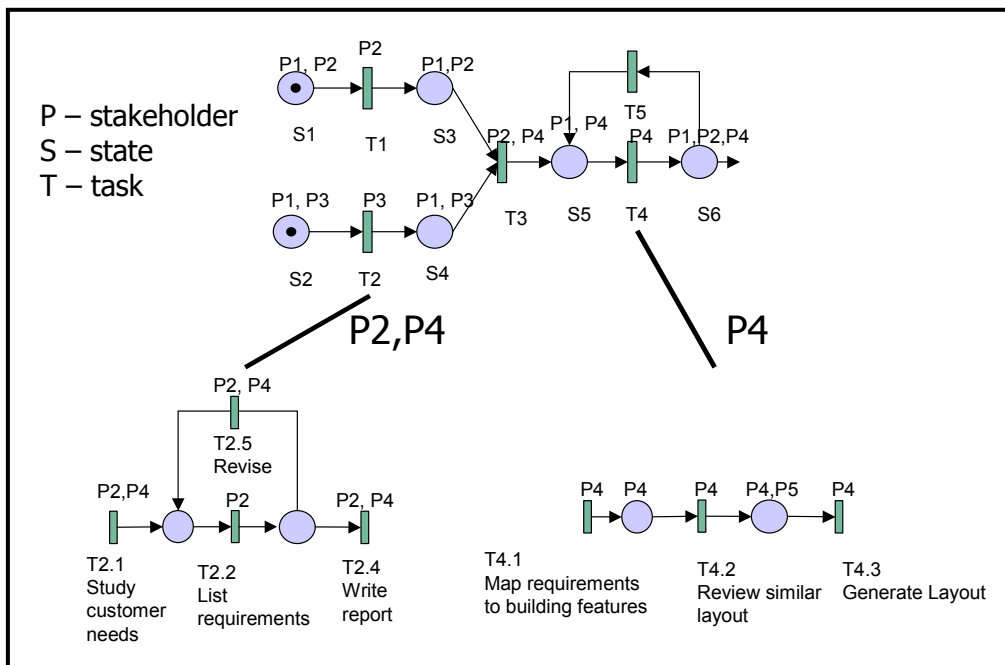


Figure 19. Design process diagram decomposition.

When a task is expanded to a new process diagram, it is possible that some new stakeholders will be considered and included in the process. For example, the architect (P4) wants to ask the customers' (P5) opinions about his building layout design. Then, the appearance of this new stakeholder (P4) might change the notes in the abstract process diagram. The roles of handling this are:

1. If a new stakeholder is involved in the new process diagram (detail process), the stakeholders who are in charge of the abstraction link will evaluate the role of the incoming stakeholders.
2. If the role of the incoming stakeholder is critical to the new process (not a specific task in that process), then his name should appear in the immediate upper level of the process.
3. The above step should be repeated in the upper level process until the role of the incoming stakeholders can be ignored.

In this example, since P5 has only a minor impact on the execution of task 4, he/she is not shown in the general process diagram. Whether to expand a task or not depends on the complexity of the process and requirements of the stakeholders. The objectives are to illustrate the process to a certain detail level so that the differences of stakeholders' perspectives are easy to be identified and design conflict can be detected.

4.3 Conflict Management

4.3.1 Basic Approaches

Conflict is the situation in which beliefs, perspectives, and/or decisions of one or more stakeholder(s) become mutually incompatible with respect to the satisfaction of some design requirements. Conflicts always occur in engineering design. Traditional approaches treat conflicts as abnormalities in the process, and devote resources to eliminate them as much as possible. Conflicts will occur even more in collaborative design as the complexity of the design process increases. It is necessary to clearly determine their roles in the design campaign. At different design stages, conflicts have different characteristics and should be treated differently. A major effort in the research is to investigate what constitutes a conflict and to understand the some common features among these descriptions. Then, one can propose a logic structure for these descriptions to form a few generic groups so that conflict taxonomy can be developed. The taxonomy of design conflict should be general enough so that it can be applied to various design problems. On the other hand, it should be expandable to accompany new con-

flicts involved in the practical design process, since it is impossible to foresee all types of conflicts before design.

Many methods and strategies have been developed to deal with different types of design conflicts. Each of them is particularly useful in its special situation. These conflict management (CM) strategies and the situations under which they will be effective must be clearly understood. These characteristics will help identify conflict management strategies, which are generic across a set of design conflict situations. Then a logic category for these strategies can be developed. Managing conflicts is never a purely technical task. The same conflict can be resolved differently at different corporations due to their different cultures. Understanding of engineering physics and corporate culture are both important to an efficient conflict management strategy. The question is how to capture those social aspects of the conflict management methods/strategies so that they can be used effectively in supporting collaborative design. Another important issue is how to provide not only the strategies for people to resolve the conflict, but also the methods to guide them to explore new strategies.

4.3.1.1 Artificial Intelligence Approach

Many AI researchers take problem-solving approaches to manage design conflict. Their approaches build searching algorithms, capture agent dependencies, or develop knowledge-base systems. Some of them view collaborative design as a distributed dynamic interval constraint-satisfaction problem and develop algorithms that use heuristics for distributed design (Tiwari and Gupta 1995). Klein (1991) introduced the concept of conflict resolution expertise. His approach used computational models that actually encode conflict resolution expertise more explicitly and use it to maintain the dependencies during cooperative problem solving. In his cooperative design model, design agents can be viewed as being made up of a design component that can update and critique designs, as well as a conflict resolution component that resolves design agent conflict. Dunsus tried to use many small, cooperating, limited-function expert systems to build an integrated system to investigate conflict. It provides ways of discovering and testing the components of negotiation, patterns of communication, functional primitives of design, and the types of knowledge needed (Dunsus 1995). Wong proposed a cooperative problem solving approach to handle the conflicts among distributed design agents (Wong 1997). He classified conflicts into schema conflicts, data conflicts, and knowledge conflicts, and proposed four modes of conflict resolution (Inquiry, Arbitration, Persuasion, and Accommodation).

4.3.1.2 Economic/Behavioral Approach

Other research works focus on the negotiation strategies of conflict management based on economic or behavioral theory. Bahler, Dupont, and Bowen (1995) introduced a protocol of evaluating compromise solutions to conflicts in collaborative negotiations. The protocol is based on the notions of economic utility by which design advice systems can recognize conflict and mediate negotiation fairly. The basic idea is to allow expressed preferences of design teams to be qualitative as well as quantitative. Another approach is concerned with global metrics for optimization, decision support, and negotiation. The coordination function is support by some optimization methodology, such as Pareto optimality and Multi-attribute utility (Petrie 1995). Game theory has been used as a typical method for generating compromise solutions in many research approaches. Vincent examined the role of game theory in the engineering design process in 1983. He examined the multi-criteria optimization task from the perspective of team design (Vincent 1983). A modified game theory approach to multi-objective optimization has been used in conflict resolution as a combination of optimization steps (Rao and Freiheit 1991).

4.3.1.3 Explicit Engineering Design Models

The engineering design models also have some mechanisms applicable to managing design conflict. For instance, QFD (Quality Function Deployment) is a structured process that establishes customer value using the “voice of the customer” and transforms that value to design, production, and supportability process characteristics (Hauser and Clausing 1988). The result of QFD analysis is a systems engineering process that ensures product quality as defined by the customers. This is essentially a methodology to solve/mitigate the conflicts among the diversified customer needs, which mainly exist in the early phases of engineering design. The Independence Axiom in Axiomatic design (Suh 1990) states that the independence of Functional Requirements must be always maintained to reduce the random search process and minimize the iterative trial-and-error process. It claims that a product design that ignores this axiom will face substantial conflicts.

To summarize, the above Section discusses the contributions and limitations of the previous works on CM. The AI-based approaches and the economic/behavioral approaches mainly focus on conflict management itself rather than its origin and influence in the whole process of engineering design. However, as both the design process being conducted and the design environment are evolving, it is difficult to use one category of methods to deal with all of the conflicts. Conflict management is highly coupled with design process modeling and or-

ganization transforming. For example, although game theory provides quite complicated methodologies to solve the conflict problems in economics, their use in engineering design requires a deep understanding of the nature of design process to adapt the game-playing models. The rightness of the analysis (e.g., to build utility functions and to determine the strategies of players) depends on the comprehension of the attributes of design participants, the design product, and the design situation. The other deficiency of these approaches is that they mainly can contribute to conflict resolution (and somewhat detection), rather than conflict prevention (which is a very effective way to handle conflict). Using the engineering design models to manage conflict is a prospective approach to solve the problem. But most of the current design models do not take supporting collaborative design as one of their primary goals. They assume that strictly following their guidelines will significantly reduce the chance of conflict.

4.3.2 Development of a Methodology for Collaborative Process Engineering with Conflict Management Capability

The central focus of this research is to enhance the efficiency of collaborative design by controlling the interplay of the design process, of conflicts, and of stakeholders' perspectives. Although "perspective" is critical for managing design interactions, traditionally it is not explicitly modeled in information systems. Thus, it is critical to form the design perspective in a structured way so that the control mechanisms can be added. Also, to relate perspective models with design process models, a design conflict model is indispensable for manipulating the entire design system. This methodology not only uses an effective way of building perspective models and process models, but also provides an analysis procedure to identify the control factors in collaborative design. The control mechanism takes the social interaction features as means to detect and resolve the conflicts. Also, it will help stakeholders to represent and adjust the design processes according to the analysis results. By adding "who" into technical decisionmaking process, it becomes a platform to support the con-construction among stakeholders.

Figure 20 shows the overview of the methodology. Since modeling stakeholders' perspective is the key issue in this methodology, the first step is to identify the stakeholders involved in the design and to help them realize their roles and purposes. These stakeholders jointly build the baseline process model, which is represented as a design process diagram. Then, by constructing the concept structure by the group interaction, the stakeholders can systematically build the Perspective Models.

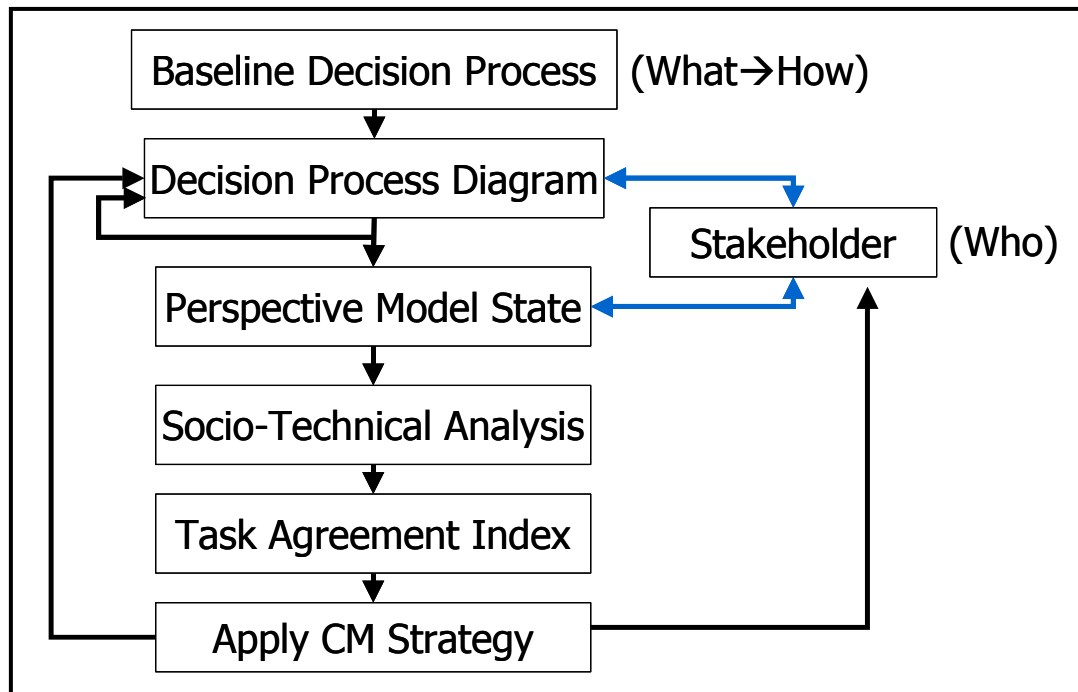


Figure 20. Steps of methodology.

The Design Process Diagram and Perspective Model States are transformed to matrices. They become the inputs of the socio-technical analysis. This analysis method will generate the Task Agreement Index to detect the conflicts in design process. Also, by referencing the analyzing procedure, it is possible to find conflict management strategies and suggest that stakeholders apply them to affect the design process and design perspectives.

4.3.3 Development of a Method for Real-Time Socio-Technical Analysis

A set of matrices can be derived from the Process Model Diagram, the Concept Structure, and the Perspective Model Diagrams. These matrices provide the basics for the analysis work.

The Incidence matrix represents the design process diagram in terms of the dependencies among the states and tasks. It also provides a way for describing the design process information in a computational format. It is possible to reconstruct the process diagram in the computer from this matrix.

The Task Dependency matrix captures the “rate of influence” among the tasks. The function of this matrix is to identify the consequence of execution or corruption of one task during the design process. The affected tasks can be directly identified from this matrix.

The Task Assignment matrix relates the stakeholders to the design tasks. In practice, the stakeholder who has right to manage the design group and design process controls this matrix. By looking at each column of task assignment matrix, it is clear to notice the stakeholders involved with a task. The rows within the task assignment matrix clearly depict stakeholders' jobs.

The State Participation matrix represents the stakeholders' participation into different states. A stakeholder "participating" in an event means that he/she plays some roles in deciding the status of an event and can make decisions whether an event can happen. By comparing the task assignment matrix and event participation matrix, the differences among stakeholders' roles are revealed. For example, although stakeholder P1 has only Task T5 assigned, he will participate in all of the states within the design process. It is clear that his role is to manage or monitor the process rather than to execute the process.

The Task-Concept matrix shows which concepts are involved (used, generated) in which task. It is generated from the concept structure and the design process diagrams. For each of the tasks, the stakeholders can specify a set of concepts that will have influences to its execution.

The Concept-Stakeholder matrix is generated from the perspective model state diagrams. It illustrates the stakeholders' perspectives toward the design concepts.

The Task Role matrix can be specified based on the task assignment matrix. It has a structure similar to the task assignment matrix, but with a different value. It is used here to depict the difference of stakeholders' influences toward all of the design tasks (Figure 21).

4.3.4 Automation of Conflict and Decision Support among Stakeholders for any Given Task

Stakeholders' perspectives are changed by interaction with others during the design process. To analyze the relationships between design process and design perspective, a mapping mechanism is developed to link the Process diagram to the Perspective State diagram. Since each design task and state will handle a set of design concepts, the relationships among task, states, and concepts can be defined. On the other hand, the Role matrix, Task Perception matrix, and Task Consistency vector were defined to represent the views of stakeholders toward the design tasks.

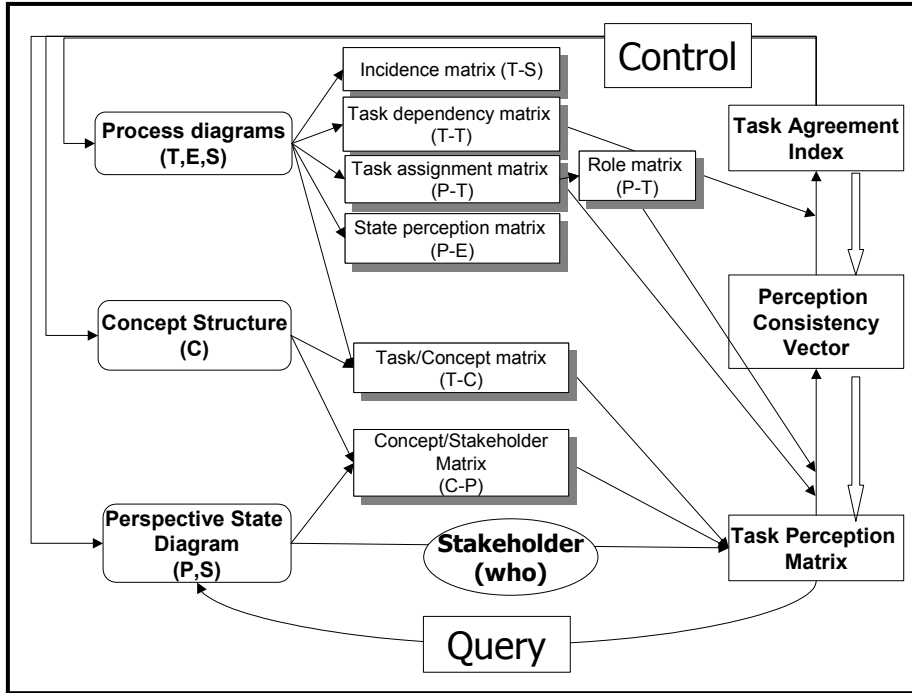


Figure 21. Matrices used in analysis.

The Perception matrix (V-T matrix) is generated by asking stakeholders to view each other's Perspective Models and to declare their attitude toward each task. As Figure 22 shows how, for each of the tasks, the concepts involved in that task from the T-C matrix can first be found. Then, from C-P matrix, all of the stakeholders who have perspectives toward a concept can be identified. After classifying stakeholders' attitudes toward all of the concepts in a task, one can generally tell which stakeholders have the same/different attitudes toward a task. It should be noted that in the Perception matrix, a person may have an attitude, but may not be assigned to the design tasks. When stakeholders view the perspective models for each others, it is possible to ask them to declare their attitudes toward the design tasks based on: (1) his own perspective model related to that task, (2) his view toward others' perspective models relate to the task.

Definition 8: A task perception matrix $M_{VT} = [m_{i,j}^{VT}]$ is defined over the stakeholder set and task set $P = \{p_1, p_2, \dots, p_m\}$ and the task set $T = \{t_1, t_2, \dots, t_q\}$ with the "attitude" value:

$$m_{i,j}^{VT} = \begin{cases} 1/n_j & \text{if } s_i \uparrow t_j \text{ (}\uparrow \text{ means positive attitude)} \\ -1/n_j & \text{if } s_i \downarrow t_j \text{ (}\downarrow \text{ means negative attitude)} \\ 0 & \text{if } s_i \text{ has no perception to } t_j \end{cases}$$

n_j is the number of non - zero $p_{i,j}$ in column j

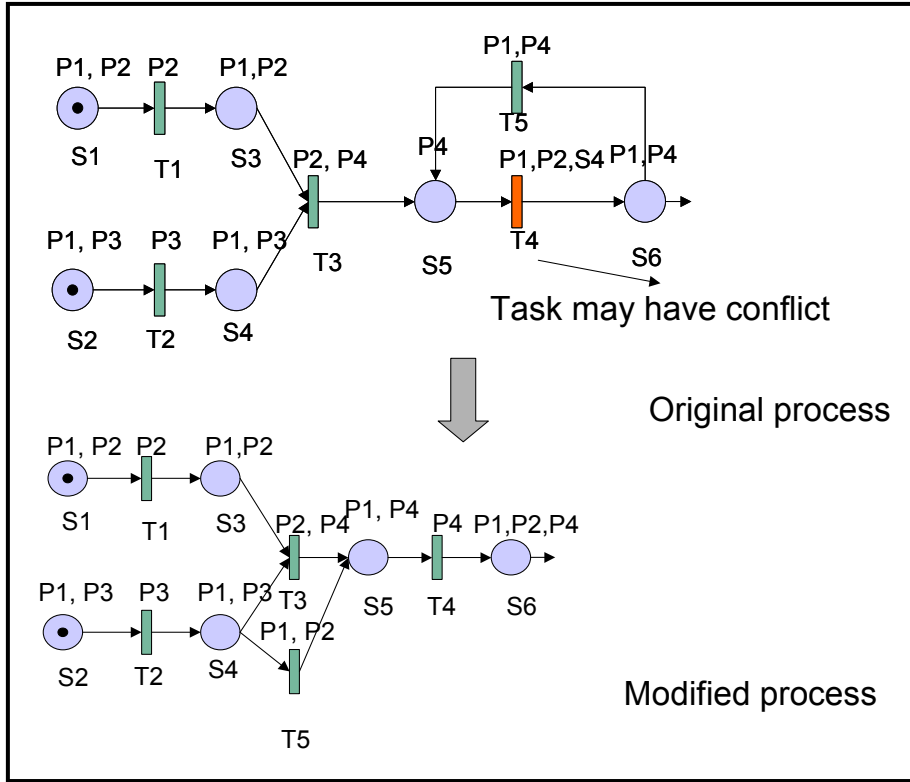


Figure 22. Adjusting process to manage conflict.

An example of the task perception matrix is shown below.

$$M_{VT} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1/4 \\ 1/2 & 0 & -1/3 & 1/2 & 1/4 \\ 1/2 & 1 & 1/3 & 0 & -1/4 \\ 0 & 0 & 1/3 & -1/2 & 1/4 \end{bmatrix}$$

Definition 9: A task consistency vector V_{TC} can be derived from the perception matrix:

$$V_{TC} = \tilde{M}_{VT}^T V_I$$

$\tilde{M}_{VT}$ is the declared task perception matrix, which can be derived from the following equations:

$$\begin{aligned} \tilde{M}_{VT} &= \hat{M}_{VT} \wedge R_T \\ \hat{M}_{VT} &= M_{VT} \wedge M_{TP} \end{aligned}$$

The operator $\wedge$ is defined as:

$$[c_{i,j}] = [a_{i,j}] \wedge [b_{i,j}] \quad \text{where:}$$

$$c_{i,j} = (a_{i,j} \times b_{i,j}) \frac{\# \text{ of non - zero in } a_{*,j}}{\# \text{ of non - zero in } b_{*,j}}$$

4.3.5 Computation of Task Consistency Indicating When-and-If “Control” May Be Necessary

It is then possible to use task perception matrix, task assignment matrix, and the role matrix to build the task consistency matrix. To get the task consistency matrix, three steps are needed. Firstly, the task perception matrix is “filtered” through the task assignment matrix to represent all of the noticed attitudes of stakeholders.

$$\begin{aligned} \hat{M}_{VT} = M_{VT} \wedge M_{PT} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1/4 \\ 1/2 & 0 & -1/3 & 1/2 & 1/4 \\ 1/2 & 1 & 1/3 & 0 & -1/4 \\ 0 & 0 & 1/3 & -1/2 & 1/4 \end{bmatrix} \wedge \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1/2 \\ 1 & 0 & -1/2 & 1/2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 & 1/2 \end{bmatrix} \end{aligned}$$

Secondly, $\tilde{M}_{VT}$ is derived from the above matrix and the task role matrix R_T . The weighting factors of influence toward design tasks are considered in this stage.

$$\begin{aligned} \tilde{M}_{TP} = \hat{M}_{TP} \wedge R_T &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1/2 \\ 1 & 0 & -1/2 & 1/2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 & 1/2 \end{bmatrix} \wedge \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1.6 & 0.8 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 1.2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1/2 \\ 1 & 0 & -0.8 & 0.4 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & -0.6 & 1/2 \end{bmatrix} \end{aligned}$$

Then, the task perception consistency vector is calculated by adding all of the values in each column of the declared perception matrix. This means to consider all of the perceptions from the stakeholders for a task.

$$V_{TC} = \tilde{M}_{VT}^T V_I = \begin{bmatrix} 0 & 0 & 0 & 0 & 1/2 \\ 1 & 0 & -0.8 & 0.4 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & -0.6 & 1/2 \end{bmatrix}^T \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -0.6 \\ -0.2 \\ 1 \end{bmatrix}$$

Thus, it is easy to identify the low consistency (or potential conflict) on task T4. The above task consistency vector only address the conflict caused by stakeholders perspectives. By considering the dependencies among tasks, one can further derive the intensity of conflicts within tasks. It is done by calculating the task agreement matrix.

$$T_c^* = M_T \otimes |T_c| = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ -0.6 \\ -0.2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0.6 \\ 0.12 \\ 0.2 \end{bmatrix}$$

$\otimes$ means to multiply all of the dependent tasks' consistency factors together. The lower the value in T_c^* , the higher the conflict intensity of that task.

4.3.6 Development of Control Strategies

After identifying the tasks with conflict, one can check back to find the point to control the conflict. The procedure of the above socio-technical analysis points out several ways to manipulate design conflict based on the above mechanism. Here the controls are added to affect the task consistency since the conflict ratios are implied in that vector. The basic method is to use various means to manipulate the three inputs (i.e., the process diagram, the concept structure, and the perspective state diagram). This provides a guideline to explore more strategies to handle the design conflicts in a comprehensive way.

4.3.6.1 Detecting Conflict

By evaluating the feasibility of design tasks, which are in the plan, but not being conducted yet, some conflicts can be prevented by notifying the stakeholders of potential conflicts earlier in the process.

4.3.6.2 Affecting Stakeholders Perspectives

By providing information to stakeholders, it is possible to change the perception matrix and therefore to increase the consistency of a task. That requires directly adjusting the perspectives (their content, purpose, and context) to maintain the integrity of the opinions toward design tasks.

4.3.6.3 Changing the Role of a Stake Holder

The task assignment matrix can be changed to modify stakeholders' roles to change the resulting task consistency vector.

4.3.6.4 Adding or Removing Stakeholders from a Given Task

It is even possible to add/remove stakeholders associated with a task to avoid the conflict situation.

4.3.6.5 Changing the Process Diagram To Reduce Task Conflicts

It is possible to rearrange design events and tasks in the process to modify both the concepts, structures, and PMSDs to control the occurrence of design conflicts.

A scenario used here introduces different conflict management approaches and their potential influences to the conflict problem. Necessary assumptions are made to simplify the analysis.

The conflict is described thus:

At the first meeting the client's design consultant states that the building is to be placed at one location on the site. The architect listens to the client's reasoning but notes that this location is not ideal from either an aesthetic or a functional point of view, since it would be too close to a major road intersection.

The Perspective State Diagram can help to highlight the dependencies and differences of views among the stakeholders. At certain design stage, conflict can

be detected by tracking and comparing the “perspective state” of different stakeholders. This section approaches two situations separately. The first is the one without assistant of the conflict detection methods. The second considers the application of these approaches.

According to the design process, the PSMDs of the design consultant and architect can be depicted at the early design stage. The design consultant first proposes the design plan; the clients and architect accept without detailed reasoning. Since there is no interaction at this stage, most of the design perspective elements of the architecture and the client are left empty. At the conceptual design stage, the contents of the PMSD elements for each stakeholder are increased. However, due to the loss of coordination, they do still not notice the difference and dependence until the next stage. When the architect is discussing the detailed features of the building with the design consultant, they realize that there might be a conflict. In this example, when the PSD elements are compared with each other at early stage, although there is still no direct meeting between the stakeholders, the system can detect a potential conflict during the design process. The attitudes of stakeholders are shown in the following matrix:

$$M_{TP} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1/4 \\ 1/2 & 0 & -1/3 & 1/2 & 1/4 \\ 1/2 & 1 & 1/3 & 0 & -1/4 \\ 0 & 0 & 1/3 & -1/2 & 1/4 \end{bmatrix}$$

Assume that Stakeholders S2 and S4 are given equal right for conflict detection. Then a potential conflict in task T3 can be detected.

$$\begin{aligned} \tilde{M}_{TP} = \hat{M}_{TP} \wedge R_T &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1/2 \\ 1 & 0 & -1/2 & 1/2 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 & 1/2 \end{bmatrix} \wedge \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0.8 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1.2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1/2 \\ 1 & 0 & -0.5 & 0.4 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & -0.6 & 1/2 \end{bmatrix} \end{aligned}$$

To control the design process and manage this design conflict, the ill-structured design process must be transferred to a good one. An approach to achieve this is to derive the mechanism to modify the consistency vectors. The obvious way is to change the perspectives of the stakeholders so that the task consistency is im-

proved. For example, if it were suggested that stakeholder P4 discuss Task T3 with P2, they may share some ideas and reach a consensus. That means the inconsistency of Task T3 have been reduced. The perception matrix is changed to:

$$M_{VT} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1/4 \\ 1/2 & 0 & 1/3 & 1/2 & 1/4 \\ 1/2 & 1 & 1/3 & 0 & -1/4 \\ 0 & 0 & 1/3 & -1/2 & 1/4 \end{bmatrix}$$

Since the designers' perspective will largely depend on the interactions among the design tasks, arranging the design process to a desired manner becomes an effective approach to coordinate the perspectives of the stakeholders. This methodology provides an algorithm to help people rearrange the design process to reconcile the design perspectives and resolve conflict. When conflicts happen, the system can identify the affected design tasks in process view from the dependency matrixes. Then a possible solution is to modify some tasks or search for a new task to remove the conflict source. Sometimes, effective handling of design conflict will eliminate the task iterations and shrink the design process. For instance, if the methodology detects a conflict in task T4 of the process diagram (Figure 22), it will suggest several ways to control that conflict by examining the perspective models involved in the related tasks. A possible solution is to add a task before state S5 so that the necessary design information is derived earlier. The process is then modified to prevent a conflict.

4.4 The STARS Prototype System

A prototype system, named STARS (Social-Technical Analysis Research System), has been developed at the IMPACT lab. It provides a web-based environment that supports the design process representation, conflict management, and knowledge integration within the design group. Its objective is not to substitute the current CAD or MIS tools, but to fill the gaps of design coordination that are ignored in the current design support technologies. Stakeholders' perspectives are modeled in the system and their roles in the design tasks are depicted. Communication tools with networking and server-client database accessing functions support the stakeholders to declare, share, and modify their perspective models.

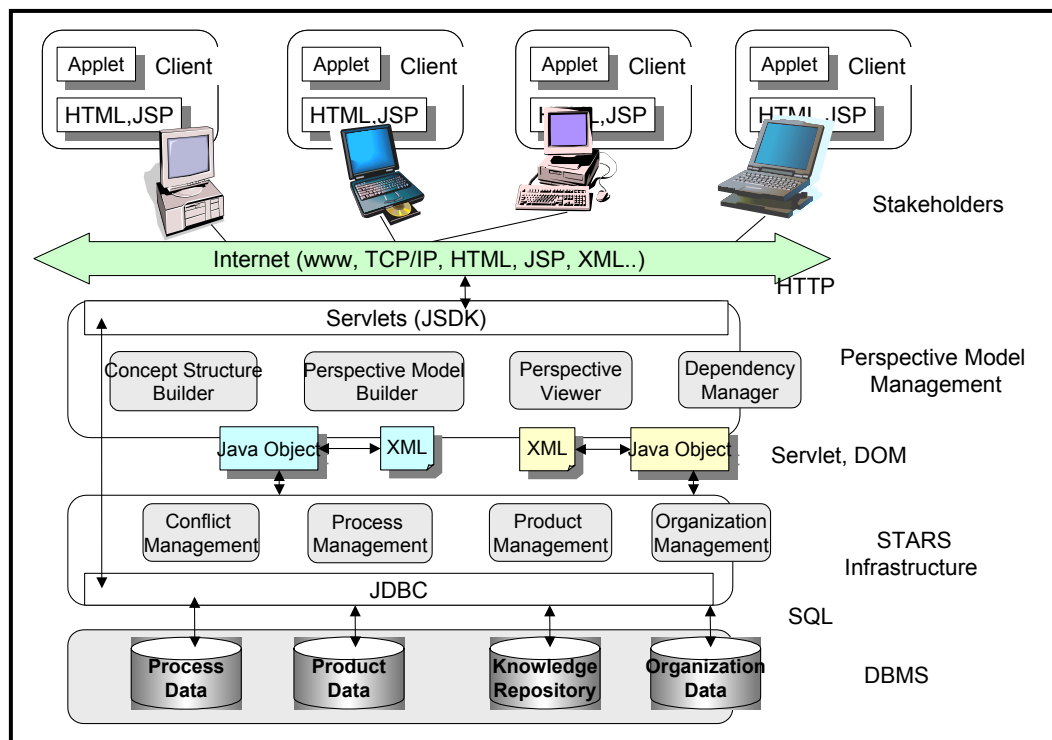


Figure 23. Overview of STARS system architecture.

4.4.1 System Architecture

As Figure 23 shows, on the client side, each stakeholder uses a set of unique web-based interfaces to declare his/her perspective and access design information. These HTML and Java applet Swing interfaces in the Internet browsers are connecting with a perspective management system on the server. To facilitate the applets' communication with the server applications, Java Servlets (McPherson 1999) are implemented on the server side. By using HTTP for communication with servlets through serializable java object, the applets are made compatible with network firewalls. Stakeholders access the product data, the organizational data, and others' perspective models when they operate their own workspaces. In the perspective management system, perspective models are transformed to XML files by Java Servlets. When analyzing the data within those files, the system uses an XML parser and DOM API to create Java object models. Thus, perspective models written in XML are transformed and can work with other Java programs.

The server provides several subsystems (e.g., Conflict Management, Process Management, Product Management, and Organization Management) to support the interactions and negotiation among stakeholders. While the functional structure and form of the product are built during design, a conflict management

system analyzes the situations in which design conflicts have occurred and applies management strategies (i.e., detection, prevention, and intervention).

The process management system is a Petri net based modeling tool for design planning, scheduling, and simulation. The product management system controls the access to various technical data and maintains its integrity. It can also exchange product data with other CAD applications by using standard product data format, such as STEP (ISO 1994). The organization management system tracks the organization structure evolution and can communicate this information to other existing management information systems. The clients can also directly access some design data from the system database. This access is supported by the use of Java servlets and JDBC to communicate with SQL. The system knowledge repository stores and updates the conflict management guidelines. Other systems can use these guidelines, and they can be transformed to XML format and fed back to the perspective interfaces of the stakeholders.

STARS has some unique features. First, by providing web browser interfaces to explicitly capture the perspectives of the stakeholders and assist their interactions, the system takes responsibility to detect conflicts among stakeholders' perspectives and thus support their knowledge integration. Second, it helps the design group to manage the decision process for design coordination by referring to the conflict management strategies. Third, it explicitly maintains social dependencies and organizational changes within the design group. Fourth, STARS traces how design knowledge is merged into the design process and captures new concepts and ideas.

4.4.2 STARS System Components

4.4.2.1 Design Perspective Model

One of the most important features of STARS is to support the representation and management of the design perspective interactions. A Perspective Model in the system is a cluster of data that represents the design perspective information (i.e., expertise, purpose, and context information toward certain concept) of a stakeholder. To get the Perspective Models, the stakeholders first collectively build a concept structure by using Concept Structure Builder (Figure 24). Then, by using the Perspective Model Builder, they declare their perspective information according to the concepts that they recognized. Concept Structure is an organization of the ontologies that stakeholders propose and use in collaborative design.

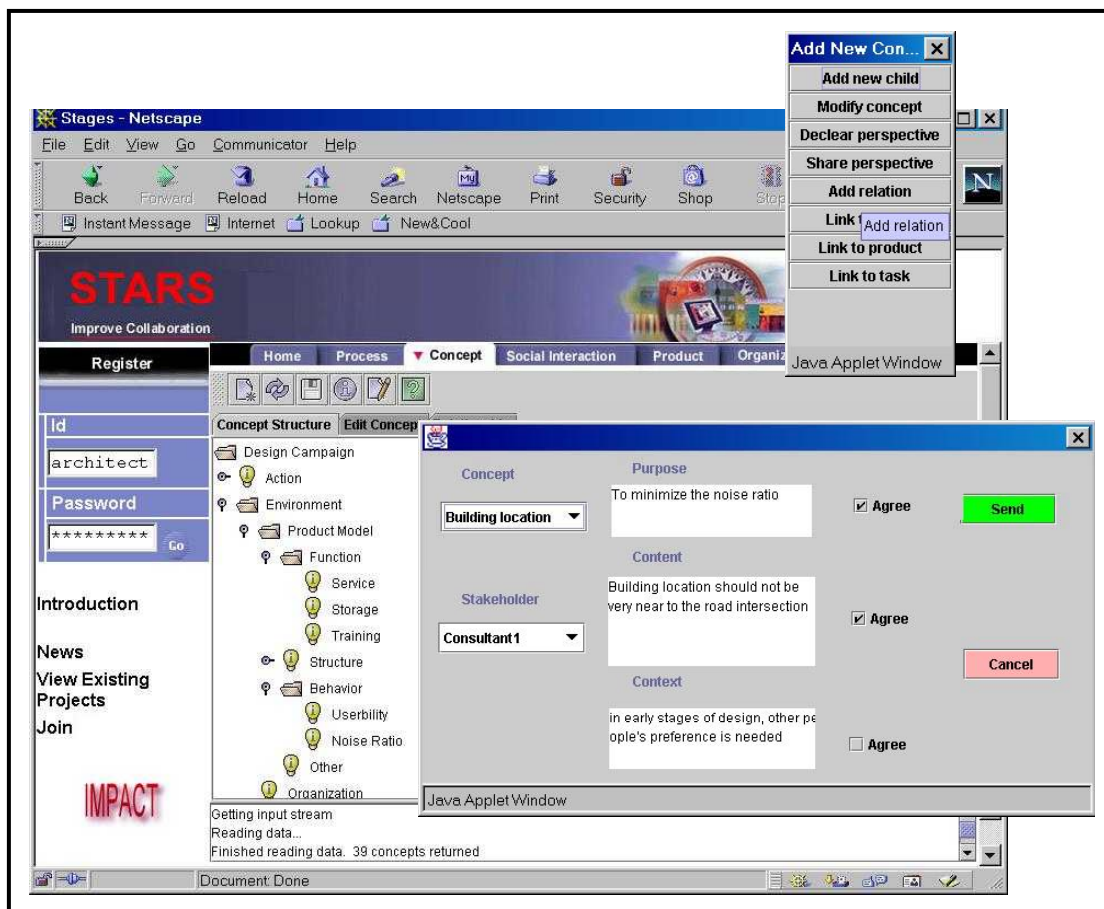


Figure 24. STARS concept structure interface.

The system uses both top-down and bottom-up construction methods (Vet and Mars 1998) to help stakeholders build the concept structure. It first provides some templates (e.g., “product function template,” “design organization template,” “conflict types,” etc.) for the stakeholders to clarify the concepts.

For more routine design, many concept structures can be extracted from previous CAD product models and preloaded in the system. When an individual proposes a new concept, he/she should first consider whether there are similar concepts in the structure. Thus, only the novel concepts can be specified and added. When stakeholders propose new concepts in design process, the concept structure is updated and used to systematically organize these concepts and their relationships. The concepts are often best generated by individuals, while the group often best performs the concept selection and enhancement. Therefore, the concept is classified into two types. “Shared concepts” are those that have been well-defined from previous design and have widely accepted meaning among the stakeholders (e.g., “Requirement List,” “Function Structure,” etc.). Only some particular stakeholders perceive “Private Concepts.” Their names or meanings are not expressed around the group.

The concept structure is built by the stakeholders. The system provides a template for them to add more concepts. Stakeholders can declare their perspectives toward a certain concept and view others' perspective information.

The perspective management system in STARS maintains a Perspective Model State Diagram for each stakeholder at a certain time. A Perspective State Model Diagram consists of all of the perspective models of a stakeholder. It is an "overall picture" of one's perception in design campaign. The changing of Perspective Model State Diagrams describes the adaptation and evolution of stakeholders' perspectives during design process. By building and manipulating the Perspective Model State Diagrams of different stakeholders, the system can help stakeholders detect and evaluate the interdependencies among their design activities (Figure 25). Dependencies of the Perspective Models are maintained in the concept structure. The change of design perspective models will be reflected in the concept structure. During design, stakeholders can perceive others' perspective models if there are dependencies.

As shown in Figure 25, a Perspective Model represented by XML can contain a group of criteria depicting his/her valuations toward the design concept. The Criteria can either be a simple string or a defined criteria object, which has specific format defined by the users. The Content in the perspective model illustrates what the stakeholder knows about the design concept. It may contain several options the stakeholder proposes to satisfy the purpose.

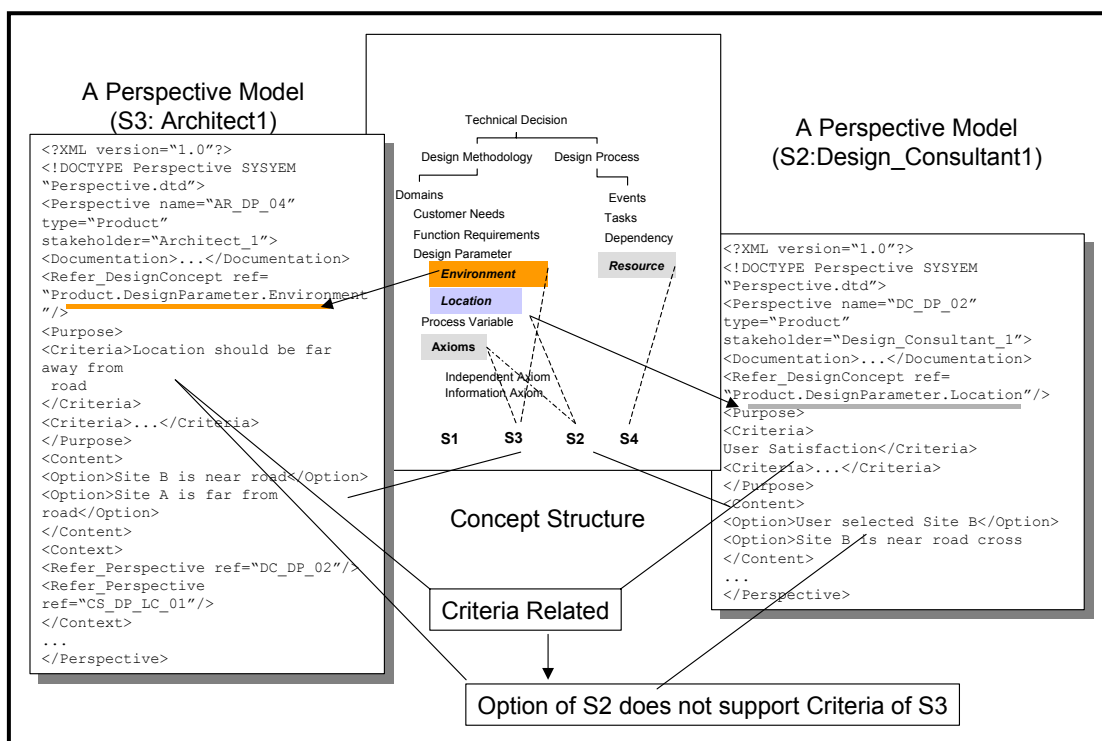


Figure 25. Perspective model dependencies.

The Context of a perspective model indicates the situational condition. It specifies the linkages to other related perspective models, which may have different time and stakeholder attributes. Stakeholders can construct and modify the data in their perspective models. They can also view and evaluate other's perspective model by referencing the concept structure. The system can notice the links among their perspective changes by tracking the modification of the concept structure.

Dependencies of the Perspective Models are maintained in the concept structure. The changes of design perspective model are reflected to the concept structure. Stakeholders can view others' perspective models if there are dependencies.

4.4.2.2 Design Process Model

Concurrency is normally encouraged in collaborative design since parallel task execution may reduce the design time and save resource. However, stakeholders are not always aware of the effects of their actions in such parallel activities. They may have limited knowledge about the overall design situation and ignore the various dependencies among local decisions. Therefore, coordination among design tasks is critical. On the other hand, note that design activities are relatively complex and unstructured compared with other processes (e.g., Computer programming, Manufacturing system). It is infeasible to force distributed stakeholders to share a very static and specific process plan. Hence, rather than to prescribe the detail jobs for all of the stakeholders, the process views provided by STARS (Figure 26) are intended to help them notice what is happening and who is doing what at any time.

Stakeholders can jointly build the design process model. The system will evaluate the task consistencies based on the analysis.

In this Petri-net-like modeling tool, the collaborative design process is represented by an organization of states and tasks. Time and resource consumption is associated with each task. As shown in Figure 26, a process diagram can be built from the formalized database. It shows the sequence of tasks and the corresponding responsible stakeholder(s), represented as horizontal axis on the diagram. The sequence of tasks on the diagram represents a chronological execution of the activities while the vertical axis shows stakeholders. Each horizontal flow of tasks corresponds to a given stakeholder playing a particular role in the design campaign. By assigning activities to a stakeholder, the relationships among stakeholders in the process become very apparent.

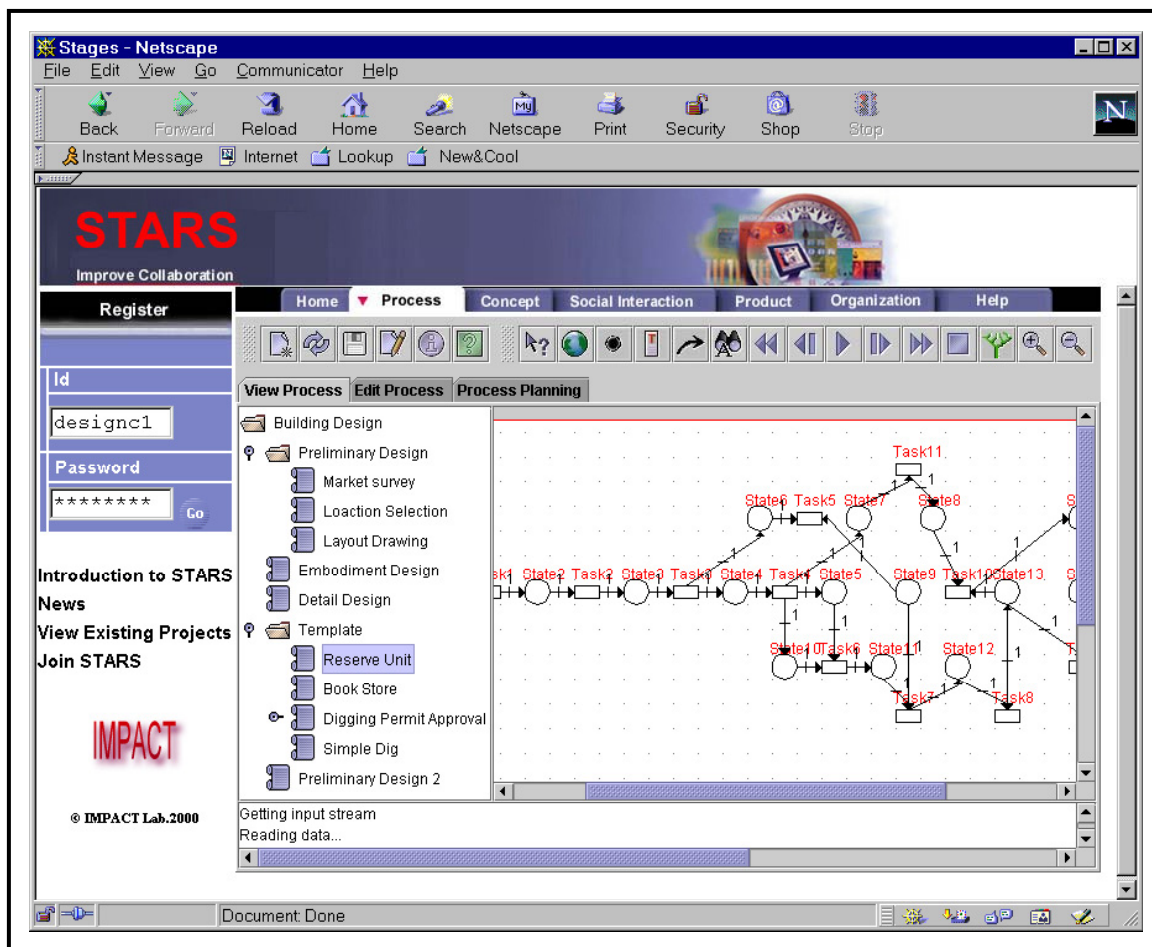


Figure 26. Joint construction of process model.

4.4.2.3 Process and Conflict Management Support

Given the process diagram, the concept structure, and the perspective models, their dependencies can be represented as several matrices (e.g., T-T, T-S, C-S, etc). Controlling the interplay among these three models provides various conflict management methods. At certain design stages, the design conflict can be detected by tracking and comparing the “perspective states” of different stakeholders. If design perspectives are not tracked, due to the loss of coordination, the chances of noticing the inconsistency and dependence are relatively small. Then, some design deficiencies are not noticed until conflicts occur at later stages.

A mechanism to track stakeholders’ Perspective Model State Diagrams is implemented by generating and analyzing these perspective models. By referencing the concept structure to the XML files, the perspective management system in the system can process the information within the perspective models. It first detects the dependencies among the perspective models and then presents re-

lated items to different stakeholders. Through Perspective Viewer, the stakeholders can detect the inconsistencies of their understandings with each other. For example, at the conceptual design stage of a building design problem, the perspective models of a design consultant and an architect are captured and represented as a series of perspective models. The two stakeholders have perspective models to their own local concepts respectively. From the concept structure, the system identifies that Stakeholder S2 (a design consultant) and S3 (an architect) have two perspective models sharing the same parent concept.

One of the criteria in the perspective models of S2 is to have the building location far away from the road. However, S3 assumes that the users of the building prefer the location B, which is near the street crossing. When the XML files are parsed for analysis, the system will notify both stakeholders of the dependence and present them with each other's related perspective information. In this case, S2 may detect that his content does not match the purpose of S3. A conflict will be initiated in the system. Conflicts can be classified in the system according to the involved stakeholder and related concepts (Lu and Cai 1999). Given a type of conflict, some general strategies can be found from the knowledge repository in the system, which are human-readable. Also, the system will suggest that the stakeholder review the concept structure and tell him which stakeholders also have perceptions of that concept. Then, by studying other's relevant perspective information and following the suggestions, the design consultant may notice his ignorance of an important design requirement and thus change his perspective. After that, the two perspective models will become more compatible. Therefore, although there is no face-to-face meeting between the stakeholders, the system provides a platform for them to handle design conflict by using the dependencies as anchor points to integrate their individual perspectives and form a shared concept and meaning.

Since conflict management requires stakeholders' negotiation, the objective of the conflict management function in the system is to help stakeholders handle the conflicts systematically rather than to automatically resolve them. By maintaining the dependencies among design concepts, design process, and perspective models, the system helps the stakeholders to notice the existence of some potential conflicts. Rather than always treating a conflict as an abnormal situation, this work takes a more comprehensive approach. In the early design stage, conflicts can be seen as an opportunity to identify team deficiencies and to create ideas. At later stages, conflicts should be prevented or resolved for the sake of efficiency. Some of the general conflict management guidelines (e.g., adjust design process, find relating stakeholders, and change their roles) are implemented in the system. Working with STARS, the users can implement more guidelines based on their experiences. When stakeholders have successfully resolved a con-

flict, they may add more suggestions in the knowledge repository to facilitate the design in the future.

In STARS, perspective reconciliation is also achieved by supporting social interaction. It helps the stakeholders to view the evolution of organization structure and realize their own roles in the project. The system keeps a social network model (Wasserman 1994) inside, which depicts stakeholders' perception of existing of each other. When two stakeholders have contact, there is a link associated with them. By helping stakeholders to share their design perspectives, the system creates channels of communication to illustrate their situation and capture background knowledge. The stakeholder can read the design conflict profile from a web-browser and perceive the convergence of design ideas. Based on the conflict management strategies, the system might achieve different conflict profiles. In short, the system helps design stakeholders manage conflict and integrate their knowledge by converging their individual perspectives.

Since designers' perspectives will largely depend on the interactions among the design tasks, arranging the design process in a desired manner becomes an effective approach to coordinating the perspectives of the stakeholders. The system provides an algorithm to help stakeholders rearrange the design process to reconcile the design perspectives and resolve conflict. When conflicts happen, the system can identify the affected design tasks in process view from the dependency matrixes. Then a possible solution is to modify some tasks or search for a new task to remove the conflict source. Sometimes, effective handling of design conflict will eliminate the task iterations and shrink the design process. For instance, the system detects the conflict between the design consultant and the architect in task T4 of the process diagram (Figure 27). It will suggest several ways to control that conflict by examining the perspective models involved in the related tasks. A possible solution is to add a task (e.g., let the architect specify the building location requirement) before T4 so that the necessary design information is derived earlier. Then the process is modified to prevent the conflict.

4.4.2.4 Prototype Applications

The major aim of STARS is to provide a platform to facilitate the collaborative design activities over the Internet. STARS has been tested using an architectural design scenario provided by CERL. It shows that using STARS has two critical effects to the design. First, the users think the web-based process representation tool helpful for them to realize the status of the overall process and to identify individual positions. Second, they realize that viewing others' perspective models provides a way to understand the meanings during communication. In fact, STARS has potential usefulness for generic collaboration over the Inter-

net. Since STARS can improve the coordination among stakeholders, it helps the different collaboration partners connected faster and easier. That is one of the key issues in supporting B2B activities on the web.

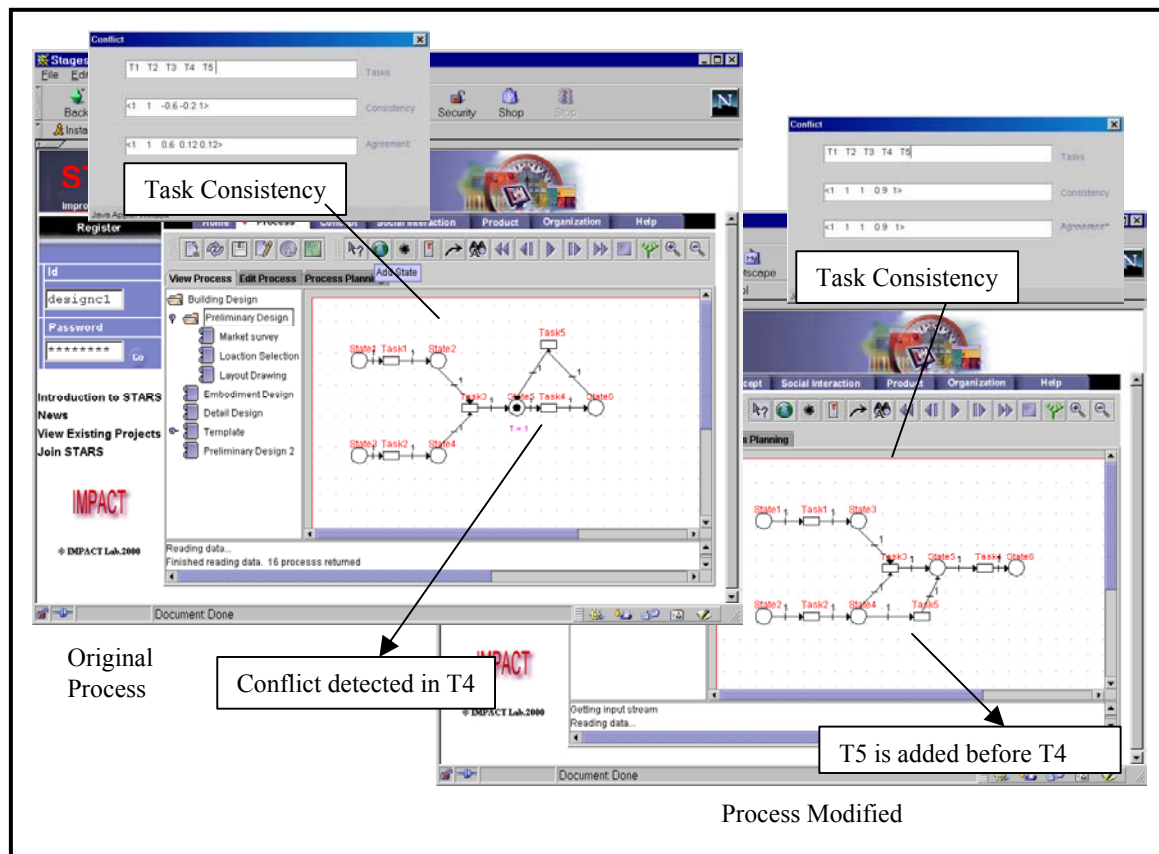


Figure 27. Conflict detection and resolution.

5 Information Sharing for Collaborative Design

This chapter presents research results for a perspective-based approach to information sharing in collaborative engineering design, which is derived from a Socio-Technical Framework. Specifically, a theoretical framework is presented that contains elements that represent and relate both the knowledge of the stakeholders and the data that encodes that knowledge. Section 5.1 presents the current state of data- and database-based approaches enabling collaboration through information sharing technology; significant obstacles and limitations are identified. Section 5.2 below presents the groundwork for the theoretical framework by presenting key concepts of the framework. Section 5.3 below presents the theoretical framework for information sharing and application interoperability through the semantic abstraction of data; Section 5.4 below presents a case study on integration through abstraction. Finally, Section 5.5 below presents some observations and issues with the use of the Extensible Markup Language (XML) for enabling information sharing and application interoperability in collaborative design technology.

5.1 Information Sharing in Collaborative Engineering Design

5.1.1 *Information Sharing in Design*

Engineering design is a collaborative endeavor undertaken by a number of stakeholders. Communication is critical within the collaborative process; stakeholders must be able to share information and exchange knowledge of both the problem space and the evolving design. Complicating the information-sharing process is the fact that stakeholders bring with them unique and individual perspectives on the engineering design campaign (Figure 28). Most information sharing (i.e., communication) within an engineering design campaign is conducted with spoken and written language. In addition, graphic languages and conventions have been developed to formalize and regularize information that cannot be easily communicated via lexical languages (e.g., English), most notably engineering drawings. With the advent of computer technology, the volume, variety, and depth of information that can be captured and processed has increased dramatically, enabling unprecedented productivity and capabilities. In fact, most of the technology infrastructure of design environments has been devoted to communication.

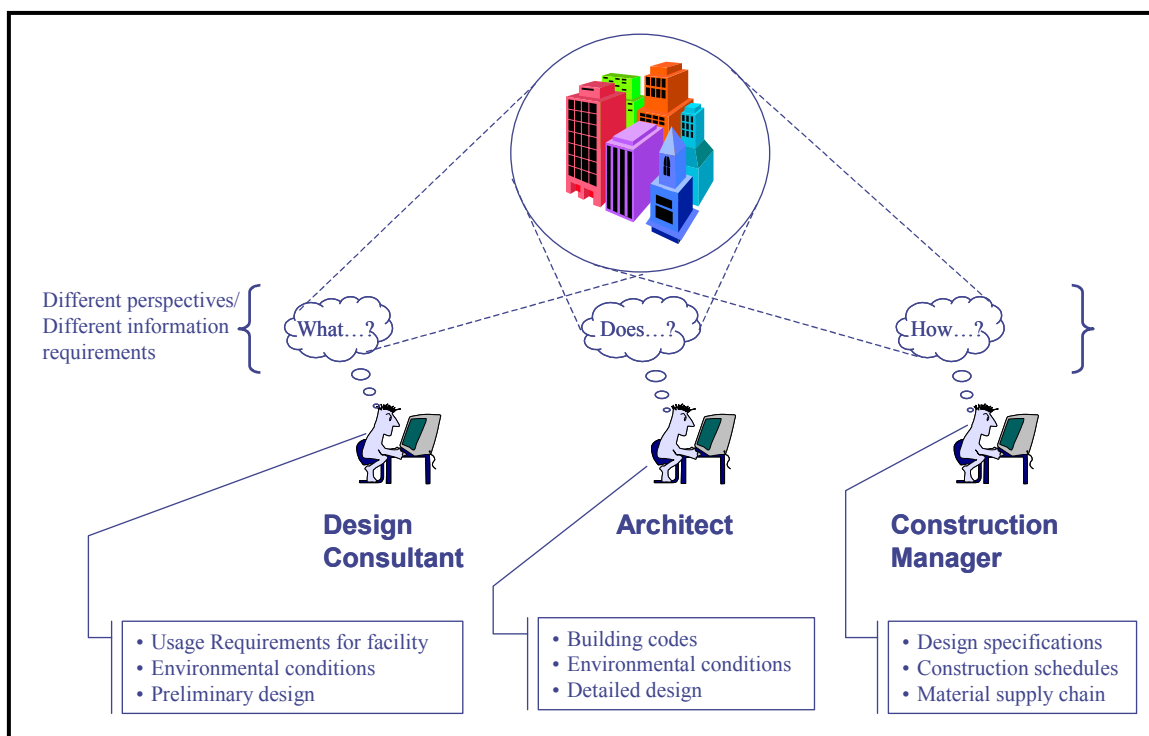


Figure 28. Multiple perspectives on same application domain.

The opportunity to improve engineering design with improved communication tools spawned a great deal of research in Concurrent Engineering, Enterprise Integration, and other areas seeking to improve engineering design by improving communication. Sharing common data, such as CAD models or, more recently, Integrated Product Models requires a common data source, as Figure 29 illustrates. Enterprise Application Integration and enterprise-wide information management, however, require a view of integrated information systems more like the illustration in Figure 30.

These views seemed to offer huge benefits in terms of the availability and quality of information that engineers need to “do their job.” These benefits, unfortunately, were never fully realized for many different reasons. It is the position of this study that the reason for the failure of integrated information technology to provide the benefits envisioned is that the focus of the research is misplaced.

Past efforts to improve the information sharing capabilities of a community of engineers via technology have focused on the development and improvement of technology. The Internet and World Wide Web, Intelligent Design Agents, and product data standards are all communication technologies, but they fail to recognize that communication is about *meaning* and meaning is about *people*.

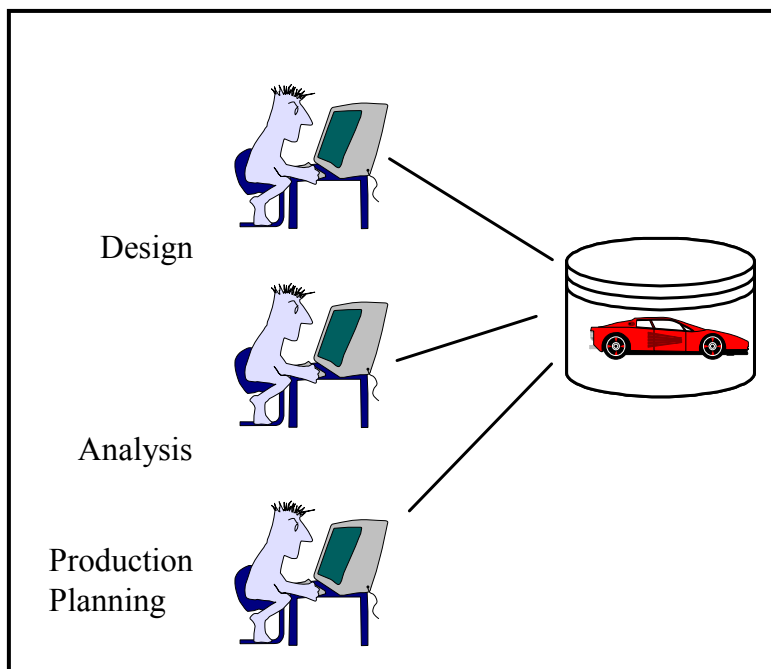


Figure 29. Integrated product data model.

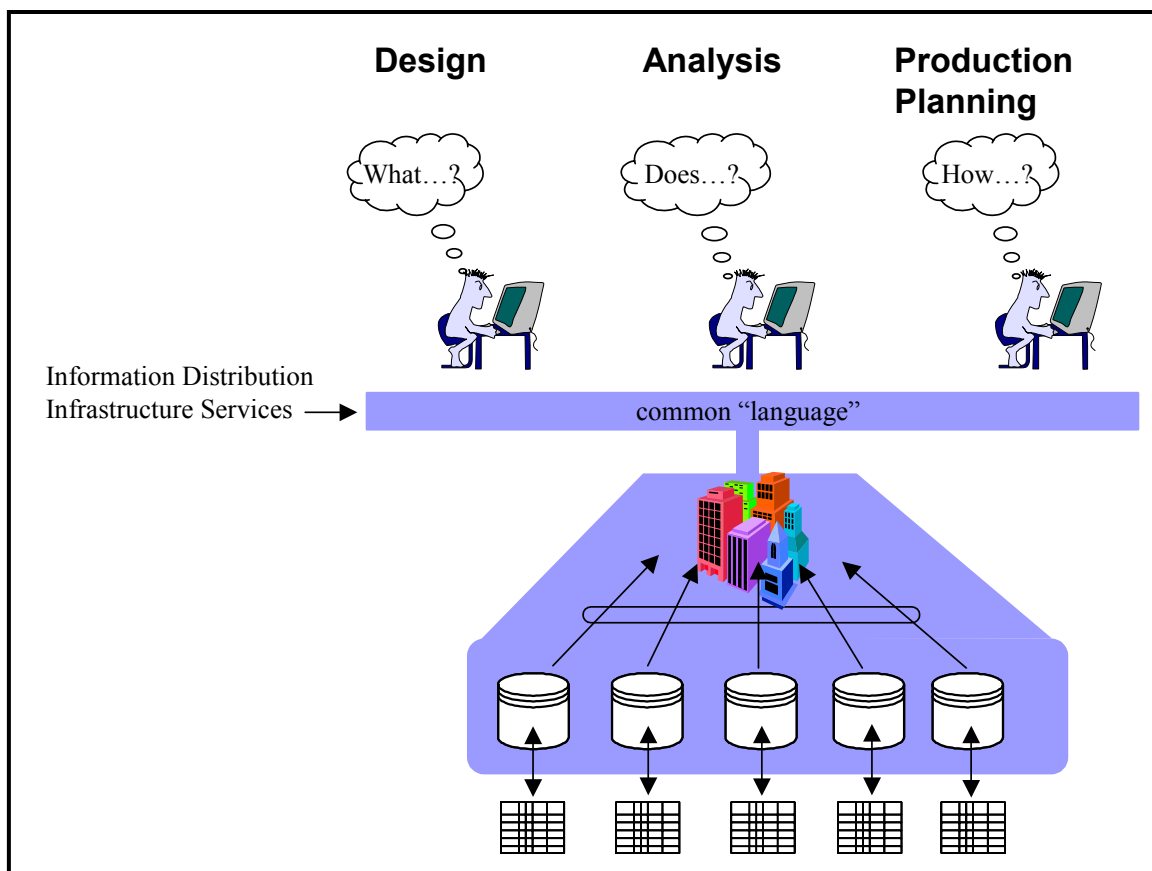


Figure 30. Integrated information resources.

As such, to improve the communication between the stakeholders within a design campaign, improvement solutions must start with the *participants* in the process—the “who”—the people who must produce, share, and consume the information. If the benefits of computer/communication technology are to be fully realized, the solutions must be based on theories of how people produce, consume, and share information and, thereby, transmit meaning; this is explored in more detail in Section 5.1 above.

5.1.2 Data Models and Application Interoperability

To access integrated data resources through an enterprise-wide information distribution service (as shown in Figure 30), the data management system architecture must be carefully planned and deployed. The key element of this architecture is the data model (or models) used to specify and describe the data contents of enterprise databases.

The term “data model” is often used in two distinct senses. In the field of database research, “data model” typically refers to a collection of concepts and operations that may be used to describe/define data and manipulations of data (e.g., the Relational Data Model). In popular usage, “data model” often refers to a particular *usage* of concepts of a data model, e.g., a database schema.

Data are the operational values that are manipulated by software applications, displayed with a name and within a context to user, and from which the user derives valid, accurate, and timely information. Figure 31 shows the distinction between “data” and “data model.”

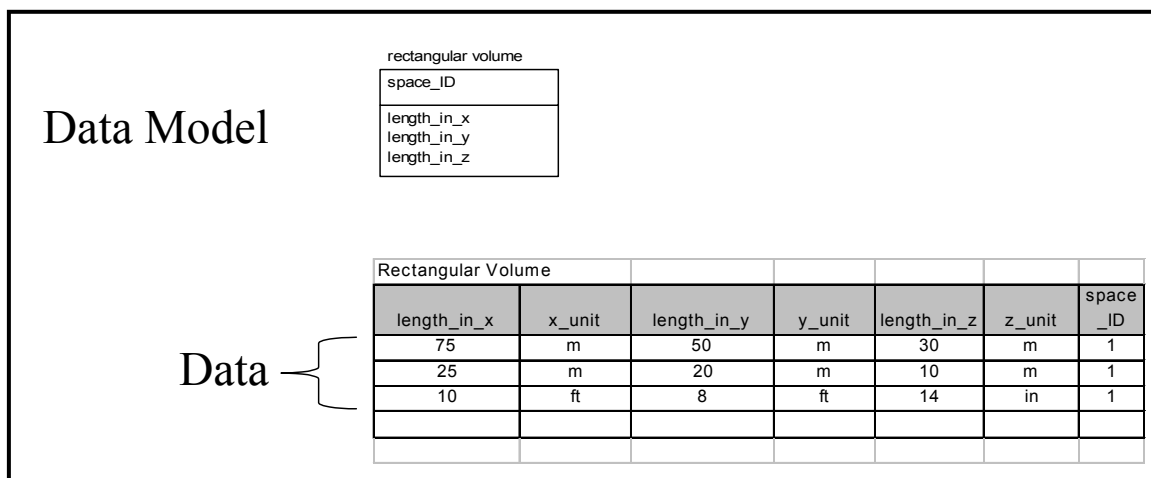


Figure 31. Data model and data.

A data model (or schema) *governs* a particular data population. Figure 32 presents a graphical illustration of a schema on the left; it is an *intentional* specification or template that is used to create instances (i.e., members of the data population). The database on the right illustrates data in a database; the arrow between the schema and the data points from a schema declaration to one (of two) instances of the declaration in the schema.

The typical relationship between a stakeholder and data is mediated through (or by) the application used to access the data (Figure 33). The data is typically tightly bound to the application, i.e., that application and *only* that application can correctly interpret and use the data. Application interoperability in this case requires translation software to convert the data from the format used by one application to the format used by another (Figure 34).

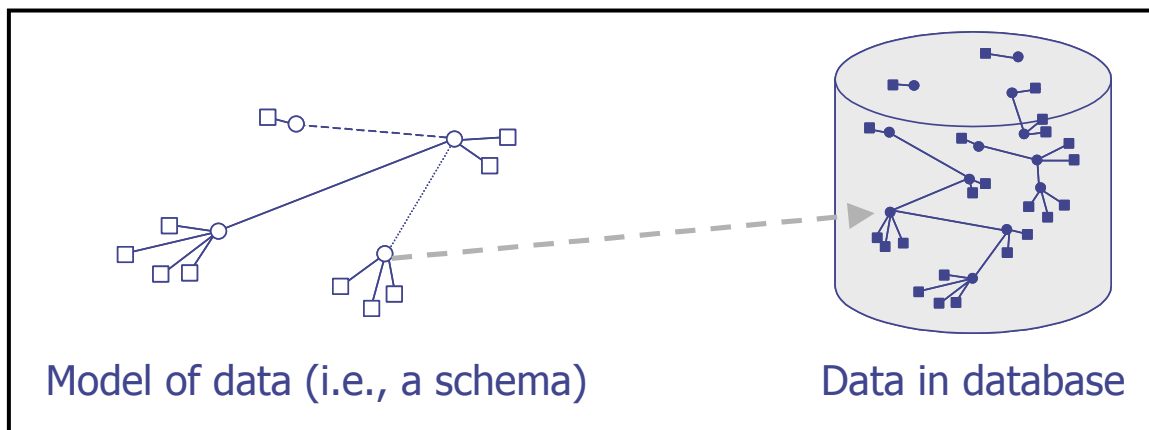


Figure 32. Relationship of data model to data in database.

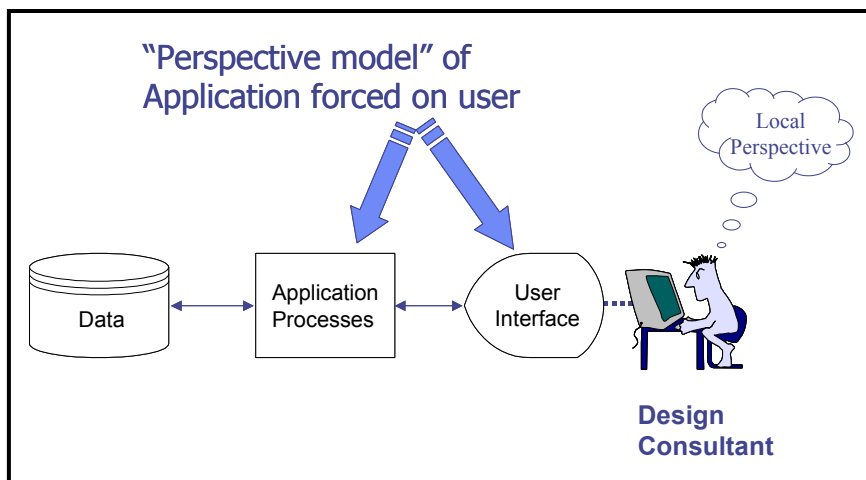


Figure 33. Relationship between perspectives and data.

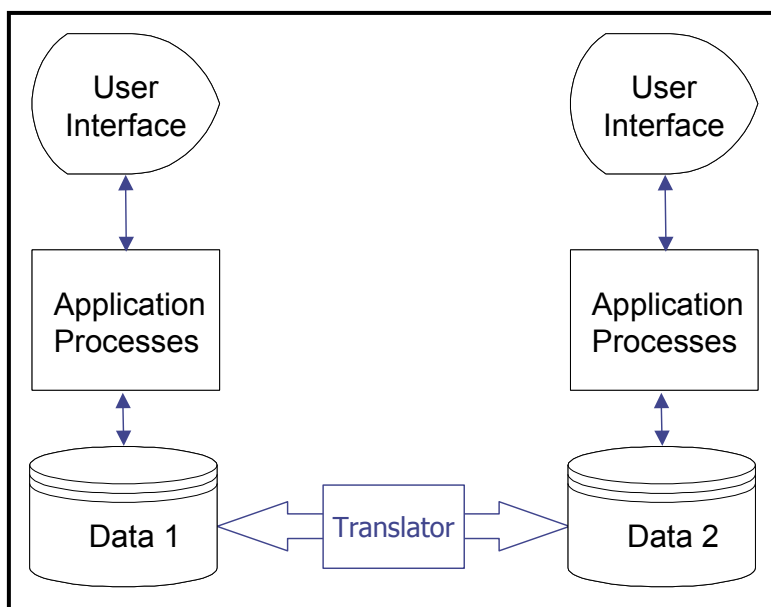


Figure 34. Translation of data.

This approach to application interoperability is referred to as “islands of automation” because each application/database combination is effectively an island cut-off from other applications. A special transportation step is required to share the work of one stakeholder with another stakeholder. As Figure 35 shows, the number of translators required to enable N application to share data (and inter-operate) is $N*(N-1)$.

System integrators logically observed that the number of translators could be reduced if all stakeholders and their applications used the same enterprise-wide data model and database (Figure 36).

Efforts to provide an application interoperability with a single, large monolithic data have not been able to effectively achieve this goal for the same reason it is impossible to create a “snapshot” model of a large enterprise: the natural progress of organizational and technological change during the time it takes to create such a data model renders the model obsolete, brittle, or of limited value upon completion. The monolithic enterprise data model is not manageable and is difficult to understand due to its size; it does not “adapt” to changing requirements, but rather must be augmented to meet new requirements.

Database Federations (Figure 37) offer a hybrid approach for providing “globally understood” data models while permitting individual applications autonomy over “their” data. Each application (or member of the federation) would publish an “export schema” that presented the data that they would make available to other applications of the federation.

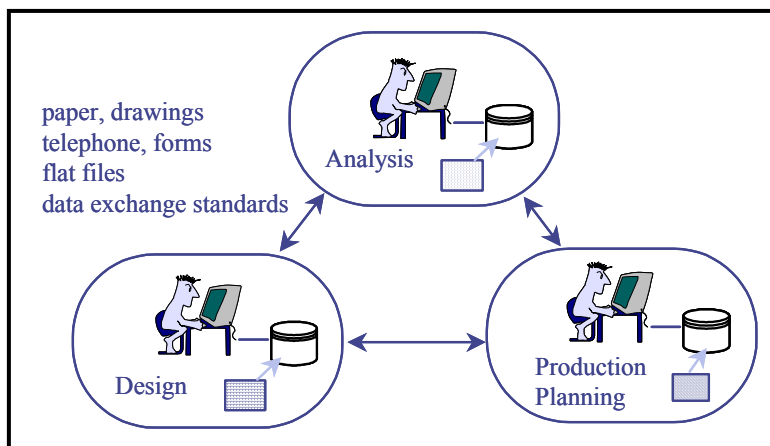


Figure 35. Islands of automation.

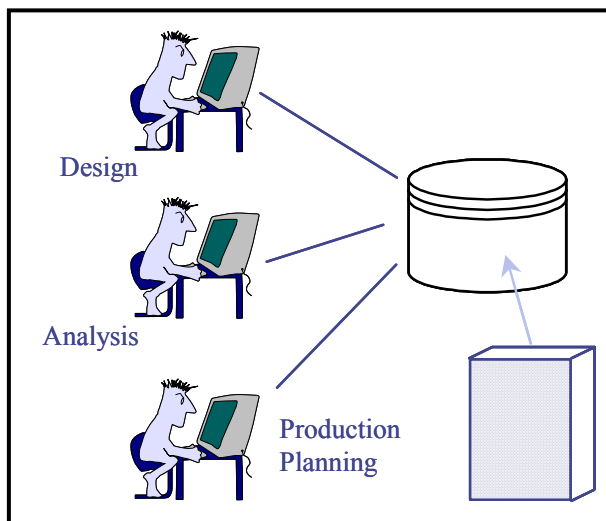


Figure 36. Enterprise data model.

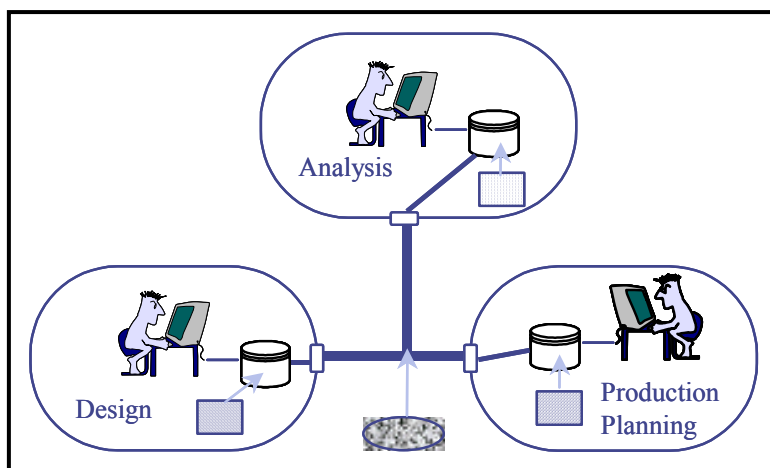


Figure 37. Database federation.

All of these approaches have strengths and weaknesses. The weakness overshadows the strength in all cases and none of these integration approaches has emerged as dominate in Enterprise Application Integration. Figure 38 shows several integration approaches and how each approach has inherent problems.

5.1.3 A New View of Information-Sharing in Collaborative Engineering Design

Systems designers trying to integrate engineering design applications to foster and promote collaborative design face two significant obstacles. These obstacles were neither evident nor apparent in the initial solutions to the problem. Furthermore, the progress of technology not only makes more sophisticated solutions possible, but also makes important and significant innovations easier to see and come by.

The introduction of the “who” into engineering design process models by the identification and representation of perspectives reveals that the two obstacles are intimately related to the nature of knowledge and communication.

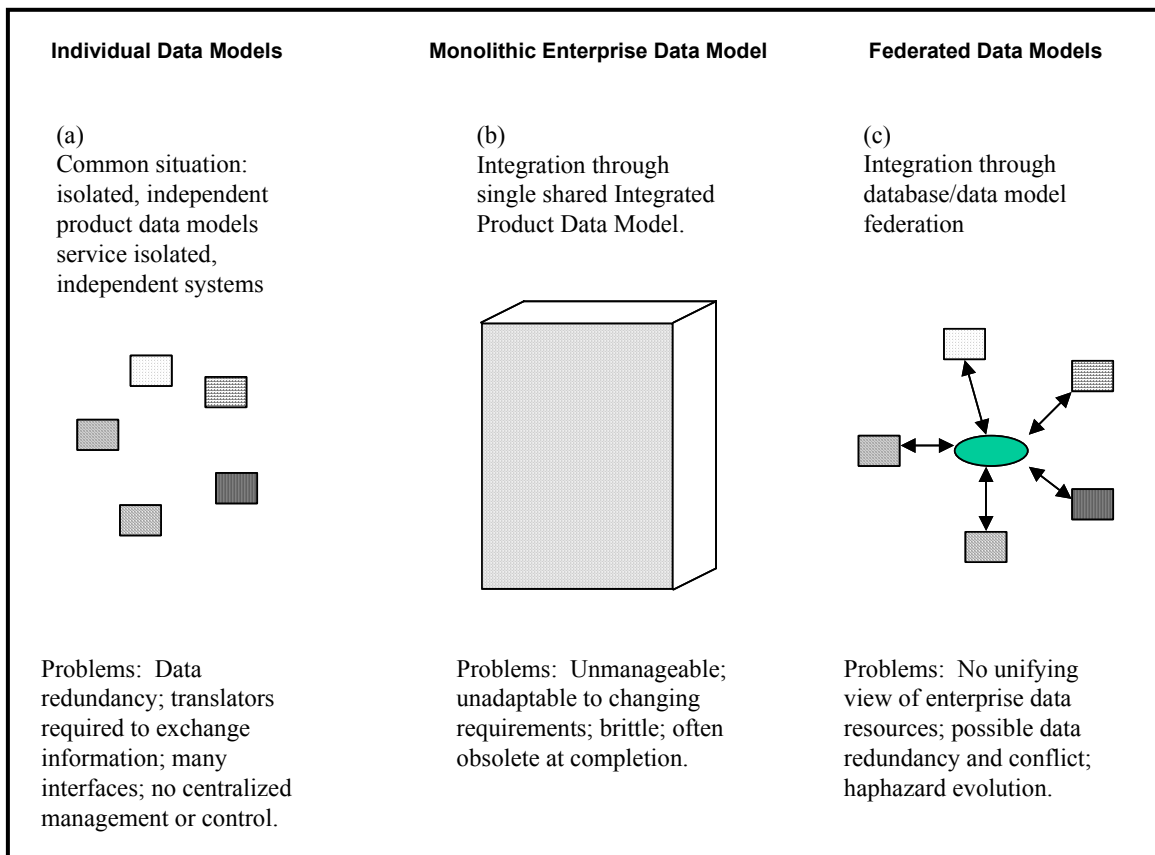


Figure 38. Integrated data model approaches.

The first obstacle is that the information requirements of a stakeholder in an engineering design campaign change on a daily basis. This is due to new task or shifting focus of work (i.e., the genesis of new *purposes*). Therefore, the stakeholder's requirements for collaborative information technology also change, and the technology should be dynamically adaptable to the new requirements. The stakeholder requires a dynamic and adaptable user interface to the data resources, E , of the enterprise. He needs an interface that mirrors his *perspectives* on the engineering design campaign.

However, not all his interfaces to the enterprise data resources need to be dynamic and adaptable. Many instances of the same task/purpose are repeatedly pursued, in which a single, stable user interface to a specialized application is both desirable and acceptable. This can be viewed as the application forcing its own "perspective" onto the stakeholder, which (perhaps surprisingly) in many design applications is the right thing to do.

The second obstacle is a little esoteric; it deals with the limited conceptions of data and data models that prevail throughout the information technology industry. Despite the fact that data almost always has some relationship or correspondence to the knowledge (i.e., the internal data store) of a stakeholder, there is a widespread belief that individual units of data *can* and *must* only have a single definition or meaning if automated applications are to be able to "understand" the data and make decisions based upon it. The problem with this view is obvious: it requires the definition of a separate and distinct data item for each shade and nuance of meaning that might be exchanged through a computer system. This results in an infinite number of data item definitions. Unlike human languages, there is no facility or approach for re-using data items in slightly different context with slightly different meanings (or entirely different context with entirely different meanings.) There is no facility for semantic abstractions or generalization for grouping and talking about "like" things, and then talking about particular instances of these generalizations.

This new view of information sharing in collaborative engineering design starts with the knowledge of the individual stakeholder and uses data model architectures that leverage linguistic mechanisms to view data more like speech. The challenge first is to get a "hold" on perspectives and then find a way to establish relationships among perspectives.

5.2 A Framework for Perspective Model Integration

This Socio-Technical Framework for Collaborative Engineering Design highlights the importance of the stakeholder in the success of an engineering design campaign. More importantly, it identifies the *perspective* of the stakeholder as being an essential anchor in communication technology that enables collaboration. Thus, the question is how to apply the dynamical model of perspectives to the design and behavior of the information systems.

There are three parts to the answer of this question:

1. The first is to determine how stakeholders interact in an engineering design campaign, how their perspectives affect this interaction, and how their perspectives evolve over the course of the campaign.
2. The second is to determine how a perspective is externalized in a manner that can be used/manipulated by information technology.
3. The third is to determine how these externalizations relate to one another.

Once these parts are synthesized into a consistent and integrated whole, the result then constitutes the requirements for collaborative information system design.

5.2.1 *Perspective-Based Collaborative Engineering Design*

This work adopted the theory of *Social Construction of Reality* (Berger and Luckman 1966) as the view of how stakeholders in an engineering design campaign interact with a purpose. In this view, stakeholders externalize their knowledge to exchange information through mediating, negotiated, conventional language, not only to exchange information, but also to reinforce the understanding and meaning of the language they use to communicate. This work wishes to capture and mimic this dynamic and adaptive behavior of language in an integrated information architecture.

This work integrates this view with existing design process models (DPMs) by introducing the “who” into the technical processes represented by DPMs. This combination produces a view of information sharing that is not *just* about communication of technical information, but *also* about interaction and negotiation to establish a *shared reality*, a common understanding of aspects of the engineering design campaign. The view here is that communication and information technology should not be tightly bound to the *meaning* that it is intended to convey, but rather should serve as a *vehicle* for facilitating the construction of

meaning and knowledge within a social community. The structure is as “meaning-neutral” a vehicle as possible.

The “local reality” that is the basis of a stakeholder’s participation in an engineering design campaign is characterized and defined here as a *perspective*. To operationalize this social construction view of collaborative engineering design, this perspective must be represented with a computational model: a “Perspective Model.”

What is needed is a framework for establishing relationships between perspective models corresponding to local realities and perspective models representing shared, community realities.

The following assertions must be maintained or reflected in the framework:

1. Knowledge ultimately resides in the mind of the individual stakeholder. Therefore, the perspective model of the individual (as his “local reality”) is the most accurate overt representation of his knowledge (assuming, of course, the competency of the stakeholder in the use of the language used to produce the perspective model).
2. “Social Knowledge” is knowledge shared by a community of two or more people and is the result of communication and negotiation. Social knowledge can be objectified as a real-world artifact (e.g., a perspective model) that can be perceived by multiple individuals and is consistently/uniformly interpreted the same way (i.e., the artifact conveys the same meaning to the perceivers.)

These two kinds of perspective models form the basis of the integrated information architecture. The methodological “glue” that holds them together is presented in section 5.3. However, before the glue is explained, some characteristics and properties of perspectives models must be presented.

5.2.2 Reification of Perspectives

Getting a “hold of” and operationalizing perspectives means externalizing the perspectives in a stakeholder’s mind into a physical manifestation that can be viewed and manipulated by agents other than the stakeholder. This process of externalizing and representing a perspective is called *reification*:

re·i·fy (rē'ə-fī', rā'-) *verb, transitive*

To regard or treat (an abstraction) as if it had concrete or material existence.*

These externalized representations are called Perspective Models.

5.2.2.1 Perspective Models

Perspectives are formed when a stakeholder becomes part of a community undertaking a design campaign and begins to interact with other members of the community. During the design campaign, a stakeholder forms a mental model of the campaign, a “local reality” (Figure 39). This model represents the stakeholder’s perspective. It is constructed and refined through learning based on the information received and is composed of (for example):

1. The current state of the design model (i.e., Integrated Product Model)
2. The current state of the campaign/project (schedules, roles, goals, resources)
3. The current state of the design environment
4. Experience, education
5. (His view of) product requirements.

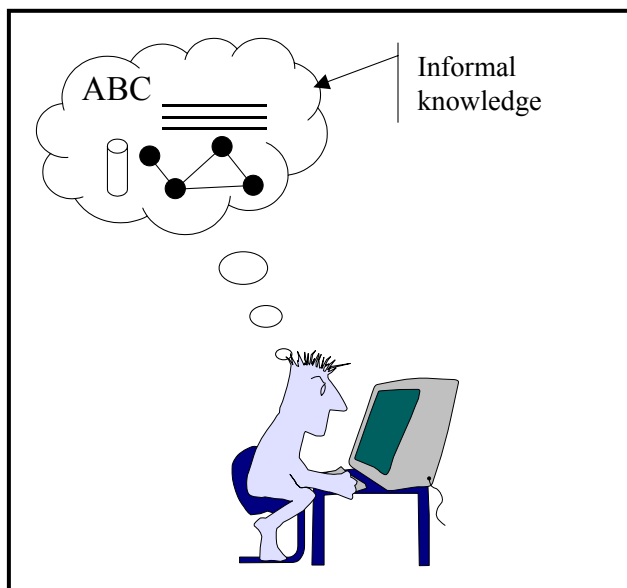


Figure 39. Forming mental model of perspective.

* Excerpted from *The American Heritage Dictionary of the English Language, Third Edition* Copyright © 1992 by Houghton Mifflin Company.

Because perspectives exist solely within the mind of the stakeholder, taking advantage of perspectives to improve the design of information technology for collaborative engineering design requires that the perspective be externalized in a form that bears some relationship to the data resources of the enterprise, E (Figure 40) and thus the perspectives of other stakeholders.

A model of the stakeholder's perspective must be interactively and dynamically captured by the communication system. This entails that a structured and semi-formal model be constructed by the stakeholder and "understood" by the system; Figure 41 illustrates the formalized representation of a perspective vis-à-vis the informal knowledge illustrated in Figure 39.

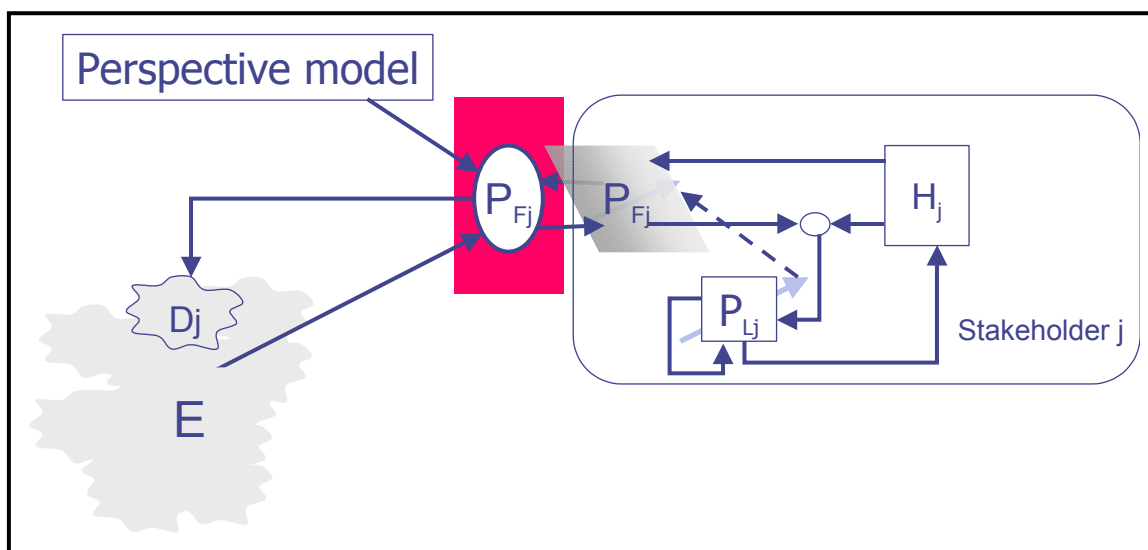


Figure 40. Externalization of a perspective: perspectives model.

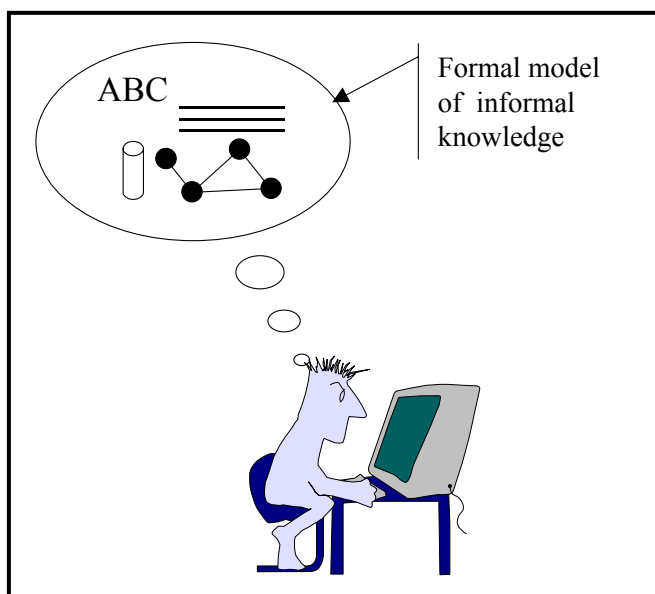


Figure 41. Formalizing perspective model.

While it is impossible to represent everything that a stakeholder “knows” with a computational model (a myth perpetuated by the technology focus of research on things such as data standards and AI), it *is* possible to build a model or vocabulary representing those things most important to the stakeholder that can be represented within the communication system. Unless the communication system has the opportunity and ability to “know” what the stakeholder knows, *collaborative* technology is not possible.

This model, of course, will need to be based on a formalized modeling language that is taught to the stakeholder to “communicate” his knowledge to the system. This work is based on the belief that an abstract ontology can be constructed to serve as “building blocks” for the model; this ontology would consist of elements such as Process, Requirement, Decision, Organization, Actor, and Resource.

Note that these perspective models are not “built” anew each time an engineering design campaign is undertaken. The perspectives of a stakeholder in a certain role in an engineering design campaign evolves, grows, and changes with his participant in different campaigns. Design Process Models and company policies, procedures, etc., also evolve over time; they, too, are representations of socially-constructed meanings and institutions (i.e., they are social constructions), and they may be used as a baseline or starting point for the “shared reality” of the community of stakeholders. This is the “everybody following the same procedure” that brings unity and coordination to a large, diverse design team.

Figure 40 does not illustrate the relationship between perspectives models and real data in a database very well. If one understands that a perspective model may be seen as a data model and also understands the relationship between a data model and data (Figures 31 and 32), then the relationship between the perspective model and E from the dynamical model of perspectives and data models/data becomes clear (Figure 42).

5.2.2.2 Multiple Perspectives

An issue that complicates the problem of capturing perspectives is the fact that stakeholders do not have a single perspective on the engineering design campaign. They actually have multiple, overlapping, and not-necessarily-consistent perspectives representing little, purpose-centered subsets of their knowledge of the engineering design campaign. Berger and Luckman (1966) note that an individual’s knowledge of “reality” actually consists of many “provinces of meaning.” As Figure 43 shows, an engineer has academic-based knowledge of science, design experience, knowledge of the organization, product requirement knowledge, and project management knowledge. In other words, his entire knowledge base is composed of different areas or domains of things that he “knows.”

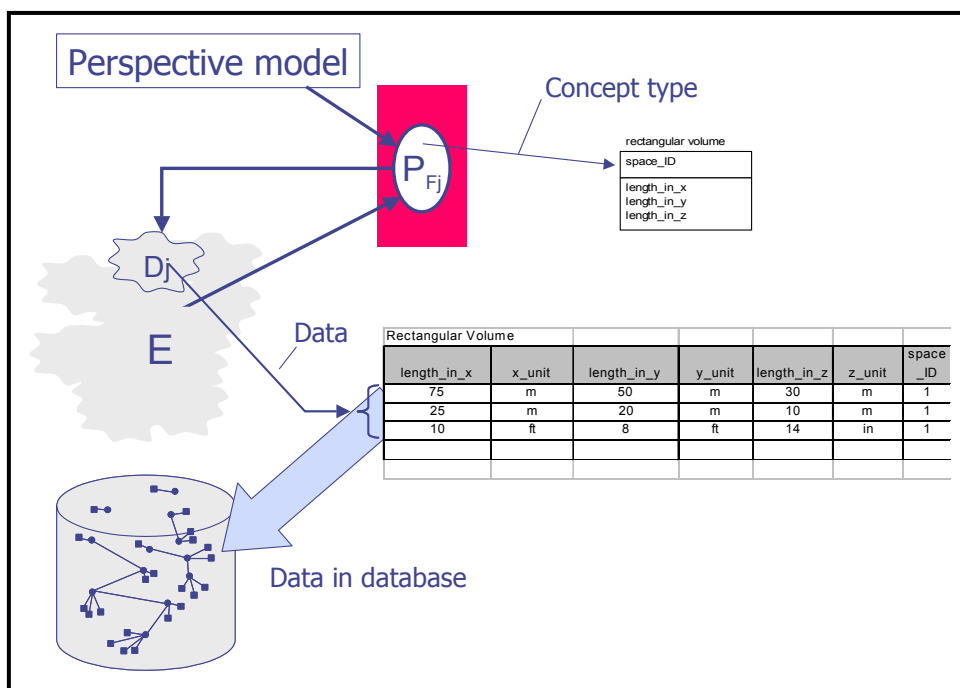


Figure 42. Perspective models, data model, and data.

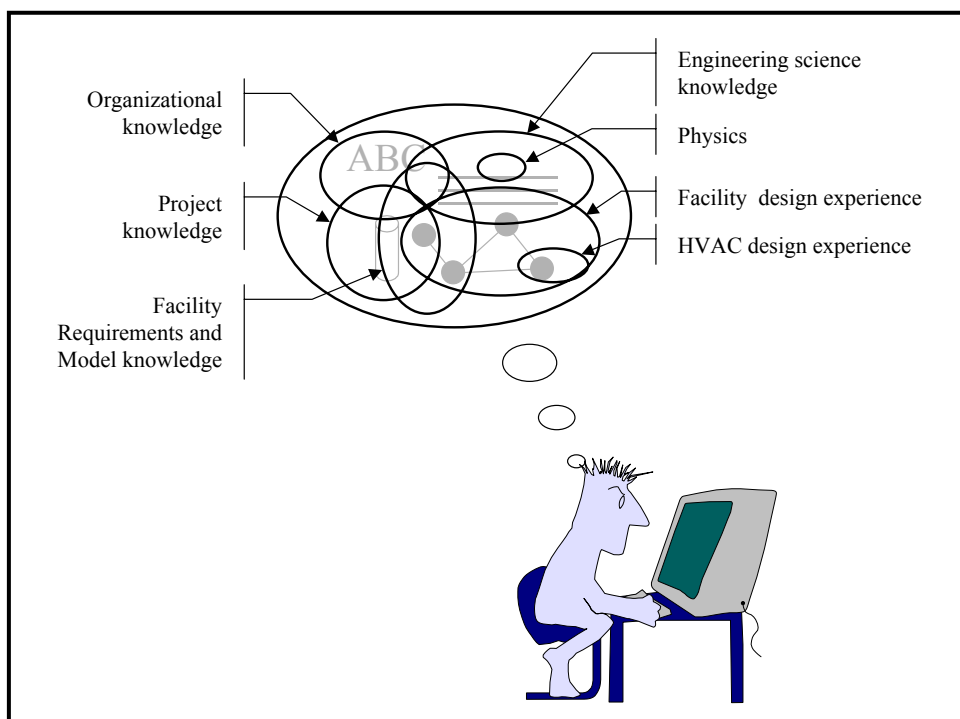


Figure 43. Multiple perspectives.

He also has a perspective that includes his understanding of the DPM employed in the campaign as well as the company procedures under which the campaign is being conducted. Methods are needed to integrate multiple stakeholder perspectives.

The problem of multiple perspectives is compounded by the fact that there are many stakeholders involved in an engineering design campaign, each with their own set of perspectives. As Figure 44 shows, the challenge of the integrated information architecture is to establish relationships between perspectives and enable conflict detection mechanisms to detect differences in stakeholder knowledge.

5.2.2.3 “Accuracy” of Individual Perspective Models

One of the goals of this work is the semantically unambiguous interpretation of data by a user of collaborative information technology. It is, after all, through the exchange of information (as conveyed by data) that individuals learn, their knowledge evolves, and they make contributions to the engineering design campaign.

Accomplishing such an exchange requires the establishment of a foundation for the accuracy of the meaning of perspective models. This foundation is the individual perspective model that represents (part of) the knowledge of a single person. The individual perspective model must be:

1. Unambiguous to the individual that constructs it
2. Of a size and character not exceeding the ability of the individual to evaluate the *validity* and *completeness* of the model.

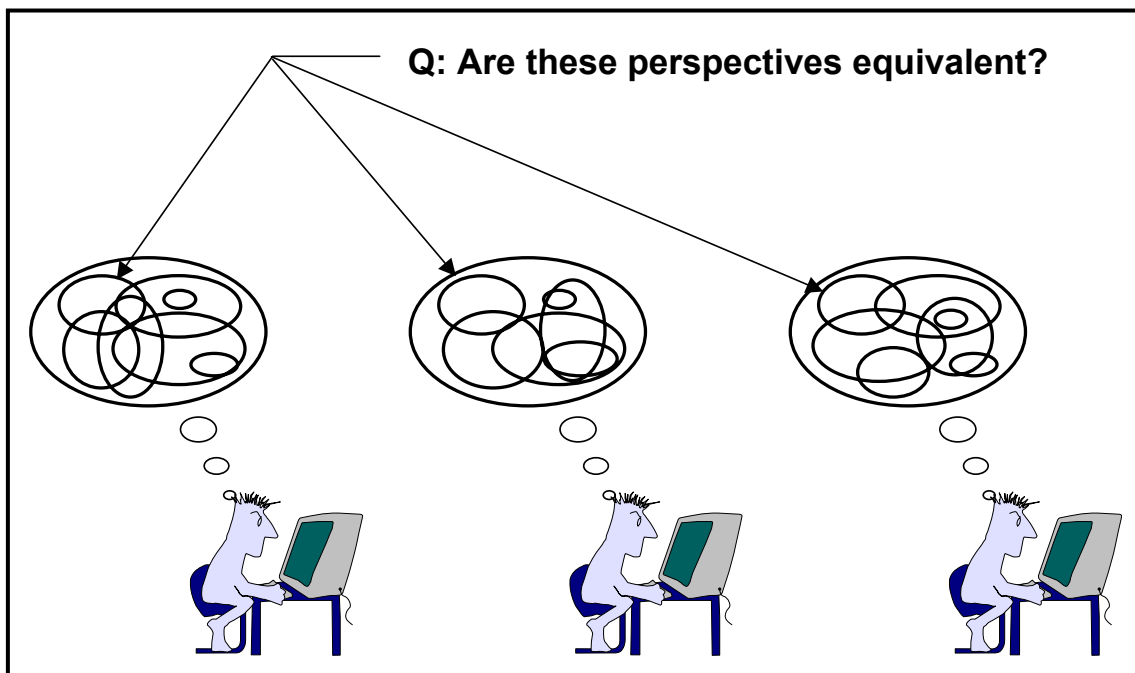


Figure 44. Multiple perspectives across stakeholders.

It is the ability to do that latter than contributes to the former. Only when an individual can ascertain that the perspective model is valid and complete can he say that it is unambiguous. Lindland, Sindre, and Solvberg (1994) define *validity* and *completeness* as:

Validity: all statements made by the model are correct and relevant to the problem.

Completeness: the model contains all the statements about the domain that are correct and relevant.

“Correct and relevant,” of course, is an extremely subjective evaluation that relies on interpretation by humans. The ability of stakeholders in the model to make these assessments is *the* crucial characteristic of perspective models. This is particularly significant as the model grows in size. Lindland, Sindre, and Solvberg (1994) observed that:

For anything but extremely simple problems, you cannot achieve total validity and completeness. Attempts to do so would require spending unlimited amounts of time and money ...

In these definitions, *problem* and *domain* are analogous to the *purpose* and *context* of perspective models.

5.2.3 The Collaborative Information Infrastructure

Putting the above theory of perspectives together into a collaborative information infrastructure requires the recognition and understanding of several important properties of perspective models and the recognition that a community of stakeholders may have its own, joint, shared perspective model that represents a shared reality of that community.

The characteristics of perspective models are discussed in Section 0 and community perspective models in Section 5.3.2. Section 5.3.3 presents the structure of the collaborative information infrastructure, and Section 5.3.4 describes the linkages between the nodes in the structure.

The structure is essentially a graph of nodes and links in which the nodes represent perspective models and link relationships or mappings between perspective models. It is easiest to visualize this as a hierarchy, but there is no requirement that it have this form.

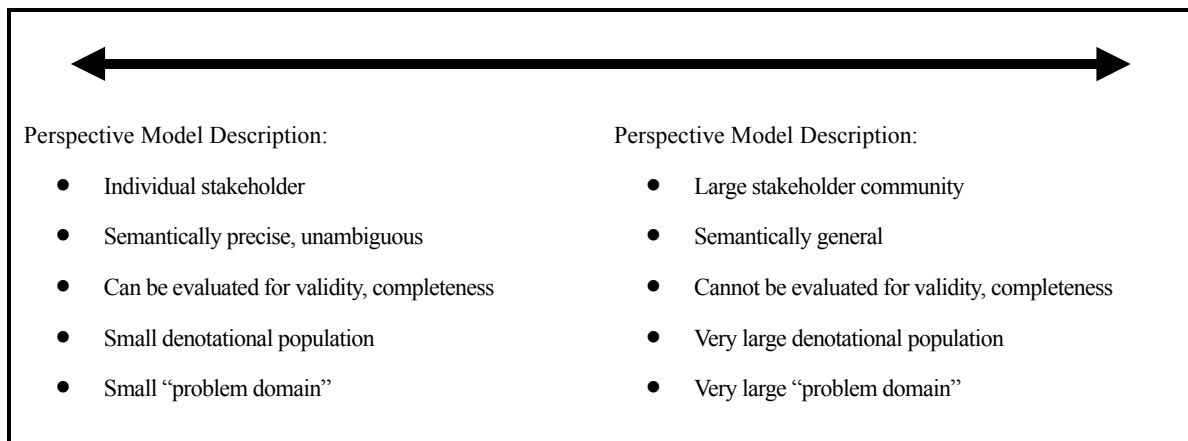


Figure 45. Variations of Perspective Model Characteristics.

5.2.3.1 Conceptual Characteristics of Perspective Models

The most fundamental characteristic of this framework is a spectrum of perspective models that vary along a cline from a:

semantically-precise (and unambiguous) perspective model with a stakeholder community of size = 1 (i.e., an individual perspective model) and relatively small denotational population,*

to a

semantically-broad perspective model with a very larger stakeholder community (upper limit = the earth’s human population?) and an extremely broad denotational population.

Figure 45 presents a simple view of the variations of perspective models. Another way to describe the variations is that it represents a spectrum from the “local reality” of a single individual to the “shared reality” of everyone in the world.

To explain the model in more detail, an explanation of some concepts is necessary to clarify the variability of perspective models over this cline.

* “Denotational population” refers to the things in the real world that concepts in the perspective model may denote.

5.2.3.1.1. Meaning Communities

Given the basis that a valid and complete individual perspective model is unambiguous to the stakeholder, how can this basis be extended to provide an unambiguous means of communication between two or more stakeholders? A shared perspective model can serve as a representation of the shared social knowledge of the community and can serve as a communication mechanism among the members of the community. The production of the shared perspective model requires the analysis, reconciliation, and integration of perspectives across perspective models, a process directly analogous to database schema integration (Batini, Lenzerini, and Navathe 1986).

The informal term “meaning community” is used here to refer to the collection of stakeholders in a shared perspective model. One form of meaning community exists in the users of the English language. Another exists in the form of standards bodies striving to define ontologies and data exchange standards; a collection of stakeholders band together to define a common data model with which to exchange industrial data.

There is not much need for a framework beyond what has already been presented if the meaning community using the shared perspective model is small. The integration of the models produces a single perspective model that all the stakeholders can determine is valid and complete. The problems arise when the meaning community grows larger and integrating perspective models becomes difficult.

5.2.3.1.2. Scope of Perspective Models

The “scope” of a perspective is difficult to describe, let alone define. Broadly speaking, “scope” refers to the domain of applicability of the model. A database schema, for example, has a scope established by the usage of the data held in the database. A banking database, for example, has a “scope” that encompasses savings, checking and other financial transactions.

The scope of a perspective (and, thereby, a perspective model) shall be loosely defined by the size of “denotational population” entailed by the model. This could also be called the “range” of the model in the same sense that the “range” of the term “whole numbers” is the set of integer values. In other words, the “range” of the model is comprised of the real-world things that can be referred to by a perspective (or a concept in a perspective).

This definition of scope is closely coupled with the meaning community of the perspective model, since it is only members of this community that can ascertain whether or not a particular real-world thing is a member of the “range” of the perspective or perspective concept. Therefore, one may also say that the scope of the perspective model also tends to grow with the size of the associated meaning community.

5.2.3.1.3. Level of Semantic Abstraction (LOSA) in Perspective Models

Semantic abstraction is a natural human cognitive ability that developed to cope with the enormous volume and variety of stimuli a person receives every day. It is natural to categorize things in the real world into classes based on characteristics of the thing; there is no need to understand how each individual *Toyota Celica* works because it is clear how *automobiles* in general work.

Parsons and Wand (1997) recognized that semantic abstraction is something that happens naturally as humans form concepts to understand the real-world. They quote linguist George Lakoff:

Without the ability to categorize, we could not function at all, either in the physical world or in our social and intellectual lives. An understanding of how we categorize is central to how we think and how we function ...

This class-member distinction will be part of perspective models, particularly in perspective models with large meaning communities. Dealing with this distinction is also problematic because it affects the ability of individual stakeholders to evaluate the validity and completeness of the model; their individual meanings of a specific concept may be “washed away” or lost in the generalization necessary to integrate perspective models within a large meaning community.

Note that the LOSA of a perspective model is functionally related to, but independent of the scope of the perspective model. If one considers the size of a perspective (roughly: the number of concepts in a perspective), the same scope can be denoted with a small model of high LOSA (i.e., a general model with few conceptual elements) or a large model with a low LOSA (i.e., a semantically precise model with many conceptual elements.) Figure 46 shows the relationships and trade-offs between size, scope, and LOSA; Figure 47 shows the same relationship mapped into a three-dimensional description space for perspective models.

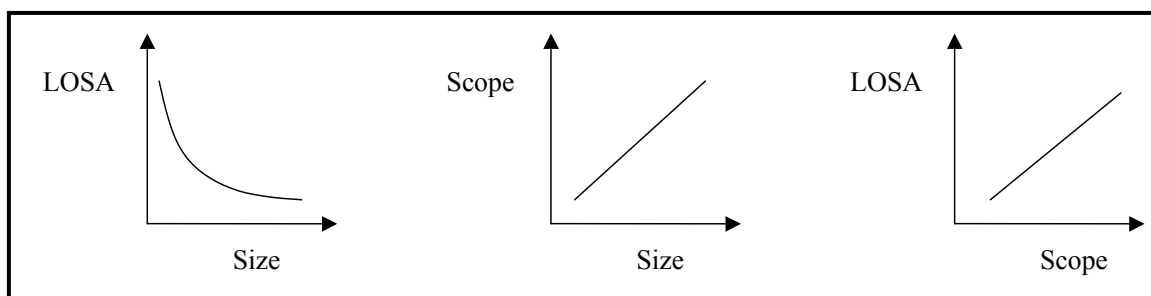


Figure 46. Relationship of LOSA, scope, and size.

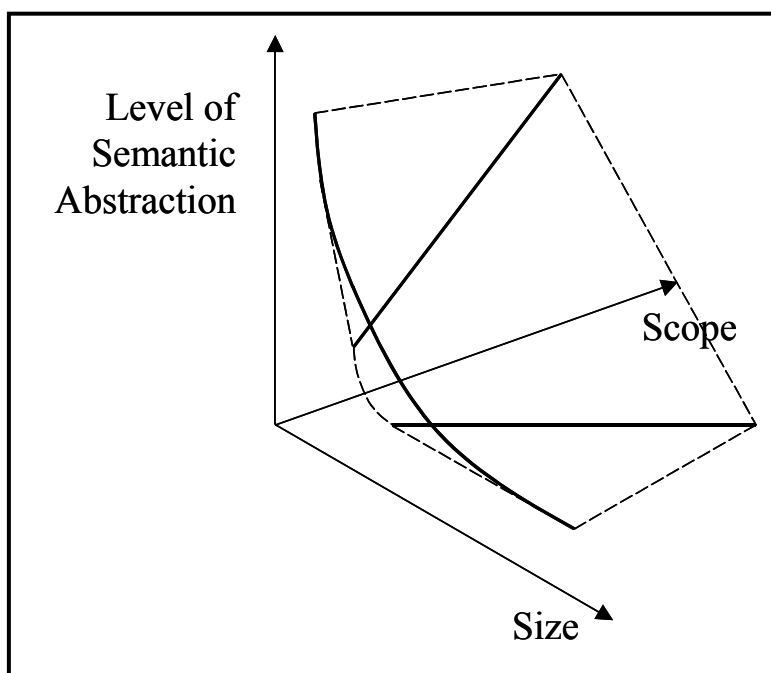


Figure 47. Relationship of LOSA, scope, and size in three axes.

5.2.3.2 Community Perspective Models

If one accepts that “local realities” are the knowledge, beliefs, etc., that stakeholders maintain in their minds and represent with perspective models, then a logical question is: What is “shared reality”? And is there a need for a model of “shared reality”?

A “shared reality” is not something that one can point at or touch—it is not a “thing” that exists in the real world. Rather, the “shared reality” that results from the social construction process is really the collection of very similar local realities held by a collection of stakeholders. The local realities are understood the same way, or mean the same thing to the stakeholders, with respect to the real world (e.g., the design environment) of which they are a part. “Shared realities” do not physically exist like local realities do (i.e., shared realities are not “brain phenomena”), but they are often manifested as procedures, models, in-

structions, guidelines, illustrations, the design environment, etc.—any kind of representational mechanism that is intended to promote the same understanding of the real world between two or more people.

However, like individual perspectives, community perspectives can be reified as a community perspective model. These models tend to have characteristics that fall on the right hand side of the spectrum shown in Figure 45. They are more abstract, have a larger usage population, and can denote a large population of real-world entities.

5.2.3.3 Collaborative Information Infrastructure Description

Figure 45 illustrates a spectrum of perspective model characteristics that are the basis for the Collaborative Information Infrastructure. The key concept for understanding the structure is that a more abstract perspective model (i.e., higher LOSA) can capture or represent the same information as a larger number of smaller, more narrowly scoped perspective models. By providing a mechanism for moving information between small scoped perspective models (those to the left of Figure 45) and more widely scoped, abstract perspective models (those to the right of Figure 45), the pathway exists for moving information between the perspective models of individual stakeholders. A individual stakeholder, then, is provided with data in exactly the form that he needs and understands.

Figure 48 builds on the previous illustration by including the perspective model characteristics described above and showing the decreasing number of perspective models required that service larger and larger meaning communities.

Figure 45 may be better presented as an “onion model,” as illustrated in Figure 49, rather than a cline or a layered model, the layered illustration presented in Figure 48. This model is a collaborative information infrastructure for perspective model reconciliation and integration.

The most esoteric characteristic of the layered model is the “point of generalization.” This “point” arises during the perspective integration process when two or more concepts (say *car* and *truck*) from different perspectives are combined and the denotation population of the new concept (say *automobile*) now subsumes the previous two. The effect is that to the stakeholder of one of the original models, the new concept does not *quite* mean what his old concept used to mean. Needless to say the further one generalizes, the harder it is for a stakeholder to “see” his original concept and, thus, his intended meaning.

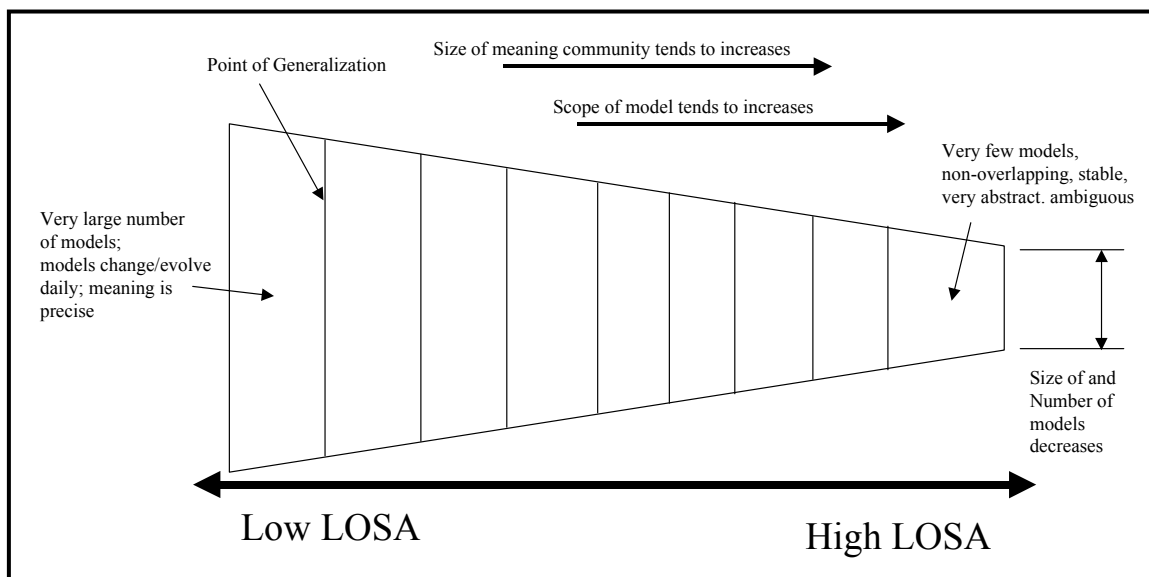


Figure 48. Layered model of collaborative information infrastructure.

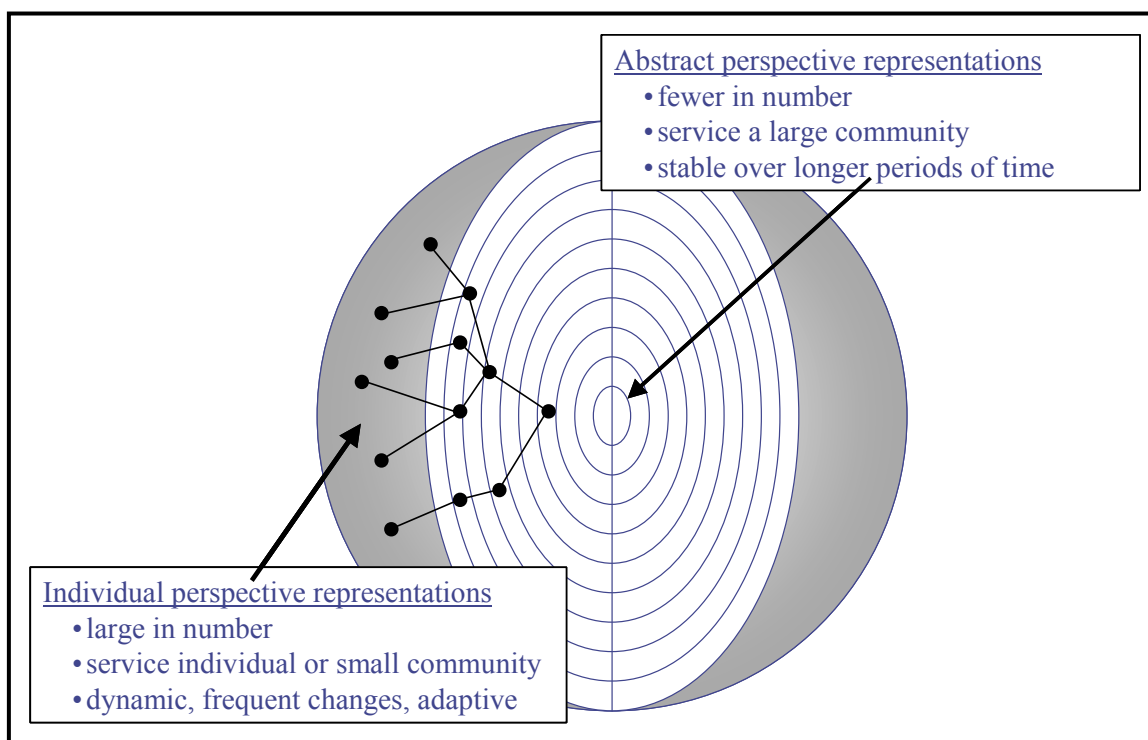


Figure 49. Onion model of collaborative information infrastructure.

Meanings and constraints that apply to perspective models of smaller meaning communities may necessarily be obscured or eliminated when perspective models are integrated. Thus, it is absolutely essential that the perspective models of the smaller communities *be maintained* as an element of the framework, co-existing with “bigger” perspective models that reflect the requirements of a larger meaning community. Integration of perspectives does not necessarily mean

that perspectives “go away”; rather, individual meanings and constraints are maintained within the framework.

5.2.3.4 Relationships between Perspective Models

This work is predicated on the belief that there should be/will be a very large number of perspective models upon which information technology applications are based. It is not feasible or possible to create a single, comprehensive “global” perspective model that serves the semantic requirements of all applications. Such a model would either be too large to be manageable, or too abstract (i.e., has a high LOSA) to be meaningful. Therefore one must accept the premise that there will be a large number of perspective models and the task is to determine the relationships among them.

The formal relationship between perspective models is called a *mapping* between the models. This work differentiates between two kinds of mapping:

1. Translation
2. Interpretation.

“Translation” is the mapping that occurs between “peer” perspective models. Peer perspective models address parts of the same problem domain (i.e., have overlapping *scopes*) and thus denote the same things in the “real world” and have the same concepts/meanings. The *contexts* of the perspective models are roughly the same. The different perspectives exist due to differing purposes, thus the information or constraints of the perspective are slightly different. Also, there may be simply the desire for a different representational structure.

“Interpretation” is the mapping between a semantically precise concept and a semantically more general concept. Interpretation occurs between points of generalization. Thus, an *automobile* in the perspective model of a large meaning community is “interpreted as” a *car* in a smaller community. The notion of “interpretation” does not currently exist within database or computer science research; it was introduced as an innovation in a product data exchange standard called STEP (Danner 1997) (Standard for the Exchange of Product model data – ISO 10303). “Interpretation” as a perspective mapping technique requires greater emphasis on the management of contexts, the importance of which with respect to data management is just now beginning to be investigated (Goh, Madnick, and Siegel 1998).

Note that some concepts in perspective models of small meaning communities will “percolate up” to the perspective models of very large communities with very

little change in meaning. These concepts are those that are present in every human endeavor. The concept of a person, for example, as characterized by name, title, and perhaps location (address) is likely to be applicable in all meaning communities, and thus is subject to the broadest standardization of meaning.

Reflecting the dynamic nature of social construction theory, the theories or perspectives and the collaborative information infrastructure, supports and engenders *frequently changing* individual perspective models. An individual's knowledge grows and evolves every day as he learns and interacts with others in his community. Therefore, his *perspective* also grows and evolves. Mapping relationships between individual and community perspective models both provides the freedom for individual perspective models to change and adapt (because they are only loosely coupled to the community model), and can easily be adapted to respond to changes in either the individual or community perspective models.

5.3 Interoperability Through Data Abstraction

Given the theoretical formulation of perspectives and their mediating and adaptive role between “internal” information and external, perceivable data, the question is how these theoretical ideas can be leveraged or applied to IT (information technology) development. What is the relationship of perspective models to data, data to users (stakeholders), and (most importantly) data to data? Research in application interoperability typically starts with the data. This work starts with the knowledge of the stakeholder.

The data-to-data relationship *is* important; it is a central focus of the research, but it must be couched within a broader theory. Socio-Technical Framework and perspectives models offer this theory.

Concretely, at the data level, this work seeks a comprehensive theory of data mapping. More broadly speaking, the object is scalable information architecture for application interoperability based on the concept of interpretation (through abstraction), which uses the data mapping technology as the “glue” for putting bits together. This is presented in three parts: a fundamental characterization of data (Axioms), a characterization of what data mapping is (Mechanisms), and then the description of a scalable integration method based on abstraction and data mapping (Methods). Axioms define the fundamental elements of the research. Mechanisms are the fundamental focus of the research. The rules and operators govern how data can be mapped from one form to another. Methods are the procedures and techniques for applying the mechanisms to foster scalable, adaptable, and integration information architectures.

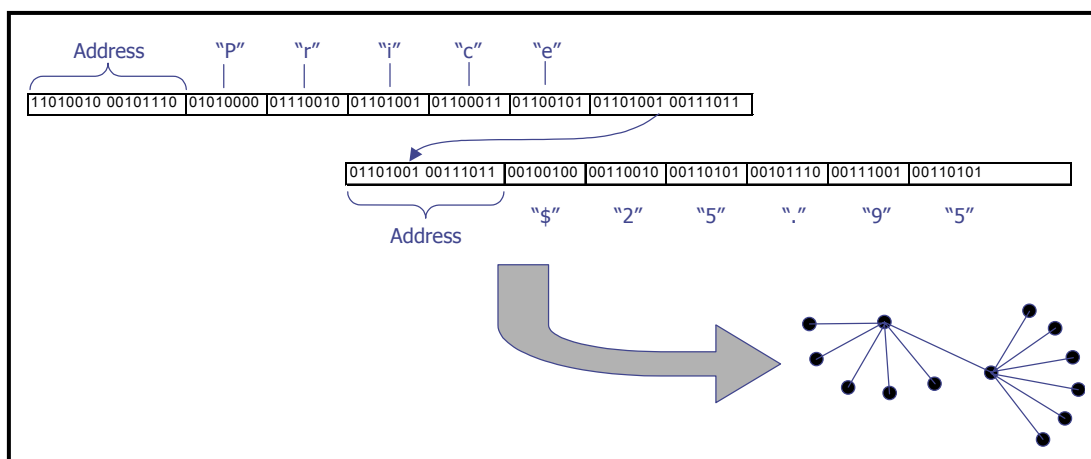


Figure 50. Data as a directed graph.

5.3.1 Axioms – Atomic Data Model

To establish formal relationships between heterogeneous data stores (thus enabling interoperability), a representation of the contents of data stores must be formulated that represents the “lowest common denominator” of all data stores. This fundamental characterization of data must be true of the representation of all digital data.

A simple directed graph (DG) is proposed here as the fundamental characterization of data. Digital data is an ordered collection of bits organized into chunks/groupings that are assigned structure/semantics/properties by the software (e.g., operating system, application) using/managing/creating the data. Eight bits make a byte; bytes are combined into groups that have differing functions, and these combinations are organized into successively more complex organized structures. Figure 50 shows this view of data.

The smallest practical unit of data to consider is usually a collection of bytes that commonly performs one of two functions: (1) representation of printable information (though appropriate interpretive and conversion processes), or (2) referencing another distinct (i.e., “addressable”) collection of bytes. These are commonly called fields. Fields are grouped together into larger groups often called a record or an instance; a record is typically the “thing” that has an address (i.e., an identifier or locator) and to which a reference field “points” to. Records are grouped together within a database or repository that is finite in size at an instant of time.

The interpretation of this view of data as a DG is as follows. There are two kinds of vertices in the graph: (1) The fields that are printable—and thus “atomic” or

“terminal”—are vertices in the graph and are called values. Values, as nodes in the DG, have the following properties: they have an in-degree of 1 and an out-degree of 0. Thus they are terminal nodes of the graph; (2) Records are also vertices and are called entities; they may be of any degree. It is easier to conceptualize a record as a vertex if one assumes that the address is the vertex rather than the collection of fields. Values and entities may be generalized as individuals, thus the set of individuals is equivalent to the set of vertices.

There is only one kind of edge in the DG data model, but it is manifested in two ways. The first, and most obvious, is the reference field that “points at” a record. This establishes a link between the record containing the reference field and another record; thus, this edge links an entity vertex with another entity vertex. The second is an implied edge between a record and a value field within the record.

Thus, a record/entity vertex has two kinds of outbound edges: those that link it with value vertices (indicated by the presence of a value field in a record) and those that link it with entity vertices (indicated by the presence of a reference field in a record). All inbound edges of an entity are reference-type edges. In the DG data model, edges are called properties.

The entire DG is called a population. The DG may not be connected, but it is finite; thus the population need not be completely interconnected, but it is bounded at an instant in time, all individuals (and properties) are part of a distinct population. Without considering the external environment of a population in too much detail, it is assumed for the present that a population has an identity and it can be referenced by things outside of the population.

This view of data is amenable to all static, declarative data stores and does not address behavioral aspects of data stores. Rather, it is a fundamental model of data values, aggregations of values (i.e., relationships between values), and relationships between aggregations. It is certainly true of the Relational Data Model (Codd 1970).

If this view is assumed as true, existing data models are merely semantic extensions of this atomic data model; other data models assign differentiable types or kinds to the values, properties, and entities, and associate behavior with the types. Thus data models, like the relational data model, are specializations of the atomic data model.

A schema further extends the semantics of the data model by instantiating and assigning types to the typed objects from the data modeling language. These

types are typically specific to a data usage domain and tell users of the data what the data “is” or “means.”

A schema is said to govern a population. “Govern” means that the schema controls, influences, or determines the behavior of the data population; it specifies the meaning and the structure of the data.

The characterization of data presented in this section is fundamentally or “atomically” representative of all data and data models. In terms of bits and bytes, all data is:

1. A finite collection of bytes (i.e., a population)
2. Chunks of bytes within this collection that can be collectively addressed (i.e., as an individual)
3. Sub-chunks of the chunk of bytes that are individual “fields” (i.e., properties) representing either:
 - a. A primitive value like a number or a character, or
 - b. A “pointer” value (or “address”) to a “chunk.”

Across different classes (or kinds) of data models there is a strong correlation between concepts that comprise each model and, thus, the terms used to refer to the concepts. This analysis, too, must leverage those same concepts, so terminology must be chosen that makes an understanding of this work and its relationship to other work clear. The following terms shall be used:

1. A Population of Individuals (scope by containment)
2. Individuals (aka “instance,” “occurrence,” or “object”)
3. Properties (of individuals).

A more precise definition of these terms will be presented in subsequent sections.

The names chosen for these concepts (Table 2) mirror the bits-and-bytes description above and recast terminology from database, object-oriented programming, conceptual modeling fields, and the rapidly developing field of web technology^{*}:

^{*} “Web technology” is a broad rubric denoting web technology standards such as XML, RDF, XML Schema, and the definition of community-based “vocabularies” or “ontologies.”

Table 2. Terminology correspondence

Term	Database	Object-Oriented programming	Conceptual Modeling	Web technology
Population	Database, data repository, data source, file	Object-base	Universe of discourse	Resource, document
Individual	Instance, record, tuple	Object	Entity	Element, element instance
Properties	Field, attribute, property	Attribute	Attribute, property	Attribute, content

5.3.2 Mechanisms – Data Mapping

If all data populations can be represented with the atomic data model, then it is possible to establish formal relationships between any two distinct populations of data. This is, of course, complicated by the additional semantics that data models and schemas introduce to the data, but atomic level must first be addressed before effects of the extended semantics can be understood and addressed.

The formal relationship between distinct populations is declared with a mapping specification; a mapping specification is comprised of mapping declarations. The purpose of a mapping declaration is like a function between sets; the domain of the function is one population, the range the other. The purpose of the research is to examine the nature and composition of these functions. A function of particular interest to this research is that of *equivalence* between populations (ultimately, semantic equivalence). This view, of course, entails the assumption that a DG/population can be considered as a set.

A design requirement for the mapping language is that it be sufficiently complete to serve as the control metadata for driving the automated conversion of data between data stores.

The term “data mapping” is used because the mapping functions are based on the atomic data model, which is strictly a simply model of data with no domain-specific or implementation-specific biases. Figure 51 shows the relationship between schemas, data, mapping specification, and conversion software.

The purpose of data is to represent information that is meaningful to or understood by a stakeholder. The assertion of the “equivalency” of subsets of two distinct populations is “information equivalency” or “semantic equivalency,” i.e., some subset of data in data store A “means the same thing” as some subset of data in data store B.

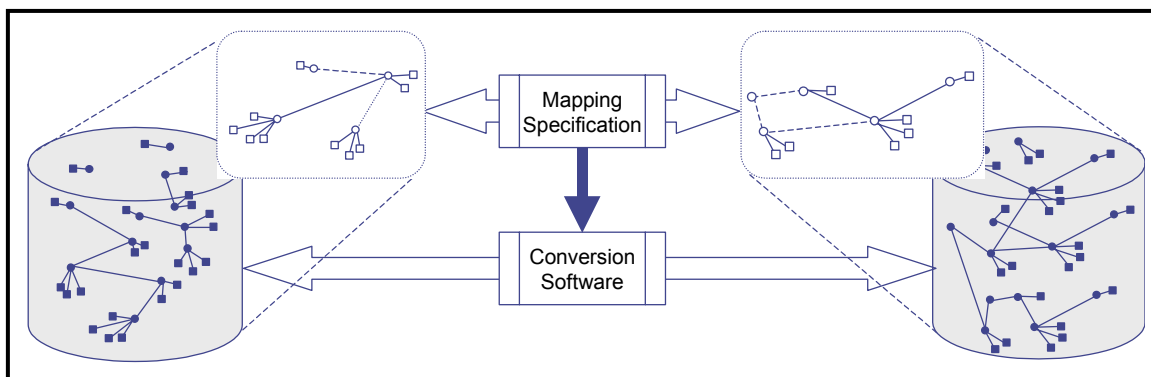


Figure 51. Mapping data between databases.

The assertion that they “mean the same thing” is a human decision; the “statement of equivalency” represented by the map must be *created by a stakeholder* who understands the data in each data store. It cannot be done automatically (though methods may be developed to facilitate the process, e.g., lexical analysis and pattern matching.) Note that equivalency is limited by (at least) mathematical invertability.

The reason that “equivalency” is of particular importance is because data within different populations often means the same thing to users of the data and it represents information in a form that can be stored for a duration of time, exchanged, or analyzed. Equivalency is just one special function. Other functions such as averages, sums, and sorts will also be part of the mapping specification language in which the transformations/relationships are asserted.

This research takes the view that “what may be mapped to what” is completely unconstrained within the bounds of the atomic data model. The only requirement is that both the domain and the range of the map are subsets of elements of the atomic data model: values, entities, and properties (including both the empty set and the maximal subset [which equals the entire population]).

The purpose of the unconstrained mapping is that there is no presupposition that the mapping model can make concerning the representational choices made to capture the semantics of the domain. All that can be said is that—at the most basic level—“this stuff over here means the same thing as this stuff over here.” The statement of this equivalence is left entirely to the individual writing the map, who supposed knows the semantics of each data population. Figure 52 shows the possible make-up of the domain and range of the map.

Source extent		Target extent
{} -- empty set, or {values}, or {entities}, or {properties}, or {{values}, {entities}}, or {{entities}, {properties}}, or {{values}, {properties}}, or {{values}, {entities}, {properties}}	Source extent maps to target extent under transfor mation T	{} -- empty set, or {values}, or {entities}, or {properties}, or {{values}, {entities}}, or {{entities}, {properties}}, or {{values}, {properties}}, or {{values}, {entities}, {properties}}

Figure 52. Data mapping domain and range.

Mapping specifications are themselves static. However, it is the intent of the research that they be executable in the sense that they can drive conversion software. Execution of the map copies, creates, and/or modifies the data in one (or both?) of the data stores. If the data populations are in active use (i.e., transactions are being lodged against them), then the only time that the equivalency can be said to be “valid” and/or “true” is in the instance immediately following the execution of the map.

The frequency and responsibility of execution of the map is outside the scope of this research.

5.3.3 Methods – The Integrated Information Architecture

The presentation thus far has covered two topics:

1. The atomic model of data
2. The fundamental structure/view of data mapping.

These models are fundamental to moving data between repositories in a way that maintains the meaning of the data. However, the result is simply a data translation paradigm (albeit more general and powerful than existing approaches). It is not enough for scalable IT integration because for n data stores, $n*(n-1)$ translations are needed when data stores are considered on a pair-wise basis, and

$$n \cdot (2^{(n-1)} - 1) \text{ (Or } n \cdot \sum_{i=1}^{n-1} \binom{n-1}{i} \text{)} \quad \text{Eq. 1}$$

translations are needed if the data stores are multiplexed. Therefore, something else is required for complex, scalable, integrated information structures. This work proposes the use of a *conceptual abstraction* of the integrated repositories that reduces the number of required interfaces for n repositories to $2 \cdot n$. Figure 53 shows the abstraction approach; Figure 54 shows the change in number of required interfaces plotted against the number of repositories.

Interpretation is a technique used in the industrial product data exchange standard ISO 10303 (STEP) (Danner 1997). It is used to map domain-specific concepts into a generic, abstract product data model (i.e., schema); the abstract data model specifies the data structures actually used to exchange the domain-specific information. In other words, there are two schemas: one specifies the information requirements of a narrow domain (e.g., automobile engine block manufacturing) and the other is an abstract product data model that can “hold” the specific information of the narrower domain. The semantics of the narrower domain are not lost in the generic model because interpretation technique sets “clues” the generic model to denote the interpretation or origination of the concept.

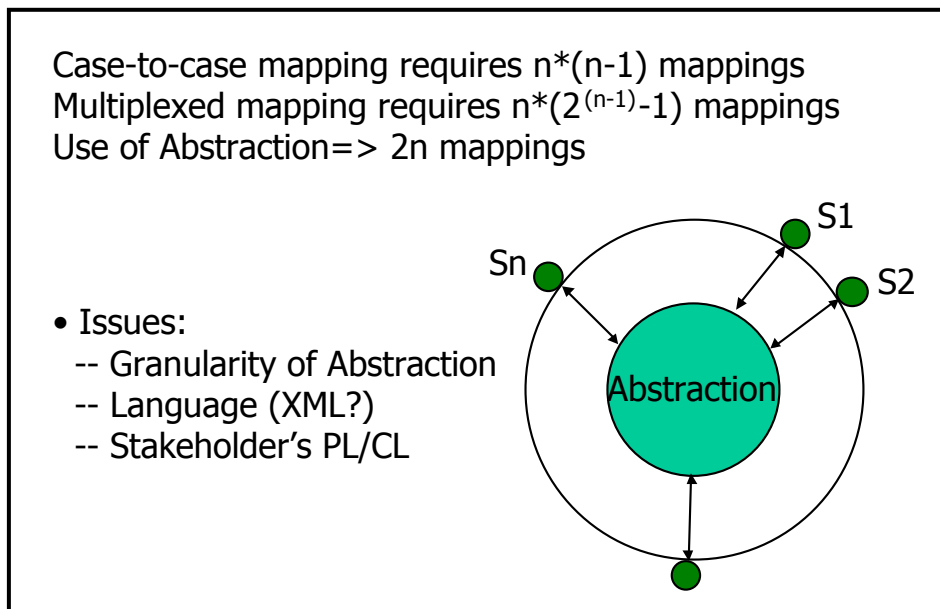


Figure 53. Abstraction as integration approach.

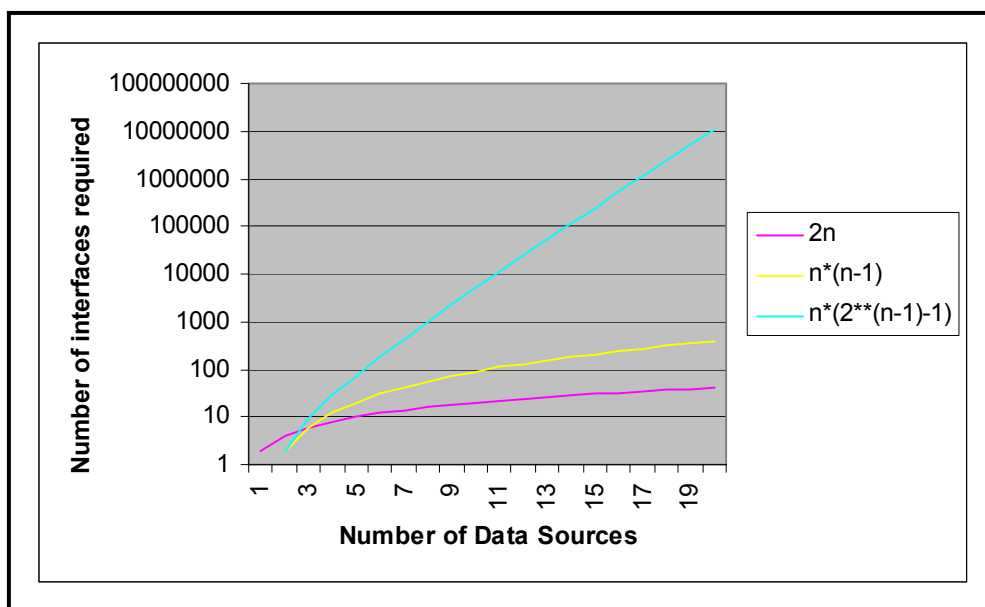


Figure 54. Number of required interfaces.

Interpretation is simply a reflection of the natural human tendency to generalize and categorize phenomena in the perceived world. The generalizations make the world easier to understand and grasp. “Interpretation” defined as a relationship between two schemas is based on two theories or meaning:

1. That the same meaning can be represented with two different schema (or ontologies), e.g., the same sentence in two different natural languages
2. That a true statement remains true even when the terms in the statement are generalized.

Figure 55 shows these theories as English natural language statements. There are precedents for interpretation in both philosophy (cf. C.S. Pierce) and in AI (cf. John McCarthy’s “lifting rules” for true assertions between contexts).

By using the technique of interpretation in an integrated information architecture, information systems can be constructed that can successfully scale up to service increasingly larger user communities in a manageable way while maintaining interoperability between the “low level” schemas (e.g., “local realities”).

The easiest form of the architecture to visualize is a hierarchy, as illustrated in Figure 56. Each perspective model is mapped through *interpretation* to a more abstract perspective model that services a broader community. In turn, this abstract community perspective model (i.e., a “shared reality”) can be mapped to an even more abstract perspective model serving an even larger community of stakeholders. (Without interpretation, this paradigm devolves to a *data warehouse* implementation.)

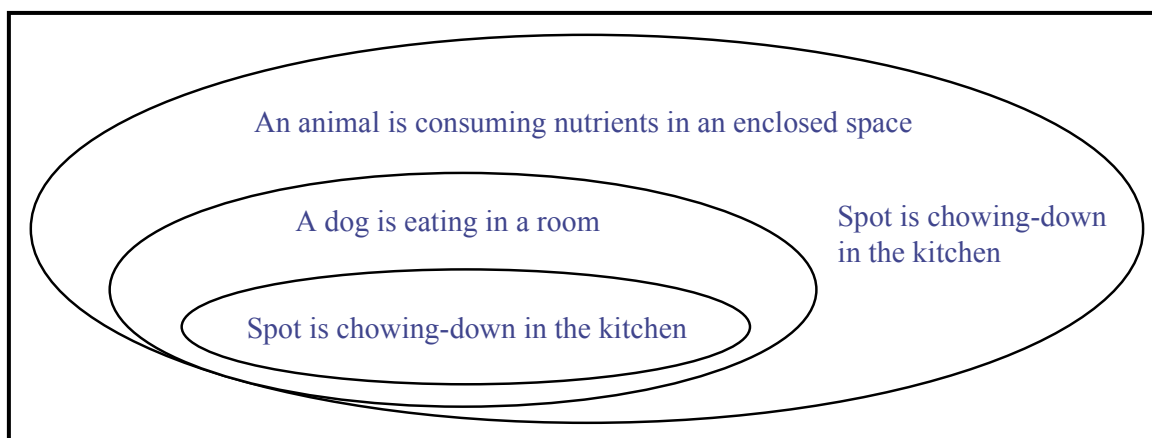


Figure 55. Truth maintenance in semantic abstraction.

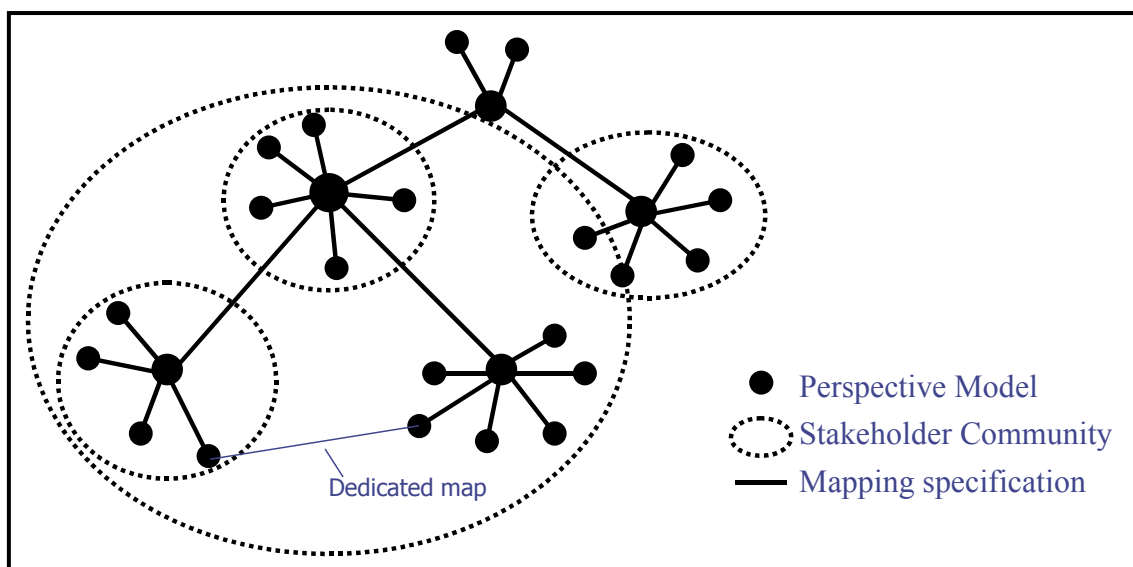


Figure 56. Integrated information infrastructure.

Although the hierarchy is the easiest to visualize and grasp, the use of data mapping as “glue” in conjunction with discrete models permits communities to be formed and evolve as requirements for the architecture change. The result is more “organic” in the sense that it adapts to environment stimulus and is complex and “messy” with respect to organization. For example, if a high-volume communication channel is needed between two stakeholders for some phase of an engineering design campaign, a dedicated map can be created to directly link the two. Once the phase is complete, the map can simply be discarded.

5.3.4 Perspective Model Analysis and Evaluation Methods for Derivation of Mapping Specification

The Atomic Data Model presented above represents data as a simple directed graph. This representation affords a number of quantitative metrics and analy-

sis techniques for evaluation and profiling of data populations and data models (i.e., schemas).

The consideration of the semantics of the data—i.e., the relationship between the data and the state of the real world as understood by a stakeholder—provides a number of qualitative metrics and analysis techniques for data populations and models.

These quantitative and qualitative techniques are introduced and discussed in this section. The ultimate objective is the application of these techniques to the automated integration or the automated derivation of mapping specifications between two data populations and their respective data models.

5.3.4.1 Purpose of Analysis

With the theoretical framework established above, the following sections outline the methods necessary for mapping, reconciliation, and integration of perspective models. Analysis of perspective models can be performed at two levels:

1. A quantitative, syntactic, structural level
2. A qualitative semantic level.

Reconciliation of perspectives requires that two kinds of analyses need to be performed:

1. Integration of perspectives of a single stakeholder
2. Analysis and reconciliation of perspectives across stakeholders.

The purposes of the analyses are to:

1. Identify, codify, and integrate individual perspectives of a stakeholder
2. Find similar/equivalent perspectives
3. Find equivalencies
4. Confirm equivalencies and resolve conflicts.

Once the analyses are complete, the relationship between perspective models is formalized by

1. Integrating the perspective models and forming a larger meaning community, or
2. Specifying the mapping between the perspective models.

The effect of the analysis and reconciliation of perspective models effectively *integrates* the models. What is significant about this approach to *integration* is that it is driven directly by the stakeholders; integration is not primarily a technology design concern, but rather a human integration concern that can be summed up in the simple formula:

$$\text{Integration} \Leftrightarrow \text{Communication}$$

The complete requirements for perspective model reconciliation and integration are not fully addressed here, but rather introduce the quantitative and qualitative approaches investigated.

5.3.4.2 Quantitative Characteristics

This work assumes that the perspective model is a statement of the stakeholder's view of "reality." Fundamentally, data and data models can be characterized as identified, bounded "chunks" of digital bits that are subdivided into "fields"; two kinds of fields serve as components of a "chunk," as described in the Atomic Data Model in Section 5.3.1:

1. Primitive data values (e.g., numbers and strings)
2. "Pointers" to other identified chunks of data.

The model, therefore, consists of elements with (primitive valued) attributes and pointers/relationships between elements. This characterization is as true for relational tables and object models as it is for semantic data models.

There are several quantitative characteristics that can be employed:

1. Graph theory
2. Dependency analysis
3. Data Model metrics.

Graph theory

Graph theory provides several mechanisms for analysis of models when viewed at a network of nodes and links between nodes. Particularly relevant is the analysis of the model as directed graph (digraph) since links between model elements are directed, i.e., they are "pointers" to other elements.

Graph theory offers two measures immediately applicable to model analysis:

1. The *order* of a model is the number of entities (nodes, objects, vertices).
2. The *size* of a model is the number of relationships (links, associations, edges).

Though these measure do not offer much insight, they provide a fast, reliable, repeatable measure for characterizing and comparing the perspective models.

5.3.4.2.1. Dependency Analysis

Since the model is a digraph, it can be analyzed with respect to dependencies among the model elements. Kusiak and Larson (1995) provide algorithms for grouping elements based on clusters of dependencies. Suh's work on Axiomatic Design (1990) is also fundamentally about dependency analysis. It is possible that coupling Suh's work on engineering design with the perspective model reconciliation methods investigated in this research would yield a powerful new and thorough approach to engineering design; this new approach would not be dependent on methods or philosophy, but only on the community of engineers undertaking the campaign.

5.3.4.2.2. Data Model Metrics

Two measures proposed by this research draw off the characteristics of digraphs and dependency analysis and are more directly focused on data models. Like *order* and *size* from graph theory, they are measures based on the structural characteristics of the model, which do not include semantics:

1. The degree of structural information encoding
2. The number of schema instance states.

5.3.4.2.2.1. Degree of Structural Information Encoding

Depending on the purpose and scope assumed by a stakeholder in the development of a perspective mode, the model can be positioned somewhere along the framework model (see Figure 48). To the "left" of the framework, the models may be very database-oriented and may be based on relational theory; these models are very "concrete" (i.e., they have low LOSAs) and the meaning of the tables (entities) and the fields (attributes) are usually very concrete, explicit, and specific. Bruce (1992), for example, presents a "high quality data model" of a market. While the meaning of the data governed by these kinds of models is very clear, the models are very rigid and brittle under changing requirements.

At the other extreme to the "right" of the framework (Figure 48) are the "conceptual" models (with high LOSAs) that purport to define a universally applicable (or very widely applicable, i.e., the model has a broad scope) set of concepts (i.e., an ontology) that can satisfy the needs of many data users. These models are very abstract and the meaning of the entities (tables) and attributes (fields) are usually very abstract, implicit, and generic. As one would expect from the dia-

metrically opposed view (with respect to concrete models), these kinds of models are very flexible and adaptable under changing requirements, but the meaning of the data governed by the model is fuzzy and uncertain.

While the qualitative differences between these models are readily apparent, there is a *quantitative* difference between them as well. The ratio of pointer-valued fields to primitive-valued fields of a concrete data model tends to be very low. The same ratio for the abstract, conceptual models is much higher. This ratio is called the *degree of structural information encoding*, or “dosie” (doe-zee). The concrete models have a low dosie because the information encoding is found mostly in the data fields (primitive-valued attributes) rather than the structure (pointer-valued attributes). Abstract models have a high dosie because the information encoding is found mostly in the structure of the model, in the relationships between entities.

Table 3 lists the results of a cursory study of three “high quality” data models and the dosies that were measured:

Both of these measures (dosies and “complexity”) have obvious implications with respect to the ability to reconcile perspective models because they affect the “understandability” of the model. Models with lower dosies and lower “complexity” will be easier to understand and implement.

5.3.4.2.2.2. Schema Instance States

“Pointer-valued” attributes in a data model represent existence dependencies among the entities in a data model. This means that a chunk of data (i.e., an *instance*) with a pointer-valued attribute cannot exist with the prior existence of another chunk of data to which it is pointing (at time of commit). If a data model consisted of five completely independent entities (i.e., none have pointer-valued attributes) then there are $2^5 = 32$ valid “schema instance states” according to that schema. Existence dependencies between entities reduces this number. Valid schema instance states are inherently controlled and enforced by the structure of the data model, but the meaningful validity of these states must be evaluated by users of the data. This becomes more difficult with abstract data models and with a large number of schema instance states.*

* Schema instance states also have an impact on the testing of databases; the number of instance states is the upper bound of test cases needed.

Table 3. Schema measures.

Model/source	Dosie	"Complexity"*
Market Model (Bruce [11])	0.34	1.063
ISO 10303-41 [28]	0.6289	1.7986
PIPPIN [48]	9.6923	3.2091

5.3.4.3 Qualitative Characteristics

While the quantitative analysis is an essential first step in reconciliation and integration of perspective models, much of the literature on data model/schema integration deals with the qualitative question of semantics: "What meaning is attributed to the model elements and the data by the stakeholders." It is these qualitative issues that pose the greatest challenge in perspective reconciliation.

Three qualitative conceptual model "measures" are intimately intertwined and cannot be "measured" in the engineering sense:

1. Scope (Section 5.2.3.1.2.)
2. Level of Semantic Abstraction (Section 5.2.3.1.3)
3. Validity and completeness (Section 5.3.4.2)

These characteristics were presented and defined above. The following observations about these "measures" may be made:

1. The *size* of the perspective model (i.e., the number of entities/concepts/objects in the model) is directly proportional to the size of the scope (of the application domain) addressed by the model.
2. The *size* of the conceptual model (i.e., the number of entities/concepts/objects in the model) is inversely proportional to the level of semantic abstraction of the model.
3. The ability of a user to evaluate or assess the *validity* and *completeness* of the model increases as the:
 - a. Size of the model decreases
 - b. Size of the scope decreases

* "Complexity" of the data model is a compound measure obtained by multiplying the number of objects/entities in the model by the dosie and taking the log of the result.

- c. Level of semantic abstraction decreases.

These relationships are illustrated in Figures 46 and 47.

These observations have a clear impact on the ambiguity, and thus the “quality” of perspective model. One of the primary research thrusts in this work is the identification of techniques to assess, evaluate, or “measure” these characteristics.

5.3.4.4 Reconciliation of Perspective Models

The methods for reconciliation, integration, and mapping of perspective models constitute the primary body of this research work. Several possible directions for this work follow.

5.3.4.4.1. Manual Model Integration

The integration of perspective models can be pursued in the manner described by Batini, Lenzerini, and Navathe (1986). This process simply brings together the members of a meaning community to discuss their individual perspectives and negotiate and specify a shared perspective model for the community. This approach produces acceptable results for small meaning communities, but becomes increasingly difficult for larger communities. In addition, it may not even be possible to bring together the members of the community due to spatial or temporal distances. Hence, automated means for reconciliation and integration to deal with the size and distance obstacles are desirable.

5.3.4.4.2. Structural and Pattern Analysis

A quantitative approach for the comparison and reconciliation of perspectives is a simple structural comparison based on the measures defined above. This could include comparisons of the order, size, dosie, or complexity of the perspectives.

By coupling the structural analysis with lexical analysis techniques, pattern analysis as pursued in research on neural networks can be applied to ascertain how perspective models “line up” with one another.

5.3.4.4.3. *Lexical and Natural Language Analysis*

To assess whether the elements of the model *mean* the same thing, some form of lexical analysis is required. This analysis could take the form of:

- identification of synonyms and homonyms of element names
- comparison of element definitions
- pattern and nature of first degree relationships of elements.

Hars (1998) and Bright, Hurson, and Pakzad (1994) have proposed methods that leverage computational linguistic theories to perform an automated semantic analysis of model elements to assess whether they “mean” the same thing. Such methods include etymological analysis (i.e., analysis of word roots for synonymity), definition correspondence analysis, and “semantic distance” per a summary schema.

5.4 Case Studies in Data Transformation through Abstraction

Theory of perspectives and interoperability-through-abstraction espoused above are both embodied in the Product Data Markup Language project (www.pdml.org). Despite its name, PDML is not a “markup language” in the same sense that HTML, for example, is a markup language. Rather, PDML is a *method* for structuring and integrating a suite of markup languages.

PDML was developed in an U.S. Department of Defense (DoD) program called Product Data Interoperability; the objective of the program was to develop the technology for integrating and exchanging product data between Product Data Management (PDM) systems over the World Wide Web. The program leverage several existing technologies, chief among them STEP and XML.

PDML is not a single data specification, but rather a structure of related specifications and tools. It is composed of the following components:

- Seven Application Transaction Sets (ATS)
- An Integration Schema
- Mapping specification between the ATSs and the Integration Schema
- The PDML Toolkit.

Figure 57 shows the relationship between these components. The following subsections present and explain the relationships between the ATSs, the Integration

Schema, and the Mapping Specifications, and discuss the relationship between these elements and the theory of perspectives described above.

5.4.1 Application Transaction Sets

Jargon, lexicons, vocabularies, and languages all develop and grow within “meaning communities”—collections of individuals to whom certain words and phrases have a specific meaning and within which the meaning is reinforced and evolves through usage over time. This is the main tenet of the theory of the social construction of knowledge (Berger and Luckman 1966), as described above. The PDML Application Transaction Sets are exactly that: vocabularies meaningful within a well-defined community—except that the community is defined as the users of a particular DoD legacy system like JEDMICS or standard like MIL-STD-2549.

Several recent papers and technical developments coming from the World Wide Web Consortium (<http://www.w3c.org>) reinforce this view of community-based “meaning” standards:

- In their vision of a “Semantic Web,” Berners-Lee, Connolly, and Swick (1999) recognize the trade-offs between local autonomy and global accessibility in the design/deployment of web data. Global protocols for access and exchange are necessary for scalability of the web, but localized standards are necessary to preserve localized, narrow-channel communication requirements. They also recognize that the definition of semantics is based on a usage community in which particular meaning and constraints are defined, built, and used.
- Context-sensitivity of names of XML vocabularies led to the development of the specification of Namespaces (Bray, Hollander, and Layman 1998). Namespaces provide a syntactic mechanism for differentiating between vocabularies developed for/by different communities.

PDML leverages the idea that semantics are local to a particular meaning community by delimiting a meaning community as the users of a particular legacy product data system. PDML was able to define a “complete and unambiguous” XML vocabulary for this community because the users had already had many years of experience using the terms in this vocabulary.

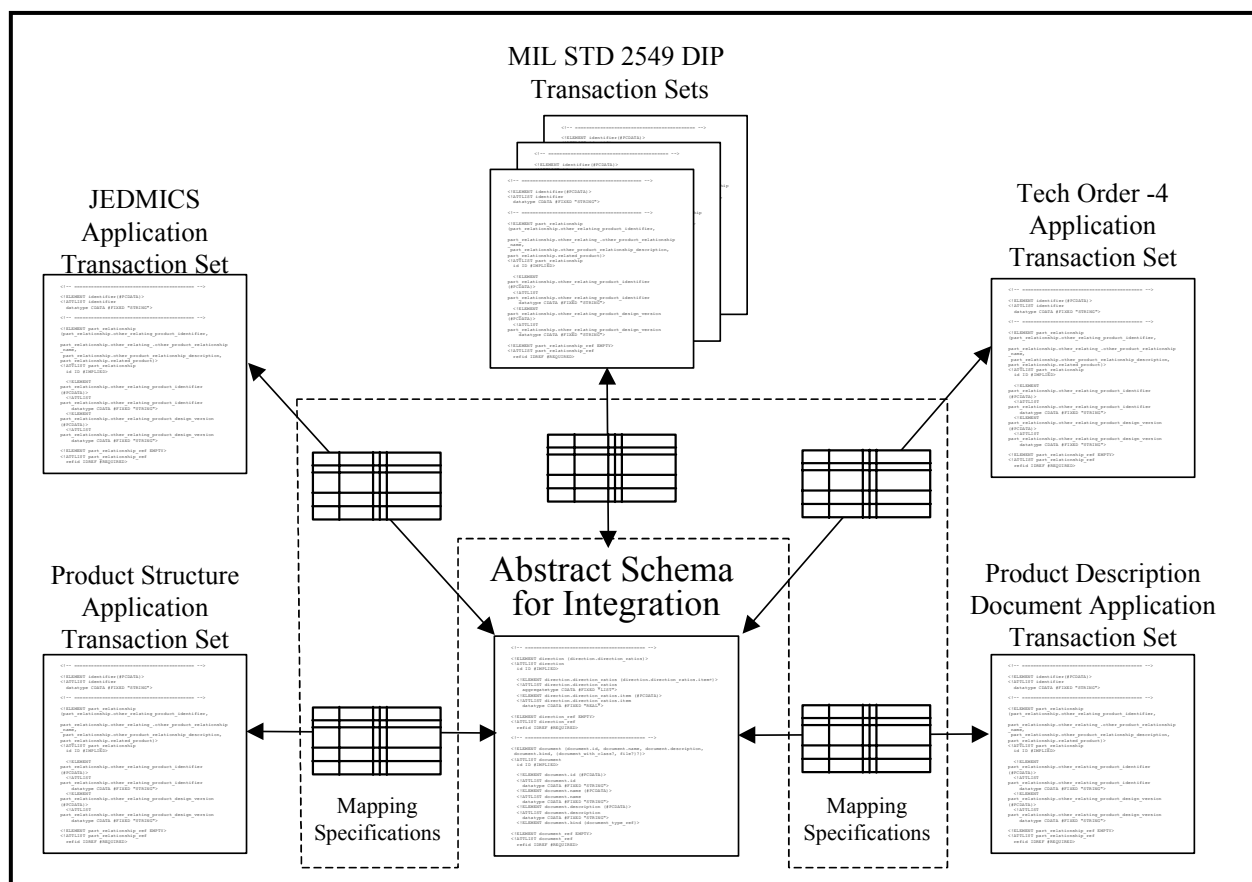


Figure 57. Structure of PDM.

For example, the Joint Engineering Data Management Information Control System (JEDMICS) is a very large (and very old) defense data system. It consists of data fields like those shown in Table 4.

Some of these fields might mean something to non-JEDMICS users like “drawing_number” or “sheet.” Who but a JEDMICS user, however, would know what a “control_code” was, or what “wsc” meant?

Table 4. JEDMICS data fields.

Drawing_number	sheet	foreign_secure
Drawing_title	sheet_revision	nuclear
cage_code	frame	wsc
doc_type	number_of_frames	safety
drawing_revision	control_code	dist
	security	master_location

The semantics of the ATS vocabulary was specified using a data modeling language called EXPRESS (ISO 10303-11) (ISO 1994). The EXPRESS schema was mapped to an XML DTD, which governed the exchange of ATS XML data. XML data can be exchanged between users using the ATS schemas if the communicating parties (i.e., software applications or stakeholders) both understand the terms in the ATS.

An ATS is an example of a Perspective Model in the following sense. The JEDMICS system is used by DoD engineering support personnel to track and manage engineering drawings of parts for equipment such as aircraft. Its purpose is to track/manage drawings. Its context is engineering logistics support (i.e., obtaining replacement parts for damaged equipment). Its content is shown in Table 4, which lists some of the data fields found in the JEDMICS database.

JEDMICS is not, however, a human stakeholder and does not have any “internal knowledge.” It is not really dynamic over time because users cannot change the meaning or organization of the fields. But it really is not static, either: fields have been added to the database and users often misuse the fields to convey other information that JEDMICS was not designed to handle. JEDMICS databases can be viewed, though, as a representation of “community knowledge.” It is a representation that is useful and meaningful to the JEDMICS user community and they can make decisions based on this data. JEDMICS is one of thousands of DoD information systems, many of which also contain data about engineering drawings, so there are other communities that have/need the same knowledge as the JEDMICS community.

5.4.2 Abstract Schema for Integration

Each Application Transaction Sets is a view (or subset) of product data necessary for DoD information system support. More precisely, an ATS is a representation of a small portion of the abstract body of information, I^* , that DoD uses in daily operation. Many of these representations semantically overlap because they are a representation of the same part of I – information such as `part_number` and `drawing_number` are common to two or more of the views. Thus, the semantics

* E is a representation of I .

of the ATSs overlap and must be reconciled and integrated if users of one ATS desire to share information with users of another, overlapping ATS.

There are two ways to do this. Traditional database texts (Elmasri and Navathe 1989) and most system integration approaches use schema merging to create a “global” schema: like concepts are identified, the definitions are reconciled and made one, and the schemas are merged. The result is a single integrated schema. This approach, however, loses some of the “flavor” of the original schemas in the integration process. As more schemas are integrated, the resulting integration schema eventually becomes too large to be manageable and the original schemas are lost in the mass of structure; another possibility is that the integrated schema is forced to a level of semantic abstraction in which the original ATS meanings are hopelessly ambiguous. Merging schemas and producing a global schema is not a scalable integration approach.

The second approach is in the technique of Interpretation as used in ISO 10303 (STEP Danner 1997) and is the approach adopted by PDML. Interpretation produces an abstract model for integration (called the Integration Schema in PDML), but the individual component ATS schemas are maintained; they are not lost or discarded after integration. A *mapping* is specified between the ATS schema and the abstract schema to bind the schemas to one another. The mapping specifies how the information that is represented by ATS data can be transformed into Integration Schema data, yet still represent the same information. The theory of interpretation is that the same meanings (semantics) can be equivalently represented using different data structures. This approach to integration is far more scalable than the first because the use of abstraction keeps schemas of manageable size while addressing more application domains.

The design of the Integration Schema is based on the STEP Integrated Resources (ISO 10303, cf. Danner 1997; ISO 1994). When an integrated, cross-application view of product data is needed, data is extracted from the source data systems using their ATSs, converted into and integrated by the Integration Schema, and then converted back to a specific ATS view. The PDML Toolkit provides the mapping and conversion capabilities that insulate the users of the individual views from the complexity of the mapping process.

The Integration Schema is not intended to be directly used for product data exchange. Rather, it is more appropriate to consider it a temporary neutral form for integration and view translation.

Unlike the more concrete Application Transaction Sets, the abstract Integration Schema is less susceptible to semantic drift due to its inherent semantic “fuzzi-

ness.” The generalized concepts and structures are meant to serve as a vehicle for carrying or conveying more precise semantics, but they “mean” more things and are thus able to accommodate a wider variety of meanings. The important difference between an ATS schema and the Integration Schema is the *scope* of the schema, i.e., that part of the real world (the application domain) that the data governed by the schema is intended to represent.

The Integration Schema is an example of a representation of a community perspective, or “community knowledge” in that it is the negotiated reconciliation of the component ATS perspectives. It has a purpose: the union of the purpose of the mapped ATSs. It has a context: the union of the ATS context. It has content: the union of ATS content. The Integration Schema *only* exists for the purpose of serving as an abstract representation for the ATS perspectives that are mapped to it.

The union operator, however, is not simply an additive process because it is full of conflicts. Combining contexts required the examination and “positioning” of the contexts relative to one another. Combining purposes obviously involves the recognition and resolution of contradictory or redundant objectives. Combining content requires semantic analysis and reconciliation/ integration strategy. But it is also important to recognize that all this conflict detection and resolution is *normal everyday human communication behavior*. It is not unusual, anomalous, or wrong.

5.4.3 Mapping Specifications and View Integration

The Application Transaction Sets are application-specific *views* of product data that define a narrow context of data usage. The Integration Schema is an application *independent* view of product data and establishes a context of product data usage that encompasses the contexts of the application views. As a view, the Application Transaction Sets can be considered as a particular *interpretation* of the Integration Schema. This interpretation is formally specified by a Mapping Specification.

A Mapping Specification is a statement of semantic equivalence. It states that the information represented by the data structures in the ATS is equivalently represented by Integration Schema data structures and rules.

Mapping is more than conversion of data between data structures. It encompasses the interpretation of data based on contextual values—a value from a single field does not *always* mean exactly the same thing (though it always *generally* means the same thing.) Based on a contextual value that indicates the

use, a field such as `document.id` could be drawing number, a tech order number, the designation of a standard or specification, or the identification of a digital file.

The DataQuest™ integration engine in the PDML Toolkit “internalizes” and uses the Mapping Specification to drive the conversion of XML data to/from the Integration Schema format.

Mapping specifications have no direct or explicit analog in the theory of perspectives. They are part of the information sharing process and the social construction process that results in community knowledge. They represent a negotiated contract of socially constructed meaning.

5.5 XML—Its Usefulness and Its Shortcomings

The Standard Generalized Markup Language (SGML) was developed by representatives of the publishing industry and standardized in the 1980s to provide a flexible and powerful ASCII character-based language for encoding the logical content of document free from consideration of presentation issue.

XML is “... a simplified subset of SGML ... optimized for the web environment, which implies data-processing-oriented (rather than publishing-oriented), and short life-span (in fact, usually dynamically generated) information.” (Goldfarb and Prescod 1998) XML is the ideal approach for deploying structured information on the web because it marries the presentation-free content management view of SGML (without many of the publication biases) to the *de facto* language syntax of web established by HTML. (HTML is an example of an *XML Vocabulary*, a set of tags defined with a specific purpose and application mind.) Figure 58 shows an example XML document.

XML is the syntax of choice for exchange of web data, leveraging a middle ground between full-blown document structure and content management offered by SGML, and the presentation mechanism provided by HTML. SGML originated in the field of text processing. The developers of the SGML language made an important realization that there is a distinction between the content of the document and the manner or style in which it is presented. HTML is a simple *application* of SGML designed to present content on the World Wide Web, which brought SGML onto the Internet as a data structuring and exchange syntax.

```
<research_paper>← Start tag
  <title>Conflict Management in Collaborative
    Engineering Design</title>
  <author>Stephen Lu</author>
  <abstract>The . . . </abstract>
  <keywords>collaboration, conflict</keywords>
  <body>
    <section_title> ... </section_title>
    <section_text> ... </section_text>
  </body>
</research_paper>← End tag
```

Figure 58. XML example.

An important feature of XML is that it is just a *syntax* for encoding data; there is no (or very little) inherent *meaning* in XML documents. Therefore, the development and application of XML vocabularies require that the tags and usage of the tags be clearly defined and consistently used to be effective as an information transport mechanism and thus enable application interoperability. In this sense, XML vocabularies are the same kind of thing as a database schema. And they are subject to all the same shortcomings that those involved in Enterprise Application Integration (EAI) have been facing for decades.

There are three primary challenges that the users of XML will face as their applications expand and grow:

1. Limitations of “standard” vocabularies
2. Integration of semantically heterogeneous vocabularies
3. Subtleties of data, meaning, and human communication.

It is a widely accepted (though naive) notion that, if an industry group can develop and use a standardized set of tags (i.e., an XML vocabulary), then applications used by industry can successfully interoperate using these tags. The underlying thinking is that interoperability is possible if one can “get everyone to use the same vocabulary as *lingua franca* for <insert favor topic>.” So widespread and accepted is this idea that *registries* for XML vocabularies have arisen (e.g., BizTalk [Microsoft], or OASIS) that act as a library or catalog of vocabularies that anyone can visit and use.

EAI lessons suggest that standardized vocabularies are not an effective or robust solution. Reasons include:

- Daily communication needs of users are too varied and dynamic
- The need for “standard” meaning is diametrically opposed to the precise, narrow meaning required to unambiguously communicating a bit of information.

Standard meanings *are* possible, however, in some situations:

- for simple, ubiquitous concepts (e.g., “person”)
- within very narrow/specific application domains.

A more significant problem that really is not recognized yet is the issue of integration of and across vocabularies. Generally, “integration” means “get everyone to use the same vocabulary.” But as shown above, different schemas/ vocabularies can represent the same information using different tag names and structures – making it difficult or impossible for these vocabularies to “interoperate.” But as vocabularies grow in size, application, and number, this problem will become more acute and more difficult to resolve. EAI has shown that integration through schema merging will not work and has yielded abstraction techniques such as interpretation.

As a vocabulary grows to meet more requirements and service more users, one of two things must happen:

1. The number of element types must grow, making the vocabulary unmanageable.
2. The meaning of the element types must become more abstract or generic, making the vocabulary more ambiguous.

Finally, an entirely research aspect of the problem deals with the subtleties of data, meaning, and human communication. The purpose of data is to convey meaning, and “meaning” is a human cognitive phenomenon; although one may say that data conveys “meaning” to a program, the *programmer* is human and must anticipate meanings of data processed by the program. The use of data by programs is subject to every error of misinterpretation and ambiguity that occurs in human language use.

Despite the fact that meaning is a human phenomenon, there is virtually no XML, EAI, or database research that touches on the vast research literature in linguistics, philosophy, sociology, and psychology on the subject of meaning. All of this research is almost completely ignored. The research presented in this report directly addresses this shortcoming.

6 Summary and Conclusion

6.1 Current Understanding

This report has outlined some new avenues investigated over the past 3 years jointly by researchers at the IMPACT Research Laboratory, School of Engineering, University of Southern California, and at the U.S. Army Engineer Research and Development Center, Construction Engineering Research Laboratory (ERDC/CERL). These research ideas collectively suggest a new, revolutionary approach to understand, study, and develop computer tools to support collaborative engineering. The approach developed here brings in new dimensions to the modeling of collaboration activities, and opens much room for further basic research. The socio-technical approach to collaborative engineering presented in this report hinges on the following basic understandings:

1. Collaboration is a human activity. Any enabling technologies that are developed need to take explicit account of the socio-technical nature of such activities by stakeholders (humans).
2. Conflict management acquires a central role in the socio-technical approach to collaboration. It involves the control, prediction, resolution and (in certain situations) even the sustenance of conflict.
3. Explicit representation of stakeholder perspectives is needed for the investigation and management of collaborative activities.
4. External data and internal stakeholder knowledge must be shared, co-constructed, and created through the dynamic models of perspectives working in conjunction with stakeholder-related data models.
5. Collaborative engineering design is modeled as a *socio-technical* construction process. A design campaign is undertaken within a design environment. Its purpose is to provide (in addition to the designed product) feedback to the design campaign and the design environment.
6. The socio-technical construction process calls for the explicit modeling of the “*who’s*” (i.e., stakeholders) and their perspectives in a design campaign as part of the design process model.
7. Quality *originates* in the stakeholder. It is *created*, through co-construction, by the interaction of stakeholders.

8. Inputs to the stakeholders in engineering design may be thought to be their belief systems, their competencies, and the constraints under which they operate. Their outputs, at any given time, are their commitments, their concerns, and the triad of goals, processes, and metrics they generate. It is through co-construction that both their inputs and the outputs are modified, expanded, and created anew.
9. *Conflicts* play a major role in co-construction and provide a basis for group progress and design innovations. What is important is the control and management of conflict, not its complete elimination from the design process.
10. Design goals, process models, and quality metrics, are interdependent aspects of any task or sub-task and are generated *simultaneously* through the co-construction of different stakeholder perspectives.

Exploration of these ideas has produced a large number of research issues, some of which have begun to be addressed in this research effort. Although significant progress has been made, the development of these new ideas requires substantial further investigation, research, and development.

6.2 Summary of Basic Research

The objective of this basic research has been to develop a *Theory for Collaboration* to support complex engineering system decisions, such as the Facility Engineering Framework at CERL. The focus of this work has been to contribute to a better understanding of human collaborative behavior, and to discover how modern IT should be designed to support these group activities. These research results will close some significant, basic knowledge gaps, and will lead to sound theoretical foundations that can be used to analytically and mathematically model, simulate, and optimize collaborative engineering activities.

This research program is based on a new paradigm, called the *Socio-Technical Framework of Collaborative Engineering*. Based on this framework, researchers have developed a system architecture with several key components, each involving and utilizing some basic modeling techniques. The main ideas behind this framework and its architecture have come (generally) from many social and organizational sciences, and (specifically) from *co-construction* process adapted from the theory of social construction. Whenever appropriate, the modeling techniques for key components are drawn from basic studies and fundamental knowledge in the fields of logic, mathematics, decision sciences, information technologies, etc. The further advancement and integration of these fundamental techniques, plus the new knowledge generated here, collectively represent basic research contributions to the science of collaboration.

These basic research activities included:

1. The Foundation: The Socio-Technical Framework for Collaborative Engineering.
2. The Components:
 - a. *Perspective Modeling* to capture stakeholders in collaborative engineering
 - b. *Process Simulation* to support distributed, collaborative activities
 - c. *Conflict Strategies* to manage collaboration
 - d. *Information Sharing* to capture, encode and relate the “data resources” of the enterprise based on evolving human perspectives
 - e. *Collaborative Information Infrastructure* to serve as a conduit for data mapping, co-construction and decision supports.
3. The Basics:
 - a. *Perspective Modeling*
 - (1) Dynamical systems
 - (2) Control theory
 - (3) New approaches being researched
 - b. *Process Simulation*
 - (1) Petri net
 - (2) Process management
 - (3) Dynamic simulation
 - (4) New approaches being researched
 - c. *Conflict Management*
 - (1) Multi-objective decisionmaking
 - (2) Utility/value theory
 - (3) Game theory
 - (4) Fuzzy mathematics
 - (5) Decision sciences
 - (6) New approaches being researched
 - d. *Information Sharing*
 - (1) Relational algebra
 - (2) Theories of logic
 - (3) Graph theory

- (4) Mathematics of inference
- (5) New approaches being researched
- e. *Collaborative Information Infrastructure*
 - (1) Networking theory
 - (2) Information theory
 - (3) New approaches being researched.
- 4. The Contributions:
 - a. Fundamental research into the above areas under the new Socio-Technical Framework will generate much new knowledge and contribute to closing the following three key knowledge gaps that are critical to the establishment of the Theory of Collaboration:
 - (1) A new information theory that directly relates to “meaning”
 - (2) Self-organizing, continuously evolving, intelligent, collaborative systems
 - (3) Computer-mediated human-to-human interactions
 - b. With such a theory of collaboration in place, it becomes possible to design, predict and control various collaborative activities, systems and environments. It will also become possible to implement practical IT systems to support these important human endeavors.

6.3 Some Future Research Directions

Further research and development efforts are needed in a few central areas to provide both the enabling methodologies and the consequent technologies for collaborative activities.

6.3.1 *Development of the Socio-Technical Framework for Collaboration Activities*

To date, this research takes on the central theme of a socio-technical framework, within which human perspectives are explicitly modeled during collaboration, to provide conflict management supports. The underpinnings of this socio-technical framework that has been developed so far need to be explored further, in particular the interactions between the various basic concepts, processes, perspectives, and co-construction strategies. Issues such as usefulness and consistency must be rigorously addressed. A proper evaluation of the framework based on the effi-

cacy of the methodologies and the implementation schemes that it has generated must be performed.

6.3.2 Development of Methodologies and Models Based on the Socio-Technical Framework

Careful assessments are needed to verify and improve the methodologies that have been developed from this socio-technical framework. Models used here for perspective templates, for setting up dynamical perspective models, and for developing this schemas in STARS need to be further evaluated and understood. Their strong points are well understood; however, their weaknesses need more exploration. This exploration requires collecting case study data from the utility of the results that these models produce in actual application domains (see Section 6.3.5 below).

To date, the dynamical modeling techniques have been used as the primary way to link this conceptual framework with functional implementations. These modeling techniques and their applicability require further evaluations. Also one needs to explore the dissection of the problem of collaboration and co-construction into pieces *other than* conflict management, process management, and perspective description and management.

6.3.3 Development of Tools Based on the Methodologies

6.3.3.1 Methods

The research emphasis here will focus on the development of enabling software technologies for collaboration activities. In other words, there is a need to create guidelines for the development of alternative tools for:

1. Process modeling and simulation tools that govern the execution of an activity campaign
2. Perspective modeling and management tools that deal with adaptive co-construction and understanding sharing
3. Conflict management tools in the execution of processes using perspectives as a way of creating new information through co-construction.

The tools must be designed to be interoperable, scalable, and capable of being integrated into the collaboration environment.

Using the understanding of collaborative activities provided by this framework, there is a further need to: (1) obtain an understanding of off-the-shelf-software tools, (2) develop criteria for evaluating and choosing commercial software, and (3) generate guidelines for developing new-generation software packages for enabling collaboration across a spectrum of engineering processes.

6.3.3.2 Tool Integration

The development of interoperable tools must extend to being able to combine their usage in a given environment so that they can make an integrated system for collaborative management. This can only be done if attention is also concurrently paid to the development of guidelines for data repositories so that they are properly structured to support such tool integration and interoperability. This also reflects on the need for proper specification languages for the description of elements of the various schemas that the tools use.

Further research needs to be done on the use of languages like XML, and the development of tools that co-construct the “semantic web,” which necessarily needs to be woven for perspective sharing, process management co-construction, and conflict resolution during collaboration.

6.3.4 Automation of Collaboration through Software Agents

The research goal is to create coordination and conflict resolution at a certain level of granularity by developing software agents that have knowledge of stakeholder perspectives and can negotiate on behalf of the stakeholders that they represent. The development of such agents would increase the efficiency of the collaboration process and allow the stakeholders to focus on problems of central importance to the completion of major activity milestones (nodes, in the Petri process model described here) while allowing issues of lesser importance to be automatically resolved by the software agents representing them. Thus the aim is to embed “intelligent” agents in the design environment that possess abilities to capture and integrate both internal knowledge and external data, and negotiate at a certain level of granularity.

6.3.5 Application and Deployment

The framework, the methodologies, the tools developed, the integration strategies deployed, and the software agents’ performance, can only be assessed by actually using the tools developed in certain real-life domain-specific environments. This is a necessary requirement for improving the understanding of this framework, for pointing out deficiencies in it, and suggesting changes. Most im-

portantly, it is perhaps the best vehicle for moving the research done to date from a purely *descriptive theory* towards a *prescriptive* theory of collaborative activity management. Several test bed scenarios therefore need to be investigated.

Bibliography

- Adler, P., A. Mandelbaum, "From Project to Process Management: An Empirically-Based Framework for Analyzing Product Development Time," *Management Science*, vol 41 (1995).
- Bahler, D., C. Dupont, and J. Bowen, "Mixed Quantitative/Qualitative Method for Evaluating Compromise Solutions to Conflicts in Collaborative Design," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, vol 9 (1995), pp 325-336.
- Batini, C., M. Lenzerini, and S.B. Navathe, "A Comparative Analysis of Methodologies for Database Schema Integration," *ACM Computing Surveys*, vol 18, No. 4 (1986), pp 323-364.
- Berger, P., and T. Luckman, *The Social Construction of Reality: A Treatise in the Sociology of Knowledge*, ISBN 0-385-05898-5 (Doubleday, New York, 1966).
- Berners-Lee, T., D. Connolly, and R.R. Swick, "Web Architecture: Describing and Exchanging Data," www.w3c.org/1999/06/07-webdata (Date of page: 7 June 1999. Found c. 1999-06-14).
- Binmore, K.G., "Why Game Theory 'Doesn't Work'?", in *Analyzing Conflict and its Resolutions: Some Mathematical Contributions*, P.G. Bennett, ed. (Oxford University Press: Oxford, 1987), pp 23-42.
- Bras, B.A., and F. Mistree, "Designing Design Process in Decision-Based Concurrent Engineering," *SAE Transactions Journal of Materials & Manufacturing*, vol 100 (1991), pp 451-458.
- Bratman, M.E., "What is Intention?", in *Intentions in Communication*, P.R. Cohen, J. Morgan, and M.E. Pollack, eds. (MIT Press: Cambridge, 1990), pp 15-31.
- Bray, T., D. Hollander, and A. Layman, *Namespaces in XML*, <http://www.w3.org/TR/PR-xml-names> (Date of page: 1998-11-17).
- Bright, M.W., A.R. Hurson, and S. Pakzad, "Automated Resolution of Semantic Heterogeneity in Multidatabases," *ACM Transactions on Database Systems*, vol 19, No. 2 (1994), pp 212-253.
- Bruce, T.A., *Designing Quality Databases with IDEF1X Information Models*, ISBN 0-932633-18-8 (Dorset House Publishing, New York, 1992).
- Codd, E., "A Relational Model for Large Shared Data Banks," *Communications of the ACM*, vol 13, No. 6 (1970).

- Crabtree, M., and J. Powers, eds. *Language Files Material for an Introduction to Language*, 5th ed. (Ohio State University Press: Columbus, 1991).
- Danner, W.F., "Developing Application Protocols (APs) Using the Architecture and Methods of STEP (STandard for the Exchange of Product Data)," *Fundamentals of the STEP Methodology*, NISTIR 5972 (National Institute of Standards and Technology, 1997).
- David, R., and H. Alla, *Petri Nets and Grafcet* (Prentice Hall, Cambridge, 1992).
- Dunsus, B.V., "Using Single Function Agents To Investigate Conflict," *AIEDAM*, vol 9 (1995), pp 299-313.
- Elmasri, R., and S.B. Navathe, *Fundamentals of Database Systems*, ISBN 0-8053-0145-3 (Benjamin/Cummings, Redwood City CA, 1989).
- Eppinger, D.S., "Generalized Models of Design Iteration Using Signal Flow Graphs," *Research in Engineering Design*, vol 9 (1997), pp 112-123.
- Goh, C.H., S. Madnick, and M.D. Siegel, *Semantic Interoperability through Context Interchange: Representing and Reasoning about Data Conflicts in Heterogeneous and Autonomous Systems* (Sloan School of Management, MIT, 1998).
- Goldfarb, C., and P. Prescod, *The XML Handbook: Open Information Management*, E.C. Goldfarb, ed., ISBN 0-13-081152-1 (Prentice Hall, Upper Saddle River, NJ, 1998).
- Grandhi, R.V., and G. Bharatram, "Multiobjective Optimization of Large-Scale Structures," *AIAA Journal*, vol 31, No. 7 (1993).
- Hars, A., "Natural Language-Enabled Data Modeling: Improving Validation and Integration," *Journal of Database Management*, vol 9, No. 2 (1998), pp 17-25.
- Hauser, J.R., and D. Clausing, "The House of Quality," *The Harvard Business Review*, vol 3 (1988).
- Hazelrigg, G.A., "The Implications of Arrow's Impossibility Theorem on Approaches to Optimal Engineering Design," *Journal of Mechanical Design*, vol 118 (1996), pp 161-164.
- Hazelrigg, G.A., "On Irrationality in Engineering Design," *Transactions of the ASME*, vol 119 (1997), pp 194-196.
- ISO, *Industrial Automation Systems and Integration—Product Data Representation and Exchange: Part 1: Overview and Fundamental Principles* (International Standard Organization, Geneva, 1994).
- ISO, *Industrial Automation Systems and Integration—Product Data Representation and Exchange: Part 11: EXPRESS Language Reference Manual* (International Standard Organization, Geneva, 1994).
- ISO, *Industrial Automation Systems and Integration—Product Data Representation and Exchange: Part 41: Integrated Generic Resource, Fundamentals of Product Description and Support* (International Standard Organization, Geneva, 1994).

- ISO, *Structuring Principles and Reference Designations, Part 1: Basic Rules* (ISO/EIC: Geneva, 1994).
- Jenson, K., *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use* (Springer, New York, 1996).
- Klein, M., "Supporting Conflict Resolution in Cooperative Design Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol 21, No. 6 (1991), pp 1379-1390.
- Krogstie, J., O.I. Lindland, and T. Sindre, "Towards a Deeper Understanding of Quality in Requirements Engineering," *Proceedings of the 7th International Conference on Advanced Information Systems Engineering (CAiSE'95)* (Springer-Verlag, Jyvaskyla, Finland, 1995), pp 82-95.
- Kusiak, A., and N. Larson, "Decomposition and Representation Methods in Mechanical Design," *Transactions of the ASME*, vol 117, No. 6 (1995), pp 17-25.
- Lewis, K., and F. Mistree, "Collaborative, Sequential, and Isolated Decisions in Design," *Proceedings of the DETC '97* (ASME Design Engineering Technical Conference, Sacramento, CA, 1997).
- Lindland, O.I., G. Sindre, and A. Solvberg, *Understanding Quality in Conceptual Modeling* (Institute of Electrical and Electronics Engineers [IEEE] Software, 1994), pp 42-49.
- Lu, S.C.-Y., and J. Cai, "Modeling Collaborative Design Process with a Socio-Technical Framework," *Proceedings of the Proceedings of 6th ISPE International Conference on Concurrent Engineering* (Bath, United Kingdom, 1999).
- Lu, S.C.-Y., et al., "A Methodology for Collaborative Design Process and Conflict Analysis," *CIRP Annals*, vol 49, No. 1 (2000).
- Lu, S.C.-Y., et al., *Conflict Management in Collaborative Engineering Design* (University of Southern California: Los Angeles, 1998b).
- March, J.G., "Bounded Rationality, Ambiguity and the Engineering of Choice," *The Bell Journal of Economics*, vol 9, No. 2 (1978).
- Mas-Colell, A., M.D. Winnston, and J.R. Green, *Microeconomic Theory* (Oxford University Press, 1995).
- McPherson, S., *Java Servlets and Serialization with RMI* (Sun Microsystems, 1999).
- Mistree, F., and J.K. Allen, *Optimization in Decision-Based Design* (Working Notes of Open Workshop on Decision-Based Design, 1997).
- Osborne, M.J., and A. Rubinstein, *A Course in Game Theory* 3d ed. (The MIT Press, Cambridge, MA, 1994).
- Pahl, G., and W. Beitz, *Engineering Design: A Systematic Approach* 2d ed., ISBN 3-540-19917-9 (Springer, Berlin, 1996).

- Parsons, J., and Y. Wand, "Choosing Classes in Conceptual Modeling," *Communications of the ACM*, vol 40, No. 6 (1997), pp 63-69.
- Petrie, C.J., "Using Pareto Optimality To Coordinate Distributed Agents, *AIEDAM*, vol 9 (1995), pp 269-281.
- Rao, S.S., and T.I. Freiheit, "A Modified Game Theory Approach to Multiobjective Optimization," *Transactions of ASME*, vol 113 (1991).
- Reschke, R., ed. *PIPPIN Pilot Implementation of Process Plant Lifecycle Functional Data Exchange*, conforming to STEP AP-221 ESPRIT-IV (Project#: 20506) Deliverable 2.1 Part 2: Logical & Physical Data Models (Task 2.1.2). ESPRIT-IV. ESPRIT-IV (Project#: 20506) Deliverable 2.1 Part 2. (1996).
- Sanvido, V.E., and K.J. Norton, "Integrated Design-Process Model," *Journal of Management in Engineering*, vol 10 (1994), pp 55-62.
- Simon, H.A., *Administrative Behavior* (The Free Press, New York, 1976).
- Steward, D.V., "The Design Structure System: A Method for Managing the Design of Complex Systems," *IEEE Transaction of Engineering Management*, vol 28 (1981), pp 71-74.
- Suh, N., *The Principles of Design*, ISBN 0-19-504345-6 (Oxford University Press, New York, 1990).
- Tiwari, S., and Gupta, "Constraint Management on Distributed Design Configurations," *Engineering with Computers* (1995), pp 199-210.
- van Griethuysen, J.J., *Concepts and Terminology for the Conceptual Schema and Information Base*, ISO TC97/SC5 N695 (International Standards Organization, 1982).
- Vet, P.E., and N.J. Mars, "Bottom-Up Construction of Ontologies," *IEEE Transaction on Knowledge and Data Engineering*, vol 10, No. 4 (1998), pp 513-526.
- Vincent, T.L., "Game Theory as a Design Tool," *Journal of Mechanisms, Transmissions, and Automation in Design*, vol 105 (1983), pp 165-170.
- Wasserman, S., *Social Network Analysis: Methods and Applications* (Cambridge University Press, New York, 1994).
- Wiest, J.D., and F.K. Levy, *A Management Guide to PERT/CPM* (Prentice-Hall, Englewood Cliffs, 1977).
- Winograd, T., and F. Flores, *Understanding Computers and Cognition* (Ablex, Norwood, NJ, 1986).
- Wong, S.T.C., "Coping with Conflict in Cooperative Knowledge-Based Systems," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol 21, No. 1 (1997), pp 57-71.
- Yoshikawa, H., "General Design Theory and a CAD System," *Man-Machine Communication in CAD/CAM* (1981).

Appendix A: List of Technical Publications

A series of technical papers have been published as a result of this research program.

- Baskin, A.B., R. Ganeshan, M. Case, S. C-Y. Lu, "Interest as a Non-Deterministic Specification of Process for An Agile Virtual Enterprise," *Conference on Agile and Intelligent Manufacturing Systems* (Troy, NY, October 1996).
- Baskin, A.B., S.C-Y. Lu, R. Ganeshan, M. Case, K. McGraw, J. Heckel, "Collaborative Workspaces: Sharing Emerging Product Models Across Disciplines, Space, and Time," *Concurrent Engineering 1997* (Troy, MI, October, 1997).
- Burkett, W., S., C-Y. Lu, "Integration of Perspective-Based Engineering Data on the Internet," Submitted to *Engineering with Computers* (2000).
- Cai, J., F. Udwadia, S.C-Y. Lu, W. Burkett, "An Evolutionary Platform for Co-Construction in Engineering Design Based on A Socio-Technical Framework," Working Paper (2000).
- Coimbatore, V., C-Y. Wang, R. Ganeshan, J. Heckel, A.B. Baskin, S. C.-Y. Lu, "Citywork: A Collaborative Maintenance Management Application," *CIRP 1997 International Design Seminar on Multimedia Technologies for Collaborative Design and Manufacturing* (Los Angeles, CA, 1997).
- Heckel, J., R. Ganeshan, M. Case, S. C-Y. Lu, A.B. Baski, "The Virtual Workspace System: An Enabling Technology for Collaborative Engineering Application," *Proceedings of the Workshop on Enabling Technologies for Collaborative Enterprise* (Boston, MA, 18-20 June 1997).
- Lu, Stephen C.-Y., J. Cai, "A Generic Collaborative Engineering Design Process Model for Conflict Management," Working Paper (2000).
- Lu, Stephen C.-Y., J. Cai, W. Burkett, F. Udwadia, "A Methodology for Collaborative Design Process and Conflict Analysis," *CIRP Annals*, vol 49, No. 1 (2000).
- Lu, Stephen C-Y., J. Cai, "STARS: A Socio-Technical Framework for Integrating Design Knowledge over the Internet," *IEEE Internet Computing*, vol 4, No. 5 (2000), pp 54-62.
- Lu, Stephen C-Y., Jian Cai, "Modeling Collaborative Design Process with a Socio-Technical Framework," *6th ISPE International Conference on Concurrent Engineering* (Bath, UK, 1999).
- Udwadia, F.S., S.C-Y. Lu, E. Griffith, M. Case, "A Socio-Technical Framework for Collaborative Engineering Design," To appear in *Journal of Engineering Design Research* (1999).

Appendix B: Facility Engineering Framework Workshop Report

Workshop Report: Facility Engineering Framework

Date: August 8-9, 2000

Place: The Construction Engineering Research Laboratory of the U.S. Army Engineering Research and Development Center

Participants:

Workshop Objective

The purpose of the workshop was to develop an in-depth understanding of the FEF theoretical underpinnings, discuss related work, and to generate a task list and schedule to make the FEF methodology generally available for use. In support of this objective, research from the University of Southern California was to be presented and its relationship to FEF project goals discussed. The applicability of the design process-modeling tool, which is under development at USC, to Building Composer and to the Digging Permit Process from LMS was investigated.

Workshop Structure:

The first day of the workshop was divided into a review session, a research presentation session, and a planning session. The review session started with a discussion of the general structure of the FEF and an example using CERL's Building Composer tool. This was followed by a summary of use-cases being generated for the Fort Hood permitting project under the LMS project. Some information was available about data warehousing work going on with the IFS system, and then future plans for the next generation of Knowledge Worker.

For the research presentation session, researchers from the University of Southern California will present elements of a *Socio-Technical* framework that has

been developer under AT23 contract over the past 3 years. Elements of this work, which have already been incorporated into FEF, were presented in some depth. Workshop participants were encouraged to visit the USC site before the workshop at <http://impact.usc.edu/cerl/>. A highlight of the presentation was demonstration of a prototype web-based tool, which is under development at USC, for managing conflict in collaborative engineering processes.

The Planning Session was be organized around moving FEF from an abstract idea to a methodology that others can to pick up and use. Questions revolved around issues such as the type of documentation that will be needed, example applications (two are currently planned), tools and utilities, and compliance certification. What kind of technology delivery mechanisms is needed? What other initiatives might use FEF as a foundation technology?

On the second day, which is optional for most attendees, researchers from the University of Southern California met with Building Composer and Land Management teams to map engineering processes into USC's prototype engineering process tool. This activity was scheduled for the morning and to map a facility design process and a permitting process.

Workshop Results – Presentations

The first day of the workshop consisted of presentation by members of CERL and USC on relevant technical projects. The speakers made the following presentations in the review session.

Michael Case

Dr. Case opened the meeting with introductions of the workshop participants. He also provides an introduction to the CERL FEF activities. The theme of his introduction was that FEF will not define a process, but rather a framework and set of tools and methodologies for promoting interoperability among Corps application systems. The need for theoretical support/underpinning for the collaboration technologies being developed under the FEF framework was emphasized, and the issue of how one might make FEF more usable across the Corps was also addressed.

Windell Ingram

Mr. Ingram, from the Corps Vicksburg Laboratory, is responsible for the Land Management Systems (LMS) project. The theme of his presentation was interoperable models and data sources. For the sharing GIS/land models and data

across systems without regard for platform limitations, the need for mechanisms and standards for interoperability with the end users in mind was emphasized. He is pursuing a “core services” project separate from LMS that can be used across the Corps and is very interested in development software product/methodologies that will actually be used and not end up as “shelfware.” Several of the ideas presented were similar to those being pursued by the USC research team.

Kirk McGraw

Mr. McGraw presented CERL’s work on building tools to work together, and use of Building Composer. This tool uses the Node-Tool (Process) engineering process model and generates the building design process using modules like the Criteria composers, the Layout composer, etc., that can be interactively used to generate the eventual building design.

Marilyn Ruiz

Dr. Ruiz presented work she is doing on managing the Digging Permit process at Fort Hood in Texas. In this LMS Case Study, there are many stakeholders in the use of Fort Hood land, and it is difficult and time consuming to reconcile the objectives of all these stakeholders when one has a mission that involves digging. The analysis aims at not only process modeling, but also integrating applications in an attempt to streamline the permitting process as well as developing suitable data repositories.

Wayne Schmidt

Mr. Schmidt presented and demonstrated the Knowledge Worker System (KWS). KWS captures and manages “institutional memory” of how processes are done, and enable knowledge workers to assign and manage tasks in an integrated and real-time environment. Synergies between the USC process model simulator and KWS were identified in the discussions.

Chuck Schroeder

Mr. Schroeder presented data warehousing activities in the Army. Forty systems are being integrated, with 55 interfaces in place and 16 more planned.

Francois Grobler

Mr. Grobler presented an overview of CERL's standards activities, in particular the activities of the IAI. He introduced the various standard organizations and the ongoing standards-related activities of SDS, OGC, SQL99, IAI and ArcInfo 8.

Stephen Lu

Dr. Lu introduced the IMPACT laboratory at USC. He highlighted the recent and current research activities at the lab, and described the long association between CERL and members of the lab.

Firdaus Udwadia

Dr. Udwadia introduced and presented the Socio-Technical Framework (STF) for Collaborative Engineering Processes and provided a detailed introduction to the research being done at USC. Collaboration technologies under STF that are being developed at USC were shown to be central to: interoperability of data and models, design automation, and development of virtual design teams. Further detailed information may be found at <http://impact.usc.edu/cerl/>.

William Burkett

Mr. Burkett presented his research on data mapping and integrated information architectures to support application interoperability in collaborative engineering processes.

James Cai

With the aid of conference call and NetMeeting technology, Mr. Cai presented his research on collaborative engineering process construction and conflict management. The process representation tool that is being developed at USC and that uses a Petri-net type model was demonstrated on a Case study developed jointly by CERL, CMU and USC. The real-time simulation capability of the tool generated considerable enthusiasm among the workshop participants.

Following the presentations, Mike Case articulated four topics for discussion:

1. Description of FEF Framework to map out issues that need to be done
2. Task and Process Definition: does XML adequately do representation of Task and Process?

3. Specification of data models—intermediate representations
4. Development of a Framework statement, assuming unlimited resources

Workshop Results – Discussions

1. 1. The following milestones were identified by the workshop team.
 - a. Framework description (ERDC)
 - b. A Proposal for process model specification (draft) – Nov (USC)
 - (1) Metadata
 - (2) Model nodes
 - (3) Tasks
 - (4) Stakeholders
 - c. Examples
 - (1) Building composer (McGraw)
 - (2) Permit project (at Fort Hood)
 - d. Tool – USC
 - e. Theoretical Underpinnings → Methodology Development → Implementation Platform, for Collaborative Engineering Processes with end user in mind
2. Mapping of data sets needs to be done with the end users in mind. Such mappings may be different for model developers, expert users of models, and decisionmakers. Here it was felt beneficial for LMS, FEF and the USC Research Team to work jointly.
3. Language for model Nodes and Tasks specification needs to be developed. The USC team was to begin taking a look at this important issue.
4. Decision Support for Engineering Processes is important and is an area where USC can contribute. FEF has already begun this with its emphasis on collaborative engineering processes; LMS is in the process of initiating and coordinating an effort in decision support.
5. The USC process modeling tool has the capability for real time simulation. On the second day of the workshop, several processes were modeled “on the fly” and the tool pointed out bottlenecks and problems that might be encountered during the execution of the processes. A one-year USC effort to develop the process tool further (especially its real-time simulation capabilities) was decided upon during discussions following its demonstration.

6. An amalgamation of the simulation capability of the USC process tool (under development) with KWS was discussed; a natural fit appeared to be present.
7. The Workshop participants made tentative plans to hold another workshop in November.

CERL Distribution

Chief of Engineers

ATTN: CEHEC-IM-LH (2)

Engineer Research and Development Center (Libraries)

ATTN: ERDC, Vicksburg, MS

ATTN: Cold Regions Research, Hanover, NH

ATTN: Topographic Engineering Center, Alexandria, VA

Defense Tech Info Center 22304

ATTN: DTIC-O

9

6/00

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 01-2002		2. REPORT TYPE Final		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Conflict Management in Collaborative Engineering Design: Basic Research in Fundamental Theory, Modeling Framework, and Computer Support for Collaborative Engineering Activities				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Stephen C-Y. Lu, Firdaus Udwadia, James Cai, William Burkett, Michael P. Case, Blessing Adeoye, and Van J. Woods				5d. PROJECT NUMBER 611102AT23	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center (ERDC) Construction Engineering Research Laboratory (CERL) PO Box 9005 Champaign, IL 61826-9005				8. PERFORMING ORGANIZATION REPORT NUMBER ERDC/CERL TR-02-2	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center (ERDC) Construction Engineering Research Laboratory (CERL) P.O. Box 9005 Champaign, IL 61826-9005				10. SPONSOR/MONITOR'S ACRONYM(S) CV-ZT	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.					
14. ABSTRACT All real-world engineering tasks involve collaborative activities among a group of human participants. The ability to understand, support, and improve collaboration is a critical factor in determining the overall cost, time, and effectiveness of modern engineering activities. Collaborative engineering tools are being introduced into the market at a rate so high that it is difficult to infuse technology in a reasoned and effective manner. Practitioners must decide which tools to adopt and to develop new and more effective processes. These decisions are made even more difficult by the fact that no body of theory exists that has been shown to describe the interaction between complex object-oriented data models, engineering processes, and human decisionmaking. The objective of this work was to develop the <i>Theory for Collaboration</i> in support of complex engineering system decisions in a highly distributed and heterogeneous environment. The results of this research will lead to a sound theoretical foundation that may be used to analytically and mathematically model, simulate, manage, and optimize collaborative engineering activities. Such a theory of collaboration will enable researchers to design, predict, and control various collaborative activities, systems, and environments, and to implement practical IT systems to support these important human endeavors.					
15. SUBJECT TERMS collaborative engineering computer aided design CAD engineering design conflict management computer aided engineering CAE					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			Michael P. Case
			SAR	176	19b. TELEPHONE NUMBER (include area code) (217) 373-7259