



**Development of An Adaptable Model for Distributed and Collaborative
Computer Aided Design**

CHEN XIAOLONG

(B. Eng)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgements

First and foremost I would like to express sincere thanks and appreciation to my supervisor, Associate Professor Wong Yoke San, for guidance, and his involvement in this research, for the technical discussions and particularly for his support throughout the course of my Master degree studies. I would not have finished this thesis without Prof. Wong's support and drive.

I would like to express my deep gratitude to Associate Professor Jerry Fuh, for the understanding, moral support and research guidance. Dr. Fuh gave me great spiritual support and encouraged me throughout this difficult but exciting journey.

Thanks to my colleague and one of the best friends, Dr Qui Zhiming, for his contribution to the proposals of this thesis and for the discussion and suggestions from Dr. Qui for the research project. I am also very grateful to fellow colleagues in LCEL for encouraging and discussion during my study: Wang Xingang, Fathianathan Mervyn, Wu Yifeng, Dr. Tang Yaxin and Lu Cong.

I would also like to acknowledge financial support provided by the Singapore Institute of Manufacturing Technologies (SIMTech), and suggestions from Dr Lu YiQiang and Dr Li Weidong, for their guidance thorough the research work, and thank to Laboratory for Concurrent Engineering and Logistics (LCEL) of Faculty of Engineering for providing the facility to complete my research.

Summary

The primary objective of the research is to develop an adaptable module for Internet-enabled Distributed and Collaborative Computer-Aided Design (DCCAD). This module is designed in a platform-independent model that enables designers to communicate and collaborate with each other across different operating systems and applications in dispersed locations. The adaptable module abstracts basic collaborative functionalities for team-design project and incorporates with web-based project management packages.

The main issues for the research are platform-independent adaptable structure, distributed design management and co-ordination, design process representation, collaborative design communication and design visualization. The system was designed with application-independent technologies. The collaborative design module utilizes extensible and adaptable programming mechanism. This enables the collaborative design plug-in and add-on be developed flexibly and to be loaded to different applications in different operating systems. The design and process representation is composed of three parts: design representation, design behavior representation and design process representation. They are described in XML (eXtensible Markup Language) and XML models are represented using Unified Modeling Language (UML) notation. Process representation and modeling were implemented using the Object-Oriented-Programming model. Design processes are well defined with elements of process such as task, dependency, event and node. Collaborative design protocols are defined for design communication. Different cases and types of communication and data exchange model

are proposed in the adaptable module. This enables users to apply different models of communication with different objectives whether peer-to-peer or client-server.

This is one of the earliest adaptable architecture implemented using the platform-independent framework, which is designed especially for Internet-based collaborative design. The developed adaptable module has been tested and demonstrated with case study in a distributed collaborative design environment.

Table of the Content

<i>Acknowledgements</i>	<i>1</i>
<i>Summary</i>	<i>2</i>
<i>Table of the Content</i>	<i>4</i>
<i>Table of Figures</i>	<i>6</i>
<i>Chapter 1 Introduction</i>	<i>8</i>
1.1 Computer Aided Design	9
1.2 Collaborative Design	10
1.3 IT supports for Distributed Co-CAD	11
1.4 Organization of the Thesis	12
<i>Chapter 2 Background and Literature Review</i>	<i>14</i>
2.1 Introduction.....	14
2.2 Collaborative Computer Aided Design Fundamentals	17
2.3 Related Research Works	20
2.3.1 Collaborative design frameworks	20
2.3.2 Other Aspects of Collaborative design	27
2.4 Discussion	32
2.5 Research Objectives and Scopes.....	34
<i>Chapter 3 System Analysis and Structure Design</i>	<i>36</i>
3.1 Distributed Collaborative Design Architecture	36
3.2 DCCAD Kernel Functional Structure	38
3.3 Data Structure Design	40
3.4 Collaborative Design Communication Protocols.....	42
3.5 Design Management Model.....	44
3.6 Summary	47
<i>Chapter 4 Collaborative Design Representation</i>	<i>48</i>
4.1 Introduction.....	49
4.2 XML and UML	50
4.3 Design Representation using XML.....	52
4.3.1 Design Part Representation.....	53
4.3.2 Collaborative Design Behavior Representation.....	55
4.3.3 Design Process Representation	57
4.4 Summary	60
<i>Chapter 5 Implementations and Case Study</i>	<i>61</i>
5.1 Introduction.....	61
5.2 Adaptive Collaborative Module Fundamental Implementations	63
5.2.1 Implementation Structure and Co-Design Process	63
5.2.2 Collaborative Central Database Design	64
5.2.3 Abstracted Collaborative Function Design	66
5.3 Application Adaptive Customization.....	68
5.3.1 AutoCAD Plug-in Implementation	69
5.3.2 MS Excel Plug-in Implementation.....	70
5.4 Distributed Collaborative Design Demonstration.....	72
5.4 Summary	78
<i>Chapter 6. Conclusions and Future Works</i>	<i>79</i>

6.1 Contributions of the research	79
6.2 Issues arising from the research.....	80
6.3 Summary of the research	82
6.4 Future directions for the research	83
References	84

Table of Figures

Figure 2.1 Johansen 2D Matrix.....	15
Figure 2.2 T-L-O 3D Space	16
Figure 2.3 Topography of CAD functions.....	18
Figure 2.4 Data View of Co-CAD Fundamentals.....	19
Figure 2.5 Web-based CS Structure.....	21
Figure 2.6 Client-Server Message Structure.....	22
Figure 3.1 Adaptable DCCAD Module Architecture	37
Figure 3.2 Message Mapping Mechanism	38
Figure 3.3 DCCAD Kernel Functional Structure	39
Figure 3.4 Design Behavior UML Denotation	42
Figure 3.5 Application Message Mapping Sample.....	43
Figure 3.6 Design Management Model	45
Figure 3.7 Design Management Solution	46
Figure 4.1 Unified modeling language notation	51
Figure 4.2 Design Part representation in UML notation.....	53
Figure 4.3 Sample of Work-piece XML Representation	54
Figure 4.4 Design Behavior Representation in UML Notation	0
Figure 4.5 Sample of Design Behavior XML Representation	56
Figure 4.6 Design Process Representation in UML notation	57
Figure 4.7 Example of Design Process Modeling UI.....	58
Figure 4.8 Properties of Task2 in Figure 7 Example	58
Figure 4.9 Sample of Design Process XML Representation	60
Figure 5.1 Processes of The Work.....	62
Figure 5.2 Implementation Structure	63
Figure 5.3 Implementation Scenario.....	64
Figure 5.4 Data Structure in UML Notation.....	65
Figure 5.5 DCCAD Kernel Structure.....	66
Figure 5.6 Web-based Project Planner Module Snap-shot	67
Figure 5.7 Add-in Implementation Pseudo Codes.....	68

Figure 5.8 AutoCAD Plug-in Structure	69
Figure 5.9 Snap-shot of AutoCAD Plug-in Menu UI.....	70
Figure 5.10 Structure of MS Excel Plug-in	71
Figure 5.11 Snap-shot of MS Excel Plug-in Menu UI.....	72
Figure 5.12 Snapshot of AutoCAD Plug-in Main UI	74
Figure 5.13 Snapshots of MS Excel Customization UI	75
Figure 5.14 Snapshots of MS Excel Plug-in Rule Definition UI.....	76
Figure 5.15 Snapshots of Web-based Project Planning UI.....	77

Chapter 1 Introduction

The global economy is pushing manufacturers to recast their product development processes and gather more inputs earlier in the design cycle. The aim is to involve a broader team of people, including customers and suppliers, experts in different disciplines so as to decrease product development costs and increase quality and efficiency of the products. Effective collaboration can result in more innovative product designs, faster time-to-market and lower product costs that give manufacturers a competitive advantage.

In modern manufacturing, concurrent engineering and computer integrated manufacturing (CIM) are becoming accepted as essential management techniques for minimizing lead time [Nee, *et. al.* 1995]. Computer-aided process planning (CAPP) used within the framework of concurrent engineering is the interface between design and manufacture. Collaborative design is an essential part of CIM. CIM is not possible without collaboration and no CIM system is complete without computer aided collaboration functionalities.

Lots of works have been done to exploit collaborative design solutions and mythologies. This thesis presents the development of an adaptable Distributed Collaborative Computer Aided Design (DCCAD) module. In this chapter, Section 1.1 introduces the background of CAD. Section 1.2 briefly presents research topic of distributed collaborative computer aided design. Section 1.3 discusses IT trends that may support developments of collaborative design systems. The first three sections provide basic background and

motivation of this thesis. The final section gives an organization overview of the rest of the thesis.

1.1 Computer Aided Design

Thirty years ago, nearly every drawing was done in pencil or ink on paper. Minor changes require erasing and redrawing while major changes often need recreating the drawing from the scratch. If a change to one drawing affects other documents, you are dependent upon someone to manually make the changes to the other drawings. CAD has fundamentally changed the way design is done.

The first graphic system was developed in mid 1950 for the US Air Force's SAGE air defense system. Dr. Patrick J. Hanratty known as "the Father of CAD/CAM" for his pioneering contributions to the field of computer-aided design and manufacturing. Developed in 1957 PRONTO, is the first commercial numerical-control programming system. While those famous manufacturers pioneered the development and utilization of CAD technologies, such as General Motors, McDonnell Douglas, Boeing, General Electric etc. Since 1981 numerous commercial programs showed up in the manufacturing environment, and provided CAD technologies. In order to win business, companies used CAD to produce better designs more quickly and cheaper than their competitors. Productivity is much improved by a CAD system that enables users to easily draw complex geometry. The speed is increased by the use of automatic fillets and chamfers. The computer will automatically “snap” particular geometric points and features. Copy, rotate and mirror facilities are also very handy when drawing symmetric parts.

1.2 Collaborative Design

Modern design process typically involves more than one person. In many cases there will be a team of designers participating in the design, along with the client. A design review draws in a possibly large number of people to inspect and discuss the result. The communication and cooperation of these people can be supported directly if allow for multiple users involving in a design process is allowed. Users have different goals, depending on their roles in the process. Designers want to modify objects in a design space, simultaneously if possible. Participants in the design review want to view and browse the design independently. Part of the communication between these users is expressed in the terms of visualization and modification in real time. Each member in the team is responsible not only for his or her primary function, but also for being an active member of the communication network, i.e. collecting, filtering, and passing along information. Human oversight and error can cause failure in the system, which is why a much research is focused on capturing and organizing information.

Collaboration is the key to increase competitiveness and innovation because it enables organizations to take advantage of core competencies as well as combine skills in financially arrangements to accomplish a better product design. According to Business Week, Yankee Group Research Inc. estimates that over the next five years, collaborating over the Internet can save companies \$223 billion by cutting transaction, production, and inventory costs [ARENA 2002].

The need for more sophisticated collaboration tools is compelling enough that many solutions have sprung up to try to fill it. But many of these solutions attempt to adapt

existing product data management platforms to a collaborative environment, an undertaking that involves enormous re-engineering to create access to an inherently closed system.

1.3 IT supports for Distributed Co-CAD

During its first forty years of existence, informatics has moved from heavy centralization to distributed systems. Initially characterized by the multiplication of isolated smaller machines, micro-informatics was soon to be transformed by the Internet; information could be exchanged between machines that function under different operating systems. The Internet is becoming increasingly more valuable in the field of computer-aided design that what we conventionally called CAD might soon be changed to IAD (Internet Aided Design) [Zhou and Krawczyk 2000], which are typical and most influential in providing Internet-related services for the CAD industry.

Web-based technologies provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. HTTP protocol is a one-direction, stateless lightweight protocol. However, it is easy to use and can traverse Internet freely. The HTTP and other protocols built on HTTP seem to dominate the web-based CAD collaboration system. A typical usage is GEOMNET [Barequet et al., 1999] The Peer-to-Peer technology is emerging. There are many advantages to use P2P structure, such as the identical structure, code reusable, high availability and flexible programming.

Another characteristic of modern software is that most engineering software applications provide user customization interfaces and Software Development Kit (SDK) which allow users access application kernel or a special file database. For example AutoCAD provides interfaces whereby users can access DWG database and message system via VBA, ObjectARX SDK. Microsoft Office provides interfaces for VBA customization.

As mentioned in this section, with the needs of distributed collaborative computer aided design and IT trends, this research is to prove the possibilities of developing an adaptable, application platform-independent module designed specially for distributed computer aided collaborative design.

1.4 Organization of the Thesis

This chapter discusses the background of the research and the underlying motivation of this research. The rest of the thesis is organized as follows.

Chapter 2 provides the principles of distributed collaborative computer aided design (DCCAD), gives an overview of collaborative design fundamentals including collaboration design mechanisms and main tasks of collaborative design, presents research issues in developing collaborative design applications, and introduces distributed collaborative design and the classification of its applications. The chapter 2 also discusses the related Co-CAD research work on mechanical engineering domain including design, visualization, project management and proposes the objective and scopes of this research.

In Chapter 3, a platform independent architecture for DCCAD is introduced. Different integration interfaces are presented. The adaptable architecture for integrating with

different applications is proposed. The DCCAD kernel has been designed for user customization. Design management and project management solutions are also provided. Design communication protocols are defined as part of the DCCAD kernel.

In Chapter 4, the information involved in the collaborative design is represented using XML, and the XML Schemas are expressed in UML notations. The content includes design parts, design behaviors and design processes. Examples of collaborative design information are also presented.

Chapter 5 introduces the implementations. Details of the implementation structure are given. The developed DCCAD adapts plug-ins, and they are demonstrated for AutoCAD and MS Excel. Collaborative design case study is then given. Design project management is presented with the relevant Graphical User Interface (GUI) design.

Chapter 6 Summarizes the development presented in this thesis, including contributions, issues arising from the development, conclusions drawn and future directions for the researches and development.

Chapter 2 Background and Literature Review

The purpose of this research is to develop an adaptable collaborative module to support Internet-enabled DCCAD. This adaptable module was designed for possible integrations with solid enterprise applications as a plug-in or add-on.

This chapter introduces the fundamentals of distributed Collaborative Computer Aided Design (Co-CAD). It provides an overview of the research status in the field of Co-CAD, research motivations and scopes of this thesis. Section 2.1 introduces concepts of distributed collaborative design. Section 2.2 discusses fundamentals of collaborative design, including system structures, functional basics. Section 2.3 introduces related research works and current status of commercial applications regarding collaborative design. Section 2.4 summarizes existing research focuses and discusses problems facing. Section 2.5 establishes the objectives and scopes of this research based on discussions in the former sections.

2.1 Introduction

When a product is designed through the collective and joint efforts from many designers, the design process may be called collaborative design. This would include dispersed functions for design, manufacturing, assembly, test, quality control and purchasing as well as interfaces for suppliers and customers [Sprow et al., 1992]. The objectives of such a collaborative design team might include optimizing the mechanical functions of the product, minimizing the production or assembly costs, or ensuring that the product can be

easily and economically serviced and maintained [Hartly 1992]. Since the collaborative design team often work in parallel and independently with different engineering tools distributed in geographical locations, even across various time zones around the world, the resulting design process may then be called distributed collaborative design.

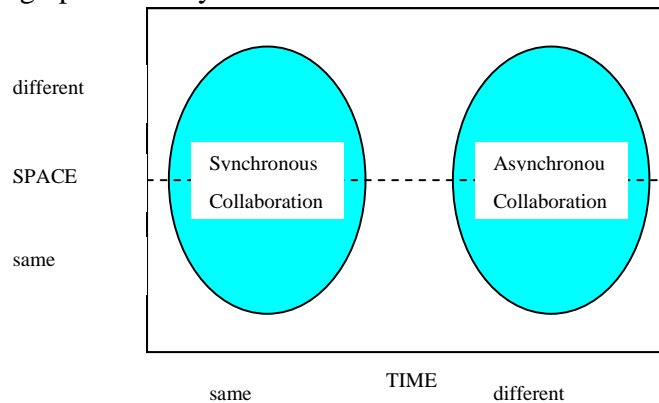


Figure 2.1 Johansen 2D Matrix

Johansen [Johansen 1988] used time-space 2D matrix to examine cooperative works. The matrix categorizes collaboration into synchronous and asynchronous patterns, as shown in Figure 2.1. This space-time matrix cannot fully represent the emerging collaboration trends. For example, collaboration may happen among different geographically dispersed companies, or within the same company but between two distributed divisions. Here we extend the matrix to a three-dimensional Time-Location-Organization space, defined as $\mathbf{O}(T,L,O)$ to describe when, where and who are collaborating.

Compared to the time-space matrix, which is a useful and concise reference to a particular circumstance, the proposed 3D time-location-organization matrix not only looks at whether participants are in the same place, but also whether they are operating at the same time or not. We also consider whether participants are in the same company or

group. In the collaboration domain, enterprises are usually concerned with the security of data flow; however, the collaboration systems' architecture and their patterns should also be taken into consideration. Very often, in the domain of one enterprise, the design platforms, such as operating systems, types of network, development tools, database systems, CAD software, etc. are usually heterogeneous, whilst the configurations and facilities are often different from company to company. This will result in different architectures and different system functionalities for collaborative design.

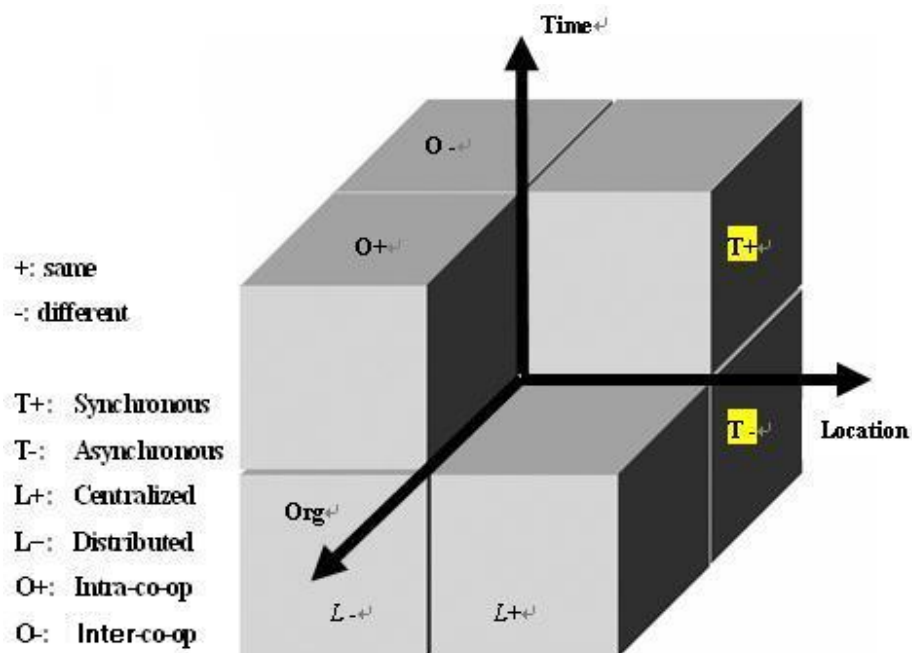


Figure 2.2 T-L-O 3D Space

As shown in Figure 2.2, based on the *time*- coordinate, we define synchronous and asynchronous collaboration. Considering the data location, modeling kernel and functionalities of collaborative design, the tasks can be centralized or distributed. For the participants of collaborative design, it can be intra-enterprise or Inter-enterprise. Due to

the different patterns of a collaboration model, it may result in different architectures and solutions for the realization of distributed collaborative design. In section 2.2 and 2.3, we will discuss these topics in detail.

2.2 Collaborative Computer Aided Design Fundamentals

Collaborative design issues are due to: different groups of people, often with different disciplines, of different enterprises, at different places, having to work together on one product. A single project often comprises of different physical parts and functional modules, i.e. electronic, mechanical, software, hardware and other subsidiary devices. Designing complex products, such as an aircraft or automobile, requires a tremendous collection of expertise, knowledge, technology and tools. Design resources are often distributed. Participants may be in different places. Integrated concurrent product development can be realized by leveraging modern information technology to coordinate people, processes, tools and technologies. Smaller engineering companies and medium-sized suppliers are looking for an inexpensive way for geographically dispersed teams to jointly develop products together over the Internet. Figure 2.2 illustrates traditional topography of CAD and CAE (Computer Aided Engineering) functions. But in the today's ICAD (Internet-based CAD) environment, those functions are very often dispersed.

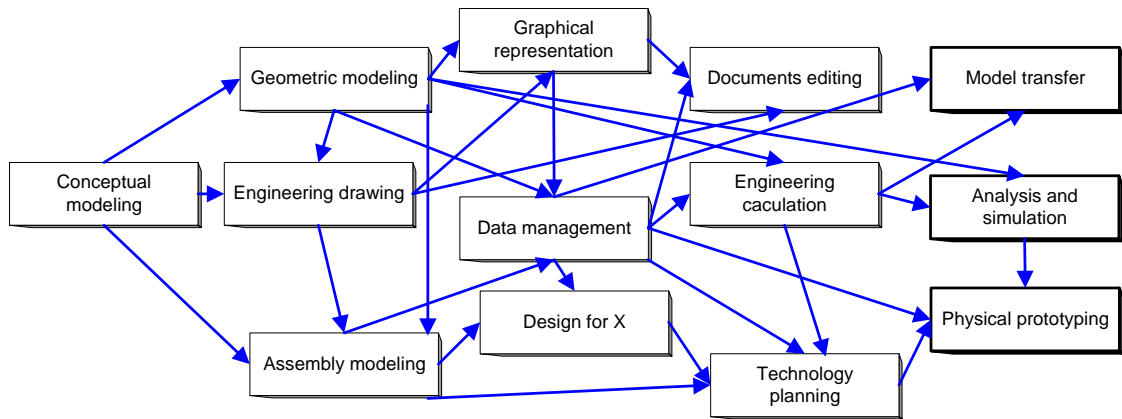


Figure 2.3 Topography of CAD functions

According to the functions and roles of users participating in the design activity, a collaborative CAD can be organized as either a horizontal or a hierarchical manner. The horizontal collaboration emphasizes on allocating a design team from the same discipline to carry out a complex design task simultaneously. The hierarchical collaboration can establish an effective communication channel between upstream design and downstream manufacturing, and it can enrich principles and methodologies of concurrent engineering to link diversified engineering tools dynamically [Li et al., 2004].

In modern product development lifecycle, those traditional CAD/CAE functions are distributed geographically cross-different application platforms. From the view of data layer, some fundamental issues need to be resolved to realize distributed collaborative design projects, as shown in Figure 2.3.

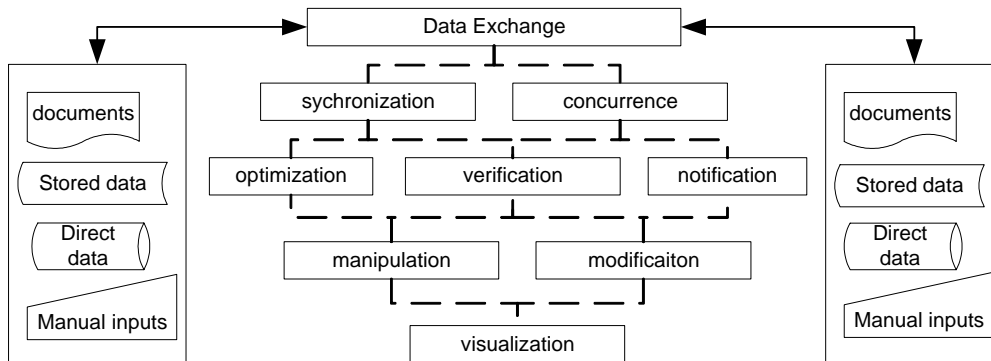


Figure 2.4 Data View of Co-CAD Fundamentals

Using modern information technologies to realize distributed computer aided collaborative design, fundamental requirements can be categorized to:

- 1). How to exchange data of different formats, in different domain definitions, from different locations via different means of communication protocols.
- 2). How to define and capture behaviours of different people of different disciplines, and organize those user interactions to notify each other. Thus making individual design behaviours in accordance with overall project process.
- 3). How to develop efficient communication protocols and methods, which make the content of design data and design behaviours understandable to each other.
- 4). How to coordinate geographically dispersed groups of participants, and the decomposed projects and tasks. Thus making the project process visualized and represented explicitly.

To solve these problems, many researchers, as well as commercial software vendors have done lots of works using different methodologies, focusing on different issues and using

different tools. In the next section, we will discuss related work carried out regarding these key issues for distributed collaborative design.

2.3 Related Research Works

Many research works have been carried out to realize collaborative design. Some researchers proposed innovative system architectures which include geometric modeling, design co-ordination, project management and design optimization modules. Others focused on some key issues of distributed collaborative design, based on current application framework, such as visualization, co-ordination, assembly, design representation and so on. This section discusses the frameworks proposed by researchers first, and then discusses other topics on collaboration functional methodologies. At the end of this section, commercial systems and their characteristics are discussed.

2.3.1 Collaborative design frameworks

There are three types of structure in collaborative design systems developed: client-server systems, agent-based systems and peer-to-peer systems. Each has its advantages and disadvantages. And different structures behave differently with different focuses.

- **Client-Server Structure**

Most proposed frameworks use the client-server (C/S) architecture to realize collaborative design systems. The Web-based system is a typical implementation of C/S structure. The Web is used by team members as a medium to share data, information and knowledge [Toye et al., 1994; Cutkosky 1993], and in some cases for product data

management and project management by integrating the Web with appropriate technologies [Numata 1996]. In some cases, the Web may only be used to monitor the design process and check the status of the working system [Shen and Barthes, 1996]. A number of the frameworks have been proposed for Web-based collaborative design systems [Numata 1996; Pahng et al., 1998; Huang et al., 1999; Tay and Roy, 2003], but most of them are still under proof-of-the-concept prototype development stage.

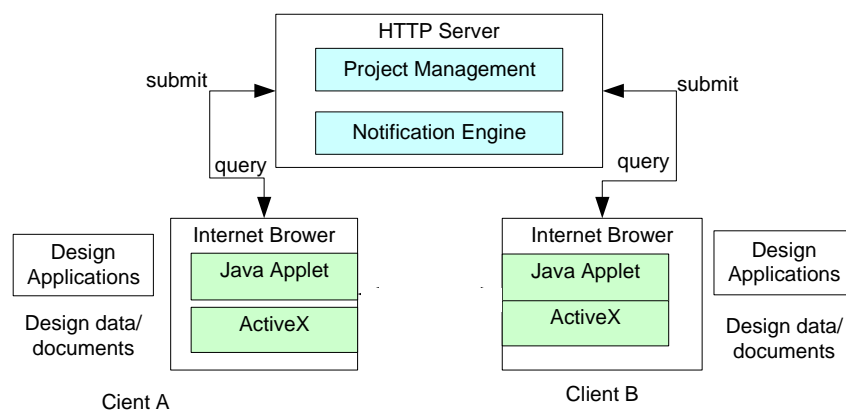


Figure 2.5 Web-based CS Structure

When Java Applets or ActiveX, are used for developing the client-side user interfaces, clients are getting stronger functions to communicate with the server side as well as other clients, as shown in figure 2.4. Because when clients query a web page that contains Java Applets or ActiveX object, the web browser will download those necessary components to the local machine first. From this point of view, web is only an access point of the system. WebEx™'s online meeting system was implemented in such structure [9]. And the cosmos's VRML viewer is also published as ActiveX component.

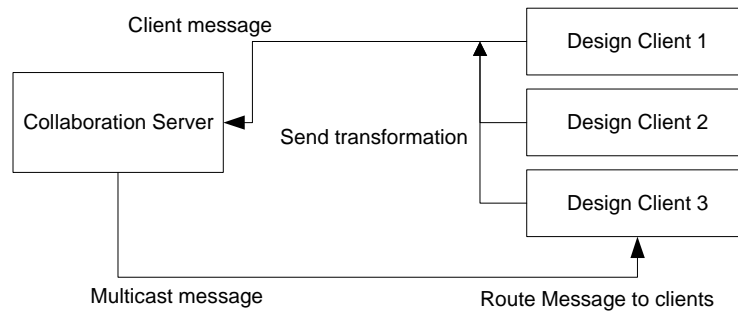


Figure 2.6 Client-Server Message Structure

Analyzing client-end and server-side functionalities, the C/S architectures for the developed collaborative CAD systems can be classified into two types: THIN Server + STRONG Client, and STRONG Server + Thin client. In the THIN Server + STRONG Client architecture type, clients are equipped with whole CAD functions and some communication facilitators. The server plays as an information exchanger to publish design information or broadcast notification messages from a client to other clients during a collaborative design process, e.g. CollabCAD™[1], IX Design™, Nam and Wright [1998], Pahng et al. [1998], Tay and Roy [2003], Bianconi and Conti [2003], MUG, etc. In the second architecture, the data structures in clients are lightweight and they primarily support visualization and some manipulation functions, such as selection, transformation, and changing of visualization properties of displayed parts, etc. The main modeling activities are carried out in a common workspace located in the server side. A thin/strong representation in the C/S respectively has been proposed to enhance the performance of the system effectively [Li, et al. 2002, 2004]. The developed systems include Alibre Design™[2], OneSpace™[3], [Van de Berg, et al., 2000], [Li et al., 2002, 2004], etc.

- **Peer-to-Peer and Grid Computing**

Peer-to-peer based architecture allows for decentralized application design, moving from centralized server models to a distributed model where each peer, independent of software and hardware platforms, can benefit from being connected to a large number of other peers. In such architectures, clients and servers have a horizontal relationship rather than the traditional vertical relationship, giving the whole peer group tremendous processing power and storage space.

The main advantage with peer-to-peer is that the system does not have a central coordinating node that can halt the whole system if it goes down or loses contact with some or all of its clients. And peer-to-peer is relatively inexpensive and fairly simple to set up and manage. The disadvantage is that it is limited in extensibility, tends to overburden user workstations by having them to play the role of server to other users, is largely unsecured, and is typically unable to provide system-wide services since the typical workstation will run a standard desktop operating system incapable of hosting any major service. The developed architecture, such as [Begole et al., 1997] and Inventor collaborative tool™, supports the sharing and manipulation of services or modules of a system by other systems. David proposed a P2P-based middleware to support content management [David et al., 2003]. Groove Networks is a P2P collaborative software platform that traverses corporate firewalls and allows secure digital collaboration. TOMSCOP is a synchronous P2P collaboration platform over JXTA [Tomomi 2004]. And IBM offers grid computing for automotive design.

It is worth mentioning that P2P computing is still in an evolving stage and much works need to be done to overcome the complex issues such as security, network bandwidth,

and architecture designs. It is a model for programming distributed systems that are characterized by an increasing decentralization, autonomy and anonymity of its peer components, where they can play the roles of both client and server, thus requiring new approaches and patterns for distributed computing.

- **Agent-based Structure**

In the client/server architecture, in order to support collaboration, the server needs to transmit the structure of the design representation so that users can pose queries about formal design concepts. To facilitate a viable design environment, it must also engage users in a dialog-like interaction that encompasses a range of activities, such as geometric and semantic product modeling, design representation, user-interaction and design browsing and retrieval. However, in order to collaborate on a distributed project, remote engineers and designers need to coordinate their efforts. This co-ordination involves translation of terminology among disciplines, locating/providing generic analysis services, prototyping services, and product management. To the degree the server is no mere repository of information, but engage users in an active dialogue with each other while providing such remote services in order to solve design problems, it may be called an agent. Agent technology may provide supports to enhance the performance of collaborative design systems [Wang et al., 2002].

Malone [Malone et al., 1997] proposes two design principles for agent-based systems, through the experience of developing “intelligent agents” such as Information Lens and Oval [Malone et al., 1995] There is a growing body of work on agent-based co-ordination techniques for engineering design. In particular, agent-based collaborative design has

been proposed as a promising approach for distributed engineering teams. PACT [Cutkosky et al., 1993] is an experimental infrastructure for concurrent engineering. The architecture is based on interacting agents. A more structured approach to agent coordination is provided by a system called Redux [Petrie 1991], which is based on a TMS (Truth Maintenance System) [Doyle 1979] with dependency-directed backtracking. The Redux ontology characterizes design in terms of a hierarchy of decisions, goals and assignments of design variables, and the rationale that supports decisions. If two decisions are found to contradict each other, the model can be used to propagate appropriate state changes to affected models. Redux is applied to an engineering framework for distributed cable harness design [Petrie et al., 1995]. The tracking of Pareto optimality is used to satisfy multiple objectives among distributed agents

Note that, different collaborative functions need different communication modes. The client-server structure is a reasonable and applicable structure to implement a distributed collaborative design system. With co-design extended systems, for manufacturing services, CSCW concept level integrations, the agent-based structure has more potential value. The peer-to-peer structure is limited in simple functionalities, such as application sharing, communication forum. Table 2.1 summarizes of those collaborative design frameworks.

Table 2.1 Summary of collaborative design frameworks

Project Name	Key Features	Structure	Tools
CollabCAD @ NIT[1]	P2P simultaneously design Share 2D and3D models	P2P/CS	OpenCASCADE Java3D
Co-DCAD @SIMTech	Feature-based DCCAD Remote simulation Thread duplicate enabled	CS/Web	OpenCASCADE Java3D
DOME @ KIST, Korea	DCOM, Multi-server	CS/Web	CORBA Java WEB
WWDDL @ KMI, Open Univ.	Distributed CBR using agents Tadzebao metphor	CS/Web Agent	WEB Java, Lisp CBR
CyberCAD @PT&NUS, Singapore	Model based Point-to-point synchronous design Web viewer	CS/Web	Java VRML
WebSPIFF	Feature-based modeler Synchronous design GIF and XML-based visualization	CS/Web	XML
MUG @Drexel Univ.	NUBRS enabled modeling Supporting shared audio and shared 3D manipulation	CS Agent P2P	JAVA DAML JMF
A-Design @CMU	Two tier representation Multi-objective optimization	Agent	Internet Lisp
RAPPID @ERIM	Marketplace approach Set-based design	Agent	Internet
PACT @Stanford	Federation architecture Wrapper for legacy integration	Agent	KQML, LISP
CADDAC @Purdue	Sever-based geometric modeling Central constraint solver. Model/View/Controller pattern	CS	STEP ACIS
*Pro/E WildFire @PTC[6]	Design repository Web-based design management Email notification	CS/WEB	WEB EMAIL
*OneSpace @Co Create[5]	Design repository Web-based design management Remote Control	CS/Web P2P	JAVA ACIS
*AlibreDesign[2]	Session duplicate Design repository	P2P/CS	Session share
*AutoCAD 2004 @Autodesk[4]	ETransmit Meet now Volo View i-drop, publish to web	CS/Web P2P	Streaming

*Commercial systems

2.3.2 Other Aspects of Collaborative design

Within different structures of distributed collaborative design systems, much work has also been done concerning various aspects of collaborative design, such as distributed geometric modeling solutions, collaborative design management and co-ordination, design visualization etc.

▪ Distributed Geometric Modeling

A method of geometric modeling is determined by its system structure, especially by whether the geometric modeling function is centralized or distributed. Qiang [Qiang et al., 2001] developed a method of using macro files instead of transmitting native CAD files to co-designers, aiming at reducing the data transmission volume. Based on the Java language and TCP/IP protocol, designers across the network can realize co-modification and visualization in geographically dispersed and disparate parties, across heterogeneous computing platforms and environments. Ni [Ni et al., 2002] presents a collaborative engine to facilitate collaborations, using an XML-based information representation where COMMAND and Parameters are formatted to streamline the information transmission within the distributed environment. Other researchers proposed using modeling services to achieve modeling requirements for collaborative design. But most of them are in still in prototype stage, because it is hard to define behaviors in designing complex products, and almost impossible to describe such modeling requests for a domain-independent product design. Case-based modeling systems are proposed by [Mervyn F. et al., 2003] to realize fixture design and agent-based assembly modeling system is proposed by [Toshiki et al., 1998].

- **Collaborative Design Management**

Managing and coordinating complex project data, project process and design behaviors are undertaken by groups of designers and other participants. Mechanisms must be provided to manage the design data, design decision, design collisions and design revision. Lu [Lu et al., 2000] used a Socio-Technical framework to manage design conflicts, and presumed that perspective views of the technical and social stakeholders are different. Wallis [Wallis et al., 1998] used a *deontic logic*-based formalism to facilitate distributed conflict management. Gilles [Gilles et al., 2000] used a ConceptIBIS model as a conflict solution in designing a terminological knowledge base. Design process management and co-ordination include project modeling and task allocation. The creation of a complex project sometimes should consider the distribution of available resources, such as design facilities, human capability, and design knowledge [Rafael et al., 2002, Denise et al., 1987]. Task and process management is not only a matter of data management. Constraint-based management considers design dependencies to coordinate each design element. There are several projects that use assembly constraint or design variable as a basic factor in the management of the collaborative design project [Lee et al., 1999; Robert et al., 2002; Chen et al., 2004; Kim 2003; Franca et al., 2002].

- **Design Visualization**

Visualization refers to a geometric model rendering as well as design behavior and design process representation [Olsen et al., 1994]. To visualize a CAD content is one of the most crucial tasks in a distributed design environment. Although current networks perform sufficiently fast data transmission rate for textual correspondence, a native CAD file

sometimes can be very large, and in the process of design, the interactions are expected to be continuous. Thus the existing network transmission rate is still far below the concurrent design requirement. Secondly, platforms and CAD software are often heterogeneous among the participants of a collaborative design project. People have to use data exchange standards such as IGES, STEP, and STL to convert data, which will result in a dramatic increase in file size.

Although using VRML format can reduce the file size to some extent when compared with the native CAD file, the waiting time of downloading a VRML file is still unacceptable and it is sometime very time-consuming. The HOOPS Stream Toolkit enables applications to stream highly compressed files containing 2D, 3D and custom data over Internet [7]. The VizStream® Platform achieves dynamically prioritized streaming by parsing the streaming order based on the user actions and communication [8].

However, streaming technologies still keep users at essentially the visualization level. Co-designers and customers sometimes require more information, such as features, than mere visualization. Aiming to solve the drawbacks of such visualization, Li [Li et al., 2003] used a B-Rep-based data transmission solution for distributed CAD. The B-Rep model is still quite concise compared to the faceted model and can precisely describe the geometry of a model. Most available CAD modeling software supports the extraction of B-Rep-based entities. Song [Song 2002] extended this method to feature-based transmission and visualization. Some related research areas such as geometric feature

extraction; geometric compression, feature recognition and representation also have been carried out by many researchers [Han et al., 1998].

Product development by human designers or intelligent agents strongly relies on information and knowledge management. Collaborative design environments are supposed to achieve the best utilization of all kinds of knowledge that are available [Rodgers et al., 1999]. The design process for complex products is a series of tasks where information is continuously being transformed. Knowledge representation is necessary for a variety of tasks involved during the design lifecycle, especially when a project involves with engineers from different disciplines at different locations. Clearly captured and represented knowledge and information will greatly help designers to understand the ongoing design process and make corresponding decisions efficiently and effectively [Gilles et al., 2002; Christopher et al., 2002]. Fleming [Fleming et al., 2000] developed a process mapping methodology for the process protocol level. The methodology enables all of the information relating to the sub processes to be represented as a series of process maps and when viewed holistically, presents an integrated generic decomposition of the processes on the high level map. Eggersmann [Eggersmann et al., 2003] introduced a model based on the identification of three types of design activities: synthesis, analysis, and decision, to design a revamped design process. Gorti [Gorti et al., 1997] developed an object oriented model for knowledge representation.

Besides those research aspects mentioned above, there also lots work have been done related to distributed collaborative design topics, such as collaborative design optimization [I. Budianto et al., 2000], [Hirokazu et al., 2003], concurrent process planning [Lee et al., 1998], agent-based workflow management [Huang et al., 2000],

Table 2.2 Summary of Research Focus Areas

R&D	Key Features	Focuses
Wallis et al 1998	<i>deontic logic</i> -based formalism	Design co-ordination
Lu et al 2000	Socio-Technical framework Conflicts management	Conflicts management
Gilles 2000	ConceptIBIS model Terminological knowledge base	Design co-ordination
Nicola et al 2000	Genetic algorithm (GA)-based search	Concurrent assembly planning,
N. Shyamsundar et al 2001	Geometric representation called the AREP Prototypical system (cPAD)	Feature based modeling
Mervyn et al 2004	Case based modeling for fixture design Java3D based modeling GA optimization	Case based modeling
Christopher et al 2002	DARPA Agent Markup Language Knowledge-based description	Knowledge Representation
Li et al 2003	B-Rep-based data transmission	CAD visualization
Song 2002	Feature-based transmission and visualization	CAD visualization
Scherer 2000	Information Logistics for Supporting the Collaborative Design Process	Design Process
Zdenek et al 2000	An evolutionary publication model Knowledge model of engineering design learning: domain, process, media	Design knowledge sharing
Lee et al, 1998	Current mold manufacturing process planning Knowledge based methodology IDEF0 structure analysis	Process planning
Huang et al 2000	Agent-based	Workflow management
ARENA	PLM integrated Versioning	Design management
SmartTeam @CATIA.IBM	PDM integrated Programmable	Design management
Vizstream @RealityWave	C/S Data Streaming View, mark-up, message.	CAD-model Visualization
Hoops Streaming Toolkit™	Provide 3D streaming APIs. Advanced compression, attribute support, object prioritization, etc.	CAD-model Visualization

geometric shape abstraction for internet-based virtual prototyping [Rajit and Ratnakar, 1998] and design knowledge sharing [Nitin et al., 2002], [Zdenek et al., 2000] etc.

2.4 Discussion

In the literatures reviewed, there are different frameworks to realize certain collaborative design functionalities. Some researchers proposed new system architectures that include most of the functions in CAD and Co-CAD. In such systems, the collaborative functionalities are often proposed all-sided. Unfortunately, these proposed geometric modeling modules are very often not adequate for application, especially when systems are designed for domain-independent collaborative design. It is difficult for end users to implement a solid modeling kernel. Domain dependent systems, such as fixture design, bearing design, hydraulic device design, they are more feasible to implement. Even those solutions provided by commercial CAD software vendors, they have their own powerful modeling kernels, such as AutoDesk, EDS, PTC and Co-create, and their collaborative design functions mostly remain on project management and design data exchange level. When dealing with CAD models in other formats, they have to rely on those standard formats, such as STEP, IGES.

Although most CAD modeling software provides Macro or LISP APIs for customized development, such functionality is limited within the usable domain of well definable operators. Where complex or irregular surfaces or operators are involved, they usually cannot be described and duplicated to other clients. And the APIs supplied by commercial solid modeling engines, such as ACIS and ParaSolid, are insufficient for distributed system development. For example, a developer cannot easily separate viewing functions from the modeling functions inside a stand-alone modeling engine because their libraries are inter-dependent.

CollabCAD uses an event-driven mechanism to realize collaborative design, because each client has its own geometric modeling functions. Many also use thread-sharing method to realize geometric modeling functions, such as OneSpace, NetMeeting, VNC. Such solution is in a way of remote control level, i.e. users use local PC accessing remote system to manipulate remote design. But network configuration and network brand often do not allow such RPC protocols.

[Lee et al., 1999] and [van den Berg et al., 2000], and research topics like Component Framework for Feature-based Design and Process Planning (CFACA) applied component technology in advanced CAD/CAM application fields [Liu 2000]. However, the life cycles of developing such applications are too long and the functions of these applications are not abundant. The matured technologies in stand-alone CAD applications, such as the data structure, file format, system framework and GUI, are not suitable to be used in a distributed environment, leaving software developers having to do much modification in the old system or even program from scratch.

A collaborative CAD system cannot be simply set-up through equipping a standalone CAD system with IT and communication facilities. Due to the complexity of collaborative design activities and the specific characteristics/requirements of CAD systems, it needs some innovations or even fundamental changes in many aspects of CAD systems, such as infrastructure design, communication algorithms, geometric computing algorithms, etc [Li et al., 2004].

The research focus is driven by above discussions, which can be summarized as:

1. Frameworks with an innovative collaborative geometric modeling kernel are difficult to implement, such as [Lee, 1999; Li et al., 2004; Liu 2000; Toshiki 1998; Rafael, 2002; Pahng, 1998; Karthik, 2003].
2. Frameworks with integration functions remain at the distributed project and design data management level. They cannot access to design behavior domain, such as [Nitin, 2002; Caldwell and Rodgers, 1998; Chen et al., 2004; Huang, 1999].
3. Solutions provided by current commercial CAD software vendors are typically restricted to their own software features, and it is impossible for them to provide commercial level interfaces, such as CollabCAD, OneSpace, AutoCAD, ProE Wildfire.

2.5 Research Objectives and Scopes

The thesis objectives and research scopes attempt to address the above problems, considering that different application platforms are involved in a distributed collaborative design environment and the requirements of design co-ordination between multi-disciplines. An adaptable distributed collaborative design module is developed, which can be customized as plug-ins for possible applications used in collaborative design.

This research aims to investigate key functionalities and technologies for distributed collaborative design and focuses on an adaptable architecture, collaborative functionalities, coordinating mechanism, collaborative design communication protocol definition, design representation and distributed design management. To achieve these objectives, necessary tasks are identified and described as follows:

- **Distributed Collaborative Design Fundamental Analysis**, that targets at identifying the collaborative design functional requirements and their characteristics.
- **Adaptable Structure Design**, which is established to suit an adaptable model and to integrate with an existing internet-enabled design system.
- **Collaboration Communication Protocol Development**, which deals with the design behavior definition and representation.
- **Design Co-ordination**, which concerns management and transmission of data in the distributed collaborative design environment
- **Design Representation**, which aims to represent design, design behaviors and design process. The design process concerns the implementation of distributed collaborative design projects and tasks.

Chapter 3 System Analysis and Design

This chapter proposes an architecture of the system to satisfy adaptability requirements of the distributed collaborative module expected of this thesis work. It also gives an overview of the functionalities of the system. Section 3.1 proposes the architecture and components of distributed collaborative design module. Section 3.2 shows the functional structure of the designed module and outlines the functionalities of every part. Section 3.3 discusses the distributed collaborative design communication protocols, which are designed for collaborative design co-ordination and synchronization. Section 3.4 proposes a design management mechanism and solution for teamwork collaboration.

3.1 Distributed Collaborative Design Architecture

To support collaborative design, computer technology must not only augment the capabilities of the individual specialists, but also enhance the ability of collaborators to interact with each other and with computing resources. However, engineering design has to address several complex characteristics, for example, diverse and complex forms of information, interdisciplinary collaboration, heterogeneous software tools, etc. And these make interaction difficult to support [Wang et al., 2002].

As discussed in Chapter 2, traditional approaches to sharing design information among collaborators and their tools include the development of integrated sets of tools and the establishment of data standards. These approaches are becoming insufficient to support collaborative design practices, because of the highly distributed nature of the design

teams, diversity of the engineering tools and the complexity and dynamics of the design environments.

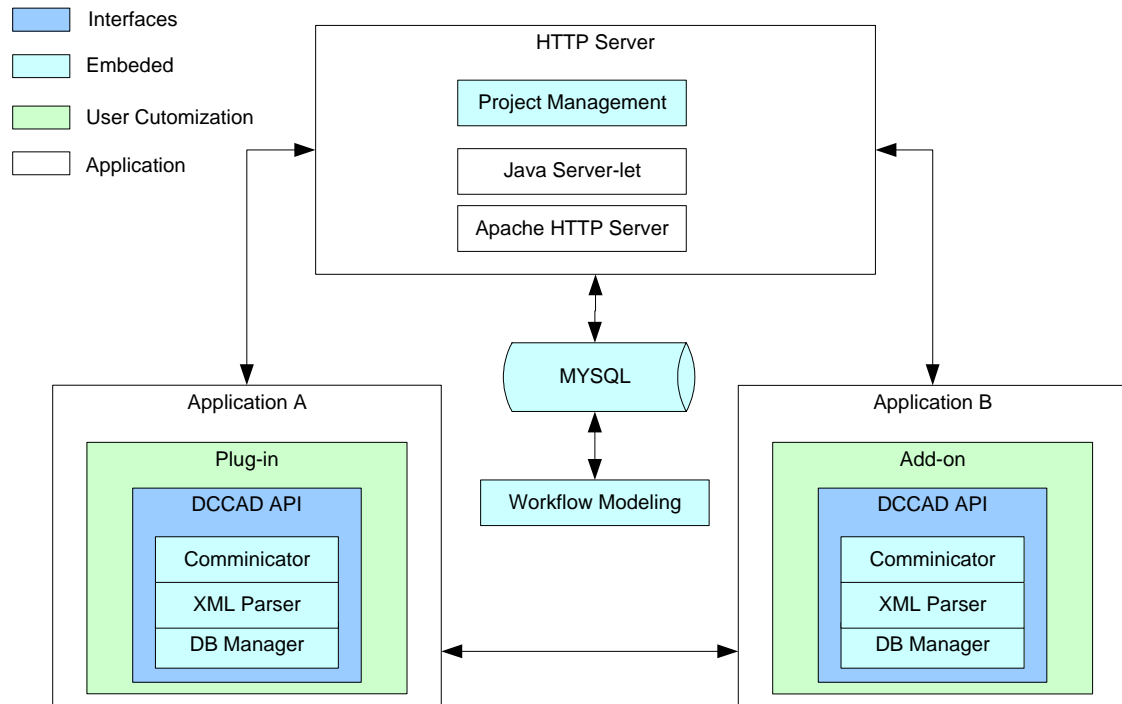


Figure 3.1 Adaptable DCCAD Module Architecture

This thesis proposes an adaptable architecture for distributed collaborative design system and application implementations, as shown in Figure 3.1. DCCAD kernel is abstracted from the fundamentals of distributed collaborative design, as discussed in Chapter 2. The kernel implemented most of the interfaces desired in the realization of collaborative design problems. DCCAD Kernel details will be discussed in next section. Because most commercial applications provide user development interfaces and customization functions, in the Plug-in or Add-on level, the customized modules are designed in charge of communicating works between applications and the DCCAD kernel. Thus operation

behaviours of different applications can be routed through the DCCAD kernel, as shown in Figure 3.2.

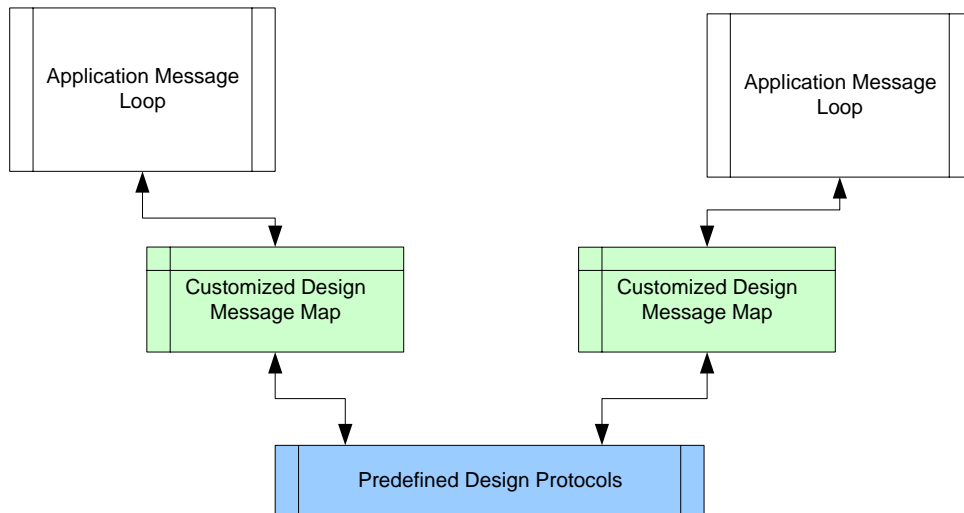


Figure 3.2 Message Mapping Mechanism

The HTTP server provides a project management pool that is accessible from different applications. The dynamic project status and design progress are represented through Web pages. Data synchronization is accomplished dynamically by the DCCAD Kernel through a designed co-ordination mechanism. The content of published data is related to the design content itself and the dependencies of the data. Section 3.4 will discuss collaborative design management solutions based on this architecture.

3.2 DCCAD Kernel Functional Structure

Considering fundamental functions of distributed collaborative design, the proposed DCCAD kernel abstracts most of those functions and wraps them into application-independent dynamic linkage libraries. Detailed components are shown in Figure 3.3.

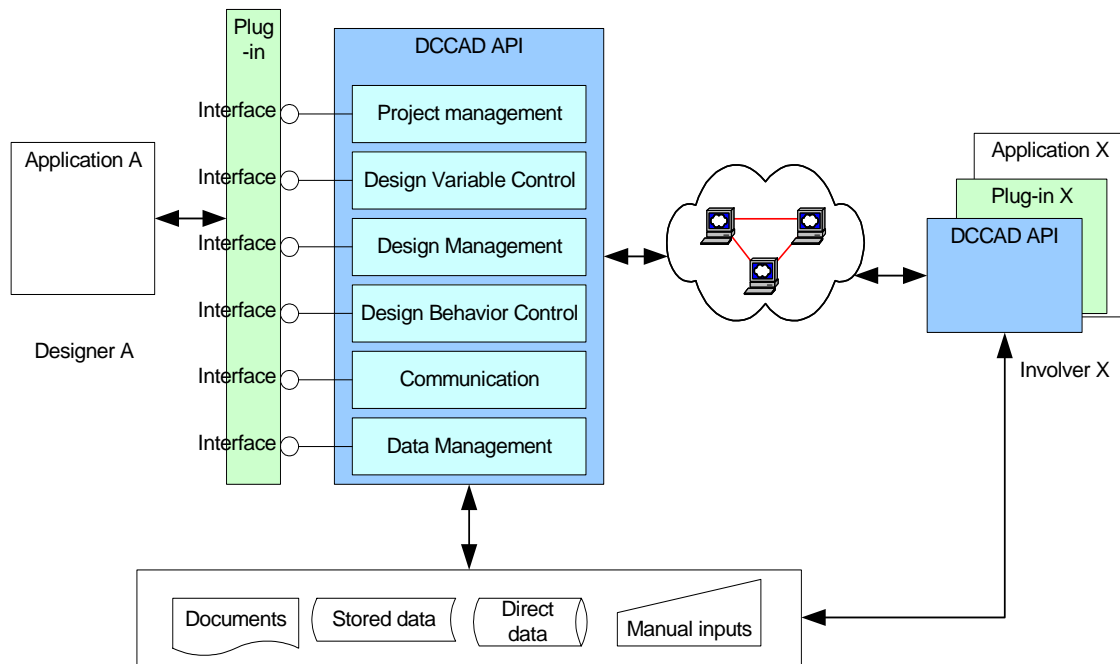


Figure 3.3 DCCAD Kernel Functional Structure

The project management module is designed for distributed project co-ordination and visualization. It uses a task-oriented modeling structure to represent a distributed collaborative design project, with dependency tags. The design variable control module is designed to manage global design parameters that have dependency with other tasks. The design management takes responsibilities of general design problems such as design requirement and design process query and updating. The design behavior control module provides the message-mapping functions. This enables different applications can capture each other's behaviors, such as design is modified, design variable is added, and variable value is set. Communication module provides different communication functions such as HTTP request, socket connection, peer-to-peer connection and message broadcast. Data management module provides interfaces to MYSQL database and XML decoding and

encoding functions. These modules function in the name-space of DCCAD kernel and itself has communication protocols and structured data.

3.3 Data Structure Design

Regardless of the nature of the work, a collaborative group needs a unified framework for data aggregation, i.e. a structure to capture and organize the information developed by the group. Within this structure, there must be a common vocabulary and behaviour; terms must be developed to describe objects and functions that are comprehensible to the group, and the data in the structure must follow a logical model with predictable behaviours and consequences. These conventions enable users to rely on the system to provide a better solution than ad-hoc communication. The data structure designed in the research work is based on the representation of a distributed design project:

{Projects, Task, Behavior, Resource, Dependency, Event, People, Design}

abbreviated as:

{PJ, TK, BH, RS, DP, EV, PL, DS}

Thus project denoted by **PJ** (x) can be defined as:

PJ (x)= **{PJ, TK, BH, RS, DP, EV, PL, DS}** where:

- PJ** = {pj1, pj2, pj3...}, set of sub-projects and parent projects;
- TK** = {ta1, ta2, ta3...}, set of tasks decomposed in project;
- BH** = {bh1, bh2, bh3...}, set of design behaviors;
- RS** = {r1, r2, r3...}, set of resources, where $r=\{r', r'' \dots\}$, resource map;
- DP** = {d1, d2, d3...}, set of dependencies;
- EV** = {e1, e2, e3...}, set of events triggered by behavior or data;
- PL** = {p1, p2, p3...}, set of people involved;
- DS** = {ds1, ds2, ds3...}, set of design items.

Based on the project elements definition and the consideration of possible application integrations, such as project management, design management, process management for distributed collaborative projects, we create data tables that can cover most of those application elements, shown as Table 3.1. It is hard to define exactly suited data structures for those applications, especially when different applications use different domain name conventions. DCCAD kernel provides data level exchange interfaces that enable users encode and decode such data elements for their customized namespaces.

Table 3.1 List of Example Tables

Key Tables	Description
PEOPLE	Project involvers
GROUP	Groups of involvers
PROJECT	Project properties
TASK	Task properties
ITERATION	Iterations in task
DESIGNVAR	Global design variables
DESIGNRULES	Rules defined for design
ACTIONS	Design actions
CODESIGNDATE	Date of different state

Task Table SQL Example:

```
CREATE TABLE Task IF NOT EXIST {
  id INTEGER not null,
  last_update DATETIME,
  name VARCHAR(255),
  type VARCHAR(255),
  description TEXT,
  disposition VARCHAR(255),
  acceptor_id INTEGER,
  pair_id INTEGER,
  created_date DATE,
  story_id INTEGER,
  estimated_hours DOUBLE PRECISION,
  original_estimate DOUBLE PRECISION,
  is_complete BIT,
  primary key (id)
  key 'story_id'
  key 'taskidx'
}TYPE=MYSIAM
```

Design behaviour data structure is denoted in UML.

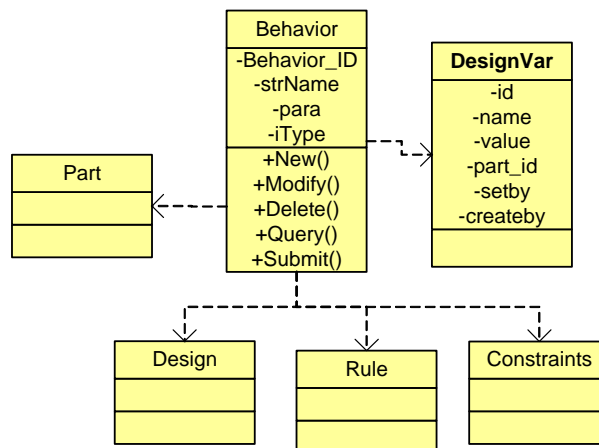


Figure 3.4 Design Behavior UML Denotation

The designed data structure will be discussed in Chapter 4 in detail, concerning the distributed collaborative design visualization and representation topic.

3.4 Collaborative Design Communication Protocols

In the namespace of collaborative design process, a collaboration group may have many clients. Each client is assigned with an identity ID. When a client is login to an active design process, the client broadcasts a packet asking for a collaboration ID with device information in it, including the name and client type. For performance purposes, the successful address is "written down" in COCADRAM and used as the default ID in the future. As would be expected from Co-CAD domain, names were truly human readable, containing spaces, upper and lower case letters, and includes support for searching. DCCAD defined ways to allow "controller" devices override the default mechanisms. The concept here was to allow controllers to provide all of this information.

In the traditional client-server frameworks, clients only talk to collaborative design coordination server. When requests arrive, the server triggers its predefined services correspondingly, such as sending email to clients, updating project or task status. In the real collaborative design world, such interactions are often unforeseeable. And such communications are triggered in the data level. Real-time design behaviors cannot be captured when necessary. A client may ask other clients to set up a unique communication channel for special purposes, rather than notify other clients or have a remote control over other clients. Thus such singleton mechanism cannot satisfy the dynamic collaborative design environment requirements.

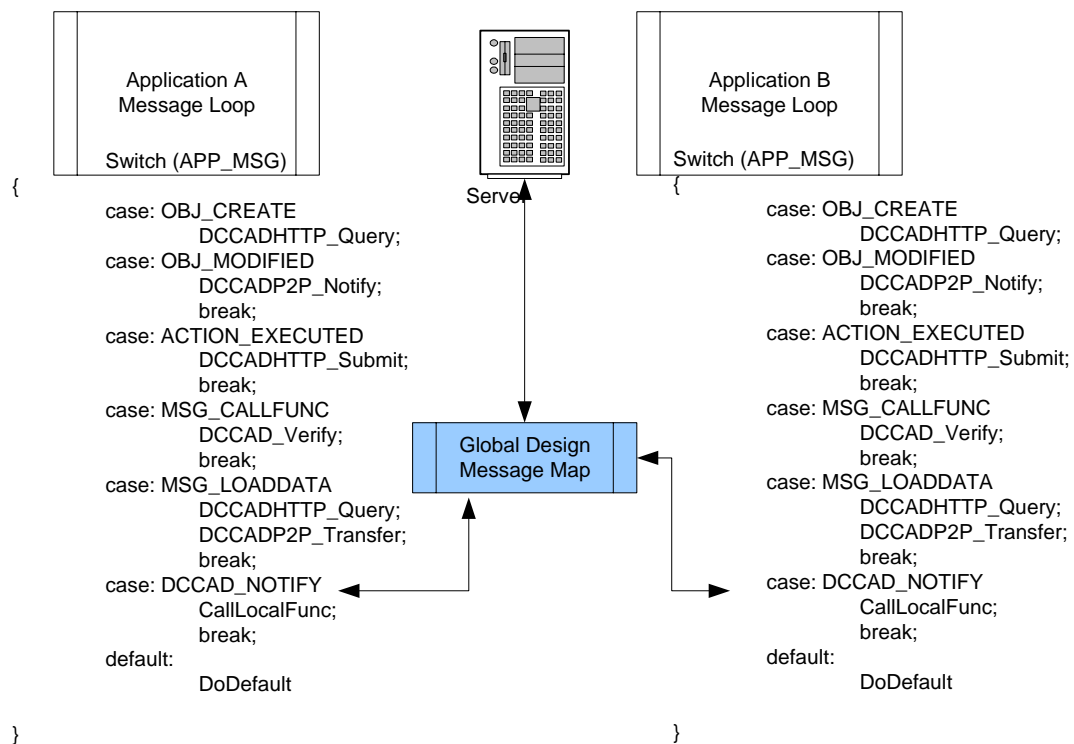


Figure 3.5 Application Message Mapping Sample

In the adaptable collaborative design module, different communication protocols are defined. Clients are able to talk to the controller as well as other clients. There are two types of connections defined in the DCCAD module: client-server and peer-to-peer. Client-server connection provides those central-control-based functions. Peer-to-peer connection provides dynamic collaboration functions. After message implementation in the plug-in, synchronous or asynchronous design behaviors are propagated to all those clients involved. Examples are shown as Figure 3.5 and Table 3.2.

Table 3.2 Design Behavior Definition Example

Design Behavior Examples	Description
LOAD_TASK	Load task from server
MARK_TASK_AS_DONE	Mark task as done
SET_TASK_VARIABLE	Set task variable
ASSIGN_TASK_TO_USER	Assign task to user
CREATE_OBJECT	Create new design
SET_OBJECT_COMMENT	Comment on design
SET_OBJECT_VARIABLE	Set design variable
SET_VARIABLE_VALUE	Set design variable value
SET TASK VARIABLE	Set task variable
SENT WORK ITEM TO CLIENT	Send item to client
SEND XML TO CLIENT	Send XML to client
CALL PROGRAMM	Call program
SEND SOAP REQUEST	Send SOAP request
SET_TASK_RULES	Set task rule
QUERY_DATA	Query data
SUBMIT_DATA	Submit data
SEND EMAIL	Send email
EVALUATE CONDITION	Verify condition

3.5 Design Management Model

In a distributed collaborative design environment, one of the key research issues is how to realize design synchronization and design co-ordination. When data in the system change, the updated information must be quickly visible to everyone who needs to see it. Most companies communicate changes in an ad-hoc fashion, i.e. big changes are communicated quickly, while small changes often languish within one small group before making their way to the group as a whole. Users need to be able to rely on the system to communicate all changes in a standardized reliable way. In addition, the system needs to have the process integrity to implement changes consistently, so that a change takes effect without the risk of being lost. Finally, the system must have referential integrity: each piece of data affected by a change must be updated, so that the consequences of a change are reflected along with the change itself.

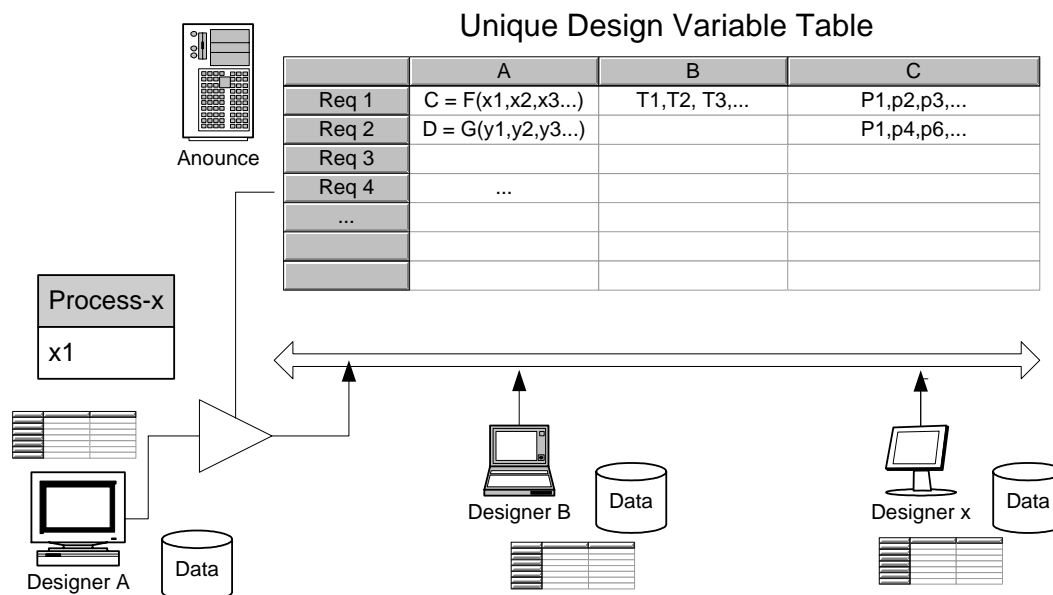


Figure 3.6 Design Management Model

To propagate design changes and modifications, this work uses a unique design variable table and design constraint list that is stored in the central database, shown as Figure 3.6.

Design variables and constraints are defined by different clients or specified by design specification in early stage. Whether the design variable should be submitted to the unique design variable table depends on if the design variable created is dependent on other designers or is referred somewhere. Those locally used variables can be set to have relations with those global defined variables. This mechanism makes it flexible when different involvers have their own namespaces.

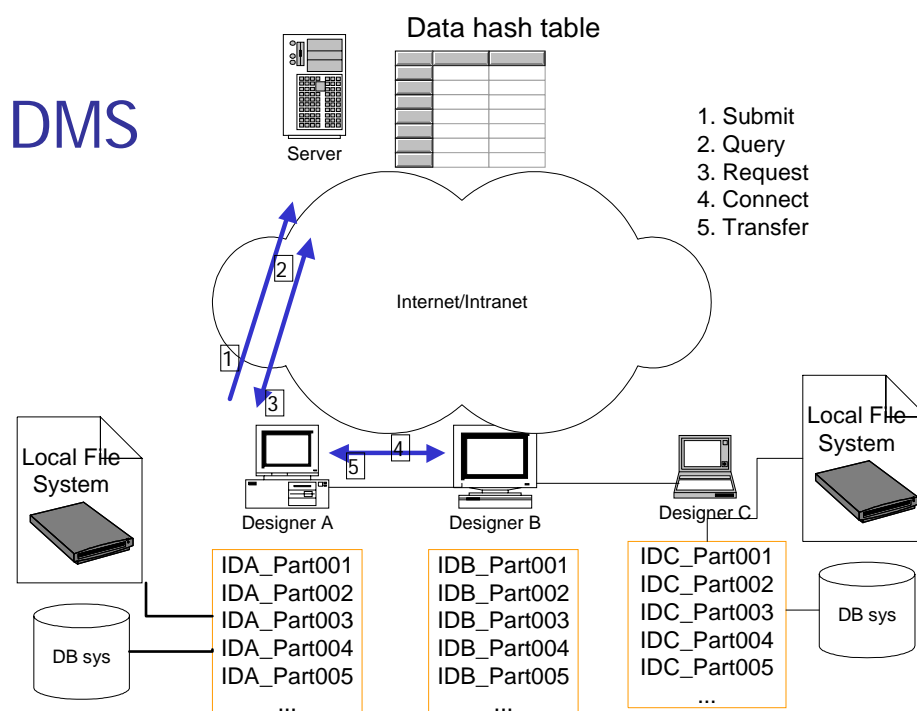


Figure 3.7 Design Management Solution

Aside from the way that information is treated in the data aggregation framework, the framework itself must be designed with the group in mind. Each individual in the process must be able to perform their key functions within the framework, and all the relevant data generated by these diverse functions must be accommodated. The products of the

collaboration must be captured and reused, i.e. key decisions must be recorded, preserved, and broadcast through out the group.

In this thesis work, the data management mechanism uses a data hash table that records the design data identity ID with client ID tags and design attributes. Whenever designers finish certain tasks and get certain versions of design. They can submit the updated version ID to the data hash table so others can quote those data. Data transmission is triggered via peer-to-peer transmission protocols, illustrated as Figure 3.7.

3.6 Summary

In this chapter, an adaptable system model for distributed collaborative design is introduced, and the data structure is presented based on this model. The key functionalities provided by the DCCAD kernel is outlined. Collaborative design communication protocols are recommended. And, collaborative design management solutions applied in the thesis work are introduced.

Chapter 4 Collaborative Design Representation

In collaborative design environment, problems can arise through the differing objectives, plans and goals that participants hold. Dynamic data representations become one of the key problems in a distributed collaborative design environment. Synchronous or asynchronous collaboration, when using a human-computer-human design system introduces special problems that are not evident in human-to-human collaboration. This is exacerbated when using a computer aided design system that needs to have a structured representation of the objects being described in the whole product development lifecycle [Perry et al., 1995]. Design representation includes product structure, design artifact relationship, design status, design changes, design events, design processes, design constraints and so on. Such representations permit problem understanding through the externalization of individual stakeholder mental models, making them available to all involved. Informed negotiation can then ensue the design ideas and iterative redesign to proceed until a final decision can be agreed upon.

In this chapter, section 4.1 introduces collaborative design information and knowledge characters. Section 4.2 discusses why XML is utilized to represent the domain knowledge and introduces graphical UML (Unified Modeling Language) notations. Section 4.3 presents the XML Schema for design artifact representation, design process representation, communication representation in UML notations. Section 4.4 summarizes the content of this chapter.

4.1 Introduction

Collaborative system design poses a number of problems, not only those encountered by individual designers in the collaborative process, but also co-ordination problems that arise from multiple agendas and understandings of a design problem. Providing tools to support the collaboration of the various stakeholders in design is therefore an important goal, which has already been the subject of research in CSCW and other disciplines [Boland et al., 1992; MacLean et al., 1989; Muller, 1992].

Design collaboration places high cognitive demands on the system. Various methods of external storage of intermediate plans, results and processes, such as diagrammatic representations, are often used. However, when a process is dynamic, these static representations are inadequate. Texts are often used to represent information that changes over time, when problems are complex and dynamic, traditional methods of representation do not work well. Confusion can occur, resulting in poor solutions. When there are several stakeholders these problems are compounded: with multiple agents, perceptions of the problem differ, as do the ways that they propose to solve them. This is made more complex because agents are often unable to unambiguously make their ideas explicit [Perry et al., 1995].

Marr [Marr et al., 1982] describes three levels of information: computational level, representation level and implementation level, at which information processing systems need to be described to account for a satisfactory explanation of the task. The three levels guide the selection of a means of data collection by specifying what is required to satisfy the needs of an adequate explanation of the observed behaviors at a cognitive level. A

theory of collaborative engineering design must specify the three areas clearly to achieve a full description of why, on what, and how the cognitive processes operate within the system. By making each level explicit, the theory becomes open to objective analysis and the possibility of empirical examination.

The aim of design representation is to facilitate the collaborative design process and design activities in a distributed environment. This research explores XML (eXtensible Markup Language) as a formal language to represent static and dynamic design information and design knowledge. This chapter presents an approach to represent collaborative design case using multiple perspectives. These perspectives, including design artifacts, collaborative design process and design communication, allow collaborative work to be applied in the broader, and more creative phases of the design process, rather than just as a walkthrough of the final design.

4.2 XML and UML

In this thesis, representation for distributed collaborative design was developed using XML, which is one of the most suitable languages for unstructured data representation. XML is a description language that supports meta-data description for particular domains and these meta-descriptions allow applications to interpret data marked-up according to this format. Besides being a recommendation of W3C, XML is used in our work because it is neutral, platform-independent, and flexible. It also allows structured data need in Web applications. This means we can collect data from various places and exchange structure rich data in various applications over existing network protocols. Today, there is also ongoing development that will include a query language in the XML standard. This

section describes the case representation for fixture design in XML using Unified Modeling Language (UML) notation.

The graphical UML notations used to represent XML model are summarized in Figure 4.1. The UML depicts a class as a rectangle with its name at the top and its attributes in the middle (Figure 4.1a). Relationships between classes are depicted as lines that link them. The inheritance relationship is identified by a hollow triangle at the end of line that indicates the superclass (Figure 4.1c), while the aggregation relationship is identified by a diamond at one end of line that indicates the owning class and by an arrow at the other end (Figure 4.1b). Multiplicity indicates how many instances of one class are related to a single instance of another class at a given point in time. For example, Figure 4.1(c) means one class whole has many parts. The multiplicity notations are located at the end of a link. The meaning of them shows in Figure 4.1(d).

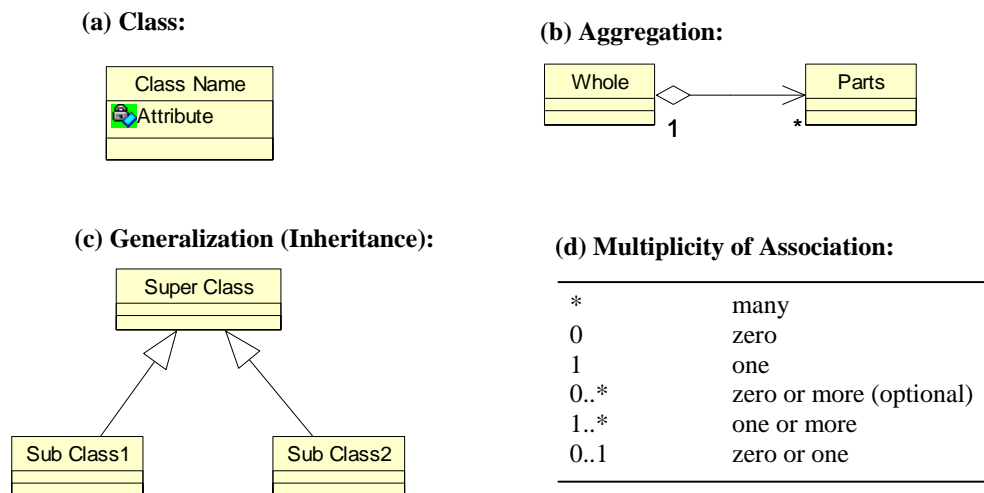


Figure 4.1 Unified modeling language notation

4.3 Design Representation using XML

This thesis presents an XML-based collaborative design representation that involves the use of multiple perspectives to support this process. These perspectives include those that:

- are produced as a result of different stages of distributed collaborative design.
- are produced as a result of applying information filters that are designed specifically for the tasks performed by individual users.
- occur as a result of different behaviours in a collaborative design environment
- occur when multiple collaborators offer their opinions on the design.
- occur when collaborators work on multiple designs.

Following sections will describe these ideas in depth, explain the specifications for collaborative design representation, as well as describe an implementation of a networked virtual design space that embodies some of these concepts.

A fully functional collaborative design representation will need to manage many different kinds of things: documents (resources, requirements, design specifications), abstract knowledge representations (experts, experience), various kinds of design behaviors and data (design data, interaction data, communication data). It is likely that different kinds of problems will require information to be organized in fundamentally different ways. What need attention includes the design part representation as well as the critical distributed collaborative aspects of design behavior representation, design process presentation and the representation of project elements.

4.3.1 Design Part Representation

In a collaborative design environment, it is important to find out how design and design requirements are represented in practice. So as the content of a design part can be understood by other participants. Requirements for a design part are the work-piece to be designed, design variables and design dependencies. Figure 4.2 shows the part representation using UML notation.

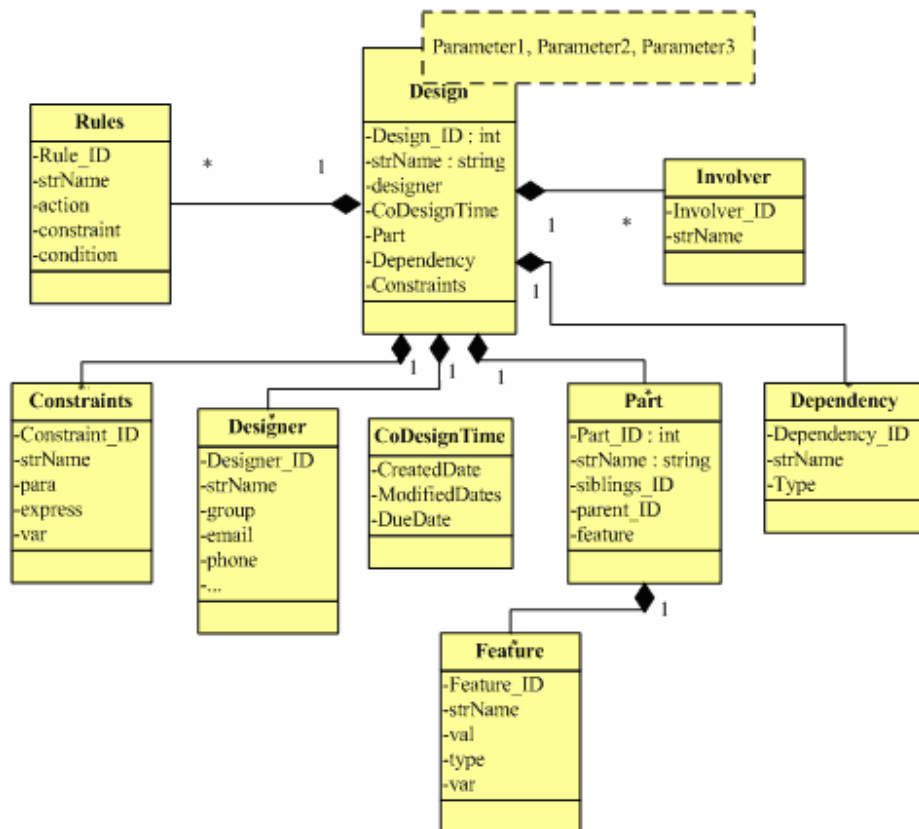


Figure 4.2 Design Part representation in UML notation

A work piece is described using feature-based geometry and material properties. The geometric information, as well as engineering information (tolerance, dimensions, etc.) is

pertaining to the features. The features include boss, pocket, hole, slot, step, and each feature can be classified further.




```

    <?xmlversion="1.0"?>
    <PARTDOCUMENT>
    <PART>
      <PARTNAME>model3</PARTNAME>
      <PARTID>32</PARTID>
      <Designer_ID>4521</Designer_ID>
      <Version>0.0.1</Version>
      <ShapeType>Prismatic</ShapeType>
      <Material>Cast Iron</Material>
      <HeatTreatment>Normalizing</HeatTreatment>
      <MachineType>Vertical Machine</MachineType>
      <DesignVariable>
        <Var>length</Var>
        <Var>width</Var>
        <Var>height</Var>
      </DesignVariable>
      <Constraints></Constraints>
      <Dependency></Dependency>
      <ProfileSize>
        <Length>15</Length>
        <Width>6</Width>
        <Height>4</Height>
      </ProfileSize>
      <ProfileFace>
    </PART>
    .
    .
    <FEATURE>
      <Feature>
        <FeatureName>Hole</FeatureName>
        <FeatureType>Through</FeatureType>
        <FeatureSubType>Rectangular</FeatureSubType>
      </Feature>
      <Dimension>
        <Width>5</Width>
        <Depth>15</Depth>
      </Dimension>
      <Position>
        <X-Coord>0.0</X-Coord>
        <Y-Coord>0.0</Y-Coord>
        <Z-Coord>0.0</Z-Coord>
      </Position>
      <Normal>
        <X-Component>-0.0</X-Component>
        <Y-Component>-0.0</Y-Component>
        <Z-Component>-1.0</Z-Component>
      </Normal>
      <FeatureFace>
        <FaceTag>1786</FaceTag>
        <FaceTag>1782</FaceTag>
        <FaceTag>1784</FaceTag>
      </FeatureFace>
    </FEATURE>
    <FEATURE>
  
```

Figure 4.3 Sample of Work-piece XML Representation

As previously mentioned, the representation was developed in XML for use in our system. Figure 4.3 shows a simple work-piece whose shape type is prismatic and its corresponding XML Schema. It can be seen the part name is “model3” from <PARTNAME> tag and part id is 32 from <PARTID> tag. The designer whose designer_id is assigned with 4521 creates the part. Part version and design variables are provided in <Version> and <DesignVariable> tag respectively. Dependency and design constraints information can be put within the defined tags also. The feature information is provided in the <FEATURE> tag. Figure 4.3 also shows hole class, its subclass and its attributes in the <FEATURE> tag.

4.3.2 Collaborative Design Behavior Representation

In cooperative design, the design process consists of a series of discrete activities. Designers do not always work together and in quite a long period they work independently with their own expertise. A design process typically involves more than one person. In many cases there will be a team of designers participating in the design, along with the client. A design behavior draws in a possibly large number of people to inspect and discuss the result. The communication and cooperation of these people can be supported directly if multiple users have an agreement on the representation of design behaviors. The users have different goals, depending on their role in the process. Designers want to modify objects in a design space, simultaneously if possible. Participants in the design review want to view and browse the design independently.

Collaborative design behavior communication between these users is expressed in the design process. The structure of the XML file in UML notation is shown in Figure 4.4.

The basic operations are: New, Modify, Delete, Query and Submit. The operation object could be Actions, Part, Constraints, Design Variable, Design rule and a design task. When a designer operates on a design, the behavior can be represented in XML. Those behaviors can thus be propagated to, recorded, queried by other involvers.

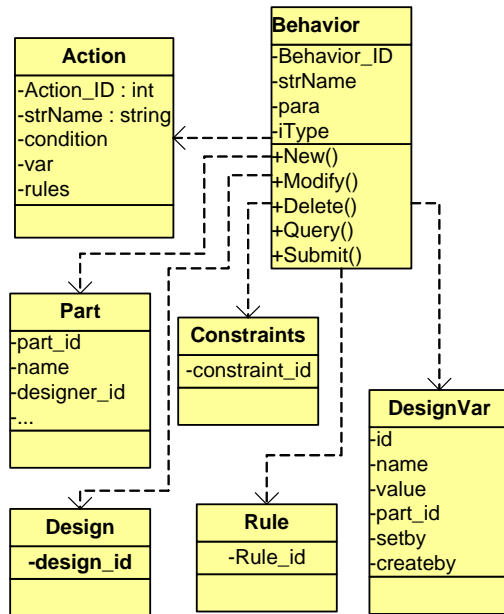


Figure 4.4 Design Behavior Representation in UML Notation

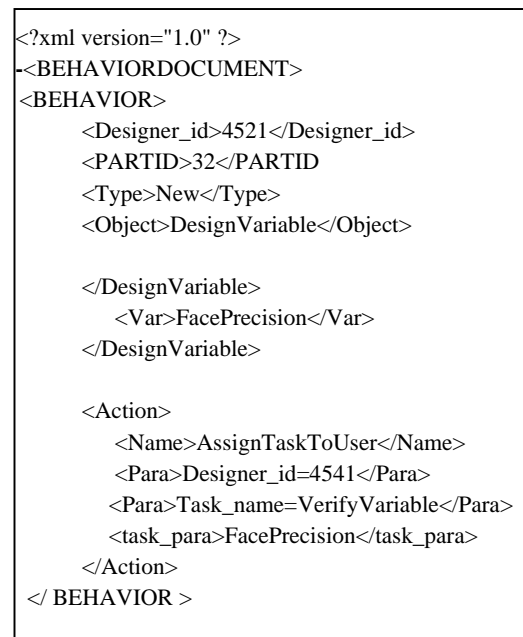


Figure 4.5 Sample of Design Behavior XML Representation

From XML schema shown in Figure 4.5, by those tags we can know new design variable named FacePrecision is created by the designer whose id is 4521 for design part with the part_id 32. The action information is in the <Action> tag. The name of the action is “AssignTaskToUser” and its associated parameters are “Designer_id=4541” and “Task_name=VerifyVariable”. This means the task “VerifyVariable” is assigned to designer with the id 4541. And the parameter for task “VerifyVariable” is the variable name “FacePrecision”.

4.3.3 Design Process Representation

Full design development may involve a large team and take several years to complete. Design, like most intellectual tasks, does not start upon sitting down at one workspace, but continues sometimes consciously and sometimes unconsciously. Process management is a vital part of enterprise systems that integrate cross-application and cross-company process flows while staying responsive to changes in business strategy. To make processes work effectively, it is very important to represent process information among workers. Designers require detailed process information without ambiguity.

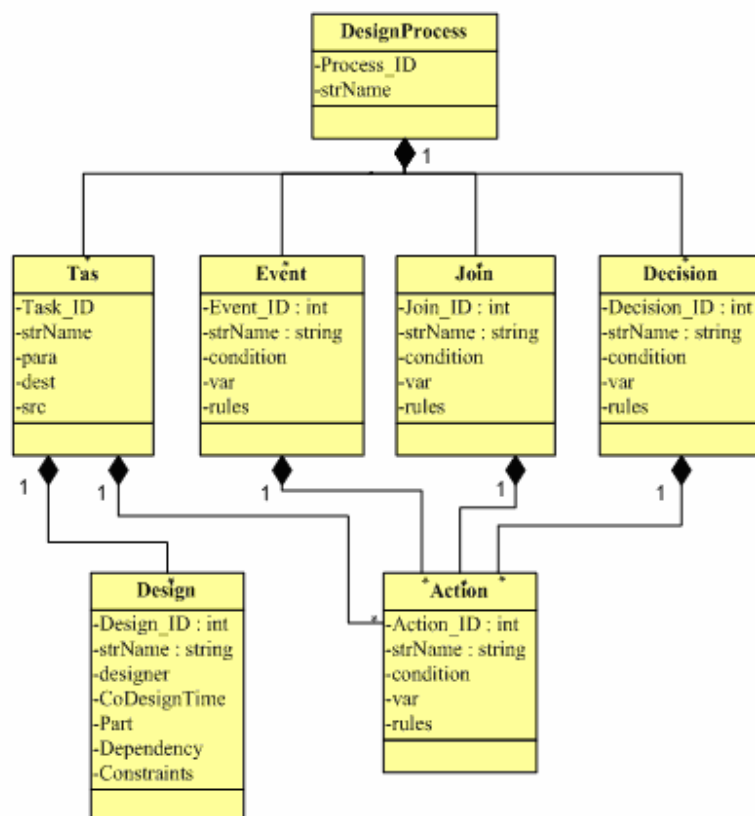


Figure 4.6 Design Process Representation in UML notation

The structure of the collaborative design process representation XML file in UML notation is shown in Figure 4.6. Besides the start and end point of a design process, there are four types of elements defined in the process: Task, Event, Join and Decision.

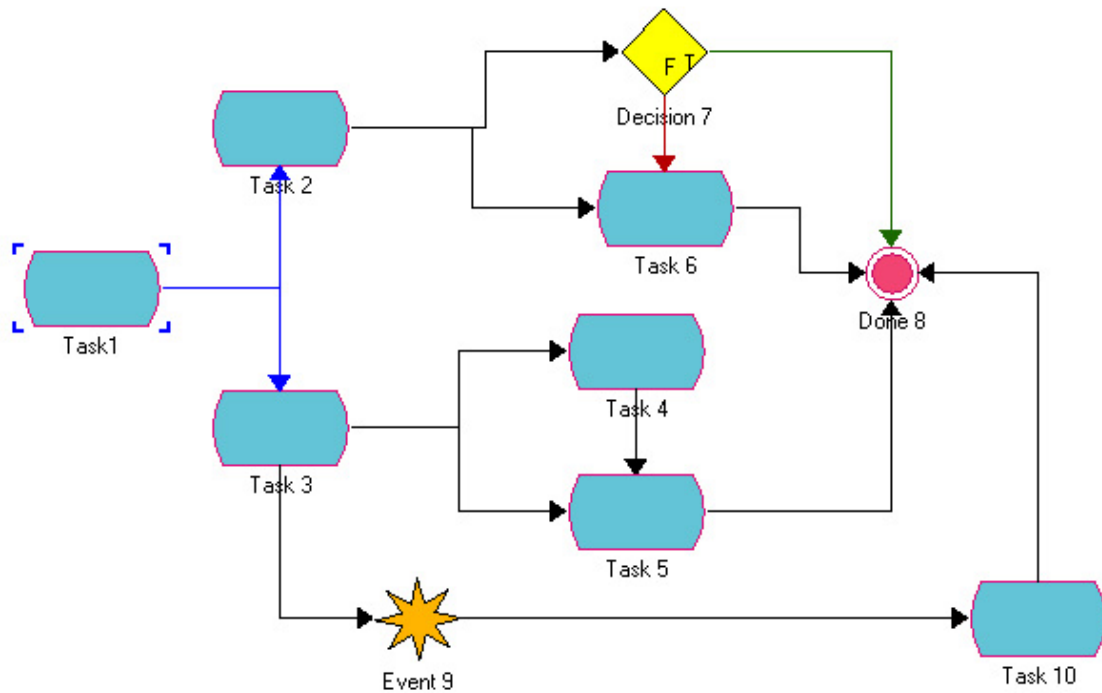


Figure 4.7 Example of Design Process Modeling UI

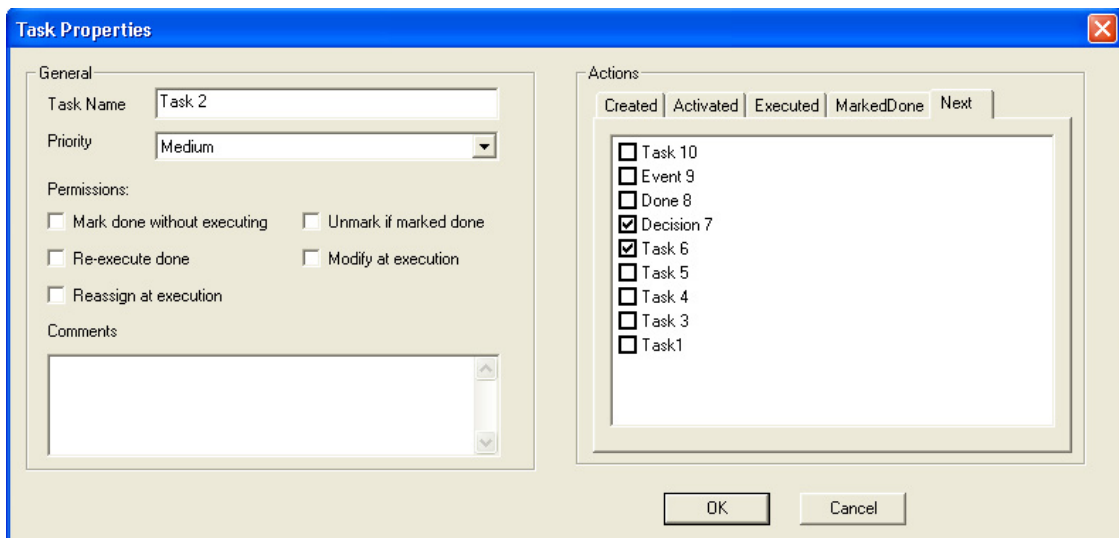


Figure 4.8 Properties of Task2 in Figure 4.7 Example

```

<?xml version="1.0" encoding="UTF-8" ?>
<CoDesign-Process version="1" build-number="0">
  <name>example</name>
  <active>false</active>
  <expiry />
  <createdby>chen xiaolong<createdby />
  <createdate>08/06/2004 02:31:23 AM</createdate>
  <changedby />
  <changeddate>08/06/2004 02:31:23 AM</changeddate>
  <notes />
  <nodes>
    <Task>
    <Task>
    <Task>
      <id>3</id>
      <x>223</x>
      <y>261</y>
      <notes />
      <name>Task 3</name>
      <priority>1</priority>
      <donewithoutdoit>false</donewithoutdoit>
      <doitifdone>false</doitifdone>
      <unmarkdone>false</unmarkdone>
      <modifiable>false</modifiable>
      <reassignment>false</reassignment>
      <nests>
        <next>4</next>
        <next>5</next>
        <next>9</next>
      </nests>
      <actions>
        <created />
        <activated />
        <executed />
        <markeddone />
      </actions>
    </Task>
    ...
    <Decision>
    <Done>
    <Event>
  </nodes>
  <variables>
    <variable>
      <name>$diameter</name>
      <notes />
      <type>string</type>
      <xmltype />
      <inmandatory>false</inmandatory>
      <in>false</in>

```

```
<out>false</out>  
</variable>  
</variables>  
</ CoDesign-Process >
```

Figure 4.9 Sample of Design Process XML Representation

Figure 4.7 shows a simple example of process representation using the defined elements. It consists of 10 nodes of four different types. As shown in Figure 4.9, the relationship is defined in <Next> tag. Figure 4.8 is a task property dialog. Corresponding properties of a task is included in the <task> tag. Design variables and their properties is expressed within <variables> tag.

4.4 Summary

The chapter examines how XML is used as a means to represent units of activity that include design parts, design behavior and design process. It outlines and develops the theoretical basis behind distributed design and demonstrates how it can be used to guide data collection and representation. The role of representations and processes is examined, and its relevance and importance to the performance and understanding of collaborative activities are highlighted.

In principle, the solution of collaborative design representation is applicable to all areas of multi-participant activity. One such area is that of engineering design, a particular instance of collaborative activity. Studies of work can be used to reveal the interactive activities, the content of artifacts, and the mechanisms coordinating the behaviors of collaborating individuals.

Chapter 5 Implementations and Case Study

This chapter presents collaborative design implementations based on the model proposed in Chapter 3 and the methods described in Chapter 4. It demonstrates how users customize their plug-in and collaborate with each other in an environment of distributed collaborative design project. DCCAD collaborative design kernel module is implemented in C++ language under Microsoft Visual Studio.Net environment. AutoCAD® and MS Excel® are chosen as the application platform for user customization based on architecture and methodologies developed in this work Autodesk ObjectARX SDK was selected for AutoCAD® plug-in implementation and VBA was used for MS Excel add-in implementation. MYSQL was chosen for central data management server. Tomcat-Apache was used for HTTP server implementation as the collaborative design project management module.

5.1 Introduction

In networked product development cycle, participants from diverse disciplines, contribute their work collaboratively through remote interaction. To illustrate, consider this scenario: a distributed team of designers are developing a welding fixture model. Design begins with an overall conceptual specification of the device, including annotations about desired functions, performance, design structures, acceptable materials, strength, weight limits, tolerance information, etc. The design teams use the network to communicate with each other and contact engineering service agents. The preliminary designs for product and its subsystems become the starting points for collaborative design. Each move in the process of collaborative design may result in proposals from others to be evaluated. This

evaluation may trigger redesign, accesses to previous design histories and corporate data, and refinements of simulation models. As the process iterates, the activities and decisions are recorded and archived until, eventually, the final product is defined and prototyped.

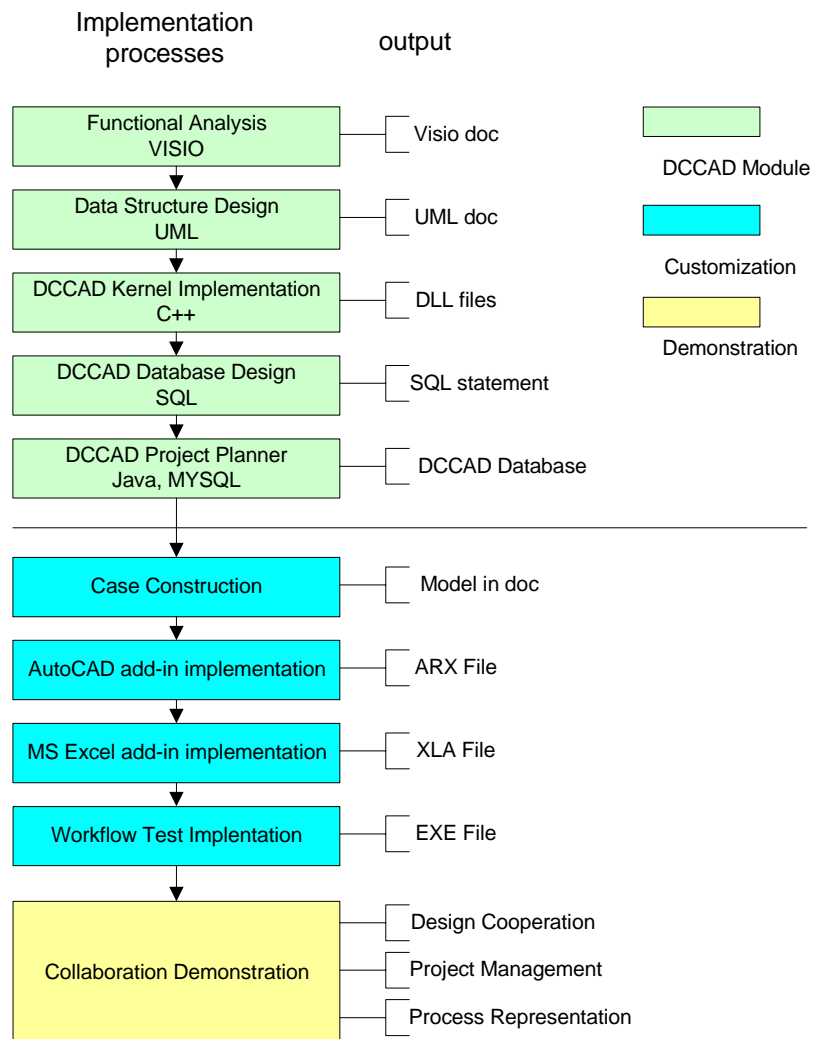


Figure 5.1 Processes of The Work

To fulfil these requirements, this thesis proposes an adaptable system model for collaborative design. The implementation work involved in this thesis includes processes as shown in Figure 5.1.

5.2 Adaptable Collaborative Module Fundamental Implementations

This section details implementation works for the distributed collaborative kernel module described in the previous chapters. Next section will illustrate how adapted add-ins were developed based on this DCCAD model and how collaborative behaviours are coordinated across different engineering disciplines.

5.2.1 Implementation Structure and Co-Design Process

The implementation uses the structure stated in Chapter 3. AutoCAD and MS Excel were used as the application customization platform to demonstrate the scenario given in the beginning of this chapter; the structure shown is as Figure 5.2.

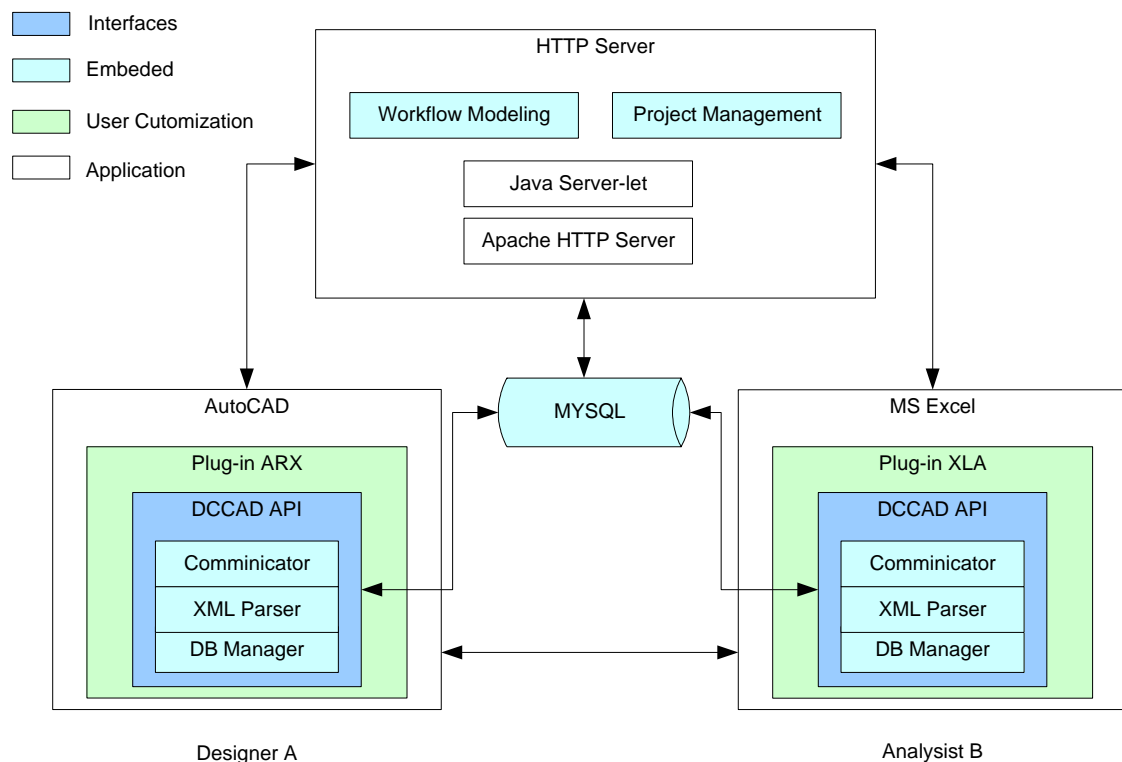


Figure 5.2 Implementation Structure

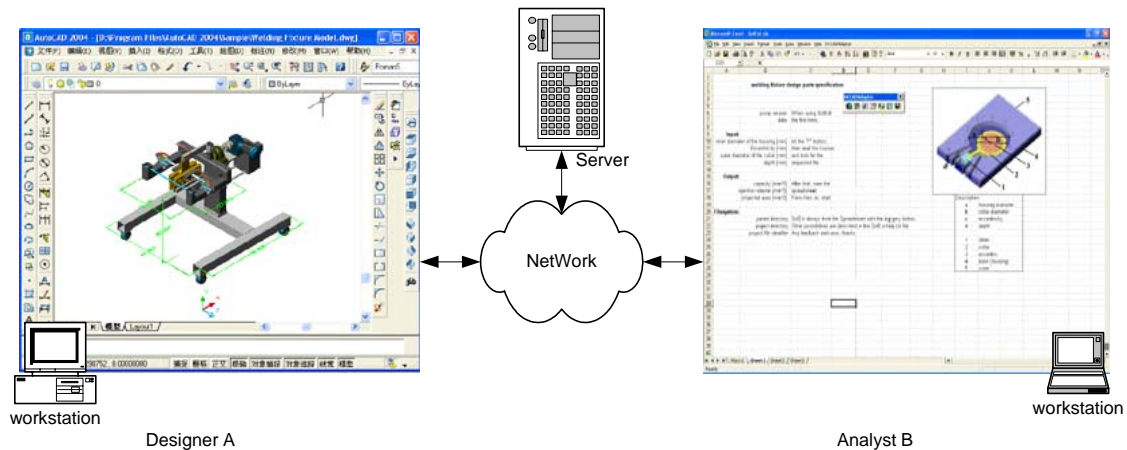


Figure 5.3 Implementation Scenario

The case assumes that: Designer A is designing fixture elements for welding. He uses AutoCAD 2004 as the geometric modelling platform for mechanical structure design. The design specification is composed by Analyst B, who is working on MS Excel 2002 professional, shown as Figure 5.3. When design or specification of the design changes, their behaviors need to be coordinated through a certain mechanism and their design data and design idea needs to be expressed and exchanged via certain tools. The following sections illustrate how the adaptable DCCAD module was constructed and implemented, how the plug-in for AutoCAD and add-on for MS Excel was customized and how collaborations were fulfilled using DCCAD modules.

5.2.2 Collaborative Central Database Design

As discussed in chapter 3, the database design is based on the distributed collaborative design project definition. The data structure of tables created in the database is dependent on the elements of the project definition. The aim of the definition is to realize collaborative design functionalities, such as project management, design management

and process representation. The data structure defined in the MYSQL can be represented in UML notation, as shown in Figure 5.4.

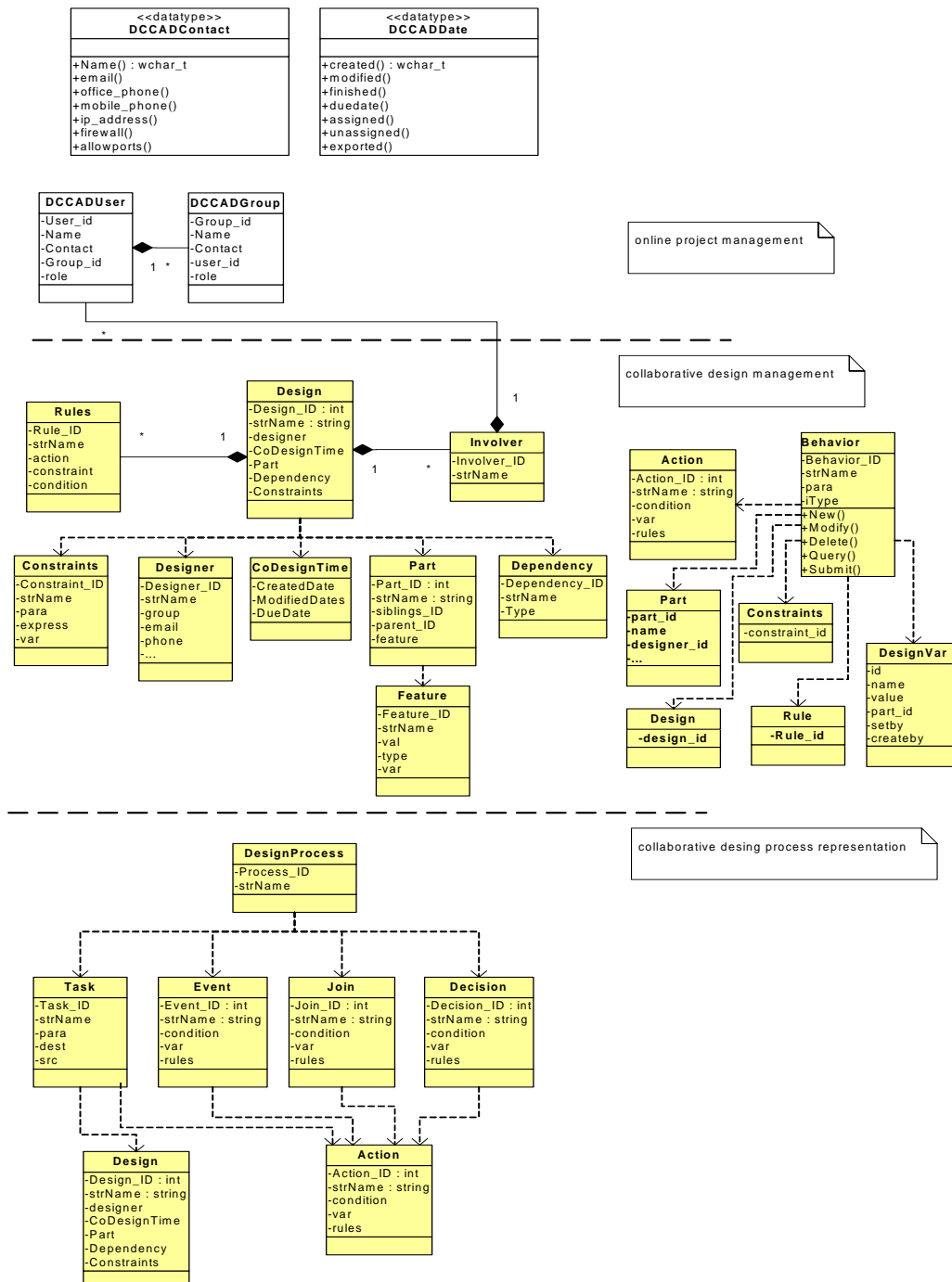


Figure 5.4 Data Structure in UML Notation

5.2.3 Abstracted Collaborative Function Design

The abstracted collaborative API is designed especially for the realization of those basic functions required in the distributed collaborative design environment. Based on those API, users can customize their applications in a more directed, easy and flexible way. And this module makes users from different departments or different domains more understandable to each other's design requirement, design operation and design behaviour. Through message mapping and data transformation, users can implement their task-oriented applications with comprehensible, unified and sharable.

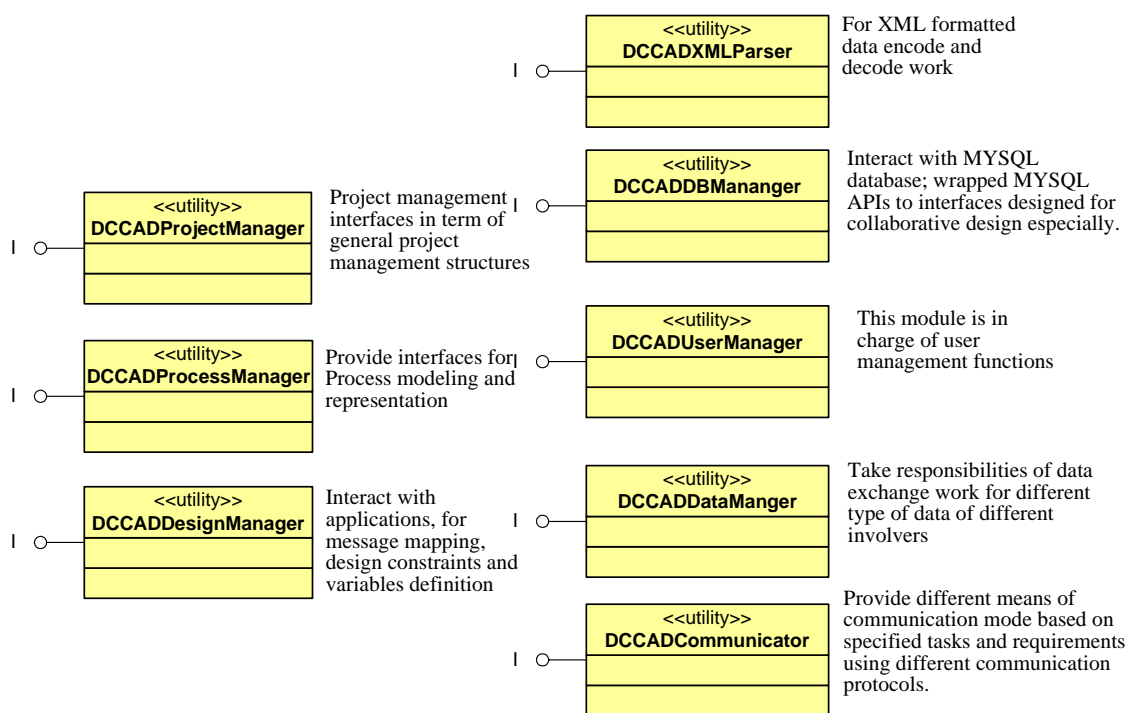


Figure 5.5 DCCAD Kernel Structure

Project management is a Java Servlet Based package. This module is a project planning and tracking tool for distributed collaborative development teams. DCCAD Project Planner is

implemented using Java, JSP, and Struts, Hibernate and MYSQL. To summarize the project planning process, first pick the features to be added (project plans) to the design iterations. The designer can estimate the effort to complete the project either at the overall project plan level or by decomposing the project into tasks and estimating from those tasks. Information about team development speed from the previous iteration is used to estimate if the team can complete the proposed project.

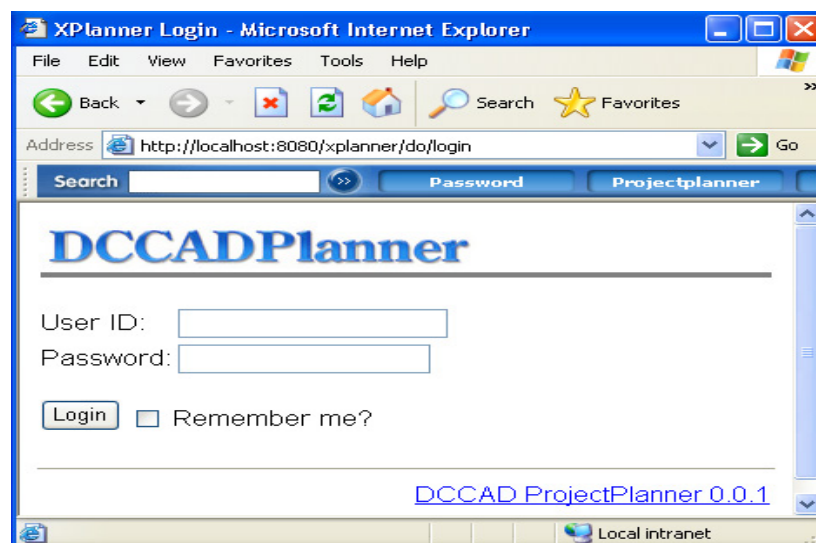


Figure 5.6 Web-based Project Planner Module Snap-shot

The DCCAD project-planning tool was created to support this process and address issues experienced in a long-term distributed collaborative project. To summarize, some of the key goals are:

- Provide an easy, flexible way for designer to track effort on the project and associated tasks.
- Provide continuous feedback of team status to participants on the Web.
- Increase team communication through a shared view of activities and their status.
- Provide automated means of providing planning metrics for iteration planning

5.3 Application Adaptive Customization

This section will demonstrate how design cooperation is realized between a designer who uses AutoCAD platform for welding fixture design and an engineer who uses MS Excel for design specification and analysis. In order to develop a DCCAD module based add-in, pseudo codes of the implementing procedure is shown as Figure 5.7

Include "DCCADBase.h"	//DCCADModule Option Explicit
Using Namespace DCCAD;	
{	Declare Function IniDCCAD Lib "DCCADKernel" () As Long
LocalUIInitialize();	Declare Function SetDCCADIcon Lib "SetDCCADIcon" (ByVal ntype As Long) As Long
DCCADObj = new DCCADObject();	
DCCADObj.DCCADInitialize();	Declare Function DCCADLogin Lib "DCCADBase" (ByVal svr As String, ByVal usr As String, ByVal pwd As String)As Long
DCCADObj.DCCADLogin(SvrName,UsrName,PWD);	
DCCADObj.DCCADGetPartList();	Declare Function DCCADGetPartList Lib "DCCADBase" (ByVal pID As Long) As String
DCCADObj.DCCADAttachPartSynchronization();	Declare Function DCCADGetValList Lib "DCCADBase" (ByVal pID As Long)
MSG msg = DCCADObj.Listen();	
Switch(msg)	Declare Function DCCADSynData Lib "DCCADDataManager" (ByVal e As Long, ByVal lpBuffer As String) As Long
{	
Case DCCADMSGxxx	Private Declare Function SetCurrentDirectory _ Lib "kernel32" Alias "SetCurrentDirectoryA" _ (ByVal lpPathName As String) As Long
DoSomething();	
Case DCCADMSGxxx	
DoOtherthing();	
Default	//Other Modules
}	Sub
}	Call DCCADModule.DCCADxxx();
	End Sub
	...
C++ Implementation pseudo Code	VBA Implementation pseudo Code

Figure 5.7 Add-in Implementation Pseudo Codes

5.3.1 AutoCAD Plug-in Implementation

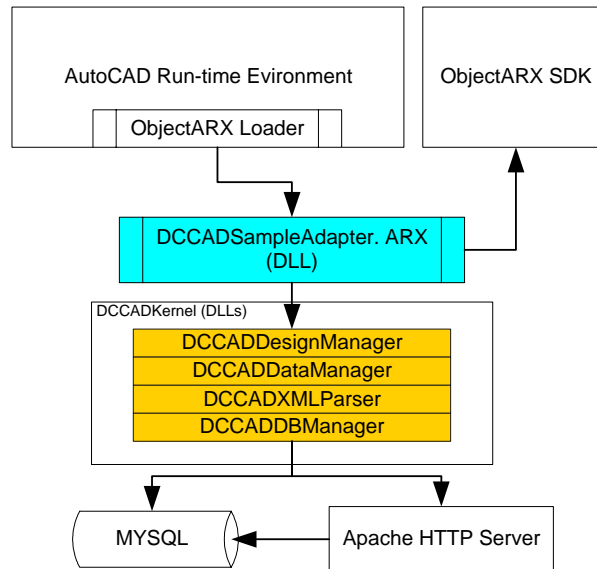


Figure 5.8 AutoCAD Plug-in Structure

Figure 5.8 shows the implementation structure of the AutoCAD plug-in based on ObjectARX SDK. The ObjectARX programming environment provides an object-oriented C++ application-programming interface for developers to use, customize, and extend AutoCAD and AutoCAD-based products. ObjectARX libraries provide a versatile set of tools for application developers to take advantage of the open architecture of AutoCAD software and provide direct access to AutoCAD database structures, graphics system, and native command definition. It enables power users to customize AutoCAD software and frees CAD designers from repetitive tasks. Smaller files, faster drawing operations, and smooth interoperability make an application built with ObjectARX, the best choice for a design software solution.

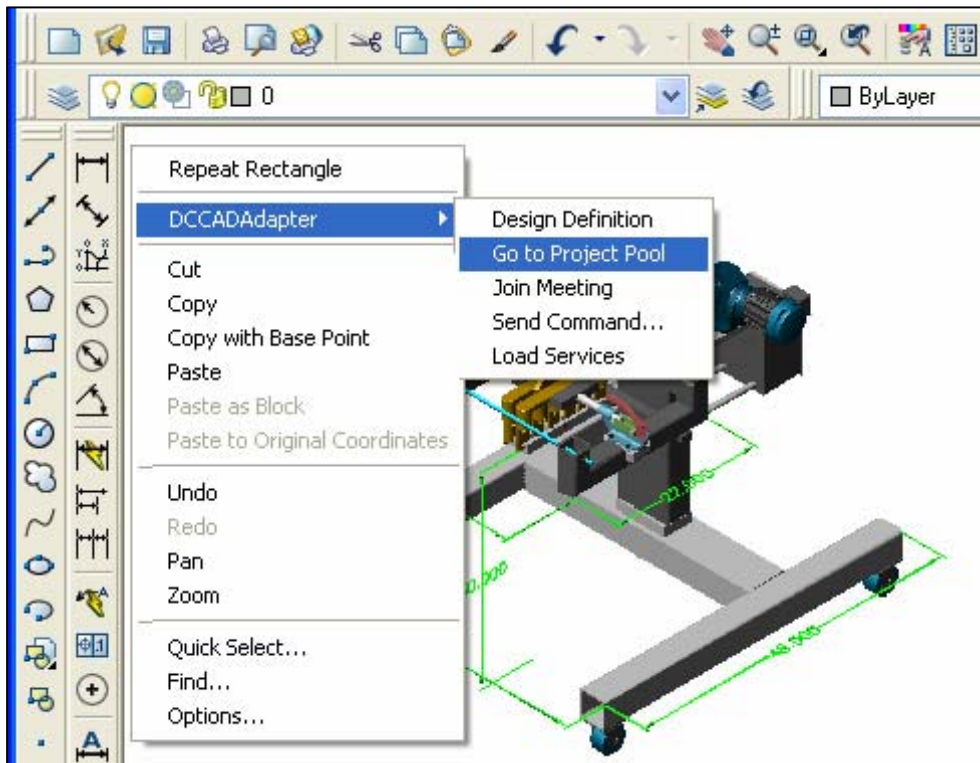


Figure 5.9 Snap-shot of AutoCAD Plug-in Menu UI

Figure 5.9 shows the pop-up menu after the customized DCCADAdapter.arx is loaded into AutoCAD application. Functions, such as design definition, participating in a project group, joining a meeting, sending a command and loading service can be triggered via the menu. The inside detailed function definition depends on user customization.

5.3.2 MS Excel Plug-in Implementation

Microsoft Visual Basic for Applications (VBA) technology has had a major impact on the software engineering largely because of its ability to allow a user more flexibility and customization. The VBA programming language was introduced in 1993 as part of Microsoft Excel and has been part of all Microsoft Office products since. Since licensing the scripting language to third party developers, it has been introduced as an embedded

scripting language in many software packages across a variety of industries. In manufacturing, software vendors are just starting to integrate VBA into their own software packages. VBA allows users to integrate programs easily with other VBA-enabled applications, allowing them to create a customized, integrated desktop that can be re-used for additional applications. To fulfil the objectives, a Microsoft Excel-based approach was chosen, whereby a design analysis system is encompassed within a workbook file. Microsoft Excel is understood, used by, and on the desktop of most engineers. Figure 5.10 shows the implementation structure of the MS Excel add-on.

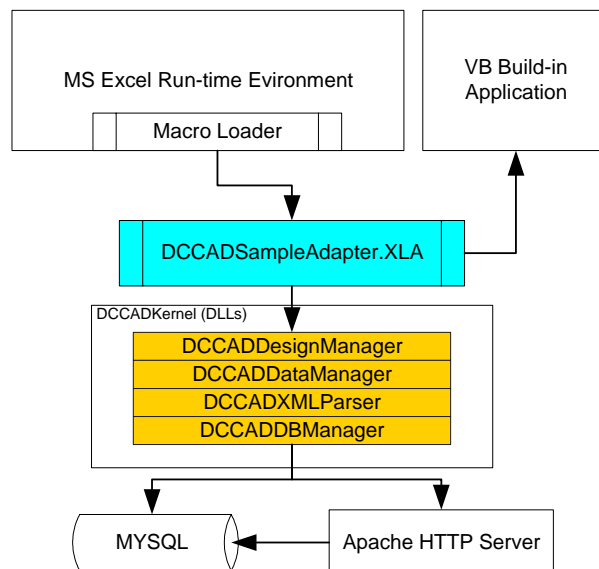


Figure 5.10 Structure of MS Excel Plug-in

Figure 5.11 show the pop-up menu after the customized DCCADAdapter.xla is loaded into MS Excel application. Functions such as opening project pool, joining a meeting, viewing history, sending a command, defining constraints and loading service can be triggered via the menu. The inside detailed function definition depends on user

customization. These functions can increase the efficiency of design processes when large repeated and similar works are involved, especially those products that depend on various unpredictable parameters customized by the user.

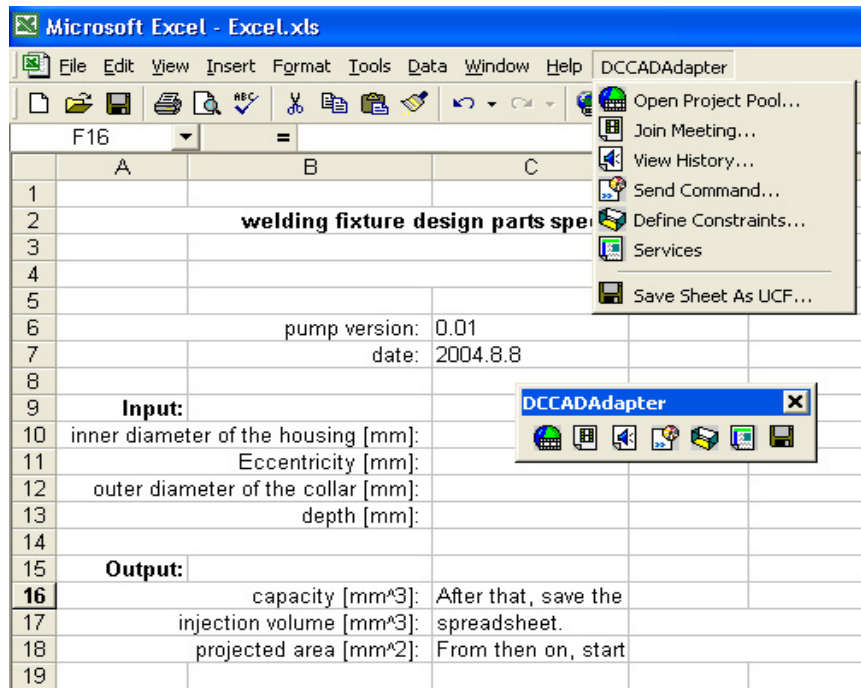


Figure 5.11 Snap-shot of MS Excel Plug-in Menu UI

5.4 Distributed Collaborative Design Demonstration

After the above implementation steps, collaboration works can be realized between the designer using AutoCAD and the engineer using MS Excel. This section demonstrates how collaborative activities are carried out based on the functionalities implemented in the DCCAD modules and customized plug-ins.

In a distributed collaborative design environment, designers who take responsibility of CAD geometric modelling work very often only follow specifications designed by other

engineers. Design results are constantly susceptible of proof. In order to reach an optimized result, design parameters need to be analyzed using simulation or optimization processes. The spreadsheet approach is convenient for calculations as well as more analysis involved using the built-in scientific functions. For more involved calculations, Excel has commercially available add-ons to interface with scientific applications including Mathematica®, Matlab®, and LabView™ for calculation and simulation. Lastly, Excel's built-in dynamic link library (DLL) calling routines allow the use of DLLs that are custom-written, for example, in C++ or Fortran.

As discussed in Chapter 3, the content of design co-ordination includes propagations of design behavior, design variable, design rule and design constraints. In a typical collaborative design scenario, such as OneSpace[3], PTC WildFire[6] and AutoCAD[4] inventor, design modification is propagated through a project repository. CAD models with markups are checked in from one side and checked out by another. Repository server set up the notification mechanism through the status of data. Such correspondences introduced vast copies of the design data in the repository. When the source contains a complex CAD model, the file size can be very large, which may take hours to check in and check out.

Instead of dealing with the complete file, this work uses a message mapping solution. When designers work with a design part, those key design variables can be bound to the entities of design, such as dimensions, features and a defined design region. When those dependent design variables are submitted to the public namespace, other designers or engineers can consequently set up a dynamic linkage with those design variables.

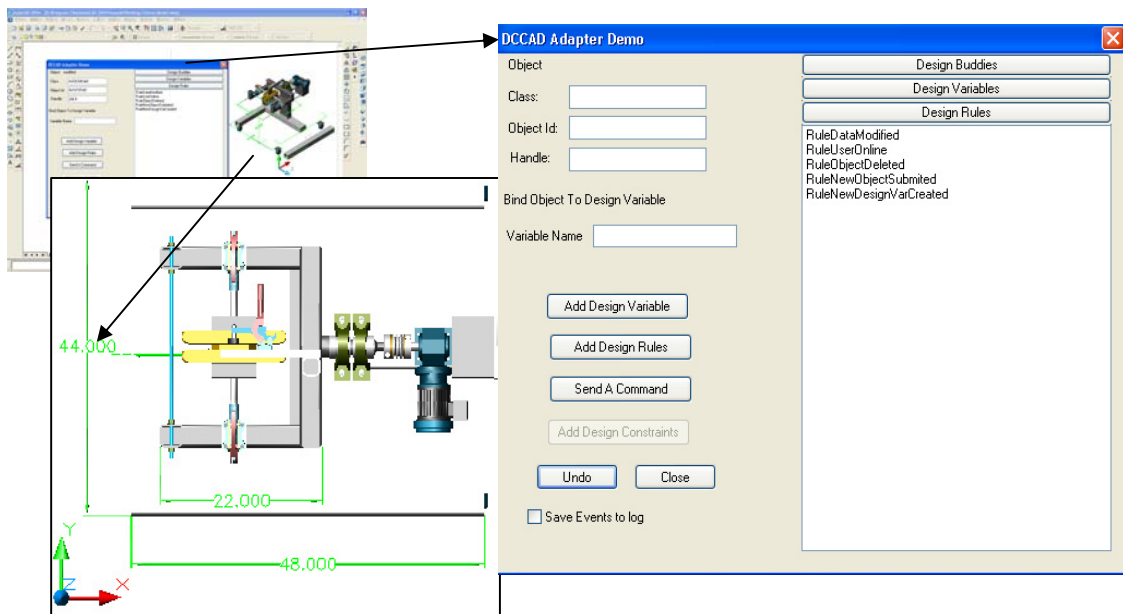


Figure 5.12 Snapshot of AutoCAD Plug-in Main UI

Figure 5.12 shows snapshots of a designer using AutoCAD, with the customization works based on the adaptable DCCAD. The design variables can be created and bound to the design entities. With the necessary parameters established, optimization and simulation calculations in place, and initial designed values in the system, the first iteration occurs. In the iteration process, the analysis system requests to perform necessary calculations, and send the feasible design variables iteratively until the major version stabilized. For a complex product design, design variable quantities might be huge. The system engineer makes top-level design decisions or modification requirements, and another design iteration is performed. The process continues, and variants of the design are considered and characterized as determined by the analysis engineer, eventually constituting a thorough examination of the design variations.

Design Variable Selection - Step 3 of 6

Selecting Design Variables
Select the cells to change in order to optimize the objectives.

Design Variables

Engine Type	Name	mplex Example! -
Number of Launches	Value Cell	mplex Example! -
InnerWidth	Minimum	0 -
OutLength	Maximum	1 -
DesignVariable_4	Type	Continuous -
	Grid Points	5 -

Add Delete

Cancel < Back Next > Finish

Model Rocket Setup

Engine Table				
Type	Thrust	Weight	Cost	Weight
1	3	0.11	\$1.20	1.1
2	4.5	0.22	\$2.20	1.5
3	8	0.45	\$2.50	1.9
4	10	0.879	\$3.70	2.3
5	14	1.25	\$5.30	2.7

Engine type	3	The engine type is a discrete design variable
Engine thrust		Having a thrust greater than the total weight is a constraint
Engine weight		
Engine cost		
Number of launches	5	The number of launches is a discrete design variable
Army men per launch	2	Maximizing the number of launches is an objective
Payload per launch	4	
Rocket body weight		
Total vehicle weight		Having a thrust greater than the total weight is a constraint
Total engine cost		Minimizing total engine cost is an objective function

Figure 5.13 Snapshots of MS Excel Customization UI

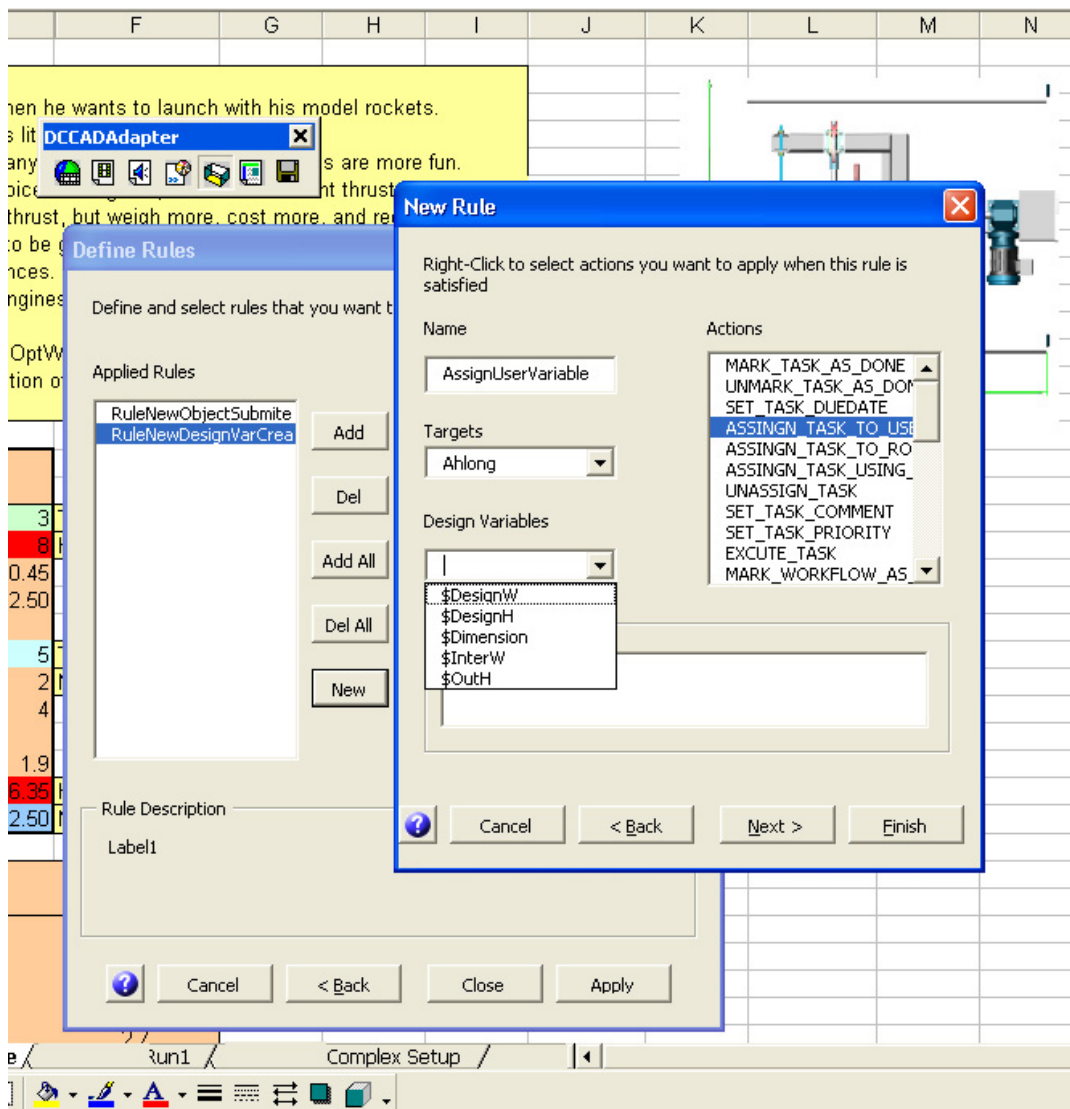


Figure 5.14 Snapshots of MS Excel Plug-in Rule Definition UI

Figure 5.13 shows the analyst working with MS Excel for optimization, based on the design model specifications, design variables and other possible parameters that determine the optimization solution. Figure 5.14 shows the snapshots of the analyst binding the design variables to the rule. The design rule will assign the optimized value every time when a result is generated. The design engineer can load the optimized or

simulated design variables. The next round design begins until a satisfactory outcome is reached.

As discussed in chapter 3, it is important to create a project management module record and visualize project structures and design status. A Web-based project management module is implemented together with the adaptable DCCAD module. The module is implemented in Java Servlet technology on Apache Tomcat HTTP server.

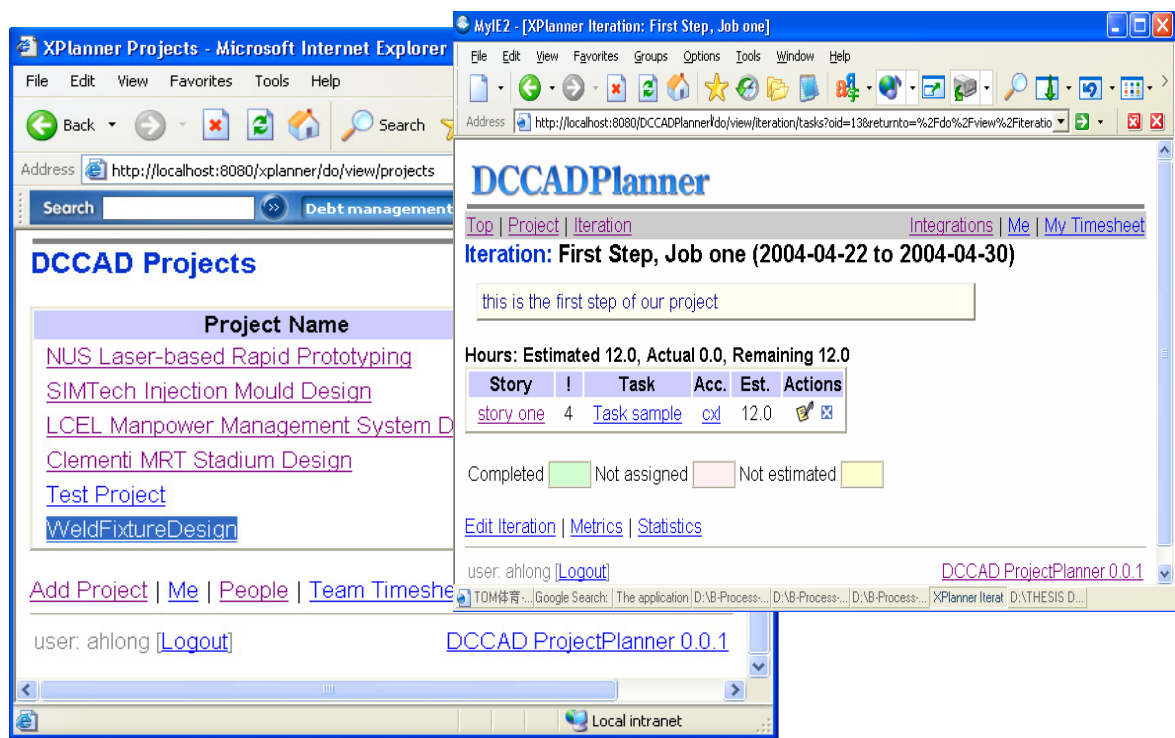


Figure 5.15 Snapshots of Web-based Project Planning UI

Figure 5.15 shows the DCCAD project management Web pages. In the pages, project participants can login to the pool with different authorization priorities. The project organization structure may differ from user to user. But the basic elements of the project are similar. A project consists of lots of sub-tasks, which may include multiple tasks or

plans. And users are assigned with different tasks with a pre-estimated process schedule. With different implementations, notifications and online discussing functionalities are suitable to be designed in this module.

5.4 Summary

This chapter presents the implementations of the proposed DCCAD research. An application-independent architecture is given for distributed collaborative design system integration and development. Based on the DCCAD communication protocol prototype and DCCAD Fundamental Kernel, AutoCAD ObjectARX based Plug-in and a MS Excel Plug-in are used to prove the methodology and framework developed in the research. A simple Java Servlets based project management module and a process modeling system prototype system are implemented to illustrate the components and functionalities required in a collaborative design environment. A collaborative design case study is given to illustrate the collaborative design communication protocols and co-ordination mechanisms designed in this work.

Due to the limitation of time and programming capability, the case study only demonstrates a simple collaborative scenario between AutoCAD and MS Excel, and with design interactions toward optimized outcomes between two collaborators. Based on the abstracted adaptable DCCAD module, more advantages can be drawn by the implementation of two design applications with integration of collaborative systems, such as PDM-based SMARTTEAM CATIA and Arena PLM-based collaborative design system. Thus more collaborative design activities can be analyzed and tested based on the adaptable model proposed.

Chapter 6. Conclusions and Future Works

6.1 Contributions of the research

The thesis presents the proposal and development generation of an adaptable model for distributed collaborative mechanical design. This is intended to provide a more in-depth understanding of the mechanisms involved in collaborative design work. The resultant prototype can be used as a solution for the development of design scenarios that are sensitive to group collaboration and co-ordination. The study covers the covert and tacit features of the work. In developing a technology to support designers, this study is required for the development of an appropriate technology to augment collaborative work practices. In this thesis, a prototype has been developed and demonstrated to show how groups of designers interact with one another, with their environments, and with the representations of design to perform problem solving. The approach adopts adaptability-oriented methods to explore the activity on the design that is involved in the co-ordination.

The research demonstrates that collaborative design is a vastly complex area. It is inherently multi-participant, and involves the use of multiple tools. The results of the research provide a description of what is needed in the case of design collaboration. It also indicates the research status and key problems. The study also demonstrates how the local behaviours and constraints interact in managing the interdependencies between the elements of the functional design system. Through highlighting the mechanisms used to co-ordinate collaborative work, the study reveals areas where a particular technology could be introduced that would increase the effectiveness of the design workers. It

therefore provides some insight when considering CSCW development to critical design co-ordination issues in engineering.

6.2 Issues arising from the research

Product development is idiosyncratic. Different organizations use different tools and applications depending on their industry, expertise, market, and budget. It is not cost-effective for companies to install copies of every application for every user because the IT overhead for installation and support can then be prohibitive. The solution has been to print data for review on paper, or to import and export data between tools. But this introduces the tasks of translation between tools and mediums, which incurs non-productive time. A collaborative tool needs to be flexible enough to work around a variety of tools, facilitating the communication between them, while remaining a cost-effective solution.

Collaboration in product development is not a new idea; designers, managers and manufacturers always need to work together to bring a product from concept to manufacturing reality. But increasing market pressures are driving organizations to work with each other in ways that challenge traditional collaboration systems. Many collaboration applications have sprung up to serve this need, but fall short due to inherent disadvantages for collaboration. A successful collaboration tool must incorporate the following fundamental principles:

- Unified framework for data aggregation that structures and organizes the data captured by the group.

- Inherent data consistency to avoid internal contradictions that degrade the integrity of the data.
- Clear access to information that provides easy access and encourages use of the tools.
- Structured access to information that provides control for what information is made visible to users and who can change that information.
- Change propagation that makes updates visible to the group and implements changes consistently.
- Accommodation of different tools to adapt specific needs.
- Integration with other enterprise applications (PLM/CRM/SCM, etc.)

The operation of groups engaged in problem solving is hard to conceptualize, because of the range and complexity of the factors that are inherent in the activity. There are various types of frameworks proposed by researchers in the past decades. But there has been no single coherent framework with which to examine collaborative design behaviors, integrating individual interaction. Traditional approaches so far either realize synchronous design via thread share, which is created in a strictly identical process, or realize cross-platform data sharing and management. Because of this, they have failed to capture what this research identifies as central features of cross-platform-design-behavior through message mapping. Its concerns are on the representation of information within the system, and the propagation and transformation of these representations in performing a given task. Nevertheless previous researches in collaborative design are synthesised and augmented in the thesis, leading to a better understanding of existing problems.

6.3 Summary of the research

The adaptable framework of distributed design has been developed to provide solution that introduces a new perspective of collaborative design. This module has been designed in a application-platform independent model that enables designers to communicate and collaborate with each other across different operation systems and applications in distributed locations. The adaptable module abstracts common collaborative functionalities for collaborative design and project management.

The main issues for the research are adaptable structure, distributed design management and co-ordination, design process representation, collaborative communication and design visualization. The collaborative design module utilizes extensible and adaptable programming mechanism. This enables the collaborative design plug-in and add-on to be developed flexibly and loaded to different applications in different operation systems. Design representation, design behavior representation and design process representation are described in XML using the UML notation. Design collaboration protocols are defined for communication purposes. Different types of communication and data exchange model are proposed in the adaptable module. This allows users to apply different models of communication with different objectives: whether peer-to-peer or client-server, broadcast or connection. A Java Servlet based project management module is provided for Web-based project management and visualization purpose.

This is one of the earliest adaptable collaborative design modules implemented using a platform-independent framework. The developed adaptable model has been tested and demonstrated with a case study in distributed collaborative design environment.

6.4 Future directions for the research

This thesis basically reached the objectives aimed by research work. But there are some research aspects can be further carried out based on the works done in this thesis.

It would be interesting to carry out more case studies to see how this work would support different applications involved in the product development process, such as PLM, PDM system. The inclusion of other stakeholders, in finance, subcontracting, materials supply, and so on would also give an insight into these relatively unexamined, although potentially critical design areas. In the thesis, these cases were not observed, and it can be seen in the case studies where the stakeholders had a direct link with the division under examination. It would therefore be useful to continue this work in more details in future.

A further research question would be to apply technologies developed from the studies, such as agent technologies, grid computing technologies, and how this would change the distributed collaborative processes of the adaptable system. Further research would help support the findings of the research.

References

- Arena Solutions, Collaborative Tools for Product Development: A New Approach. Arena Solutions white paper, 2002
- Barequet , G., Duncan, C. A., and Gooddrich, M. T., Bridgeman, S. S., and Tamassia, R., 'GEOMNET: Geometric Computing over the Internet', IEEE Trans on Internet Computing, Vol. 32, pp. 21-29, March-April 1999
- Begole J, Struble CA and Shaffer CA, 1997, Leveraging Java applets: toward collaboration transparency in Java. IEEE Internet Computing, 1-2, pp.57-64.
- Bianconi F and Conti P, 2003, Collaborative product modeling in heterogeneous environments: an approach based on XML schema. Proc. 10th ISPE International Conference on Concurrent Engineering: Research and Applications, Portugal, pp.303-310.
- Boland R.J., A.K. Maheshwari, D. Te' eni, D.G. Schwartz & R.V. Ramkrishnan (1992). Sharing perspectives in distributed decision making. *CSCW'92 Proceedings*. ACM Press: 303-313.
- Caldwell N.H.M., Rodgers P.A., WebCADET: facilitating distributed design support, in: Proceedings of the IEE Colloquium on Web-based Knowledge Servers, London, UK, 1998, pp. 91-94.
- Chen Li, Tingjin Wang, and Zhijie Song, a web-based product structure manager to support collaborative assembly. *Journal of Computing and Information Science in Engineering*, March 2004 Volume 4, Issue 1, pp. 67-78
- Christopher D.C Christopher D. Cera, William C. Regli, Ilya Braude, Yuri Shapirstein, A collaborative 3d environment for authoring of design semantics, *IEEE Computer Graphics and Applications*, Special Issue on "Computer-Aided Design", 22(3): 42–55, 2002
- Cutkosky, M.R., Engelmores, R.S., Fikes, R.E., Genesereth, M.R., Gruber, T.R., Mark, W.S., Tenenbaum, J.M., and Weber, J.C., "PACT: An Experiment in Integrating Concurrent Engineering Systems", *Computer*, V26, No.1, January 1993, pp28-37.
- David Hausheer, Burkhard Stiller: Design of a Distributed P2P-based Content Management Middleware. In Proceedings of 29th Euromicro Conference, IEEE Computer Society Press, Antalya, Turkey, September 1-6, 2003.
- Denise J. Ecklund, Earl F. Ecklund Jr., Robert O. Eifrig, and Fred M. Tonge. DVSS: A Distributed Version Storage Server for CAD Applications. In Proceedings of the

13th International Conference on Very Large Databases, Brighton, England (VLDB '87), pages 443--454, September 1987.

- Doyle J., A truth maintenance system. *Artificial Intelligence* 12 231—272, 1979.
- Mervyn F., Kumar A. Senthil, S. H. Bok, Nee Andrew Y. C.: Development of an Internet-enabled interactive fixture design system. *Computer-Aided Design* 35(10): 945-957 (2003)
- Fleming Andrew, Angela Lee, Rachel Cooper, Ghassan Aouad, the development of a process mapping methodology for The Process Protocol Level 2. *Proceeding: Product and Process Modelling in Building and Construction*, Gonçalves, Steiger Garcao & Scherer (eds) 2000 Balkema, Rotterdam
- Franca Giannini, Marina Monti, Domenico Biondi, Flavio Bonfatti, Paola Daniela Monari: a modeling tool for product data management in a collaborative design environment, *Computer-Aided Design* 34 (2002) 1063-1073.
- Gilles Falquet and Claire-Lise Mottaz Jiang. Conflict Resolution in the Collaborative Design of Terminological Knowledge Bases In *proc. EKAW 2000 (International Conference on Knowledge Engineering and Knowledge Management)*, LNCS, Springer-Verlat, March 2000.
- Gilles Falquet and Claire-Lise Mottaz. A Model for the Collaborative Design of Multi-Point-of-View Terminological Knowledge Bases In R. Dieng and N. Matta (Eds) *Knowledge Management and Organizational Memories*, Kluwer, 2002.
- Gorti S. R., Gupta A., Kim G. J., Sriram R. D., and Wong A., An Object-oriented Representation for Product and Design Processes *Computer-aided Design*, V30(7), P487-501, 1998
- Han, J. H. e Requicha, A. A. G., 1998, Modeler-independent Feature Recognition in a Distributed Environment. *Computer-Aided Design*, 30(6), p453-463
- Hartly J, *Concurrent engineering*. Cambridge, MA: Productivity Press, 1992.
- Hirokazu Takizawa and Rieti, Coordination Costs and the Optimal Partition of a Product Design, *RIETI Discussion Paper Series 03-E-014*, June 2003
- Huang, G.Q. K.L. Mak, Web-based morphological charts for concept design in collaborative product development, *Journal of Intelligent Manufacturing* 10 (3/4) (1999) 267-278.
- Huang G.Q., S.W. Lee, K.L. Mak, Web-based product and process data modeling in concurrent "design for X", *Robotics and Computer-Integrated Manufacturing* 15 (1) (1999) 53-63.

- Huang, G.Q K.L. Mak, WeBid: a Web-based framework to support early supplier involvement in new product development, *Robotics and Computer Integrated Manufacturing* 16 (2/3) (2000) 169-179.
- Huang G.Q., J. Huang, K.L. Mak Agent-based workflow management in collaborative product development on the Internet *Computer-Aided Design* 32 (2000) 133–144
- Hutchinson, J., Koch, C., Luo, J., and Mead, C. (1988). Computing motion using analog and binary resistive networks. *IEEE Computer magazine*, 21:52—64
- Hutchins, E. 1995 *Cognition in the Wild*. Bradford: MIT Press.
- I.Budianto J.Olds, A collaborative optimization approach to design and deployment of a space based infrared system constellation, *Proceedings of 2000 IEEE Aerospace Conference* 18-25 March 2000 Bigsky, Montana.
- Johansen, R. *Groupware: Computer support for business teams*. New York: The Free Press, 1988.
- Karthik Ramani, Abhishek Agrawal, and Mahendra Babu, CADDAC: Multi-Client Collaborative Shape Design System with Server-based Geometry Kernel, *Journal of Computing and Information Science in Engineering*. 2003 Volume 3, Issue 2, pp. 170-173
- Kim K.Y, PhD thesis, "Assembly Operation Tools for e-Product Design and Realization", University of Pittsburgh, 2003.
- Cheng K., Pan P.Y., D.K. Harrison, Web-based design and manufacturing support systems: implementation perspectives, *International Journal of Computer Integrated Manufacturing* 141 (1) (2001) 14-27.
- Lee J Y, H Kim, S B Han, Web-enabled feature-based modeling in a distributed design environment. *Proceeding of DETC99/DFM-8941*.
- Lee Rong-shean, Yuh-min chen, Hsin Yucheng and Mindah Kuo, A framework for a concurrent process planning system for mold manufacturing, *Computer Integrated Manufacturing Systems* Vol.11 No.3 P.171-190, 1998
- Li J, S M Gao, X Zhou: Direct incremental transmission of boundary representation. *Symposium on Solid Modeling and Applications 2003*: 298-303
- Li WD, Ong SK and Nee AYC, 2002, Establishment of a distributed design environment. *Proc. 9th ISPE International Conference on Concurrent Engineering: Research and Applications*, UK, pp.605-612.
- Li W.D, Y.Q. Lu and W.F. Lu, Collaborative Computer-Aided Design – Research and Development Status, *Proceeding of CAD2004 at Pattaya, Thailand*.

- Liu, Q., Tay, F.E.H., Tan, K.C., Kwong, K.K., (2000) CyberCAD: Internet Distributed Interactive Collaborative Design, Proceedings of the Internet Workshop 2000, Tsukuba, Japan
- Lu S. C-Y., A methodology for collaborative design process and conflict analysis. CIRP Annals 2000.
- Eggersmann M., Gonnet S., Henning G.P., Krobb C., Leone H.P., Marquardt W. Modeling and Understanding Different Types of Process Design Activities, Latin American Applied Research 33:167-175 (2003)
- MacLean A., Young R.M. and Moran T.P.. Design Rationale: the argument behind the artefact. *CHI'89 Proceedings*. ACM Press: 247-252. 1989
- Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A. and Cohen, M.D., "Intelligent information-sharing systems", *Communications of the ACM*, **30** (2): 390-402, 1987.
- Malone, T.W., Lai, K.-Y. and Fry, C., "Experiments with OVAL: A radically tailorable tool for cooperative work", *ACM Transactions on Information Systems*, 13 (2): 177-205, 1995.
- Malone, T.W., Lai, K. and Grant, K.R., "Agent for information sharing and coordination: A history and some reflections", *Software Agents*, edited by Bradshaw, J.M, AAAI Press, 109-143, 1997.
- Marr D., Vision. W.N. Freeman & Co, San Francisco, 1982
- Muller M.J. Retrospective on a year of participatory design using the PICTIVE technique. *CHI ' 92 Proceedings*. ACM Press: 455-462. 1992
- Nam TJ and Wright DK, 1998, CollIDE: a shared 3D workspace for CAD. Proc. 1998 Conference on Network Entities, Leeds, UK, pp. 389-400.
- Nee, A.Y.C., Whybrew, K. and Senthil Kumar, A. (1995), Advanced fixture design for FMS, Springer-Verlag, c1995.
- Nicola Senin,, Roberto Groppetti, David R. Wallace, Concurrent assembly planning with genetic algorithms, *Robotics and Computer Integrated Manufacturing* 16 (2000) p65-72.
- Nitin Sawhney, Saul Griffith, Yael Maguire and Timothy Prester. ThinkCycle: Sharing Distributed Design Knowledge for Open Collaborative Design. *International Journal of Technologies for the Advancement of Knowledge and Learning (TechKnowLogia)*, Jan 2002, Vol. 4 Issue 1. pp. 49-53.

- Shyamsundar N., Gadh Rajit, Internet-based collaborative product design with assembly features and virtual design spaces. *Computer-Aided Design* 33 (2001) 637-651
- Numata, J., Lei, B. and Iwashita, Y. (1996); Information management for Knowledge Amplification in Virtual Enterprises, *Proceeding of IEMC 96*, p281-285.
- Olsen GR, Cutkosky M, Tenenbaum JM and Gruber TR, Collaborative Engineering based on Knowledge Sharing Agreements, In *Proceedings of the 1994 ASME Database Symposium*, 1994.
- Pahng GDF, Senin N and Wallace D, 1998, Distributed modeling and evaluation of product design problems. *Computer-Aided Design*, v30 (6), pp.411- 423.
- Perry, M. and Thomas, P. (1995) Externalising The Internal: Collaborative Design Through Dynamic Problem Visualisation. *Ancillary proceedings of HCI'95*, Huddersfield, August 1995
- Petrie, C., Webster, T. and Cutkosky, M.R., Using Paretooptimality to coordinate distributed agents, *AI/EDAM*, 9: 269-281, 1995.
- Qi Zhou & Robert J. Krawczyk, From CAD to iAD A Survey of Internet Application in the AEC Industry. College of Architecture, Illinois Institute of Technology. ACADIA 2001
- Qiang L., Zhang Y. F., and Nee, A. Y. C., A distributive and collaborative concurrent product design system through the www/Internet, *International Journal of Advanced Manufacturing Technologies* (2001) 17:315-322.
- Rafael Bidarra, Niels Kranendonk, Alex Noort, and Willem F. Bronsvooort, A Collaborative Framework for Integrated Part and Assembly Modeling. *Journal of Computing and Information Science in Engineering*. December 2002, Volume 2, Issue 4, pp. 256-264
- Rajit Gadh and Ratnakar Snothi, Geometric shape abstraction for internet-based virtual prototyping, *Computer Aided Design*, Vol 30, No. 6, PP.473-488, 1998
- Rodgers, P.A., A.P. Huxor, N.H.M. Caldwell, Design support using distributed Web-based AI tools, *Research in Engineering Design* 11 (1) (1999) 31-44.
- Rodgers, P. A., Huxor, A. P, et al, 1999, "Design support using distributed web-based AI tools", *Research in Engineering Design*, Vol. 11, No. 1, pp. 31-44.
- Shen W, J.P. A. Barthes, An experimental environment for exchanging engineering design knowledge by cognitive agents, *Proceedings of the IFIP TC5/WG5.2 international conference on Knowledge intensive CAD volume 2*, p.19-38, January 1997, Pittsburgh, Pennsylvania, United States

- Song, Research Report, CAD Lab, Seoul National University. 2002
- Sprow E, Chrysler's concurrent engineering challenge. *Manufacturing Engineering* 1992, 108(4): 35-42.
- Scherer R. J., Information Logistics for Supporting the Collaborative Design Process, HANNOVER 2000 - Portuguese Pavilion. Monday, 11 September 2000.
- Tay Feh and Roy A, 2003, CyberCAD: a collaborative approach in 3D-CAD technology in a multimedia supported environment. *Computers in Industry*, 52, 127-145.
- Tomomi Kawashima, Ma Jianhua, Hosei University, TOMSCOP - A Synchronous P2P Collaboration Platform over JXTA, 24th International Conference on Distributed Computing Systems Workshops - W1: MNSA (ICDCSW'04) March 23 - 24, 2004 Hachioji, Tokyo, Japan
- Toshiki Mori and Mark R. Cutkosky, Agent-based Collaborative Design of Parts in Assembly Proceedings of DETC 1998 ASME Design Engineering Technical Conferences September 13-16, 1998, Atlanta, Georgia, USA
- Toye, G., Cutkosky, M.R., Leifer, L.J. et al. SHARE: A methodology and environment for collaborative product development. *The Int. J. of Intelligent and Cooperative Inio.* 5~yst. 3, 2 (June 1994), 129-53.
- Van den Berg E, Bidarra R and Bronsvort WF, 2000, Web-based interaction on feature models. Proc. Seventh IFIP WG 5.2 Workshop on Geometric Modeling: Fundamentals and Applications, Parma, Italy, pp. 319-320.
- Wang L, An approach to collaborative design and intelligent manufacturing. In Proceeding of International Joint Conference of SCI'99, vol7 (Industrial Systems), 1999, p. 431-7
- Wang Lihui, Shen Weiming, Xie Helen, Neelamkavil Joseph and Pardasani Ajit Collaborative conceptual design: state of the art and future trends. *Computer Aided design* v34 (2002) 981-996.
- Wallis A, Haag Z. A Muti-agent Framework for distributed collaborative design, In Proc. of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprise, 1998. P828-7.
- Zdenek Zdrahal, Paul Mulholland, John Domingue and Marek Hatala, Sharing engineering design knowledge in a distributed environment, *Journal of Behaviour and Information Technology*, 19(3), pp. 189-200. 2000

Zhou, and Krawczyk, R. (2000). A proposed working model for Internet aided design.
Ph.D. proposal, College of Architecture, Illinois Institute of technology.

Websites:

- [1] www.collabcad.com
- [2] www.alibre.com
- [3] www.onespace.com
- [4] www.autodesk.com
- [5] www.cocreate.com
- [6] www.ptc.com
- [7] www.hoops.com
- [8] www.realitywave.com
- [9] www.webex.com