Development of an Online Collaborative Working Environment for Design and Manufacturing

Yu Xiong

A report submitted in partial fulfillment of the requirements of Nottingham Trent university for the degree of Doctor of Philosophy

This research programme was carried out in the Advanced Design and Manufacturing Engineering Centre, School of Architecture, Design and Built Environment, Nottingham Trent University, Nottingham, UK

March, 2008

Content

Nomenclature

DCOM	Distributed Component Object Model
EAI	Enterprise Application Integration
EJB	Enterprise JavaBeans
GUI	Graphical User Interface
HTML	HyperText Mark Language
HTTP	HyperText Transfer Protocol
J2EE	Java2 Enterprise Edition
JAXR	Java API for XML Registries
JSP	Java ServerPages
JDBC	Java Database Connectivity
ODBC	Open Database Connectivity
RMI	Remote Method Invocation
SMEs	Small and Medium Sized Enterprise
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SMEs	Small and Medium Sized Enterprises
UDDI	Universal Description Discovery and Integration
URL	Uniform Resource Locator
VE	Virtual Enterprise
VMI	Vendor Managed Inventory
VO	Virtual Organization

WSDL	Web Services Description Language
WSRF	Web Service Resource Framework
WWW	World Wide Web
XML	Extensible Markup Language

Abstract

This research is to develop a novel collaborative working environment (CWE) for manufacturing and design using advanced Web/Internet technologies such as Web Service, Grid Service and other related software tools/packages. To achieve the above, the following research modules are developed by the author:

A service oriented framework for computer aid design, which acts as an online collaboration system, has been developed with the utilisation of the latest technology, Web Service. The concept of Service-Oriented Architecture has been implemented in the framework. Users from anywhere in the world can join the design process from their PCs, no matter what operation system they are using. The service-oriented system has the capability of going through firewalls and can afford multi-users due to the characteristics of Web service. Also the loose-coupling structure makes the system very easy to be updated.

Another module for the CWE is to solve the software sharing problem when the platform is used among several geographically dispersed users or organisations. A software package bank system has been developed, which utilised the ideology of service oriented approach and successfully solved traditional problems in this field.

Based on the outcomes mentioned above, the research finally developed a more powerful infrastructure using Grid service, which is a further development of Grid computing and Web service. The Grid service is considered to be the most important future solvent for Internet. Since it is based on Web service, the new Grid-enabled system would integrate all the aforementioned Web service systems, which enable

the CWE to be a high-performance, powerful, flexible and robust environment for collaboration. As a case study, a Grid service based system utilising the CWE for inventory management has been developed, which proves the feasibility of the novel approaches developed by this research.

Acknowledgement

I would like to express my sincere gratitude to my Director of Studies, Professor Daizhong Su, whose endless guidance, support; encouragement inspired confidence in me and greatly smoothed the way through my study.

Sincere thanks are due to my supervisors; Ms Martin Philips, Dr. Peter Fitzgerald and Dr. Ian Solomonides for their advices, comments, backing and interests in my work.

I would like to thank the EU-Asia Link Project for financial support which made this research possible and enjoyable. And also, thanks for the Chinese Education Ministry for the Chinese Government Awards for Outstanding Self-Financed Chinese Students Aboard to recognize this research.

I have received a tremendous amount of support from administrative and technical staff from Nottingham Trent University, during the course of my research work, for which I am sincerely grateful. There are, however, certain persons who deserve a special mention; Miss Julie Bradshaw, Miss Melanie Bentham-Hill, Ms Doreen Corlett, Dr Philip Breedon and Mr Gary Griffiths.

Special thanks to all my colleagues and friends who made these years memorable for me. At last, I would like to thank my father, Prof. Zhongkai Xiong, and my mother, Ms. Fuling Wu, for their immense love and supports.

Publications

Book Chapter:

Yu Xiong and Daizhong Su**,** Resource Packing to Fit Grid Infrastructure, Lecture Series on Computer and Computational Sciences, 2, ISBN 90 6764 425 0, VSP/Brill Academic Publishers, The Netherlands

Journal Papers:

1. **Yu Xiong** and Daizhong Su, Collaborative Working Environment for Inventory management. International Journal of Technology Managemen**t,** (Invited Paper, accepted)
2. **Yu Xiong** and Daizhong Su, Wrapping Legacy Applications into Grid Services: A Case Study of a Three Services Approach, Lecture Notes on Computer Science, Volume 3865, July, 2007, pp520-pp529
3. Daizhong Su, **Yu Xiong**, Shuyan Ji and Yongjun Zeng, Framework for a Collaborative Working Environment, International Journal of Production Research, Volume 49, No. 9 May, 2008 pp2363-2380
4. **Yu Xiong** and Jiajia Nie, A Cooperated Adversing Investment Model in Assembly Supply Chain with Multiple Suppliers. International Journal of Learning and Intellectual Capital, (Accepted))
5. **Yu Xiong** and Daizhong Su, A Web-Service Based Approach for Software Sharing, Lecture Notes in Computer Science, Volume 3865, Feb 2006, Pages 215 – 224
6. Daizhong Su, Jiansheng Li, **Yu Xiong**, Yongjun Zheng, Collaborative Design and Manufacture Supported by Multiple Web/Internet Techniques, Lecture Notes in Computer Science, Volume 3865, Feb 2006, Pages 483 – 492

Conference Papers

1. **Yu Xiong**, Ji Liu, Fitzgerald Peter and Daizhong Su, Development of a Service-Oriented Software Package Bank, 9th International Conference on Computer Supported Cooperative in Design, 24-26 May, 2005, Coventry, United Kingdom
2. Wenjie Peng, **Yu Xiong**, Shuyan Ji and Daizhong Su, Development of a Virtual Research Institute using Web/Grid techniques, 8th International Conference on Computer Supported Cooperative in Design, 26-28 May, 2004, Xiamen, China.
3. Daizhong Su, **Yu Xiong**, Shuyan Ji, Yongjun Zeng and Jiansheng Li, EU Asia IT&C And Asia-Link Programmes Supported Research In Web-Based Collaborative Work Environment and Its Future Development, Proceeding of International conference on Advanced Design and Manufacturing, 8th-10th, Jan, 2006
4. **Yu Xiong**, Daizhong Su and Jiajia Nie, A Cooperated Advertising Investment Model in Assembly Supply Chain with Multiple, Suppliers, Proceeding of International conference on Advanced Design and Manufacturing, 8th-10th, Jan, 2006
5. **Yu Xiong** and Daizhong Su, A Grid Based System for a Manufacture/Retailer Virtual Organization, Proceeding of International conference on Advanced Design and Manufacturing, 8th-10th, Jan, 2006

List of Figures

Chapter 1

Background and Introduction

In this chapter, the background information is introduced first, highlighting the needs for carrying out the research; and then, the introduction to the project and the layout of the thesis are given, providing necessary information about the research presented in this thesis.

1.1 Background of the research.

Internet technologies has revolutionary significances over the industry in human history. Internet technologies have influence on many aspects of human life, create new living, new working style, new strategy for country, and new economic.

This research is seeking to apply advanced Internet technologies to develop a Web enabled environment for manufacturing industry to enhance their capability by utilizing a wide range of information technology. The information technology for manufacturing enterprises has been developed rapidly, which can be categorized into three groups: the first is related to production information, which is mainly about CAD (Computer Aided Design)、PDM(Product Data Management) and the application of related technologies, providing a solution to “what to do”, The second is related to manufacturing information management includes MRP (Material

Requirement Planning)、MRP II (Manufacturing Resource Planning) and ERP (Enterprise Resource Planning) software of information management, which helping enterprises improve the efficiency of management to answer the question “when to do and where to do”; The third is related to the technique information, which includes CAPP（Computer Aided Process Planning） system and MPM（Manufacture Process Management） system to solve “ how to do”.

CAPP technology produce the production process recur to group technology and expert systems by logical judgment and rationale. Intellectualized CAPP system can inherit and study the expert's experience and knowledge to direct design process; it can remedy the shortcomings of the skilled and experienced experts.

Attaining information such as information of customer in time is a very important function of designing products and their future sells, if the information of customer received is not enough, it is then unsuccessful because of the market for the products is not enough. Information technology can help manufacturers to get the customer information, for example, the survey to the potential customers through Internet. How much information could be attained greatly depends on how advance the information system and the adopted mechanism are.

Information technology not only helps enterprises in the business management to improve their capacity of many aspects, but also can help enterprises to optimize flow management, change business mechanism, constructs a operation platform with highly efficient coordination to support each part of business such as production, research, development, sales, logistics and services, and more important, it can achieve a higher level of supporting innovation strategy of products.

Information technology such as CAD, CAPP, CAM and PDM rescuing staff of design and research department from multifarious handiworks, then they can devote more time and energy to the innovation and development of products, because these fast and efficient tools, information, knowledge and technology were injected into products, the adding value of products increases faster.

Enhancing design, production and sales by integrating Internet technology with information technology is urgently needed, due to the major reasons listed below:

First, competition and cooperation become the main objective of sustainable developing of enterprises in the 21st century. It is the trend of development of economic globalization and informatization, also is the result of surpassing traditional competition theory and competition pattern.

Second, the pattern of individual competition among enterprises is changing to the pattern of collective competition among supply chains or alliances. The focal point of competition is how to achieve operational mode of cooperative business, mutual trust and win-win mechanism, maximize profit and value of enterprises.

Third, some of large enterprises become international enterprises, from national, partial and imperfect competition tends to international and multifarious competition, and looking for space of more competition, coordination and profit.

Fourth, until now most enterprises still use single system or stay on the information isolated island, have not created the SCM (Supply Chain Management) system which is based on cooperative business, so it is hard to configure resources of enterprises even estate chains and support development of enterprises, and is short of responding from technology.

This section is concluded that, many technologies such as CAD/CAPP have been utilized to aid manufacturing management so as to align the different enterprise in a supply chain to be an alliance, and to increase their competition ability; however, still many enterprises have not been informationized because they found difficulties in utilizing the technology or felt the limitation of the technologies. The technology to give enterprise more facilities is necessary, so that more and more information isolated islands could be connected.

1.2 Introduction of the research.

This research is related to two projects supported by the EU (European Union): Asia Link programme [1] and Asia IT&C programme [2] respectively. The former is for collaboration in postgraduate research student training and the later is for the development of a Web-enabled intelligent environment for collaborative design and manufacture, both of which aim at enhancing research collaboration between the EU and China. The consortium members include two Universities and one company from China, and four universities and three companies from the EU states (UK, Finland, Spain, Slovenia and Romania). As part of the two EU Asia projects, this research is related to develop a Virtual Research Institute (VRI) for the Asia Link Project and a Web enabled working environement for the Asia IT&C project, in order for geographically dispersed research teams, universities and industrial partners to collaborate over the Internet. The outcome of the research in VRI is utilized to implement a Collaborative Working Environment (CWE) to support multi-discipline collaboration. CWE is the ultimate outcome of this PhD research.

With the rapid development of Web/Internet techniques, several virtual institutions have been developed, such as the virtual school developed by Malalasekera et al [3] for delivering a Master of Science degree course in automotive systems engineering, the Virtual Learning Portal at The Nottingham Trent University (TNTU) in the UK [4] and the virtual learning environment at Lappeenranta University of Technology in Finland [5]. Comparison of software development tools for such purposes has been given by [6]. Different from the existing virtual institutions/schools, the VRI in this project is not only for virtual teaching and learning, but also is much research related, and, in particular, an Internet based platform for geographically dispersed teams to conduct collaborative tasks is provided. Web/Internet based collaboration has been an important research area and has attracted researchers' great attention. For example, Yoo and Kim conducted research in Web-based knowledge management for sharing product data in virtual enterprises [13], Krebs and Ionescu developed the DISCIPLE System for Collaboration over the Heterogeneous Web [14], and McKinley, et al, reported their development of Pavilion, a middleware framework for Collaborative Web-Based Applications [15]. The authors' research team has been actively involved in this research area with number of achievements, such as a CGI (common gateway interface) approach developed for remote execution of a large size software package [7,8], a prototype electronic catalogue built which applies JDBC techniques and creates CAD files dynamically [9], and a Web-supported system for collaborative design [10]. Those existing systems and achievements enhanced the utilization of existing techniques for Web/Internet based collaboration. However, they have limitations: some of them are designed for information sharing only, some are just used for applications based on browser, and some are ineffective and not user-friendly for information exchange between clients and the server. According to the literature survey, there has not been any effective powerful Internet based collaboration system covers the total production process with integration of clients'/designer interaction, supplier/manufacturer relation, CAD/CAM/CAE, design and manufacture; and, in addition, considering the interactions of all parties involved in the total process, including users/customers, industrial partners and research institutions. The Internet based collaborative platform being developed for the VRI is aimed to overcome the problems.

In the development of the VRI, NTU is responsible to two modules of Virtual Research Institute, which are actually the fundamental of all distributed systems:

(1) Software package bank: This is to store software packages contributed and shared by the consortium members.

(2) A platform for online collaboration: this will enable the project partners to collaborate online for certain research tasks.

The two modules can achieve important features of distributed system, interaction and sharing, which can finally serve a Collaborative Working Environment (CWE). CWE is a platform which is expected to integrate a wide range of applications, research outcomes and resources. Chapter 3 has more specific introduction about CWE. At certain stages of the research, the author intentionally collects information from China, which is because the research is based on a collaboration project with partners in China.

Based on the introduction and background previously mentioned, this research is seeking for answers to the following questions:

1. What are the current disadvantages in traditional solutions in distributed system information sharing? How could the new approach solve the problems?

2. What are the disadvantages for traditional solutions of distributed interaction? How the new approaches will address the disadvantages.

3. How to make the approaches work together on distributed design and manufacturing?

4. How to validate the system which integrating all the approaches in real industry environment?

1.3 Aims and Objectives

Aims:

- Develop an approach for Grid-enabled resource sharing and interaction.
- Establish an Internet-based Collaborative Working Environment with case study.

Objectives:

- Review Web enabled technologies and tools.
- Evaluate the existing Web-techniques and methods developed for the proposed research and integrate appropriate ones into the infrastructure.
- Develop a new online collaboration environment with the most advanced and latest techniques. This module will assist the designer dispersed all over the world collaborate with each other.
- Review the existing software sharing and legacy code policy.
- Implementation of the policy, which solve the legacy code copyright and software sharing problems.
- Design and implement a framework for Collaborative Working Environment with Case Study.

1.4 Structure of the dissertation

The dissertation consists of 9 chapters. Brief descriptions of chapter 2 to chapter 9 are given below.

Chapter 2 - Research Methodologies

This chapter will introduce how the author intends to carry out this research which involves the evaluation of different research methods and general review of the researches and decide which research methods should be adopt based on the nature of this research..

Chapter3 Review of Distributed Technologies and the Research in Collaborative Working Environment

This chapter reviews the possible distributed technologies, compares their features and finds out their advantages or disadvantages, and then, decides to use which technology, and explains why the author intend to use this specific technology. Also, the existing CWE systems are reviewed and the latest approaches are identified for utilization in this research.

Chapter 4 - A Service Oriented Framework for Computer Aid Design

A Service-Oriented Framework for Computer Aid Design has been developed to enable geographically dispersed team members to collaborate online over the Internet. The advanced technique of Web Services has been utilized for the users to provide and to request the services, and the most popular CAD software AutoCAD is used as vehicle to illustrate the methods developed which are in principle applicable for other CAD software packages such as Pro/Engineer. The new system can be operated in computers located in different locations with different operation systems.

Chapter 5 - Service Oriented Software Package Bank

A Service-Oriented Framework for software package sharing has been developed to enable geographically dispersed team members to share programmes online over the Internet. The advanced technique of Web Services has been utilised for the users to provide and to request the services. The new system can satisfy the requirements of copyright problem, maintenance and other issues. The user of the software can get description files and build their own client application to access the shared application via Internet. This means that the software owners can share their applications with geographically dispersed members of Asia-link project efficiently without caring about the issues existing in the traditional way. The layout and concrete implementation will also be discussed in this chapter.

Chapter 6 - Wrapping Computer Resources to Fit Grid Infrastructure

To make the previous mentioned technology adapt the latest Grid architecture, *Web Service Resource Framework* has been applied and a method has been developed for packing existing resources, including different types of software, hard disk spaces and other devices, to fit the Grid architecture. The later part of this chapter describes the implementation of the theories and shows a real example to convert a standalone, DOS environment single user application into a Multi-User, interface-friendly and Network supported grid resources/Services.

Chapter 7 - Development of a collaborative Working Environment using Grid Service Technology

To utilize the advantages of existing and emerging Internet techniques and to meet the demands for a new generation of collaborative working environments for the Education/Business use, a framework with an upperware-middleware architecture is proposed and implemented, which consists of four layers: resource layer, middleware layer, upperware layer and application layer. The upperware contains intelligent agents and plug/play facilities; the former coordinates/controls multiple middleware techniques such as Grid computing and Web-services, while the latter are used for the applications, such as previous developed online Computer Aided Design, to plug and loose couple into the system. The method of migrating legacy software using automatic wrapper generation technique is also integrated. This CWE integrated the previous research done by the author and finally implemented and described in this chapter.

Chapter 8 - Grid Service Based Collaborative Working Environment for Inventory management

In this Chapter, a case study is presented which is supported by Grid infrastructure CWE. This system could enable huge numbers of manufacturers and retailers all over the world to cooperate with each other within a Virtual Organization by sharing inventory information. This system supports two models of inventory management so that the Quality of Service could be enhanced. With the help of switching between the two models, manufacturers and retailers could be facilitated when they are in a changing business environment. With the support of Grid computing, the system is more sophisticated and has more advantages than any other systems by more accurate calculating and more quick response. This system is also extensible to integrate more functions from multiple disciplines, compatible to the most advanced existing technologies, and adaptable to any future technology.

Chapter 9 – Conclusions and Future Work

This chapter concludes the research reported in this thesis, by giving a brief summary , highlighting its novel feature and major contribution, as well as proposed further work .The aims and objectives highlighted in this chapter provide the evidence for the next chapter on carrying out the research methodology.

Chapter 2

Research Methodology

To achieve the aims and objectives of this research stated in Chapter 1, a set of research methods for this research need to be identified, un-proper methodology may mislead the research from the aims and objectives. So in this chapter, the author will first state the importance of the role of research methodology in this research. And then align the research content with the proper research methods to draw out the whole research methodology of this research.

2. 1. Importance of Research Methodology in this research

The aim of a PhD course is not only to contribute to a research area, but also to understand a way to do research and to get the capability to conduct future research. Therefore the research methodology is as important as the research itself.

It may be true, as McGrath [4] stated that it is not possible to do an unflawed study. Any research method chosen will have inherent flaws, and the choice of that method will limit the conclusions that can be drawn. So to conduct a research, multiple research methods should be evaluated and some of them will be eventually adopted to utilize their advantages and avoid the disadvantages.

This chapter shows how the author intends to carry out the research and achieve the objectives of this project stated in Chapter 1. The following paragraphs explained why the research methodology is necessary.

(a). The author needs to consider how the system is designed and eventually implemented because the method affects the results. A result is convincing only when the research methods are properly chosen. For instance, from a technological view point, if Java is chosen rather than C++, the outcome of your achievement will be different, because they place emphasize on different users and operation systems. If you choose to implement an object-oriented system rather than process-oriented system, the influence of the system will be also different (and more widely accepted). Knowing how the system was developed helps the reader evaluate the validity and reliability of the results, and the conclusions drawn from them.

(b). Often there are different methods that the author could use to investigate a research problem. In the system design and implementation, one could develop the whole system then test the whole system and find out the problems one by one, or one could develop the system module by module, and then test the system when a module is developed. The methodology should make clear the reasons why the author chose a particular method or procedure.

(c). The research methods must be appropriate to the objectives of the study. For example, if C language is chosen with its process-oriented system design and programming, of course the system would not meet the requirement of “Cross Platform” and “Universally” features specified for the collaborative work environment which is the major outcome of this research.

(d). The methodology should also discuss the problems that are anticipated and explain the steps taken to prevent them from occurring, and the problems that did occur and the ways their impact was minimized. This test is very essential, when the system is designed, how to testify it? By theory or by case study?

(e). In some cases, it is useful for other researchers to adapt or replicate the methodology, so often sufficient information is given to allow others to use the work.

2.2 Essentials of this research

To make sure which research methodology should be applied, the essentials of this research should be considered. So the following first briefly summarizes this ourline of the research concept, then points out the challenges of this research, and finally reaches a conclusion of which system architecture to adopt.

2.2.1 Outline of the research concept

The theories to build Virtual Research/Collaboration platform have been widely studied and developed. D Makola [1] has brought out a Collaborative Orthopaedic Research Environment project which provides an infrastructure that combines clinical, educational and research activities in a Virtual Research Environment for orthopedic researchers to collaborate in the design, analysis, and dissemination of experiments. A Service Oriented Architecture was employed. Sabina Jeschke etc. [2] have presented a concept for a new type of collaborative working environments for knowledge gain and research, focusing on the field of nanoscience and nanotechnology. Web service was also adopted in the system developed.

The Virtual Research Institute is a virtually provided system to facilitate the online collaboration of researching and teaching. First of all, it is a virtual organization system. It has similar functions and structures to other virtual organization systems. The Virtual Research Institute also enables geographically dispersed research teams to work together. They can share knowledge with each other cross the Internet. The outcome of the research should not be limited to the Virtual Research Institute itself, but be applied in other collaborative working environment.

For bio-pattern analysis, the Virtual Research Institute was utilized to integrate a large number of researchers and doctoral students into the Network to create a critical mass of specialists to identify and address barriers to progress, and to co-ordinate, lead and promote the development and take-up computational intelligence techniques (for instance, about 100 researchers and 85 doctoral students from 25 partners may be integrated at a given time) [5] .

In this research, the online collaboration platform could enable researchers to collaborate simultaneously. The software package bank could enable the inter-organization software sharing, and the concept could be applied to any other resources (hardware etc.). The platform can share resources and connect people from different locations all over the world. The platform should be universal and the fundamental framework could be utilized by any other research domain, such as teaching, research, business, manufacturing etc.

2.2.2 The research challenges

Because of the above concept and the requirements from the projects, the following requirements need to be met at least:

(a). The online research collaboration requires real time inter-operations.

(b). The cross platform features are required because participants are probably using different operation systems.

(c). The system should be extensible and flexible enough, because there will definitely be many future requirements to be satisfied, in that case, the system should be given the function to integrate future plugins.

(d). The system also needs to support a huge number of users with stability and constancy.

To satisfy all the requirements, not only the technology, but also the system architecture needs to be carefully selected. The existing systems can hardly satisfy all the requirements and many of them are limited by traditional technologies, this will be reviewed in later chapters.

2.2.3 The system architecture

The reality in IT enterprises is that infrastructure is heterogeneous across operating systems, applications, system software, and application infrastructure. Some existing applications are used to run current business processes, so starting from scratch to build new infrastructure is not an option [9]. In the quickly changing business environment, new business requirement, new business model, new produce/customers etc. require new supporting functions from the system. An ideal system should support the use of current applications and in the meantime accommodate any un-predictable future functions. SOA (Service Oriented Architecture) with its loosely coupled nature allows enterprises to plug in new services or upgrade existing services in a granular fashion to address the new business requirements, provides the option to make the services consumable across different channels, and exposes the existing enterprise and legacy applications as services, thereby safeguarding existing IT infrastructure investments, and this feature is treated as a main module in this research. The case study of the research will use the SOA to solve a SCM (Supply Chain Management) problem, mainly in inventory contract and manufacturing. An enterprise employing SOA could create a supply chain composite an application using a set of existing applications that expose the functionality via standard interfaces.

SOA is both an architecture and a programming model, a way of thinking about building software. On the horizon, however, there are even more significant opportunities [6]. Principally, there is Grid computing, which is much more than a computing solution; it also will provide a framework whereby massive numbers of services can be dynamically located, relocated, balanced, and managed so that needed applications are always guaranteed to be securely available, regardless of the load placed on the system.

2. 3 Research methodology and stages in this research

2.3.1 Consideration of the research methods to be taken

There are many research methods for different research areas, for example, there are the research methodologies for Bio-Medical Research [7], or for Health Science [8] etc. The are also many research methodologies in the research areas of Information system research and development, a large number of research methodologies have been identified by Galliers [3], as shown in the following table:

Laboratory Experiments	Subjective/Argumentative
Field Experiments	Reviews
Survey	Action Research
Case Studies	Role/Game Playing
Theorem Proof.	Descriptive/Interpretive
Forecasting	Futures Research
Simulation	

Table 1.1 Research Methodologies

This research, it includes a development of framework for a system, the author intends to solve the problem theoretically, and prove the theory by laboratory experiments and a Case Study. This is the main approach for the author to carry out the research process. In this case, the following research methods should be applied in this research.

1.) **Reviews** will be conducted because the technology is developing so quickly, it is necessary to know the progress made by other researchers. So before the research is started, the author needs to review the relative researches and gain more related knowledge and ideas for this research.

2.) After a whole picture of the research area is developed, the next step will be to identify the problems and weakness of the current research. The advantages of existing research will be incorporated into this project while disadvantages will be discarded. After that, the author's own theorem to solve the research problems will be formed. Finally, the author will need to theoretically prove his potential solutions. This is called "**Theorem Proof".**

3.) Further to the theory proof, some other work needs to be done. The theory need to be tested by laboratory experiments, and this work should be divided into many periods: (I) Divide the system into modules (in this system, two critical modules are required, which are the solutions to sharing and interaction in a distributed system). (II) For each module, the proper technology need to be studied, this research requires a strong technology ingredient: the proposed structure of module needs to be compatible to future technology and adapt to next generation work. The author may be unlikely to know how to use the technology to build the system beforehand, because this technology maybe a very recently released one. So the study of the software development package is very necessary. (III) After the study of the technology then the laboratory experiments need to be done to testify the theory. (IV) For different modules of the system, the structure may be also different, so, repeat the periods 1, 2, 3. (V) The framework of the system forms when it integrates several research modules; these modules are implemented and tested in laboratory and this step is so called **Laboratory experiment.**

4.) **Case Study:** From the previous steps, outcomes are achieved and the system is developed, but it is necessary to test them using a real application, which is a case study, to show the utility of the proposed system. The case study must be convincing, and the application area should be carefully selected.

2.3.2. Research stages

First, a literature review should be done and a proper research stage should be carried out, and only after that, the preparation of the research could be adequate and convincing. The stages of the research are:

I. Review Web enabled technologies and tools.

II. Evaluate the existing Web-techniques and methods in the areas related to the proposed research and integrate appropriate ones into the infrastructure.

III. Design the framework of the new online collaboration environment and detail its development plan.

IV. Development of online CAD system and design the system components including request module, response module and middleware module.

V. Investigate into the software sharing methods and legacy code policy.

VI. Implementation of the policy to resolve the legacy code copyright and software sharing problems.

VII. Integrate the previous achievements with other functions into CWE framework.

VIII. A Case Study to demonstrate the tools and methods developed and to highlight the research outcomes.

This research starts from reviewing the Web enabled technologies and tools. The research will then evaluate the existing Web-techniques and methods developed for the proposed research and integrates appropriate ones into the infrastructure. After that, a new online collaboration research platform with the most advanced and latest techniques will be developed. The developed application will be tested and evaluated by laboratory experiment. For the laboratory experiment, because new technology will be applied, the author must start from using the basic test example first, then go to develop a simple interaction application, if successful, a more comprehensive development of application will be conducted to accomplish the mission, if the application cannot satisfy the requirements, the research will be kept focusing on fixing the problems until all the requirement are satisfied. After completing the online research platform, the next task which will be to review the existing software sharing and legacy code policy. This implemented method will also be tested and evaluated following the requirements for this module. After the two essential modules of distributed system are designed (sharing and interaction of resources/applications). After this, a whole online collaborative working environment is designed based on the two essential approaches. The system needs to be applied to facilitate different areas. However, the author will concentrate on one area to make a case study to prove the utility of the system. The case study will be conducted on inventory control. According to the research methodology, the next chapter is literature review.

Chapter 3

Review of Supporting Distributed Technologies and the Research on Collaborative Working Environment

There are some potential technologies that could be utilized to develop the system, and they have some similar functions to each other. These technologies include: Java-RMI, CORBA, DCOM, Web Service and Grid Computing. The first three technologies are categorized as traditional technologies, and the later two are recognized as SOA (Service Oriented Architecture) technologies. In the first section of this chapter, the author will review these technologies and compare them to attain the advantages and disadvantages. This will help to get a clear idea on which technology to choose. In the second section, the author will review the current researches of collaborative working environment. These literature reviews provide a direction on how this research will be developed.

3.1 Analysis of possible technologies and introduction of SOA technologies

This section first analyses the drawbacks of current technologies (Section 3.1.1) and then introduces the advanced SOA technologies (section 3.1.2). These technologies are all popular distributed technologies and potentially can be utilized in this research. The author will try to identify the more suitable one(s) after get a clear impression on the advantages and disadvantages of each of the technology.

3.1.1 Evaluation of Traditional Web Technologies

(1). JAVA RMI

Remote Method Invocation (RMI) facilitates object function calls between Java Virtual Machines (JVMs). JVMs can be located on separate computers - yet one JVM can invoke methods belonging to an object stored in another JVM. Methods can even pass objects that a foreign virtual machine has never encountered before. By allowing dynamic loading of new classes as required, Java RMI provides a very easy solution for the talk between server/client [10].

The drawback of RMI technology is that both the Client and Server side need to be implemented using Java Language, if one of the side is using C++ or other language, the invocation will not happen. One of the necessary features of CWE is to support geographically dispersed users who are using different platforms and applications developed by different technology/language, so the CWE should not only support JAVA developed applications. To avoid such a problem, Common Object Request Broker Architecture (CORBA) becomes a choice of developing distributed system.

(2). CORBA

CORBA is OMG's (Object Management Group) open, vendor-independent architecture and infrastructure. It enables computer applications to work together over a network. Using the standard protocol IIOP (Internet Inter-ORB Protocol), it is claimed that a CORBA-based programs from any vendor, on almost any computer, operating system, programming language, and network, can interoperate with a

CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network [11].

CORBA is applicable in many situations, because of the easy way that it integrates machines from so many vendors, with sizes ranging from super computer to handsets. It is the middleware of choice for different kinds of systems. One of its most important uses is in servers, which must handle large number of clients at high hit rates with high reliability. These systems are supported with specialized scalability fault-tolerance. But it is not used just for large applications; specialized versions of CORBA run real-time systems, and small embedded systems.

The drawback of the CORBA is obvious: Firstly, IIOP protocol does not use standard port, and this makes CORBA can hardly go through firewalls. Secondly, the client and serve side are not so loosely coupled so the development of Server/Client using standards from different vendors will cause trouble. This means that different vendors actually are developing their own standards though and this is not the case as the original claim of the CORBA inventor. CWE requires a very flexible infrastructure which is widely compatible to different systems; it also requires a very high loose coupling module to guarantee the influence. CWE is developed to support the next generation network. So the drawbacks of CORBA have greatly limited the influence of CORBA. The other similar technology is DCOM, which is a Microsoft technology;

(3). DCOM

Distributed Component Object Model (DCOM) is a Microsoft technology for software components distributed across several networked computers to communicate with each other [12].

DCOM is a major competitor to CORBA. Proponents of both of these technologies saw them as one day becoming the model for code and service-reuse over the Internet.

However, the difficulties involved in getting either of these technologies to work over Internet firewalls, and on unknown and insecure machines, meant that normal HTTP requests in combination with web browsers won out over both of them. As a Microsoft technology, DCOM cannot support the applications in other platform/Operation System, and this becomes a limitation of the spreads of this technology. It is obvious that DCOM cannot satisfy the requirements of CWE either should cross platform or go through firewalls.

3.1.2 Analysis of SOA technologies

As we can see from the discussion of traditional technologies from section 3.1.1, current/traditional technologies more or less cannot fully satisfy the requirements of CWE (The requirement and characteristic of CWE are discussed in section 3.3 and in Chapter 8). The following two technologies are following Service Oriented Architecture (SOA). SOA is a flexible, powerful and cost-efficient way of building, operating and evolving ICT-intensive solutions for use by business, science and society. SOA has the potential to: 1. builds on existing industry practices, trends and emerging technologies; 2. provides the rules and methods for combining services into an ecosystem that promotes collaboration and self-organisation; 3. brings benefits of increased agility, lower overhead costs and broaden availability of useful services for everybody, shifting the balance of power from traditional ICT players towards intermediaries and end-consumers of ICT [13].

SOA technologies are recent emerging technologies which could provide revolutionary solutions to industry applications. Also, different from traditional Object Oriented technologies (Java, CORBA etc.) SOA solutions are more promising.

The author tries to author that the SOA is the right structure which could be employed to support the infrastructure of CWE. Web service and Grid are both introduced as below.

(1). Web Services

The term Web services [14] describes a standardized way of integrating Web-based applications using the XML（eXtensible Markup Language），SOAP (Simple Object Access Protocol), WSDL (Web Service Definition Language) and UDDI（Universal Description, Discovery, and Integration）open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall.

Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across a network, with descriptions to the interfaces. Following the description, developers can then add the Web service to a GUI (such as a Web page or an executable program) to offer specific functionality to users.

Web services allow different applications from different sources to communicate with each other without time-consuming custom coding. As all communication is in XML, Web services are not tied to any one operating system or programming language. For example, Java can talk with Perl; Windows applications can talk with UNIX applications. So web service becomes an ideal choice to this research. Web service is an emerging but matured technology; it can provide many great solutions to industry applications. However, from a larger and higher perspective, the web service can give solution to specific problems, but cannot renovate the traditional network, though many concepts have been brought forward such as virtual organization, the web service cannot efficiently support those macro-conceptions. The scientists so put a lot of efforts on another technology, Grid Computing.

(2). Grid Technology

Grid Computing is recognized as the next generation network. It brings out concept in current stage. Scientists think that, instead of WWW (World Wide Web), the next generation network will be built under a GGG (Great Global Grid) infrastructure [15]. No specific technology is assigned to implement the Grid. After years of exploration, scientists think that Web Service is the most suitable carrier to support grid and then, finally brought out the official standard for Grid Framework [16].

Origination

The Internet was invented to link all the large, federally funded computing centers, so that the resource sharing will be achieved. When it comes to the terminology of Grid, the same concept is applied, however, extended, to create a more generic resource sharing context.

Fully functional proto-Grid systems date back to the early 1970s with the distributed computing systems project at the University of California, Irvine. David Farber was the main architect. This system was utilized to merit coverage and Cartoon depiction in Business Week on 14 July 1973, however, this technology was finally abandoned in the 1980s as the administrative and security issues involved in having machines that you did not control to do your computation were seem as insurmountable.

Only in the 90s, the father of grid, Ian Foster, Carl Kesselman brought together all the ideas of grid and created Globus Toolkit. Globus Toolkit provides tools include not only CPU management, but also storage management, security provisioning, data movement, monitoring and a toolkit for developing additional services. Additional services are based on the same infrastructure including agreement negotiation, notification mechanisms, trigger services and information aggregation.

Characteristics and Definition of Grid

In short, the term grid has much further reaching implications than the general public believes. While Globus Toolkit remains the de facto standard for building grid solutions, a number of other tools have been built that answer some subset of services needed to create an enterprise grid.

Grid should have the following features:

- Large-scale resource sharing,
- High-performance orientation.
- Collaboration
- Eliminate the Client/Server mode.
- Organize user dynamically, and detect the invalid node automatically.

Grid computing is often confused with cluster computing. The key difference is that a cluster is a single set of nodes sitting in one location, while a Grid is composed of many clusters and other kinds of resources (e.g. networks, storage facilities).

For functionality, one can classify Grids into several types: (1) Computational Grids (including CPU scavenging Grids) which focuses primarily on computationally-intensive operations. (2) Data Grids or the controlled sharing and management of large amounts of distributed data. (3) Equipment Grids which have a primary piece of equipment e.g. a telescope, and where the surrounding Grid is used to control the equipment remotely and to analyse the data produced.

So grid computing is recognized as a middleware, the third generation of Internet, a virtual super computer and a seamless integrated environment for computing and collaboration. Grid computing belongs to distributed Computing.

Though grid has gained above reorganizations, scientists still cannot clearly define it, because different scientists have different view for it: the "father of grid" Ian Foster defines follow A Three Point Checklist [17]:

- Computing resources are not administered centrally.
- Open standards are used.
- Non-trivial quality of service is achieved.

IBM defines Grid Computing as *"the ability, using a set of open standards and protocols, to gain access to applications and data, processing power, storage capacity and a vast array of other computing resources over the Internet. A Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of resources distributed across 'multiple' administrative domains based on their (resources) availability, capacity, performance, cost and users' quality-of-service requirements*"[18]

Grid computing reflects a conceptual framework rather than a physical resource. Grid technology aims to develop an environment in which individuals or institutions can access computers, databases and experimental facilities simply and transparently, without having to worry about where these are located. [19]

From the introduction of Grid, it is obvious that grid can provide strong support to build the CWE in this research. Features like heterogeneous, universal, compatible etc. are just the same as required by the CWE.

3.2 Analysis of current applications of Grid.

In large aircraft company, many designers are designing the same airplane for its different parts; however, each part of the aircraft is very complicated, with hundreds of thousands of accessories. For each of the accessories, it has large amount of parameters. In some cases, the designers need to see the simulation of the aircraft they designed; however, each designers only have the data of the accessories of their own parts, and if they want to view the effect of the whole plane, they got problem, even super computers cannot afford to simulate the whole piece of aircraft based on so many data. Now, grid can take this job. Grid can integrate all the computers and make use of each of them to calculate and accomplish each part of the job. Each CPU of the computer in the grid is involved, each of them are responsible to simulate its own party of job. The screen of the requester will only collect the simulated result of each part and compose them to be a whole plane and display on his screen. For example, the designer of wing simulates only the wing parameters and form a picture in his own computer first; then send the result (the picture) to the requester, just like other designers (for plane head, plane rear), the requester composes all the pictures together and display it on screen. This is illustrated in Figure 3.1:

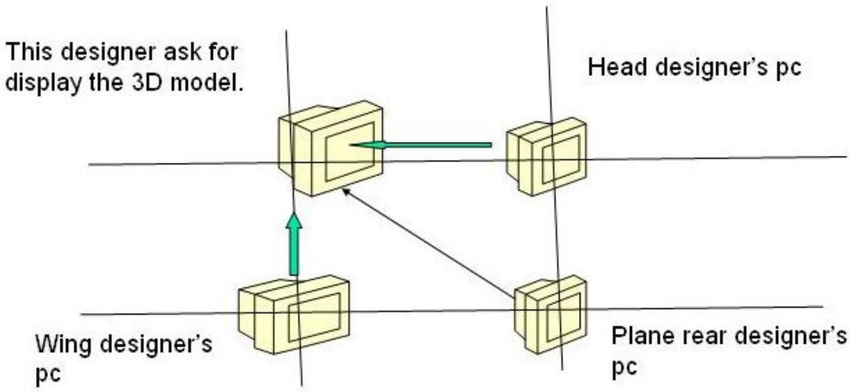


Figure 3.1 Grid Support simulation design

Also, the Grid could enable the equipment to be a node of the infrastructure, as shown in Figure 3.2:

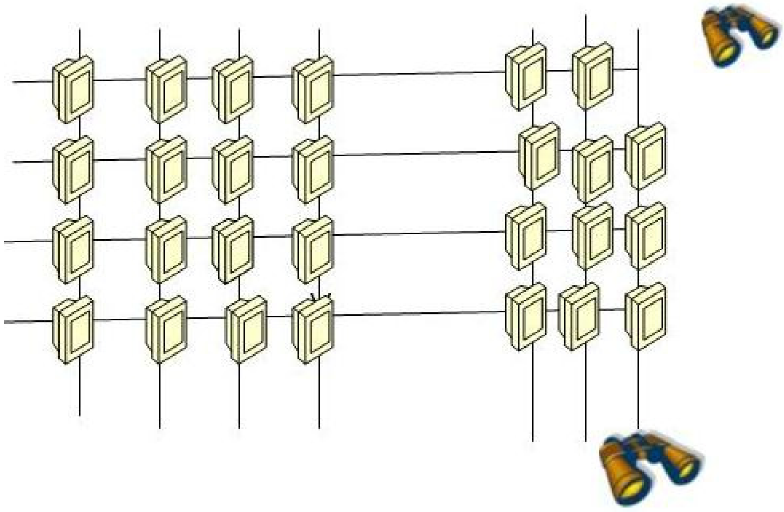


Figure 3.2 Make every astronomer has his "own" virtual telescope

Suppose there is a very valuable and unique astronomy telescope in an astronomical observatory. Usually, the scientists who want to use the telescope need to visit the astronomical observatory, and this costs a lot of time, money and energy. However through using grid technology this problem can be solved as the telescope can link to Grid as a node, and provide the service to authentic users, the user can remotely control the telescope in his screen, and move the mouse to adjust the angle of the telescope and watch the scene just from the user's screen. As shown in Figure 3.2, the telescope is linked into the grid system and is recognized as a node of the grid.

Other typical application of Grid includes the Open Science Grid [20], for large-scale scientific research, TeraGrid [21] to create an integrated, persistent computational resource. ClimatePrediction to predict the weather [22]. Access Grid for large-scale distributed meetings, collaborative work sessions, seminars, lectures, tutorials, and training [23].

3.3 Review on Collaborative systems and working environments

The term CWE [24, 25](Collaborative Working Environment) has many similar terminologies such as CSCW [26, 27] (Computer Supported Collaborative Work) system, WEE (Web Enabled Environment) [28, 29] etc. but all of them are describing one thing, which is simply put by: to help geographically dispersed people/firms to collaborate on tasks over Internet.

CWE are widely studied by many researches and applied in many areas. Mariano Navarro is designing a collaborative platform for working and living in rural areas. They provide a collaborative platform for rural communities; develop a common methodology for rural living lab developments and assessing benefits of results. This plays an assistive role to policy makers, addressing which EU policies are needed for innovation and rural development in 2010.

eProfessional Collaboration Space(ECOSPACE) pursues the vision that by 2012 every professional in Europe is empowered for seamless, dynamic and creative collaboration across teams, organizations and communities through a personalized collaborative working environment. It will design and develop an open standard with SOA, and a collaboration upperware and services, the system will support the creation of new tools and simplify the complexity of collaboration in dynamic work environments and which enable users for creative and knowledge intensive tasks.

Various of other similar systems, which may not explicitly called CWE, have been studied. These systems are used and give tremendous impact in education, healthcare, manufacturing, transportation, retailing, pure services, and even war [30].

Mihaela Ulieru et. al. propose a recursive multi-resolution collaborative architecture, based on multi agent coordination mechanisms as a solid foundation for the development of web-centric cooperative application in global manufacturing. This application is illustrated on a supply chain example [31]. Ke-Zhang Chen et al. have developed a Virtual Manufacturing system to provide a reliable foundation for future

practical manufacturing with a much better prospect of success, a shorter lead time, and a much lower investment cost [32]. Bala Iyer et al. describe an architecture that defines a service-oriented, web service based approach to model management, which could help the decision making process in distributed environment [33]. Lai et al. provides a demonstration of harnessing the power of the Internet to enable collaborative manufacturing in such a setup
where the key components such as the human resource personnel and the manufacturing equipment are geographically separated [34].

With the rapid development of IT and Web/Internet techniques, network supported collaborative working environment (CWE) and related techniques are emerging as a viable alternative to the traditional design and engineering process automation.

Researchers and industrialists have been paying great attention to the research in this area. For example, Yujun and Jinsong [35] introduced a paradigm called Internet-based collaborative product development chain. This chain connects users in geographically different locations to carry out product development activities simultaneously and collaboratively using a series of tools, including product management tools, workflow management tools, and project management tools. Chenga and Fenb [36]develop a prototype of Web-based distributed problem-solving environment to facilitate computer aided engineering technologies. It can also support networked collaboration such that scientists around the world could interactively, visually and experimentally explore their daily design work through the proposed system. Younas, et al [37] presented a protocol in order to improve the efficiency of Web services composition, which was based on the peer-to-peer paradigm, to exploit the capabilities of underlying networks such as the processing carried out at the network nodes. Zhang and Gong [38] design and develop a virtual Geographic Environment(DVGE) system, which is an Internet-based virtual 2D and 3D environment that provides users with a shared space and a collaborative platform, for publishing multidimensional geo-data, and for simulating and analyzing complex geo-phenomena.

More and more research findings have been continuously reported, indicating the substantial progress in this research area. However, our literature survey reveals that although middleware techniques, such as CORBA, Web services and Grid have been utilised in the current CWEs and related applications. The combination of such existing and emerging middleware to form a more powerful and flexible CWE is still a challenge research.

As an important research initiative, the European Union set up the 'Collaborative Working Environments' as one of the strategic objectives for research funding under the FP6 IST programme in 2005. In this programme, EU sponsored 13 projects in a wide range of areas including catalyzing rural development, distributed virtual manufacturing enterprises, virtual engineering for SMEs, digital factory for human-oriented production, strategic innovation, advanced robotic systems, human operators in emergency/disaster scenario and others [39] In the European Union's ICT theme of the 7th Frame Programme [40] it has been stated that *"the current Internet, mobile, fixed and broadcasting networks and related software service infrastructure need to be progress accordingly in order to enable another wave of growth in the on-line economy and society over the next 15 year"*. "*The challenge is to deliver the next generation of ubiquitous and converged network and service infrastructure for communication, computing and media.*" All those demand new collaborative working environments and associated technologies.

As part of the Asia-Link and Asia IT&C projects supported by the European Commission, the research in Web-based collaborative design and manufacture was carried out by the international project consortia led by the author's team. The related work conducted includes the development of a Web-enabled environment [41] online collaborative computer aided design and manufacture [42] effective remote-execution of large size programs [43] and Web-enabled distributed product design [44][45]. All these become essential basis for this research. Among all the essentials to construct an environment for collaborative working environment, sharing, reusability and interoperability of resources are the extremely important; they are the inherent requirement of enterprise operation [46].

3.4 Review on Web Based Collaborative Design

Online Collaborative CAD has been widely studied by researchers in different aspects, a collaborative CAD system needs two kinds of capabilities and facilities: distribution and collaboration. These two terms emphasize the different aspects of a system: physically and functionally. The former separates CAD systems as geographically dispersed and expands them to support remote design activities, and the latter associates and co-ordinates individual systems to fulfill a global design target and objective [47]. In literature, Li et al. summarizes the recently related works from three aspects—visualisation-based collaborative systems, co-design collaborative systems and CE (concurrent engineering)-based collaborative systems. Around these aspects, about 100 papers and 30 commercial systems/international standards published or launched recently are discussed in their publication [48]. They also analyze current researches and development statuses and issues, underlying algorithms, mechanisms and system architectures. In their papers, the future trends and challenges are explained and compared in detail.

From a technical view, there are many related researches utilizing different concept and technique to achieve web based collaborative CAD. But in this research, to achieve the integration in final CWE infrastructure, a SOA approach may be a better choice considering the drawbacks of other technologies. In current researches, many SOA approaches for collaborative CAD related topics have been done. Yan Wang has proposed a new feature-based modeling mechanism—document-driven design—to enable batch mode geometry construction for distributed CAD systems

[49]. In his research, a semantic feature model is developed to represent informative and communicative design intent. Y. Fei et. al. present a collaborative working framework of CAD systems based on web services and described the architecture of the framework in detail. They discussed several key technologies used in their framework including services division method, which guarantees high feasibility in maintaining, and the deploying policy of distribution services for the poise of web servers' burden [50]. However, they are all describing a whole system and those systems may not able to integrate with or to be integrated by other systems.

Apart from Web services, other technologies are also utilized such as agent technology. O. Chira proposes a framework aims to optimise design process operation and management by supporting the dialogue among distributed design actors [51]. Qi Hao et. al. present an industrial case study in the development of a collaborative e-Engineering environment for mechanical product design engineering by applying intelligent software agents, Internet/Web, workflow, and database technologies [52]. Agent technology is very helpful towards a interaction intensive system, from these literatures. They describe a whole system which could utilize the agent smoothfully, however, in this system, it is a very loosely coupled environment, and any function in the environment should be easily added or removed from the system. In that case, the use of agent technology in the systems described in those literatures may not be the backbone to support the CWE which will be proposed by the author. So the main technology to implement the functions in this research will be SOA technology.

3.5 Review on legacy system reusability from Web

As reusability of resources is also a key feature of a Collaborative Environment [46], countless researches have concentrated on giving solution to this issue, Diego Bovenzi et. al. have present an integration approach that exploits black-box techniques for capturing the dynamic and static models of a legacy system user interface, and reproducing them on client devices with the support of a software wrapper. The wrapper is designed to satisfy service stability, data integration, and application integration requirements, and exploits State Chart Diagrams and open standard technologies, such as XML and XSL [53]. This approach gives some hints to the author's research to be conducted, however, the drawback is that, the loose coupling feature is not fully explored and this approach may only support limited applications. If that approach is implemented by strong web services components, the problem will be solved, but that research does give a clue for this research. Y. Zou et.al present a stack of interrelated protocols, such as SOAP, WSDL, and UDDI, to support service integration using the Web as the medium. Specifically, they aim for a framework that allows for the identification of reusable business logic in large legacy systems in the form of major legacy components [54]. Again, this approach is imperfect consider that it only support HTML, and the advantages of web services have not been fully explored, similar researches have been down by Gerardo Canfora's research [55]. Web services do not only support the interoperation between systems (the service part) and resources (the legacy part), but also support the interoperation between services themselves, and this character should put the development of wrapping legacy system onto a more advanced level. Leonid Erlikh [56] has given an overview of the basic concept and a wide knowledge introduction towards this area in a applicable perspective view. Grace Lewis et. al analyze that the specific interactions should be considered, which will be required by the SOA and any changes that need to be made to the legacy components. They have recently helped an organization evaluate the potential for converting components of an existing system into services that would run in a new and tightly constrained U.S. Department of Defense SOA environment. Their paper describes the process that was used and outlines several issues that need to be addressed in making similar migrations [57].

Migration of form based legacy systems towards service-oriented computing is a challenging task, requiring the adaptation of the legacy interface to the interaction paradigm of Web services [57]. Currently, almost all the researches concentrate on how to expose the legacy application, according to this, how the services should be developed. There is hardly a research concentrate on how the web services set should be constructed so that they could serve the legacy system by cooperation.

3.6 Concluding remarks

In this chapter, a review of the fields is presented, which is relevant to the subject of this research project. First the potential technologies have been introduced and compared to help on the decision how to choose a proper technology to develop the system. For the Grid technology, some examples of different application areas have been given. As mentioned, the system needs to be universal, dynamic and extensible; so SOA architecture should be employed, then Web/Grid Service Technology becomes the choices.

Also, some of the current related researches have been reviewed, these literatures not only give proof that how important this research is but also provide the author more ideas on how to develop this system. This research is so worth being conducted that not only the current projects, but also the EU projects are concentrate on this area of research. There are several related researches have been done by the Advanced Design and Manufacturing Engineering Centre in Nottingham Trent University, which provide a strong basis for the CWE proposed in this research, the big challenge is how to make them integrated and work together for the CWE. So the two most important issues which make the resources work together in CWE come into the light, which are the sharing and interaction among the resources and systems in the CWE, together with wrapping legacy systems to fit the CWE.

For the interaction, the author intends to solve the problem starting from developing an online collaboration system of CAD, which requires an intensive interaction between the participants. The current researches all describe individual systems and seldom of them consider the future integration with other systems, and there are also hardly any researches talking about real time multiple user systems. The common thing of current researches is that, they solved the interaction problem in many kinds of technologies, but that is all, no future consideration.

Also, the current researches of the reuses of legacy system, especially the SOA approaches have been discussed and the author gets a clear idea on the advantages/disadvantages of them. Though there are many similar researches, but the author identified the big concern which other researches hardly pay any attentions to, which enable the function, not only a single service, but also multiple services to serve one application and this is to fully explore the power of web services. This chapter has provided a sufficient literature basis for this research. This approach is considered not only for wrapping, but also for resource sharing. According to the above issues, the next chapter considers the interaction in a distributed system.

Chapter 4

A Service Oriented Framework for Computer Aided Design

The research presented in this chapter aims to solve the interaction problem in the Service-Oriented environment and provide a communication support for the whole Collaborative Working Environment. Interaction is one of the most important issues in a virtual environment. In this Chapter, this method proposed by the author solved this problem in online collaborative CAD.

4.1 Introduction

Nowadays, computer aided design (CAD) packages, such as AutoCAD and ProEngineer, have been widely applied in design engineering, which greatly speeds up the design process. But most of such CAD packages are of a single location application only. With the development of the Internet technology, there have been demands for geographically dispersed designers and associated engineers to collaborate online. To meet the demands, the online CAD has attracted researchers' and software development companies' great attention. For instance, AutoDesk Company announced the availability of Autodesk Streamline, which is a hosted service for instantly sharing digital data and allows users to access design information anywhere, any time without ever downloading a CAD file, printing a drawing or opening a CAD package. Designers and engineers could instantly share the design information contained in CAD files with anyone inside or outside the company [58]. However, it does not provide an online environment for the users to interactively working on the CAD drawings in real time, such as instantly modifying the drawings online and allowing the users/designers to discuss the modifications with the aids of audio/video capabilities. To overcome the problems, researchers have been making efforts to combine Internet techniques with the CAD. For example, the State Key Laboratory of Mechanical Transmission at Chongqing University [59] developed a prototype online CAD system by integration of AutoCAD with the Microsoft Netmeeting, which enables several AutoCAD packages in different locations to communicate with each other. In addition, their users can instantly modify the drawings online and discuss the modification by speaking to each other while seeing each other on the screen. Of course, this is an important achievement; however, it is based on Microsoft Windows and cannot run on other systems such as Linux and Unix; Also, networks are often protected by firewalls, which could not be easily penetrated by the system. The system may work well within a Local Area Network (LAN), but not other networks rather than the LAN.

The Advanced Design and Manufacturing Engineering Centre at Nottingham Trent University has been involved in two European Commission supported projects, Asia IT&C and Asia Link. In the Asia IT&C project, a Web-Enable Environment (WEE) for collaborative design and manufacture is being developed, and in the Asia-Link project, a virtual research institute (VRI) is being established, both of which are aimed to enhance the collaboration between the EU states and China in research and higher education in engineering. Within the WEE and VRI, the Web-based collaborative CAD is an important task to complete. To achieve the objective, a Service-Oriented Framework for Computer Aid Design (SOF-CAD) has been developed, which is presented in the dissertation. The SOF-CAD enables geographically dispersed team members to collaborate online over the Internet with advanced features of overcoming the drawbacks of the existing online CAD systems mentioned above. The technique of Web Services has been utilised for the users to provide and to request the services, and the most popular CAD software AutoCAD is used as vehicle to illustrate the methods developed which are in principle applicable for other CAD software packages such as Pro/Engineer.

4.2 Overview of the SOF-CAD

4.2.1 The main features of Web-services utilized in the SOF-CAD

With Web-services functions, a Web application can provide services for thousands of users at the same time. The system can mix Net, Java, C++ and various other languages together. Adopting unified eXtensible Markup Language (XML), which is the most important feature of a Web Service application program, used for data description [60], making the service oriented architecture possible. The services can be described as Web entity and receive all information transmitted in SOAP (Simple Object Access Protocol) in the XML format and is easy to go through firewalls [61]. The structure of the service-oriented system is plantable and flexible to meet the constantly changeable demands to update and append required additional functions. It can also integrate the existing software (programs and packages) and avoids the tedious tasks to re-write the software in a required format in which they were not originally developed. Therefore, the system is not only powerful and reliable, but also flexible and extensible, in order to meet the increasing new requirements.

The web service has the following characteristics which have been utilised in this research：

- Relying on enterprises or organizing existing technologies, Web Services can turn enterprise existing digital resources, packages, application systems into services [62]. After that, their interfaces are described in WSDL (Web Service

Description Language) files, which help the client to understand what the services are and how to access them [63]. In this way, all the resources become the services that can be accessed via the Internet.

- Web Services have real-time feature. The change of one resource will reflect immediately to others that are sharing the same service.

- Web services have dynamic characters. Any authorized user could join the services without affect on other existing users.

- Web services can be built in another service, or they can work in coordination. A service user can consume a lot of other services at the meantime. Some service themselves can call at other services. Several services can work in coordination to conduct one or more jobs [62].

Generally speaking, the Web has given a good interaction platform between human and computer programmes, while web service is giving a better environment for interaction among programme themselves [64]. That is why web service is chosen to be the intermedia technology of the CAD collaboration system.

4.2.2 Main structure of the SOF-CAD

In this system, the technology Web Service plays a key role of connecting an individual site, where the CAD package is located and related services are available, to the Internet, with multimedia functions of audio/video and message board provided by the VRI. The JNI (JAVA Native Interface) carries out the tasks of dynamic data processing and links with the AutoCAD and Web Service. The designer operates the AutoCAD to produce drawings and makes changes which are instantaneously sent, with the aid of JNI and Web Service, to the collaborative designers at different sites. With VRI's multimedia functions, the designers can communicate online with each other in the ways of speaking, writing messages on the board, and seeing each other on the screen during the collaborative design process.

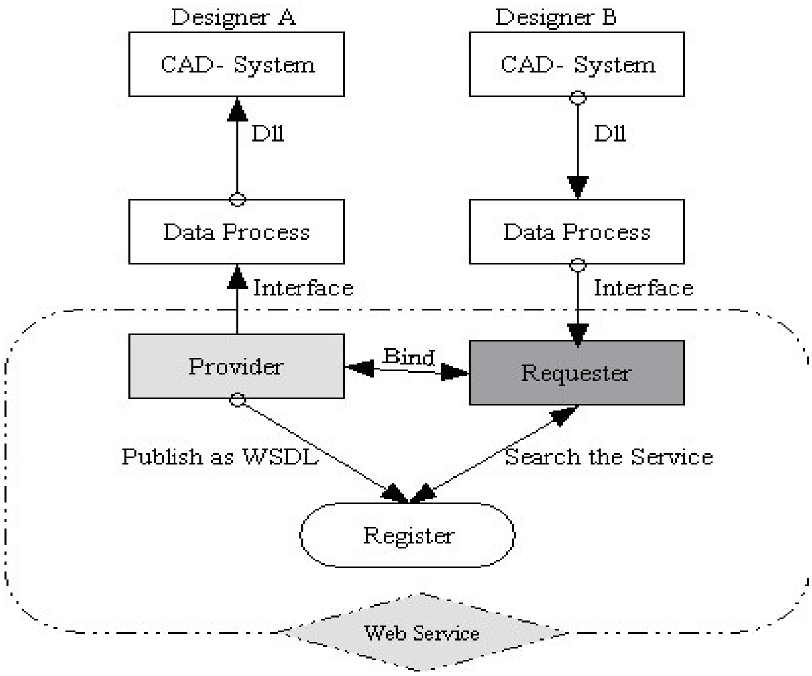


Figure 4.1 Operation procedure of the SOF-CAD

The operation procedure of the SOF-CAD is illustrated in Figure 4-1. In order to load a CAD model as a Web service, the provider (Designer A) of the CAD model generates a service-describing file in WSDL format, and then registers the service with the WSDL file to the service registration server. The WSDL file contains all the information of the service including the location, lifecycle and functions of the service. In a normal situation, when a client requests a service, the client, i.e., the requester of the service may do not know where the service is, so he/she searches the registry server for the information, if it is found, the registry server sends the requester (Designer B) the WSDL file. With the information provided in the WSDL file, the client (requester) can find the service. Then the service provider, which is in charge of service allocation, would produce an instance of the service for the requester and, also, the provider can produce more instances if there are more requesters. As shown in Figure 4.1 when designer B makes a change in his/her

AutoCAD drawing, the change would activate outer DLL (Dynamic Linked Library) program to capture the change information and then invoke a Java interface, which would create the request with these parameters as SOAP message. After that, the message would be transferred to designer A, who is providing the service, and then, the SOAP message is analyzed by designer A's system. This activates AutoCAD at Design A to make the same change in its drawing to meet the request.

4.3 System components

The SOF-CAD system consists of three modules:

- Request module: This is for a computer to send request to service via Network. Before that, this module needs to capture the change of drawing and describe the request of change with JAVA.

- Middleware module: This module is used to generate SOAP messages by wrapping JAVA request and send the SOAP message to the aimed computer, where the received messages will be unwrapped.

- Response module: This is for the computer to response and to react when receiving a request from another. When the parameters have been captured and analyzed, they will be used by the ActiveX automation and if all the procedure goes successfully, the response module would return a signal to inform that the mission is completed.

Figure 4.2 shows how the three modules work together in the procedure. It has to be noted that when the connection is established between two computers, each side can act as either client or server. For the case shown in Figure 4.2, in one action, when designer B makes a change in the CAD drawing, the system sends the request using request module, This contains the information of the changes, to computer A through middleware module. Computer A then calls its response module to action. In this scenario, computer B is considered as a server and computer A is a client. After that, if designer A wants to make a change in the CAD drawing, the request module in computer A is invoked, in this scenario, designer A is considered to be a client, and computer B becomes the server. So that whoever provides the service does not decide if he is client or server; but the specific process decides this. Another thing should be addressed is that; the request module and response module exist in both the service requester and service provider. They only deal with data processes and have nothing to do with data transmission through network.

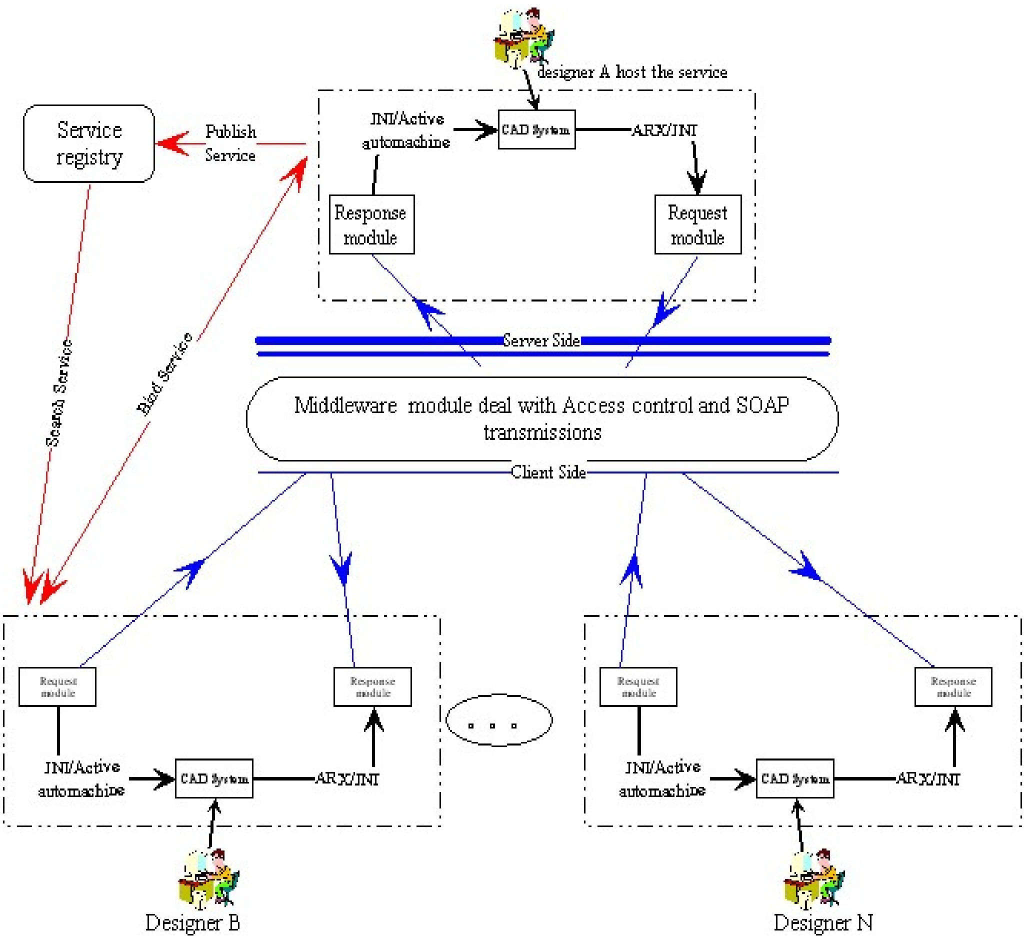


Figure 4.2 System modules and data transmission flow

4.3.1 Request module

Suppose that in a gear designing process, the first step is to draw a circle in the AutoCAD in local PC. After this is done, the information of the change need to be send as a request to the aimed computer, the system then need to call the Java function outside AutoCAD via the C++ program, which exists in the form of Dll. The procedure can be divided into the following two sub-procedures: (1). Invoke outer programme with ARX. (2) convert C++ datatype to Java datatype using JNI (Figure 4.3).

(1). Invoke outer programme using ARX.

ARX is a powerful AutoCAD integrated toolkit, which can be used to develop new AutoCAD commands as users want. It can help to integrate outer program into AutoCAD as well, and so, it becomes an ideal choice for the SOF-CAD system to be developed. In the SOF-CAD system, the information of change can be perfectly exported from AutoCAD database by the ARX programme [65].

When a user starts AutoCAD, he should load the ARX package. When he makes a change, the ActiveDoc->isModified() function is activated and the ARX package export the information of change as parameters to outer Dll programme.

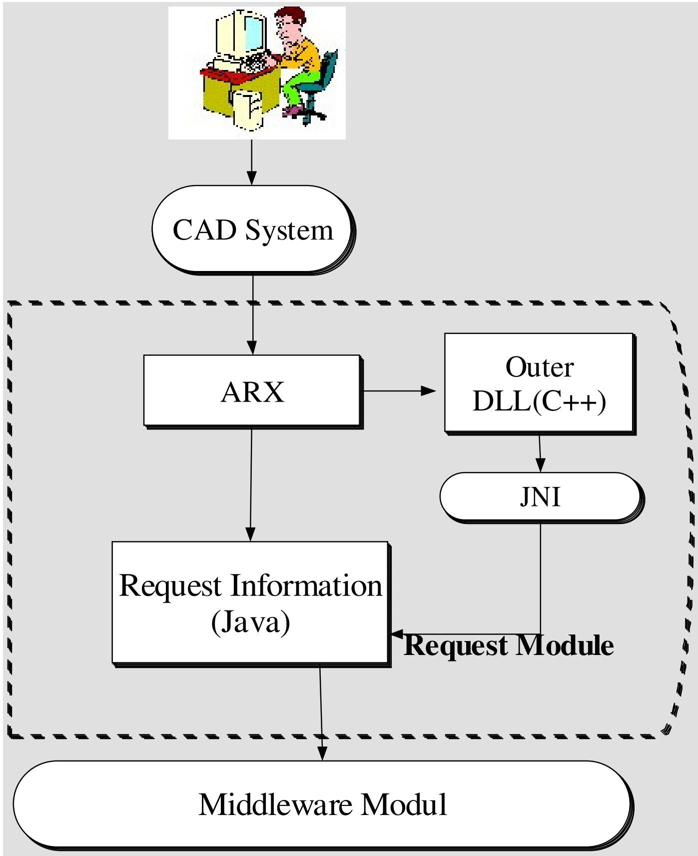


Figure 4.3 Data process of Request Module

(2). Convert C++ datatype to Java datatype using JNI

In the first step, the AutoCAD has activated the outer Dll, and in this step, the Dll would call Java programme by JNI (Java Native interface), which acts as the bridge between Java and other languages, for example, JNI is doing well to communicate C++ and Java.

All Java programme can only run within JVM (Java Virtual Machine), even when running in a native application. The Dll is written by C++, it must include functions to create a JVM and to initialize it. To assist to do so, the SDK (Software development kit) includes a JVM as a shared library file (jvm.dll or jvm.so), which can be embedded into native applications. The java programme is used to be packed into SOAP request and send the request to the middleware module.

4.3.2 Response module

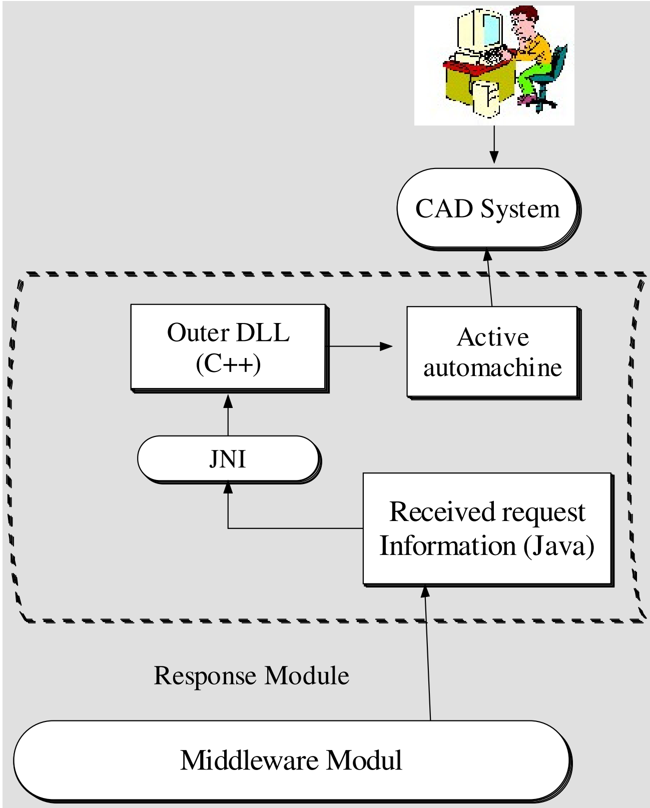


Figure 4.4 Structure of Response Module

This module is also located in both service requester and service provider, which is mainly deal with received request from the different location. Similar to the request module (Figure 4.4), two processes here should be addressed:

(1). Convert Java datatype to C++ datatype.

When information of request is received, it needs to be converted to C++ data type so that it can be recognized by the next step. The parameters in the information are transferred into Dll after they are converted.

(2) ActiveX Automation

When the service is requested by the other users, the Java code of the provider will call C++ file at the same time. In fact, the C++ file is of the Dll format. The Java file calls the special function in the C++ code. This C++ code will ask AutoCAD to react, as shown in the sample, to draw a circle in request module. The circle appeared in the provider's AutoCAD is the same as the one drawn in request module; it is done by the system based on Internet rather than local user.

4.3.3 Middleware module

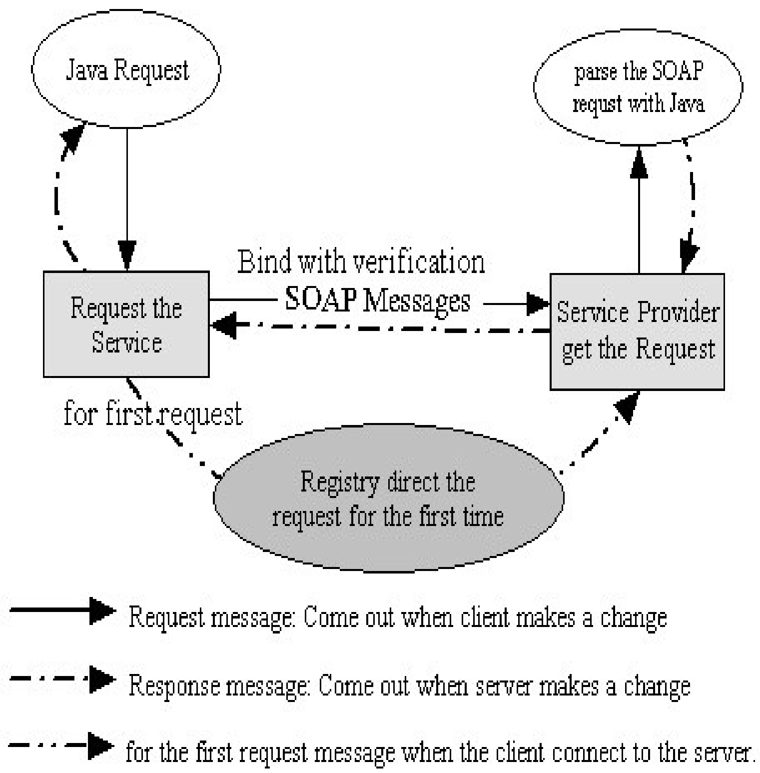


Figure 4.5. Process of the Service Module

This is the module to verify and transfer SOAP messages (Figure 4.5). Service provider needs to register their services to the service registry so that the information could be found by other requesters via the Internet. Then the service registry directs the service requester to find the right location of service. After that, service requester and Service provider are bound to each other and ready to correspondence.

The parameters captured by the program in client are first packed in XML format. This is then ready to be transferred in SOAP message so that the service provider could receive and analyze it [66][67].

Security is also taken into consideration in this module. There is a table that is used to store the ACL (Access Control List) in the service catalogue. In the current version, PKI (Public-Key Infrastructure was used to do access control [68]. Each entry in the ACL table contains certificate information of user who can use the shared application and an application identifier. Only a certificated user can go through this module.

4.3.4 Discussion.

The three modules works respectively from each other in accordance with one of the main character of web service; loose coupling. Maintenance can be made separately too, which means that any modification to one module does not effect the cooperation among the three modules. Multiple threads applied in web service and activate in each module to enable the function which sends parameters while listening to others.

There maybe some exceptions happen, for example, it is possible that when a user send a request, there are no service provider, in that case, the system will send a alert to him about this

4.4 A case study

Following the aforementioned system design and implementation, a case study is conducted to testify and illustrate the use of the system. It includes three computers with Pentium IV 2G processors located at three sites and linked to Internet. Each of them is installed with AutoCAD version 2004, Axis, Tomcat 5.0, and Object ARX package. Each computer is also equipped with video and audio facilities, a video camera and a microphone. Please note that a firewall is turned on to test the system intentionally.

The system is operated using the following procedure:

- Each designer starts AutoCAD on his/her local computer.

- The SOF-CAD System loaded.

- The leader of this case opens a drawing or draws a new drawing in AutoCAD on his/her computer. Then the drawing is sent automatically to the participants and is automatically displayed in AutoCAD in their local computers.

- The participants then can discuss the drawing using functions of audio/video, whiteboard.

- Meanwhile, the participants can modify (add or delete entity) the drawing on their local computer. Only the entity modified in local AutoCAD will be transferred automatically and simultaneously to remote AutoCAD and renew the same entity of the drawing on each computer.

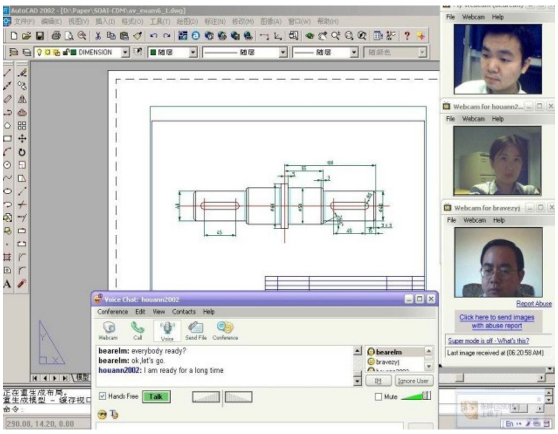


Figure 4.6 Screenshot from Designer Xiong

The task of the collaborative CAD conducted in this case study is to modify a shaft design online amongst three geographical dispersed designers Xiong, Hou and Zeng (from top to bottom shown in Figure 4.6). The collaborative design is conducted within the Virtual Research Institute (VRI) environment [69] where the audio and video facilities are provided for online communication via the Internet. When the designers get into the system, the designers can see each other from the right side panel on their own computers (Figure 4.6) and can speak online as well. Figures 4.6 to Figure 4.8 were captured during the process of the online design to illustrate the procedure mentioned above.

Before the online collaboration, the leader of the design, Designer Xiong deployed the service and published it on the web. To initiate the online design, he opened a drawing in AutoCAD on his computer and sent invitation to Designers Zeng and Hou to join the design. After receiving Xiong's invitation, Designers Zeng and Hou started the SOF-CAD system, which began looking for the service, and then connected to Designer Xiong after found the service. A tiny control panel appeared in top left front of AutoCAD informing that "System started…" and Designers Zeng and Hou could see the drawing on their AutoCAD.

After viewing the drawing, Designer Zeng was not satisfied with the shaft-end design, and made the following modification: the length of shaft-end was changed from 45mm to 40mm and its diameter was changed from 40mm to 35 mm. All the changes were simultaneously displayed on the three computers, as shown in Figure 4.7.

After looking at the modified drawing further, Designer Hou thought that there should be an oil storage hole which would be more suitable. She added this to the drawing, shown on her computer, and Designer Xiong and Zeng could immediately find the changed drawing on their computer, as shown in Figure 4.8.

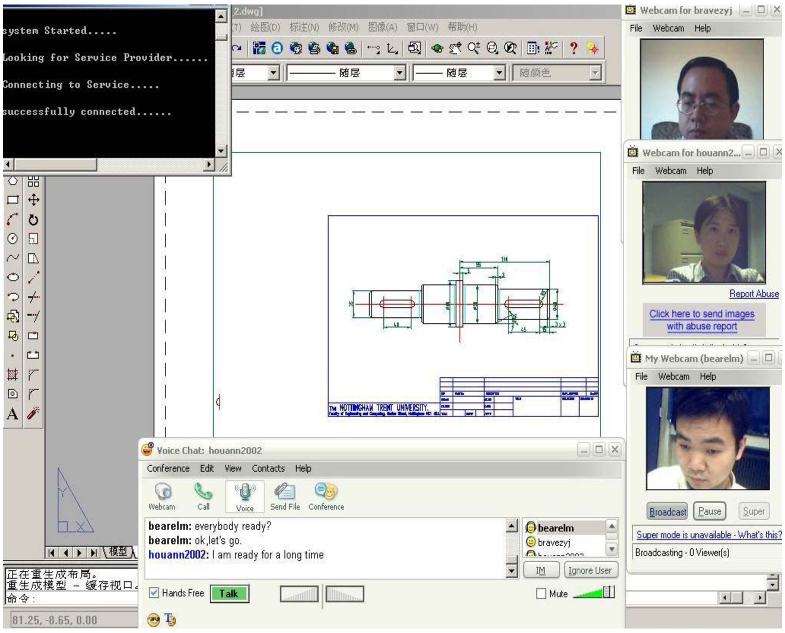


Figure 4.7 Screenshot from Designer Zeng

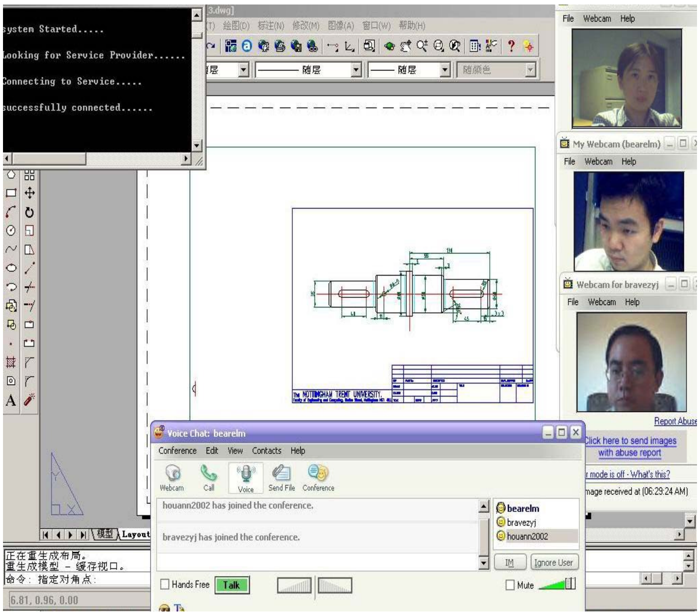


Figure 4.8 Screenshot from Designer Hou

4.5 Evaluation and limitation

The case study indicates that the SOF-CAD system can meet the expectation described in section 4.2.1. However, as a feedback from the users, the real-time interaction has a latency; by adopting web services the system takes the advantages of cross platform and loosely coupling structure,; the current Web service protocol do not solve the latency issue well. The other limitation is that, the development of the system does not consider the concurrent control among the participants. If multiple designers are changing the same object in the drawing during a short period(1 seconds), the system can only recognize the latest modification from one of the users. Another user's feedback is that, the deployment of the system is complicated. All this shows that there is a distance from current research to the commercial software.

4.6 Conclusion

The SOF-CAD system is strong and reliable due to the design of SOA structure. The system already helped the group on design and research collaboration. The service oriented architecture, which is loosely coupled, makes the system much less fault occurring in comparison with traditional systems. The feature also provides a potential possibility for the communication between AutoCAD and ProE, and the direct connection among softwares is not necessary with web service support during collaboration. Most notable for its cross platform system characters, it runs well on both Linux and Unix system. JAVA, SOAP, WSDL technologies are used for data transmission and descriptions. These technologies also make the system can reach all over the world ignoring the existence of firewall.

Another thing must be noted is the system is based on open source tools like Tomcat and Axis, which could be downloaded everywhere without care about the copyright problem compared to Netmeeting which is bound to commercial software Windows. The third generation of the Internet will be more effective on CAD areas. The development of SOF-CAD system is the first step of the research on CWE, where the Service-Oriented interaction will play a very important role. After the solution of resource interaction has been shown in this chapter, the next chapter will consider the other important aspect of distributed system, the resource sharing.

Chapter 5

Service Oriented Software Package Bank

This chapter is to present a proposed solution for the Asia-Link software package bank module. The solution must be efficient and distinct from other existing ones. As is discussed in the literature in Chapter 3 and will be further discussed later in this chapter, many traditional problems which exist currently in software sharing should be solved. The related issues to this research include resource sharing, reuse of legacy application. The wrapper method toward legacy applications is also a very important function in the CWE infrastructure, which could make the system accommodate more resources.

5.1 Introduction

The conventional way of software sharing is to give the binary or source file to the users. However, the software owner may neither want to give the binary file nor the source code to other people for the following reasons：First, the executable file could easily be cracked using anti-compile tools and delivered to other people who have no rights to use it. This situation may be worsened when source codes are released to as the copyright would no longer be protected.

Another reason for software owner's reluctance to share files is that the software may be released with a lot of copies being used in many different places. If a serious bug is then discovered, it could take a lot of time to update all the copies. Sometimes the software development team may want to recode the old version of the program to make it adapt to new requirements, and the spread of the old version makes the update difficult. Moreover, some software needs to request information from a database belonging to another organization, who may not wish to give permission to share the resources with their users other than the service providers.

To overcome the above drawbacks, a new way to share software, i.e. service-oriented software sharing, has been developed by the author. This method neither gives the binary nor the source file to users; instead, it packs the binary files of the software resources, so called shared applications, within a Web service and provides users an interface of the service. The user of a shared application receives the description file of the interfaces, with which the user could build a client application to access the shared application via Internet. The software owner does not need to worry about the software sharing problems mentioned earlier and the software can be shared efficiently.

However, most of the existing software programs/packages have not been designed for this purpose and do not follow the right structure in so doing. To resolve this, a method to package the software based on the Web service technology is developed by this research.

To understand this further the following begin with a brief introduction of the Web-service based approach for software sharing. This is continued by presenting the architecture of service-oriented software sharing system and its implementation, followed by a case study to illustrate the approach developed.

5.2 Service-Oriented software sharing and Web Service

5.2.1 Requirements for the service-Oriented Software sharing

In the service-oriented method for software sharing, the term ‘service’ refers to a logic view of the real application defined in terms what it does, typically carrying out the business logic of an application. The service interface is an end-point where the particular business logic of an application could be invoked over a network. The description file of the service interface is accessed by the service requester. A client application (requester) should be built follow the description of the interface so that it can interact with the service to complete a task. This method enables the service requester to share not only the business logic of an application, but also the resources associated with the application. It provides enough facilities for users to allow their client applications to integrate such services for their use. By sharing software in such a way, the drawbacks of the traditional methods for software sharing are overcome, and software can be efficiently shared.

Although this new idea of software sharing sounds simple, the implications are often subtle. There are some requirements that should be taken into consideration:

* It should support access control, so that the software owner may allow only selected people to share his/her software.
* Since many existing legacy applications have not originally been designed for distribution purposes, the issue of how to convert these applications into distributed ones has to be considered.
* The software performance is also a very important issue. The approach should balance the functions and the efficiency.

5.2.2 The role of the web service in service-oriented software sharing

Web service is chosen to meet the challenges of service-oriented software sharing. Although many distributed technologies can do this, web service is to be the most suitable one. Non-web service technologies, shown in figure 5.1, can provide various of ways to interact with the applications. These have to be coded separately for each way, e.g. People have to code IDL (Interface Define Language) for the CORBA way. However what is coded cannot be used by Servlet/JSP architecture [70][71], and so it is to others. Generally speaking, it has to be coded again and again for every kind of client that wants to invoke the service. Although the type of client can be regulated to appoint ones, it still greatly hinders the flexibility of building client applications. The advantage of web services should be taken into account to meet the needs of the service-oriented software sharing.

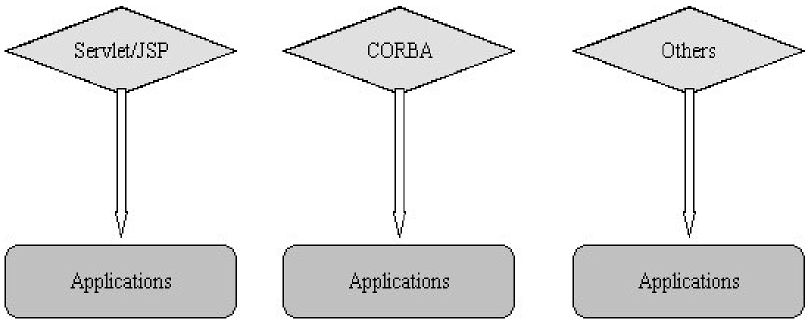


Figure 5.1The scenario of non-web service technologies used in software sharing

Web service is the latest technology and provides a SOA software design. A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has a interface described in a machine-processable format called WSDL (Web Service Description Language). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

For the Web service approach, once the shared application is packed into a service, the uniformed service interface, which is provided by WSDL [63], is published. After such WSDL files are discovered by other people, it is not necessary for them to understand the internal structure of the service, if they can build their own applications which interact with the service interfaces. The service can be visited by programs written in JSP/Servlet, CORBA or others over Internet, they can even be directly visited by an unknown application. Multiple users with different client applications can visit the same service at the same time as shown in Figure 5.1. For example, while some people are using a web browser to invoke the service, other users can use their own customized client-side programs to call the service. By using web services in software sharing, a legacy program can easily be integrated into the system as a Web service package with distribution features. The same service not only is available for multiple users, but also has multiple ways of accessing.

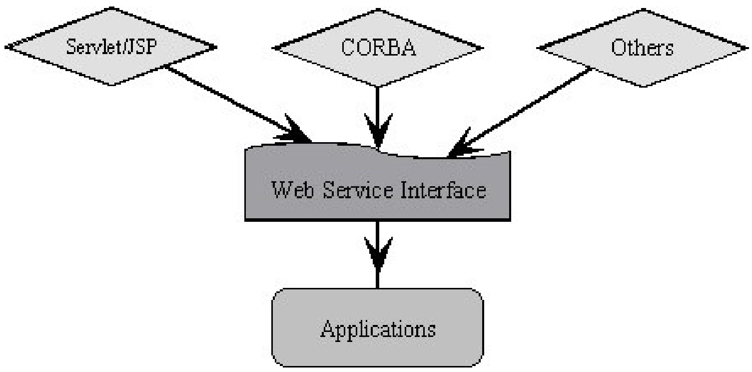


Figure 5.2 Different clients access applications via web service interfaces

Figure 5.2 shows the structure of a web service approach. These discussions have analysed requirements of the service-oriented software sharing. Earlier research, in Chapter3, found web based research adopted web service to resolve the re-use of legacy application, but their approaches cannot fully exert the capability of web service. The first requirement is easily achieved by using web service just like other approaches. In this research, the author tries to design a software bank component to meet the rest of the needs. The aim of the software bank is to implement a service-oriented software sharing system for people who want to share and use applications in different areas and departments via the Internet using the new software sharing method.

5.3 Constituents of the software bank component

The Service oriented software sharing system has to be designed to meet the requirements stated in the previous section. The first requirement could be easily achieved by using Web service. The other requirements are met by careful design of the system components as detailed in this section.

5.3.1 Constituents of a service-oriented software sharing system

In the service-oriented software sharing system developed using Web-service technique, there are three types of services:

* Application proxy service,
* Application proxy factory service, and
* Application manager service

Besides the three services, there are two types of applications:

- *Shared applications* which are the software packages/programs, such as CAD packages, design tools and calculation programs, to be shared amongst the collaborative team members and users, and

- *Client applications* interacting with the shared applications, which are built with the service interfaces description files and are used by the people, who want to access the shared applications.

Application proxy service (APS). The APS provides a surrogate for the process to run a shared application and to control the access to it. It models functions presented in the shared application and exposes the functions as Web services. Actually, an application proxy Web service is similar to a running process of a shared application, but it is accessed remotely. The APS also manages the resource used by the shared application, monitors the situation of the shared application, and handles security and other issues. A client application interacts with an APS to share an application.

Application proxy factory service (APFS). It may be a common situation that the shared applications are not originally designed for service-oriented software sharing. Some of them even cannot be used in a multi-user environment where the service-oriented software sharing method is to be utilised. A single APS cannot serve multiple users, so there would be multiple services to serve different clients. These APSs should be managed, and the resources used by these services should be managed too. The APFS, which is similar to the process management in a local Operation System, is designed to meet the needs of managing the creation and lifecycle of an APS. The resources used by these services are also managed by the APFS. Each shared application has an APFS to create multiply APS instances. The APS and APFS pattern not only make it possible for an application to be packed with a web service, but also make a single user application pretend to be a multi-user one, which is similar to research results presented in [72].

Application manager service (AMS). In an Internet environment, everyone can access the same single Web service using his/her own client application at the same time. Although the APS and APFS are used to make a shared application pretend to be a Web service based on a multi-user one, it is in fact a single user one. The number of clients that these applications could serve at the same time is not as good as a pure Web service one. So the AMS is built to confront this performance issue. Another reason of building the application manager service is to support the access control mechanism.

5.3.2 Architecture of the service-oriented software sharing system

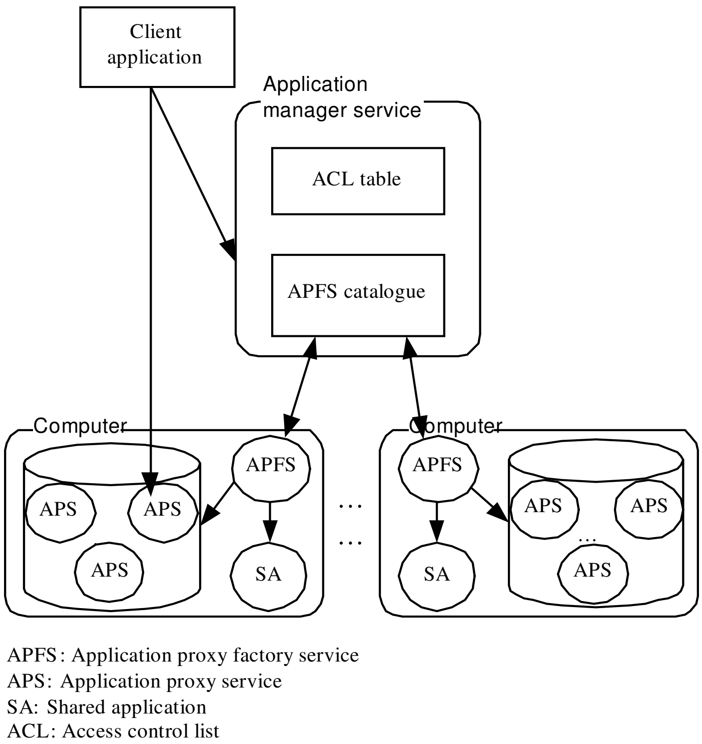


Figure 5.3 Architecture of software bank

Figure 5.3 shows the architecture of the service-oriented software sharing system. There potentially are multiple shared applications installed in multiple computers. Each of them is installed with the APFS. The process of activating the shared applications are represented by the APS created and managed by the APFS. All the APFSs are registered to the AMS and assigned an application identifier. The AMS has an ACL table for security control.

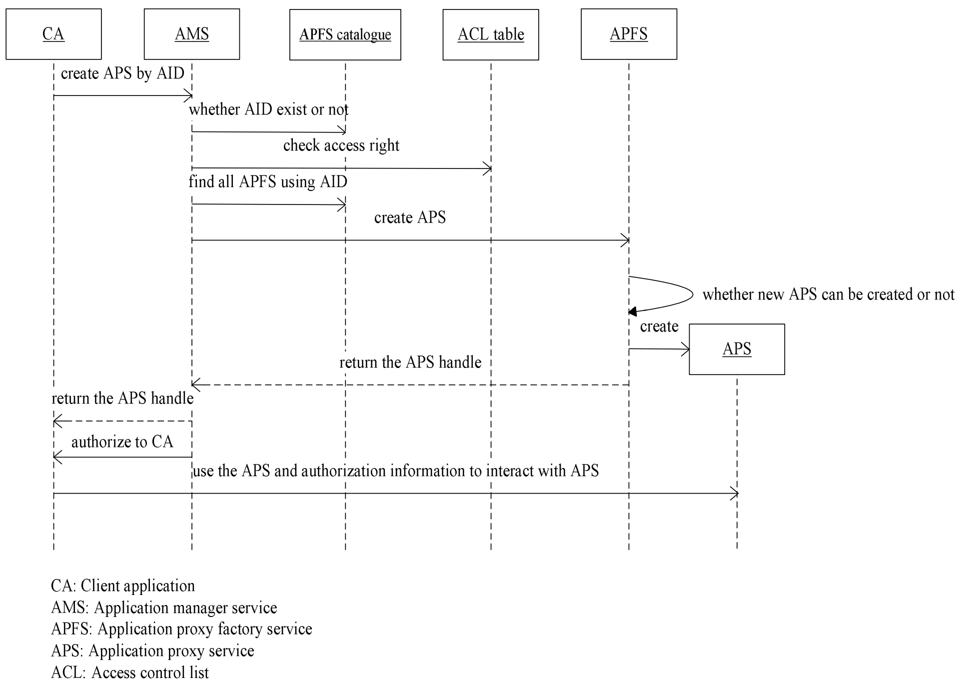


Figure 5.4 Process of interaction

Figure 5.4 shows the process of interaction between client applications and other components within the system. When the user launches a client application, the client application issues a request to get an APS from the AMS. When the request arrives, the AMS looks up the APFS registered in the catalogue using the application identifier specified by the client application. If the APFS exists in the catalogue, the AMS then checks whether the client application has the right to use it. If the access right checking is successful, then the AMS interacts with the APFS to create an APS.

It's possible that multiple APFSs are associated with one shared application. These services may be installed in different locations. When an APFS is requested to create an APS by AMS, it should determine whether a new APS is to be created or not. If the APFS cannot create a new APS, the AMS is then informed, and hence the AMS chooses another APFS instead.

After an APS is created by an APFS, the AMS returns the handle of the newly created APS to the client application. The AMS authorizes the client application with a license to access the APS. Then the client application uses the handle to access the shared application. After the client application completes its task, the APS used by the client application terminates, and the resources allocated to the APS are then released accordingly.

During the process of interaction described above, the client application cannot access the shared application without authorization. This performance issue is solved by the AMS and APFSs, and the service-oriented software sharing is thus achieved.

It would be a challenging work to assure the reliability of the system, such as handling the exception that one APS is down while it is being processed. This is currently being dealt with in the authors' on-going research.

5.4 Implementation of the approach

The major tasks for implementation of the Web-service based approach for software sharing include packing an application with an APS and implementation of the APFS and AMS. The development of the methods for conducting the tasks is detailed in this section.

5.4.1 Pack application with APS

The APS represents a process of activating a shared application. When a request arrives, the APS delegates the request as local function calls to the associated process. The primary issue of this service is how to pack an application with an APS. Four strategies are developed to pack different types of applications as categorized below:

- *Applications with distributed features.* In this case, most applications could be packed with Web service by wrapping the interfaces exposed for distribution purposes. With these interfaces, the APS could be easily built.

- *Applications built with component technology*. If all the business logics of the application reside within a single component, then the solution is to simply pack the component object and expose the web service interface. If the business logic resides in multi-component, a new business object should be built. A new business object interacts with multi-components and exposes all of business logic interfaces, and can be packed with APS.

- *Applications using shared libraries*. In this case, the solution is to build adapters to call functions within the shared libraries. Then build a high-level business object to interact with low-level adapters, finally, expose the business interface with the APS.

- *A standalone executable file as an Application.* This kind of application is very difficult to be packed into services though it is still possible to do so. In most cases, it is recommended to rewrite the application. However, in some cases, applications are really very important and it would be costly to rewrite, some solutions could still be available. For example, Microsoft Visual Studio Net provides some tools to convert a standalone executable file into DLL, and then the strategies mentioned above are applicable.

In other situations, some applications use input stream to read data and output stream to write results. These standard I/O streams could be replaced with or redirected to another input or output streams. For example, in Java language, the System.setIn() method could be used to replace standard input stream with an input stream, even a network input stream. So an aided application could be written to replace or redirect the I/O stream of the original application, then expose interface using Web service. Another case is that applications use command line arguments to read data. To deal with this, an aided program with functions to execute files by specified command line arguments are necessary; such functions exist in most programming languages.

For applications, most of which are Graphical User Interface (GUI) applications, do not use standard I/O functions, they could be converted into shared libraries. If users want to reuse these applications instead of rewriting them, which often results in an unimaginable high cost, the solution is to build a bypass library to replace the library needed by the application. The bypass library interacts with the APS. Due to the complexity of the work involved, it is not recommended.

5.4.2 The implementation of APFS.

APFS has a function to create an APS. When a creating request arrives, the application proxy factory should check the status of the local computers to determine whether a new APS could be created or not. After a new APS is created, the APFS allocates the resource required by the APS, and then registers the APS to a proxy service catalogue. This catalogue is stored in the XML format.

The APFS should manage the lifecycle of all the APSs. It periodically checks whether an APS in the service catalogue is still active. If an APS is inactive, the application proxy factory would dispatch a timer for this service. The inactive APS would be removed. If its TTL (Time To Live) is over, the associated shared application process should be terminated. The resource allocated to this process is then released. In current version of the software sharing system developed by the authors, each application factory service is associated with only one shared application and accessed only by the AMS.

5.4.3 The implementation of AMS.

The main function of the AMS is to deal with the performance and security issues. The AMS provides a catalogue containing URIs of all registered APFSs. The application identifier identifies all the APFSs of a shared application. As mentioned in the above sections, each APFS represents a shared application in a computer. So the application identifier is used to look up all the APFSs of a shared application.

The APFS catalogue is implemented via an interface, which connects to a relational database, typically a MySQL database. Each APFS is registered as a database entry containing an application identifier and URI of the application Proxy factory service. When looking up a specific shared application, the URIs of APFSs with the same application identifier should be returned. Then the AMS interacts with the APFSs using these URIs to create an APS.

In order to achieve higher performance, a shared application could be installed in multiple computers with an APS associated with each one. Then all the APFSs are registered to the AMS. When creating an APS, the AMS polls these APSs and chooses one, which has reported that there are enough resources to create a new APS.

There is also a table used to store the Access Control List (ACL) information for a shared application in the factory service catalogue. In the current version, the AMS use PKI (Public Key Infrastructure) [68] for access control. Each entry in the ACL table contains certificate information of each application identifier with its authorized users who could use the application. When a client application requests to use a shared application, it sends its certificate information in a SOAP header to the application manager. Then the application manager checks whether the client application could use the shared application by using the ACL table. If the client application has access right, the AMS signs a certificate to the client application. This certificate expires after the client application finishes its tasks. In the last step, the AMS uses the APFS to create an APS, and then returns the handle of the APS to the client application, which finally uses the certificate given by the application manager to interact with the created APS.

5.5 Conclusion

A Web-service approach for software sharing has been proposed, based on which a service-oriented software sharing system has been developed. It enables geographically dispersed team members to share their software resources, such as CAD packages, design tools, analysis software and calculation programs, over the Internet. Different from traditional methods of software sharing, the Web-service based approach neither gives the binary nor the source file to users; instead, it packs the binary files of the software resources within a Web service and provides users an interface of the service. It hence meets the requirements of copyright protection, reuse of legacy code, and better performance.

This research utilized the application proxy service, proxy implementation service and application manager service to conduct a software sharing task. Nowadays, software sharing is a common issue for collaboration over a network, therefore, the Web-service based approach developed by this research has a great potential for applications in a wide range of areas. The structure of the three service approaches is very promising and should be extended and utilized in a more wide range. After discuss the web service based approach, the next chapter will conduct the research on the improvement of this method and also, a case study will be conducted to testify the system. This approach also provides an important basis for CWE.

Chapter 6

Wrapping Computer Resources to Fit Grid Infrastructure

Based on previously mentioned approaches, in this chapter, a way to wrap resources using Grid technology is presented. Note that due to the similarity between Web service and Grid service, the 3-service-made up component introduced in chapter 5 is still valuable and utilized in this research, but in a Grid context. This work is also a turning point for the research from Web-service level to Grid-service level, which can accommodate large scale resource sharing and intensive interaction. A case study is conducted to testify this approach.

6.1 Introduction.

Grid computing is an emerging technology, which enables the utilization of distributed computing and resources to create a single system image, granting users seamless access to vast IT capabilities. Just like an Internet user views a unified instance of contents via the Web, a Grid user essentially sees a single, large virtual computer. The concept of virtual organization is strengthened in Grid, which has the potential to change dramatically the way in which we use computers to resolve problems, similar to that Web has changed the way we exchange information [73][74].

Although Grid has significant advantages, it still has not gained applications in practice as widely as expected, due to the problem of utilizing existing resources. Many resources, including both standalone and Web-enabled applications, currently in use are not initially designed for Grid. Users or designers of these applications/resources are reluctantly to rebuild them to adopt the new structure, because the rebuilding could be expensive in time and cost. To overcome the problem, in this research, the latest Grid architecture, *Web Service Resource Framework* (WSRF) [75], has been applied and a method has been developed for packing existing resources to fit the Grid architecture.

6.2 Converting Resources into Grid Supported Services within WSRF

WSRF is a set of Web service specifications and conventions designed to standardize their representation. In this research, it is utilized to coordinate Web service and Grid. Different from Grid, Web service is a matured and practically-proven technology. With Web service, the request and the response are encoded in XML. An XML formatted remote procedure call is sent and received by the *XML-Remote Process Call* function. The *Simple Object Access Protocol,* which is also an XML technology, is utilised to remotely manipulate an object. The existences are published through a number of different Web service protocols, including *Universal Description*, *Discovery and Integration* and *Web Services Description Language*, which enables the Grid computing to be easy to deploy [76].

Resources play an important role in distributed systems. But majority of existing resources are not ready for Grid, an emerging new technology. While most of programs are following the Object-Oriented structure that is still very popular, the Service-Oriented ones would have not got a sufficient market for lacking of compatible resources.

In this research, a similar method has been developed to pack existing applications into Grid resources with three types of services including:

- *Application proxy service (APS)* which provides a surrogate to control the access to a shared application. It models functions presented in a shared application (SA) and exposes the functions as services. Actually, an APS is similar to a running SA process and can be accessed remotely. The APS also manages the resource used by the shared application, monitors the situation of the shared application, and handles security and other issues. A client application will interact with an APS to share the application.

- *Application proxy factory service (APFS).* Some of Applications may not be used in a multi-user environment (a service-oriented software should be a multi-user environment). A single APS cannot serve multiple users, so there will be multiple services to serve different clients. These APSs should be managed, and the resources used by these services should be managed too. The APFS, which is similar to the process management in local operation system, is designed to meet the needs of managing the creation and lifecycle of an APS. The resources used by these services are also managed by the APFS. Each shared application has an APFS to create multiple APS instances. The APS and APFS patterns not only make it possible to pack an application with a web service, but also enable a single user application to support multiple users.

- Application manager service (AMS). In an Internet environment, everyone can access the same web service from any client end at the same time. Although APS and APFS are used to make a shared application pretend to be a web service for multi-users, it is in fact a single user one. The number of clients that these applications can serve at the same time is not as many as that of a pure Web service. So AMS is built to confront this performance issue. Another reason of the building application manager is to support the access control mechanism.

There may be multiple shared applications installed on multiple computers. Each of them is installed with the APFS. The running process of the shared application is represented by the APS created and managed by the APFS. All the APFSs of a shared application should be registered to the AMS and assigned an application identifier. The AMS has an *Access Control List* to control the security.

From the client's view, when the user launches a client application, the client application will issue a request to get an APS from the AMS. When the request arrives, the AMS will look up the APFS registered in the catalogue using the application identifier specified by the client application. If the APFS exists in the catalogue, the AMS will check whether the client application has the right to use it. If the access right checking were successful, the AMS would interact with the APFS to create an APS.

There may be multiple APFSs associated with a shared application. These services are installed in different host for performance reasons. When an AMS sends a request to an APFS to create an APS, the APFS should determine whether a new APS could be created or not. If this APFS can not create a new APS, the AMS will be informed, then the AMS will choose another APFS to try.

After an APS has been created by an APFS, the handle of the newly created APS will be returned to the client application by the AMS. The AMS authorizes the client application with a license to access the APS. Then the client application will use the handle to access the shared application. After the client application completes its task, the APS used by the client application will be destroyed, and the resources allocated to the APS will be released.

During the process of interaction described above, the client application cannot access the shared application without authorization. The performance issue can be resolved by AMS and multiple APFSs. The service-oriented software sharing can thus be achieved.

6.3 Resource Packing.

Similar to the packing method in the last chapter, different types of resources should be treated differently as described bellow.

A. Standalone software packing.

Three kinds of standalone applications are considered:

(1). Applications built with component technology

If all the business logics of an application reside in a single component, the solution is to simply pack the component object and expose the web service interface. If the business logic resides in multi-components, a new business object that interacts with the multi-components and exposes all of business logic interfaces should be built. Then the business object can be packed with the APS.

(2). Applications using shared libraries

In this case, the adapter is built first to call functions within the shared libraries, a high-level business object is then built to interact with low-level adapters, and, finally, the business interface is exposed with the APS. In some situations, shared libraries can be converted into components, and then this method can be used to describe it.

(3). Standalone executable files

This kind of applications is the most difficult for packing. Although it is possible to convert these kinds of applications into a web service, it is not worth doing so in most situations because of the difficulties. Ideally, rewriting the application could be a better choice. However some applications may be important and cost a high price to rewrite, therefore, the conversion has to be considered with different cases described below:

- Some of these applications can be converted into a shared library by tools; for example, Microsoft Visual Studio .Net provides some tools to convert standalone executable files into *Dynamic Link Library*. In this case, the previously packed shared libraries with APS policy can be used.

- Some applications use input stream to read data and output stream to write results, such as applications using standard I/O stream. In many platforms and programming environments, there are functions which can replace or redirect the input stream or output stream with another input stream or output stream. For example, Java language can use System.setIn() method to replace standard input stream with any other types of input streams, even network input stream. So another application can be written to replace or redirect the I/O stream of the application, and to expose the interface using Web service. Other cases are the applications that use command line arguments to read data. In many programming languages, there are functions which can execute a standalone executable file and specify the command line arguments.

- The remaining applications (most of them are *Graphical User Interface* applications) can neither use standard I/O functions nor be converted into shared libraries. To reuse such applications instead of rewriting them, it often results in a high cost. In order to reuse these applications, a Bypass library should be written. Then the library needed by the applications is replaced with the Bypass one. The Bypass library will interact with the APS. As can be seen, this method will incur high cost and not be efficient, therefore, it is not recommended.

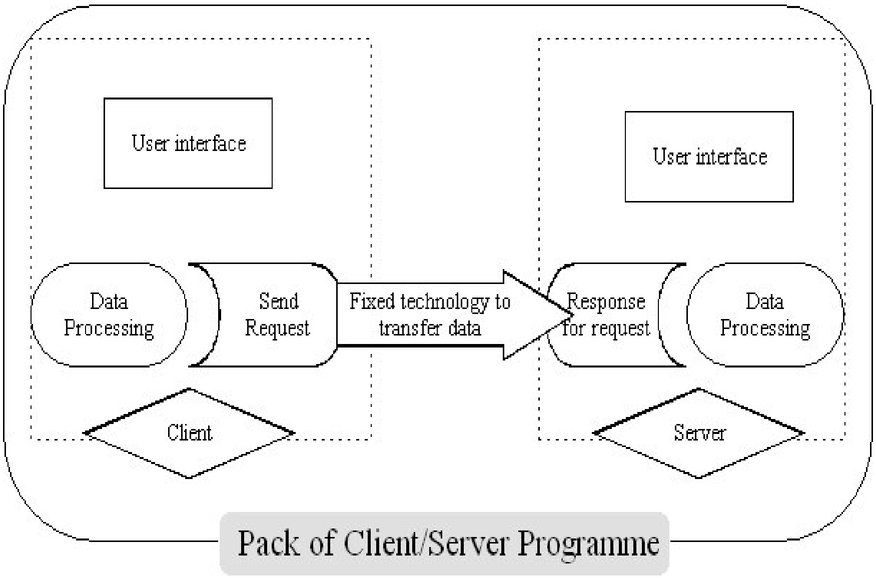


Figure 6.1 Traditional Network Program Architecture

B. Network enabled software packing

There are applications that are already network-supported; however they are developed with different technologies that are not service-oriented. Such applications cannot be viewed as a single programme as the standalone one does.

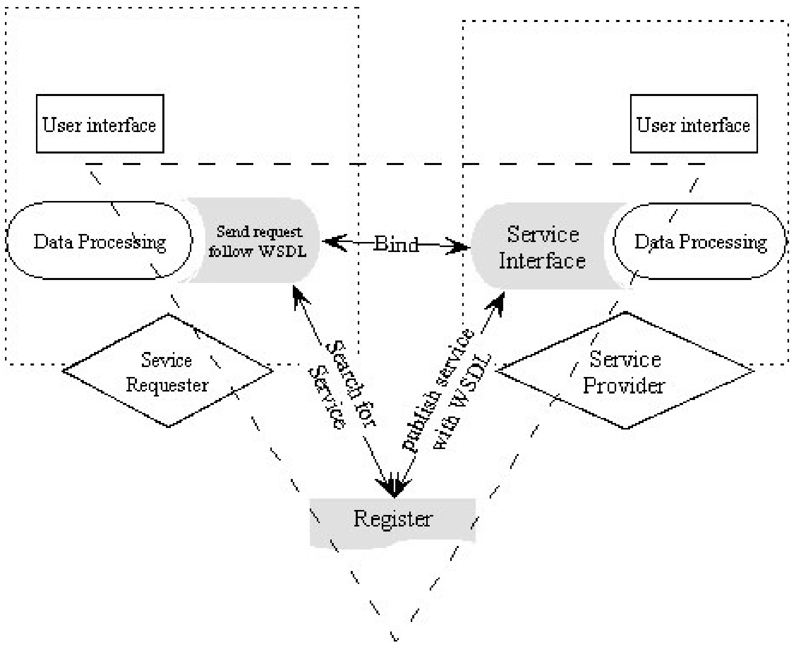


Figure 6.2 Service Oriented Programme Architecture

Most of the applications have two modules in a traditional view: client and server, as shown in Figure 6.1, and both of the two sides have to programme with compatible technology. The client and server have to be born in the same time; people cannot code the client program if they do not know the server code well. Usually, such program is coded by the same group of people for both of the two sides. However, to change it into service oriented, part of the code does need to be re-coded.

As shown in Figure 6.2, the structure of the applications includes the service requester module, the service provider module and the middleware module (sending request, service interface and register within the triangle area). This helps to change it into a Service-oriented application. No matter what the program language or technology has been used. Only the request and response functions are to be replaced. They can be rewritten following the service way, but the core code of data processing will still be reserved to work in the new model. Following this structure, multi-user access can be achieved.

C. Hard disk space and other resources packing

Hard disk space can be shared in network as a service, and its disk functions can be operated as well, such as *copy, paste, view property, sharing limitations* and even *delete* according to the authorization level. Participant, who wants to share his hard disk space to others via the service, needs to write a program in the format suitable for the service including the disk operation functions mentioned above. These functions are described in XML and published on the service registry as mentioned in Section 2. At the beginning, the user may do not know the location of the space provider; however, after having found the XML service description file in service registry, he is redirected to and bound with the provider, and is able to operate the shared spaces with the functions provided. The GridFTP is utilized in this research, which provided a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks [77].

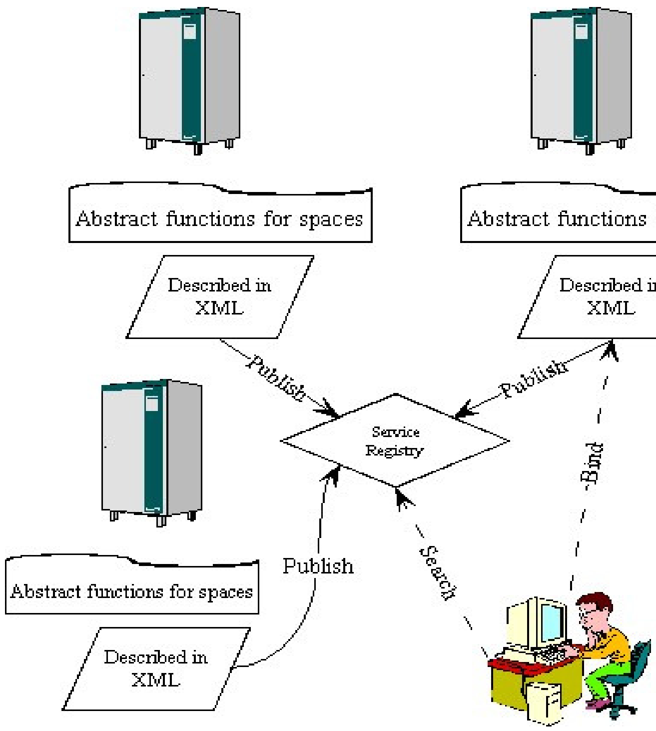


Figure 6.3 Service-Oriented Space sharing

As shown in Figure 6.3, the requester, i.e., the user, may not be aware of which provider he is bound to. According to his request, he could be bound with two or even more providers. For the user himself, he cannot feel that he is using resources on a huge network but feels that he is using his own computer and find a space which seems located in his own hard disk.

Just the same as the utilisation of the hard disk space, any other resources can be represented as applications and converted to be services in the Grid environment. Their functions are abstracted out first, packed as described in Section 6.2, and then published in service registry so that can be found by users.

All the resources described above can be wrapped using the application manager service, application proxy factory service and application proxy service models. The services should be stateful, and WSRF [78] can present such type of services. Take the application proxy service for example, each application proxy service should be associated with one active process of shared application, and also invoked by only one client application. If the application proxy service is stateless, the aim can not be achieved. The same situation happens in the implementation of the application proxy factory service.

6.4. System Implementation

In this section, a legacy application for gear design optimization, GearOpt [5], is used to illustrate the Service-Oriented Architecture (SOA) based approach for LAW. Rather than using Web service, Globus Toolkit is utilised in this research to develop the system so that the system can support Grid Computing. The original GearOpt is a single user version without distributed features. As the demand of using this application for gear design is increasing, the author decides to wrap it and enable it with network features, then, the Service-Oriented Architecture is employed.

The original application has the functions of Genetic Algorithm (GA) and numerical analysis for gear strength calculation to the British Standard BS 436. As shown in Figure 6.4(a), the two data files are used to setup the optimization specifications (goals, weight factors, population size and number of tests); the ‘exe’ file, which is the GA program, conducts the optimization and the numerical analysis program is invoked by the GA program in the optimization process to calculate the tooth strength. After the calculation, the results are stored in a folder named “demo”. This GA application was built long time ago. Compared with nowadays applications with friendly GUI interfaces, this application is not convenient to use for local users, as it lacks of network invoking features. However this program is still useful and powerful; to protect the copyright, the application owner is reluctant to distribute it away to other unauthorized users. To solve the two problems, the authors decide to wrap it using the service-oriented approach based on Grid.

The GA and numerical analysis programs are the core of the software. So the aim is to wrap the GA program and the numerical analysis program with service and to build a GUI client application to interact with the service provider. The input parameters of the GA program are stored in the '.dat' files and the location of the files is specified from command line arguments. In the Java language, the class Runtime has a method called exec. This method executes an executable file and specifies the command line arguments. After the GA program calculates the gear parameters, it produces the results as a file in its working directory. The Runtime.exec() method also specifies the working directory; so the input and output of the GA application can be redirected.

6.4.1 Implementation of Client Interface

As shown in figure 6.4(b), the Grid Portal (client application) is used by remote users to capture the input parameters. The parameters are obtained by the client step by step from the input of user (This process will be shown in the later experiment session). As could be see in Figure 6.4(b), the interface is pretty much friendly to the users. With the help of the Grid Portal, the user input will be generate as a dat file and this file will be transferred to the server.

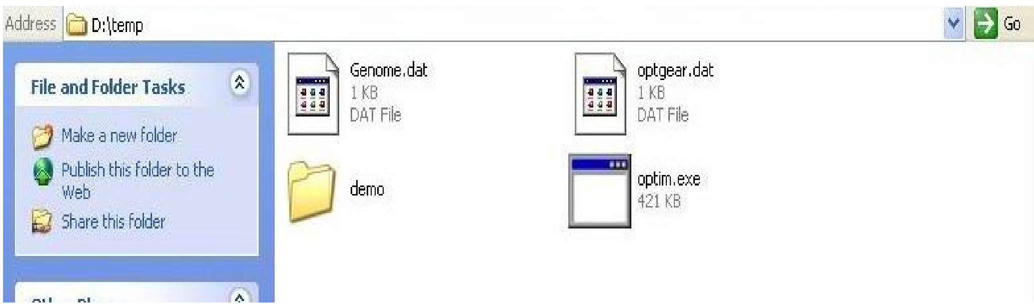


Figure 6.4(a) Original Application

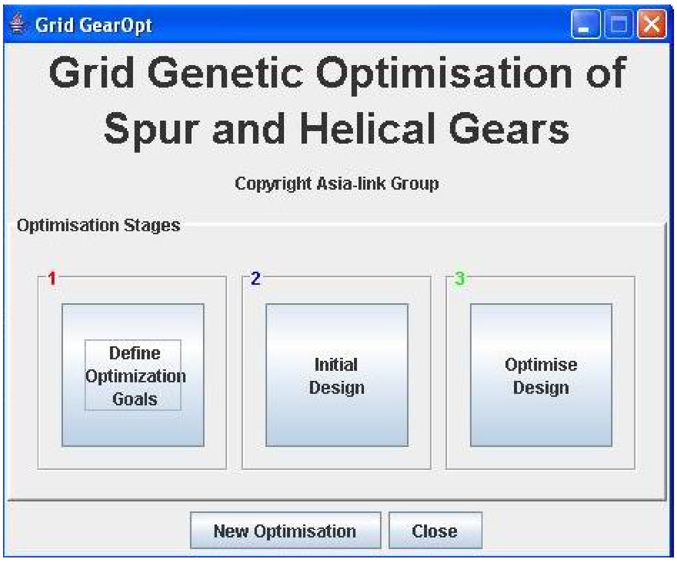


Figure 6.4(b) Main panel of the application

6.4.2 Implementation of services

The major tasks for implementation of the service based approach for software sharing include packing the application with an APS and implementation of the APFS and AMS. The development of the methods for conducting the tasks is detailed in this section.

6.4.2.1 Application packing

The APS represents a process of activating a shared application. When a request arrives, the APS delegates the request as local function calls to the associated process. The primary issue of this service is how to pack the GA application with an APS.

The GA application is a standalone executable file with the input from two dat files and the output to specified files in a folder. This kind of application is very difficult to be packed into services. In Java language, the System.setIn() method is used to replace standard input stream with a network input stream. So aided application was written to redirect the I/O stream of the original application, and then expose interface using service. APS is assigned to GA program by executeFile=D:\\temp\\optim.exe. What's more, the original optim.exe is for single user, it can only be executed once at one time in a single computer since it is developed. The authors need to change it into a multi-user enabled application, so before the APS invoke the application, APS should appoint the application run in one thread rather than in process, so that the application can simultaneously be executed many times in back office. For application in each thread, a separate folder to store its output and input is assigned.

6.4.2.2 The implementation of APFS

GA APFS has a function to create an APS. When a creating request arrives, the application proxy factory should check the status of the local computers to determine whether a new APS could be created or not. After a new APS is created, the APFS allocates the resource required by the APS, and then registers the APS to a proxy service catalogue. This catalogue is stored in XML format. APFS is assigned with one APS by apsURL=http://192.168.1.58:8080/ogsa/services/spb/APS. APFS should also specify the output location by outputDIR=D:\\temp.

6.4.2.3 The implementation of AMS

The main function of the AMS is to deal with the performance and security issues. The AMS provides a catalogue containing URIs of all registered APFSs. The application identifier identifies all the APFSs of a shared application. As mentioned in the above sections, each APFS represents a shared application in a computer. So the application identifier is used to look up all the APFSs of a shared application, in the authors' experimentation, three computers are equipped as service provider:

gearopt=http://192.168.1.58:8080/ogsa/services/spb/APFS,
http://192.168.1.57:8080/ogsa/services/spb/APFS
http://192.168.1.56:8080/ogsa/services/spb/APFS

The APFS catalogue is implemented via an interface, which connects to a relational database, typically a MySQL database. Each APFS is registered as a database entry containing an application identifier and URI of the application Proxy factory service. When looking up a specific shared application, the URIs of APFSs with the same application identifier should be returned. Then the AMS interacts with the APFSs using these URIs to create an APS.

In order to achieve higher performance, a shared application could be installed in multiple computers with an APS associated with each one. Then all the APFSs are registered to the AMS. When creating an APS, the AMS polls these APSs and chooses one, which has reported that there are enough resources to create a new APS.

There is also a table used to store the Access Control List (ACL) information for a shared application in the factory service catalogue. In the current version, the AMS use PKI (Public Key Infrastructure) [65] for access control. Each entry in the ACL table contains certificate information of each application identifier with its authorized users who could use the application. When a client application requests to use a shared application, it sends its certificate information in a SOAP (Simple Object Access Protocol) header to the application manager. Then the application manager checks whether the client application could use the shared application by using the ACL table. If the client application has access right, the AMS signs a certificate to the client application. This certificate expires after the client application finishes its tasks. In the last step, the AMS uses the APFS to create an APS, and then returns the handle of the APS to the client application, which finally uses the certificate given by the application manager to interact with the created APS.

6.5. The deployment of Services

```
- <service name="spb/APS" provider="Handler" style="wrapped" use="literal">
    <parameter name="operationProviders" value="org.globus.ogsa.impl.ogsi.FactoryProvider
      org.globus.ogsa.impl.ogsi.NotificationSourceProvider" />
    <parameter name="persistent" value="true" />
    <parameter name="instance-schemaPath" value="schema/GridGearOpt/APS/APSService.wsdl" />
    <parameter name="schemaPath" value="schema/ogsi/ogsi_notification_factory_service.wsdl" />
    <parameter name="baseClassName" value="org.globus.ogsa.impl.ogsi.GridServiceImpl" />
    <parameter name="handlerClass" value="org.globus.ogsa.handlers.RPCURIProvider" />
    <parameter name="className" value="org.gridforum.ogsi.NotificationFactory" />
    <parameter name="allowedMethods" value="*" />
    <parameter name="factoryCallback" value="org.globus.ogsa.impl.ogsi.DynamicFactoryCallbackImpl" />
    <parameter name="instance-baseClassName" value="edu.ntu.APS.impl.APSImpl" />
    <beanMapping qname="ns:GenomeVO" xmlns:ns="urn:APSService" languageSpecificType="java:edu.ntu.vo.GenomeVO" />
    <beanMapping qname="ns:OptGearVO" xmlns:ns="urn:APSService" languageSpecificType="java:edu.ntu.vo.OptGearVO" />
    <beanMapping qname="ns:ResultVO" xmlns:ns="urn:APSService" languageSpecificType="java:edu.ntu.vo.ResultVO" />
  </service>
```

Figure 6.5 Deployment document of APS

```
- <service name="spb/AMS" provider="Handler" style="wrapped" use="literal">
    <parameter name="baseClassName" value="edu.ntu.AMS.impl.AMSImpl" />
    <parameter name="allowedMethods" value="*" />
    <parameter name="persistent" value="true" />
    <parameter name="schemaPath" value="schema/GridGearOpt/AMS/AMSService.wsdl" />
    <parameter name="authorization" value="self" />
    <parameter name="handlerClass" value="org.globus.ogsa.handlers.RPCURIProvider" />
    <parameter name="className" value="edu.ntu.AMS.AMSPortType" />
  </service>
```

Figure 6.6 Deployment document of AMS

After the implementation, the services need to be deployed and published. A service description file should be written for each service to be recognize by the server container—Globus Platform. *.wsdd file is used to deploy services. In this research three files are written as Figure.6.5 represents the Deployment file of APS, Figure.6.6 represents the deployment file of APFS and Figure.6.7 represents the Deployment file of AMS.

```
- <service name="spb/APFS" provider="Handler" style="wrapped" use="literal">
    <parameter name="baseClassName" value="edu.ntu.APFS.impl.APFSImpl" />
    <parameter name="allowedMethods" value="*" />
    <parameter name="persistent" value="true" />
    <parameter name="schemaPath" value="schema/GridGearOpt/APFS/APFSService.wsdl" />
    <parameter name="authorization" value="self" />
    <parameter name="handlerClass" value="org.globus.ogsa.handlers.RPCURIProvider" />
    <parameter name="className" value="edu.ntu.APFS.APFSPortType" />
  </service>
```

Figure.6.7 Deployment document of APFS

6.6. Experiment

After the implementation of services and the preparation of deployment file, it is time for experiment. AMS is deployed in one central computer while APFS, APS are deployed in three different computers.

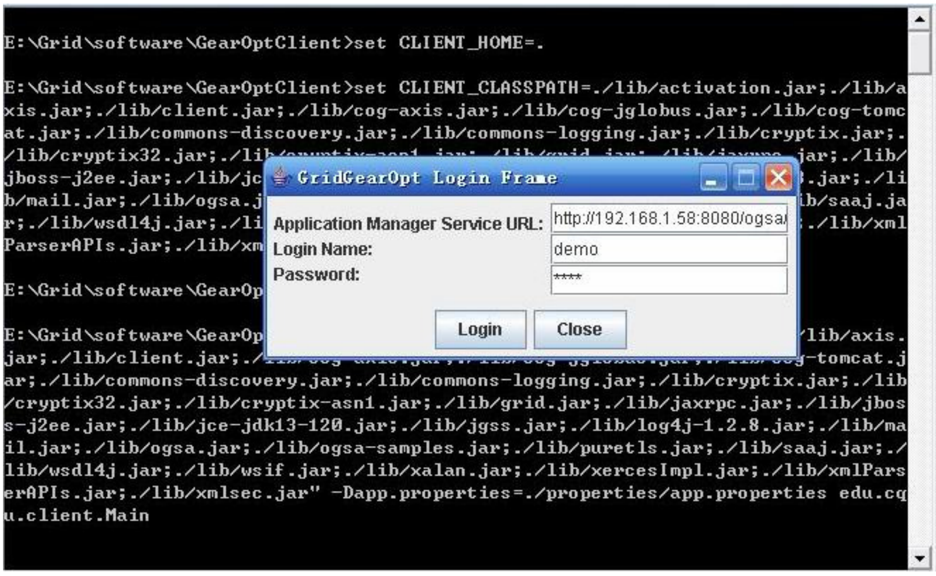


Figure 6.8 The login frame

Figure. 6.8 is the login frame when the application in client is started. After login into the system, the main frame (Figure. 5.5 in section 5.4) comes out. Then choose “Define Optimization Goals”, and input parameters in “Optimization Parameters” Panel, when ready, click on “continue”.

The list shown in Figure.6.9 is for the user to identify which design parameters are to be optimized. Selection of these parameters will include them in the optimization process. Non selection will freeze the values to those defined in the initial design. The process is capable of optimizing the performance of both variable and fixed centre distance gear pairs.

There are five goals: facewidth, contact ratio, centre distance, equal stress and equal slide/roll. The goals of the search are defined by adjusting the importance of the 5 fitness functions shown in Figure 6.10

The characteristics of the GA, i.e., population size and number of tests are set by the user with the aid of the form illustrated in Figure 6.11, which is the last panel of Define Optimization Goal, after click finish, the screen will change to entrance panel again (Figure. 5.5 in section 5.4), then it’s time to setup initial parameters of the Gear, click on “Initial Design”

The initial design provides the starting point of the optimization search, as shown in Figure6.12 and Figure 6.13. For this purpose, the basic configuration of the gear must be provided, including geometry, performance and material information. These are prompted for user input or default values are provided.

After click “finish”, figure 6.14 is shown until the final result comes out (Figure 6.15).

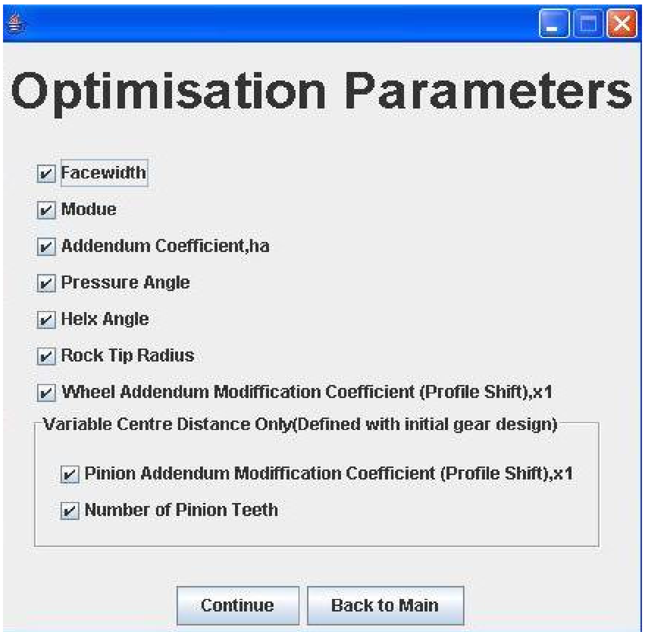


Figure 6.9 Parameters Panel

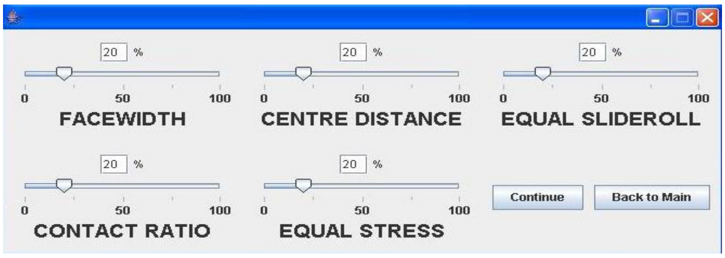


Figure 6.10 Parameter of Fitness

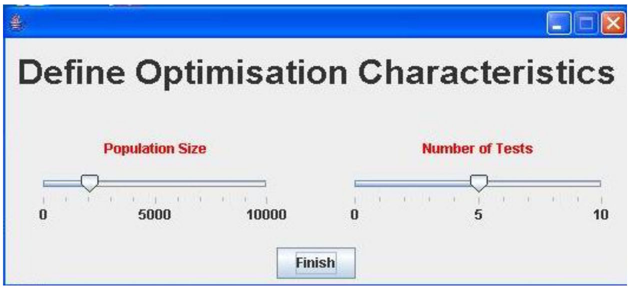


Figure 6.11 Population size

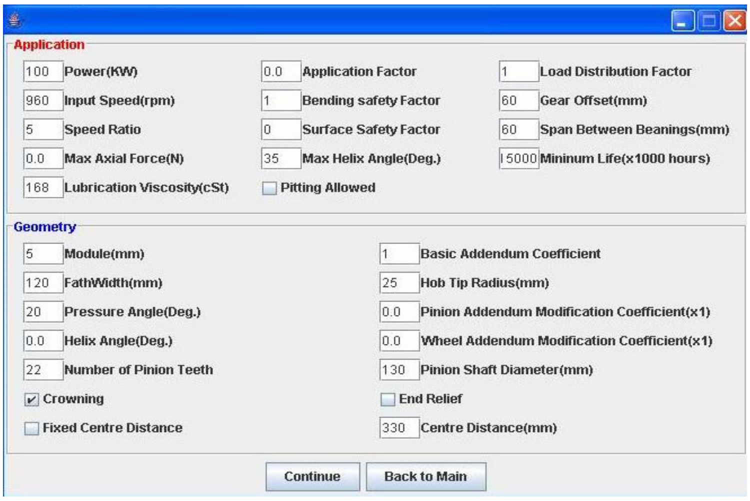


Figure 6.12 Parameters of Gear

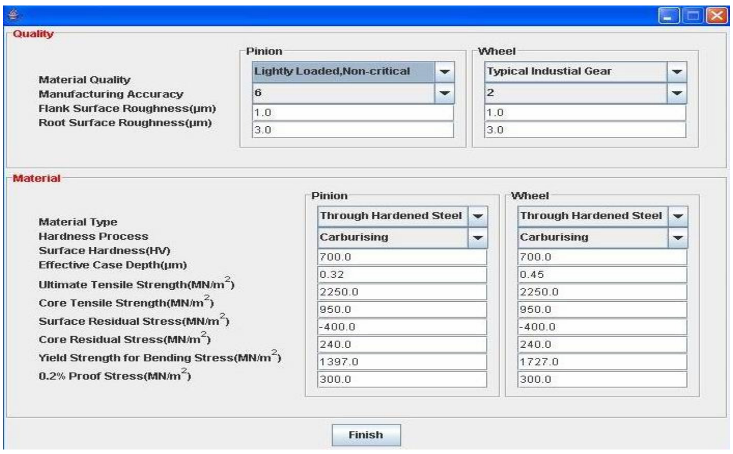


Figure 6.13 Material and Quality

It should be noticed that, the experiment is done using three computers located in three different locations, and they are providing the service in meantime.

Figure 6.14 Waiting panel

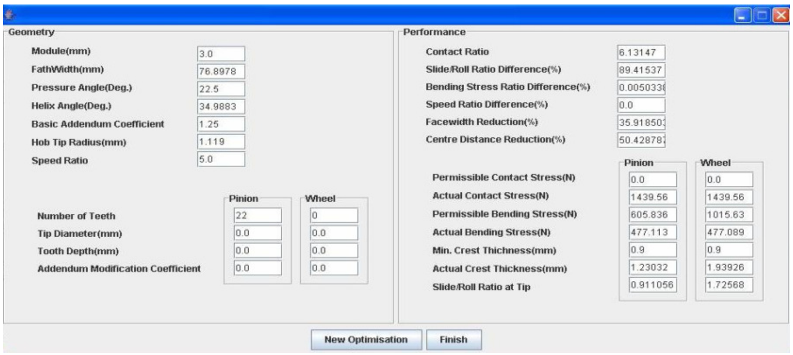


Figure 6.15 Result panel

6.7 Evaluation and limitations.

After the case study, the author received several feedbacks from the users. Generally the users felt positive that the service provided a much better quality of service. But some of the negative feedbacks were (1) if the input parameter requires a long processing period, the user cannot know the progress, some calculations took 1 minute, while some others took hours of calculation. A more friendly user interface should be considered such as making the users be aware of the percentage of progress made. (2) It would be much better if utilising the grid computing to conduct parallel algorithms in the future, so that the calculation process would be faster. (3) A more simple way to deploy the client is expected to be available. (4) Apart from the four types of application mentioned in the previous chapter, the solutions for other types of application need to be discussed. (5) The licence issue has not been solved. License issue should be a detailed project which need to be carried out. From an overall rating by the users, this application has got a higher satisfactory from the users than the SOF-CAD research module. And this is because this approach caused less interaction and fewer faults. All the feedback will be taken into consideration in future research.

6.8 Conclusion

To utilize Grid computing in a more practical way and to enrich the Grid resources, a method of converting traditional resources to services has been successfully developed. It can be used to pack different resources to fit the Grid architecture, including the packing for standalone software, network enabled software, hard disk space and other devices.

The outcome of this research has greatly enhanced the usability of Grid, and proved that existing traditional resources can be re-used for Grid with the method developed and compatible with most other advanced techniques.

The researches conduct so far successfully solved the resource interaction and sharing in a distributed system. So the next chapter discusses how to integrate these researches in a single distributed system, which has advanced features. The necessary features will also be discussed in next chapter.

Chapter 7

Framework of a Collaborative Working Environment for Integrated Design and Manufacture

7.1 Introduction

Sharing and interaction are actually two of the most important issues in a distributed system, the previous researches conducted by the author have targeted the two issues in a innovative SOA way; what is more, the two achievements also solved many problems brought by current approaches. So the previous researches have formed a solid basis to build a powerful collaborative working environment.

Based on all of the above mentioned research experiences together with the literatures in Chapter 3, a new CWE framework with an upperware-middleware structure is proposed, which works as a mediator and facilitator to provide collaborative services for the development of flexible, scalable and adaptable tools and applications in design and manufacturing engineering. This enables geographically dispersed team members to effectively collaborate with each other through the aid of seamless integration of Information and Communication Technologies (ICT).

The research includes the development of an overall structure of the framework, the methods how the upperware interacts with the underlying layers and the applications, and the middlewares to be developed in various Internet/Web and ICT techniques such as Grid, Web services, mobile agent and wireless computing. The techniques for resource wrapping as well as resource publishing, resource searching and resource locating are also proposed for the CWE. Section 7.2 is the overview of the proposed 4 layer approach. In section 7.3, the wrapper method introduced in Chapter 5, is re-emphasized to show its great contribution to the CWE. Section 7.4 discusses CWE related approaches done by ADMEC, which could be integrated and assist the CWE. Section 7.5 introduced the operational function of the system.

7.2 Overview of the CWE framework

The structure of the framework is shown in Figure 7.1. The upperware interacts with the applications and middlewares in the underlying layer, to provide specific services for collaboration combining the basic services provided by the middlewares and the tools.

A distributed approach is proposed which combines various middleware techniques including Grid computing, Web services, P2P and intelligent agents to enable the following features:

- Seamless collaboration: it supports resource sharing in dynamic and collaborative working environments and integration of distributed systems. It can support the share of controls of the contact, facilitates cooperation among clients in a network; enable a distributed searching network, facilitates contact grouping, and permits easy introduction of value-added services.

- Virtuosity: It enables interoperability across borders, encourages scientific/business collaborations between organizations, and allows remote job submission and management.

- Autonomy: It allows services to be executed without requiring the monitoring or continuous interaction of clients. The aim is to create services that display intelligent autonomous behaviour by being able to complete an assigned task by appropriately choosing from a set of possible strategies.

- Security: Information services provide information about available resources and status of resources, while security services provide single sign-on, authentication, and authorization and secure data transfer.

To achieve all the above is a novel and challenging task. No current existing technology is capable of providing all these features. In order to implement the integration and collaboration amongst the partners, the system has to be provided with the following four layer architecture, as shown in Figure 7.1 Mobile Agent technology is integrated in this approach to achieve the aims, this technology will be introduced in section 7.2.3.

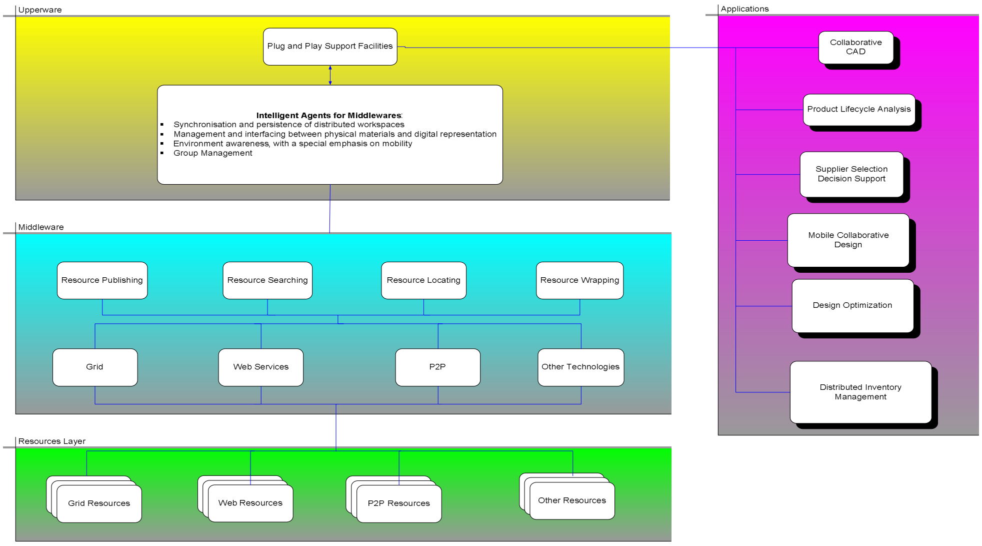


Figure 7.1 Structure of the proposed framework

7.2.1 Resources layer

The bottom layer is the Resource layer. The physical resources would be wrapped into specific technology enabled ones so that they could be recognized by the collaborative working environment, this system should support four resources: Grid, Web Services, and P2P. These resources exist independently in the system.

7.2.2 Middleware layer

The middleware is a class of software technologies to manage the complexity and heterogeneity inherent in distributed systems. It is defined as a layer of software above the operating system but below the application programs. This layer provides a common programming abstraction across a distributed system and what's more, it connects parts of a distributed application with data pipes and then passes data between them. It has two parts:

The first part mainly aims to coordinate three main enabling middleware techniques including Grid, Web services, and P2P. For the large computing or sharing of the distributed resources, Grid computing is the ultimate vision in distributed computing, spanning locations, organizations, hardware and software boundaries to provide tremendous power. Different techniques have different advantages in different fields. The task to determine when and which technique is to be used will be carried out in the upperware layer.

The second part is to provide resource management in the system. It will provide functionalities to publish, search, locate and wrap to coordinate the resources, and to control the transmission of all feature model resources in the system. Traditional resource discovery is based mainly on centralized and hierarchical modes, which are inefficient in finding other resources. In the management system of the proposed framework, each node periodically notifies the network of its presence, discovering its neighbours at the some time, which resolves the above problem; existing resources, no matter where they reside in a network, are integrated together and allowed to communicate with each other. The resources are encapsulated into services and each component can be found and invoked.

Basic functions for the possible types of resources should be produced here and all the resources in the system are operated and managed. The main functions are described below:

- *Resources registration, discovery and allocation*. In the system, all the resources are packed into recognizable resources and registered in a registry centre so that they could be found. Instead of storing the service information on their own servers, the client, who wants to publish services, makes the information available to registration server. The service requester can be re-directed to the publisher's service by the registration server..

- *Index and repositories system*. The research focuses on creating an indexing system of clients and of available services. The initial proposition is to create an index of all clients and their resources on large indices in a central server. When a query is sent to a registration server, the server looks-up the index; if the query can be resolved, then the server sends a message to the originator to inform where to get the service or file. Although this method can provide the best performance for resource discovery, the server in centralized indexes and repositories system is expensive, the bandwidth and hardware required to support large networks of clients are expensive too. If the server in the system fails to function properly, it brings down the network. To avoid the above problem, the system has a cluster of servers, and, hence, if one server fails, the rest of the servers will continue to support the network. Recent court rulings cast serious doubt about the liability involved in using centralized servers to index resources in a peer based network [79]. This research adopts the decentralized server method to avoid the drawbacks of current centralized index approaches. The ability to monitor and enforce this requirement is quite challenging, and may be too much of a risk. Further research will be conducted to determine possible alternative index or discovery schemes to handle resource discovery and use.

- *Resource Wrapping*. Some resources may not be directly associated with any kind of the middleware technology, so resource wrapping should be taken into consideration, which migrate different CAD/CAM legacy software systems to distributed collaborative resources in CWE. This research has been conducted and introduced in Chapter 5 and Chapter 6.

7.2.3 Upperware layer

The 'upperware layer' interacts with the applications and middlewares to provide specific services for collaboration combining the basic services provided by the middlewares and the tools. It includes two main parts, intelligent Agents and Plug and Play supporting Facilities.

(a) Intelligent agents for middleware

The intelligent agents are employed to enable the following main features:

- Coordination of the utilisation of multiple middleware techniques. Some resources may involve more than one kind of Web/Internet technique, so the coordination should be taken into consideration. For example, if a client finds a Grid Service and this service needs frequent interaction between the two parties, then mobile agent technology should take this into consideration. Using two or more kinds of the advanced technologies is a challenging task in such a layer. The research focuses on modular design, resource sharing and data communication using different technologies. This layer also can intelligently tell what the suitable technologies for the selected resources are, and then use the right method to carry out related work for the request from the user or system.

- Process control of concurrence and consistency as well as synchronous and/or asynchronous messages. Process, elements (security, services, monitoring, etc) can be shared across applications to provide horizontal services to decouple these reusable application components, facilitating more rapid changes in these processes.

- Group Task Coordinator is responsible for the task assignment of each requester, monitoring the action state of each requester and accepting the communication request and service from each provider.

- To provide an interface to the worker centric design tools and establish the connection between the design tools and the middleware techniques specified in the underlying layer.

(b) Plug and play facilities

The framework is built on top of service component-connection modules where new functionality can be simply inserted and used. It supports the reuse, integration, management and execution of distributed software elements and packaged applications, and allows movement of data and services.

It supports plug and Play, to make the system better distributed will enable developers to deliver rich Internet applications. The applications or resources could be easily plugged into the supporting facilities to enable them work well within the collaborative working environment.

7.2.4 The applications

The upperware provides facilities for applications to plug into this layer. They will be adapted or produced to follow the standards specified in this layer. "Customizable" plug-in functions are utilised in the system to provide a flexible mean to add functions into the system in the future to fulfil tasks for other collaborative work areas. Necessary applications to the design and manufacture are to be implemented by all of the partners using their own systems, which will ensure the use of heterogeneous systems. More and more services could be added to the system no matter what collaborative work areas they belong to (for example, Virtual Learning System, E-business system).

As for the collaborative design and manufacture, related tools for enabling fast and flexible manufacturing can be integrated into the platform, such as online semantic collaborative CAD/CAM, Virtual design engineering and mobile devices.

7.3 Migrating legacy software systems to Web-enabled collaborative resources through an automatic wrapper generation technique

Because this approach has been detailed described in Chapter 5 and Chapter 6, so this section mainly introduces how this approach contributes to the whole system. The objective of this part of the research is to develop and to implement a generic service wrapper for automatically migrating different legacy software systems to distributed collaborative resources in the CWE.

The innovative changes of Internet environments and the introduction of Web Services eases the transition from mainframe based centralized legacy systems to more flexible service-oriented distributed systems. Existing legacy software systems use system software and application software designed and implemented with older technology (that is, without distributed concepts, object technology, and Web services technology). Although legacy systems were implemented with older technology, they may still provide value by performing crucial work, and they usually represent a significant investment and years of accumulated experience and knowledge.

7.3.1 The wrapping method

There are three strategies for introducing Web middleware concepts into existing legacy software systems, i.e. redeveloping from scratch, reengineering approach with code conversing and wrapping with unified interface. Each approach has advantages and disadvantages. Wrapping provides the best compromise. It is a method of encapsulating that provides clients with well-known interfaces for accessing legacy systems. The advantage of wrapping is that legacy systems become part of the new generation of applications without discarding the value of legacy applications. Such wrappers also allow application owners to safely expose existing systems to both internal and external users without revealing their essential (and proprietary) code, and can work with a variety of communication methods via configurable protocols.

To use wrapping, application developers must understand and implement the interfacing techniques to legacy systems. The goal of this research is to investigate different types of problems that should be solved in order to wrap legacy systems and suggest an effective wrapping method. It is necessary to construct extensible wrapping template classes for various interface types and research and develop a wrapper generator in order to alleviate an application developers' burden in providing their services. By using this method, it is possible to extend the usefulness of legacy applications by facilitating their migration to service-oriented distributed environments with minimum effort. The previous three services approach has categorized few types of legacy applications.

Exposing the legacy application in a collaborative form of service components such as a node application on Grid, a service based on Web Services and so on, makes it available to clients outside the Intranet using respective standard Protocols. This procedure is adaptive to the selected underlying architecture such as Grid, or Web services. The wrappers are used as tools facilitated in the upperware layer. The basic functions of the wrapper are to receive calls of incoming protocol such as SOAP, to prepare messages before invoking the legacy application, and to invoke the legacy application.

7.3.2 Issues to automate wrapper generation

To migrate legacy systems into collaborative environments, the following issues have to be addressed:

- Variety of the interfaces to legacy systems. There are many interfacing styles in legacy systems. They have different implementations to each other and dedicated proprietary. Thus it is difficult for server-side application developers to implement wrapper objects for legacy systems, even when understanding some of the interfaces to the legacy systems. To cope with this problem, the Wrapper Template Classes and an automatic wrapper generator set are developed to free the providers from understanding and implementing various interfacing techniques.

- Representation of interfaces to legacy systems. To generate a wrapper automatically, a server-side developer or provider should submit interfacing information for legacy systems to an automatic wrapper generator. Thus, some representations are required to describe easily the interfaces to legacy systems. For this purpose, the WSDL (Web Service Definition Language) is used. In any case, a server-side provider should describe the services of legacy systems with WSDL for clients; in other words, the server-side provider has to describe the interfaces to wrapper tools. The Extended WSDL will provide a way of describing the services of legacy systems for clients and another way of describing actual interfaces to legacy systems for a wrapper generator.

7.3.3 Benefits of the wrapping approach

The benefits of the wrapping approach are as follows:

- Enhancement of reusability. By using previously developed programs through wrapper tools, they are reused at the level of executable codes.

- Reduction of software development cost. The previously developed programs have high reliability since they have been used and tested for a long time. Rather than redesign and redevelop programs with the same functionality, wrapping legacy systems should reduce both development and testing cost.

- Location transparency of server applications. Service wrapper tool for legacy applications have a location transparency feature. Thus, although a wrapper service can be located anywhere on the network, the provider who uses a wrapper service is able to construct client applications without any consideration about the location of legacy applications.

- Code security and easily joining. The wrapper automation generator helps to avoid the need of knowing about Web technology by resource provider and without revealing resource code to wrapping developer.

- Flexible wrapper for different resources to different interface. Resources are heterogeneous, i.e., written in different language, running in different platforms. In addition, due to using multiple middleware and multiple protocols in this environment, Wrapper must be facilitated with adaptive procedure for different resources to proper wrapping interface. This is dependent on the need of collaborative resource function. For example, if it is computing resource, it will be wrapped in Grid protocol and interface; if it is interactive commercial CAD software system, it will be wrapped in Web services, and so on..

7.4 Utilisation of the Framework for Collaborative Design and Manufacture

This PhD project is part of an on-going research project conducted by the Advanced Design and Manufacturing Engineering Centre (ADMEC) [80], which is to develop a collaborative work environment for design and manufacture. The total process of collaborative design and manufacture has been considered, including formulation of product design specification, supplier selection, conceptual design, detail design manufacture, inventory management after sale services and recycling. The CWE framework is utilised to integrate the following applications developed by the ADMEC and the consortia of Asia IT&C and Asia Link projects:

(1) A service-oriented approach for online software sharing. It enables the users to share software resources over the Internet. It consists of shared applications, client applications and three types of services: application proxy service, proxy implementation service and application manager service. With the aids of the services, the client applications interact with the shared applications to implement a software sharing task. The approach satisfies the requirements of copyright protection and reuse of legacy codes. A case study of online sharing a software package of gear design optimisation was conducted to demonstrate the approach developed [81].

(2) Web-service supported collaborative CAD. With this approach, the geographically dispersed team members can collaborate over the Internet to conduct computer aided design with tenures of online modifying CAD drawings. The advanced technique of Web Services is utilized for the users to provide and to request the services, and the most popular CAD software AutoCAD is used as vehicle to illustrate the methods developed' The method developed is in principle applicable for other CAD software packages such as Pro/Engineer and Solidworks. It can be operated in computers with different operation systems located in different locations [41].

(3) A Grid based system for a virtual manufacture/retailer organization (inventory control). It enables a large number of manufacturers and retailers to cooperate with each other within a Virtual Organization by sharing inventory information. This system adopts a mode based on inventory management to enhance the Quality of Service, which would greatly facilitate the business processes between the manufacturers and retailers. With the support of Grid computing, the system is more powerful than other existing systems and has considerable advantages of more accurate calculations and more quick response [82]. In the next chapter, this module

will be detailed analyzed as a case study to show how the grid working environment could help future supply chain management.

The three projects are directly related to this PhD project and the support to inventory management is taken as a case study to illustrate the utilisation of the framework, which will be detailed in the next chapter.

7.5 A Test Example

The basic functions of the four layer system have been implemented. This system can search and locate the services, invoke and automatically make the searched services as a plugin to the system. The modules in the four layers are implemented and tested using a simple service example. This laboratory experiment provides a applicable proof on whether the CWE is feasible before carry out the case study application presented in the next step (Chapter 8)

Within the test example, the framework is implemented with JAVA and Globus Toolkit. For the convenience of users, the design of the interfaces has borrowed some ideas from the interface of Microsoft MSN Messenger. A simple service was produced to test the functions of searching, locating sharing etc. The example service is to provide an adding operation service to the users, which requests input of two parameters from the service consumer and returns the result from the service provider to the consumer. It tested the interaction between service provider and service consumer; also, it proved the functions of service publishing, service locating, service searching of CWE. The test procedure is illustrated below.

1.) Login Frame:

Figure 7.2 Login frame

This nick name will be used together with the IP address which is the unique identification of a user. After filled the input of the name, the "Enter" button allows the user to login.

2.) Main Frame:

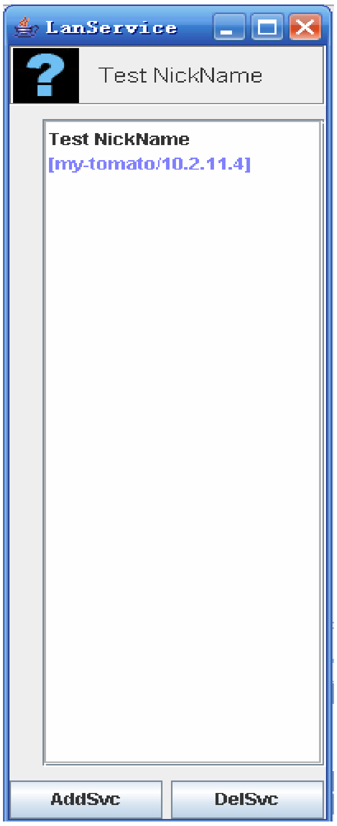


Figure 7.3 Main frame

After the login process, a frame similar to MSN appears, which is the main frame, the main frame shows all the friend list in the network, the user could check which services are shared by which friend by double click on the friend's name, every friend's name shown on the main frame is also follow the format of [Nicename/IP Address]. The following figure is the screen print of framework; it's only one friend in the list because it's only for test. My-tomato is the username and 10.2.11.4 is the IP address, those two identifiers became unique to identify a user. The two buttons at the bottom of the frame is for adding or deleting the services.

3.) Panel of adding service.

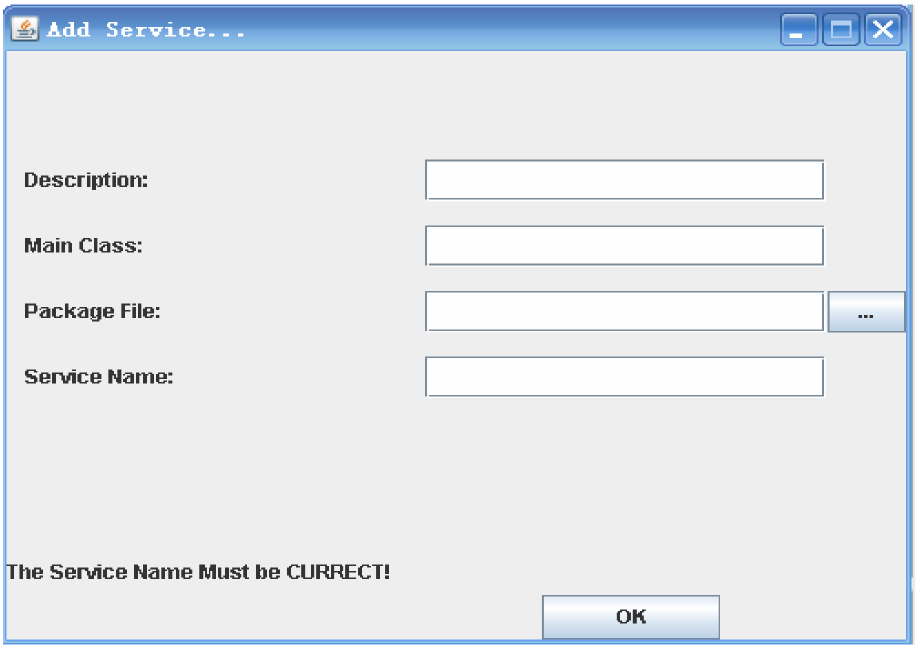


Figure 7.4 Panel for add service

Click on the "AddSvc" Button, the adding service panel appears, which enables user to add his own services to the Grid. The parameters include:

Description, Describe the service to other users.

MainClass, The Class name in the client where the main function is.

PackageFile, The location of service package.

ServiceName: The name of the service.

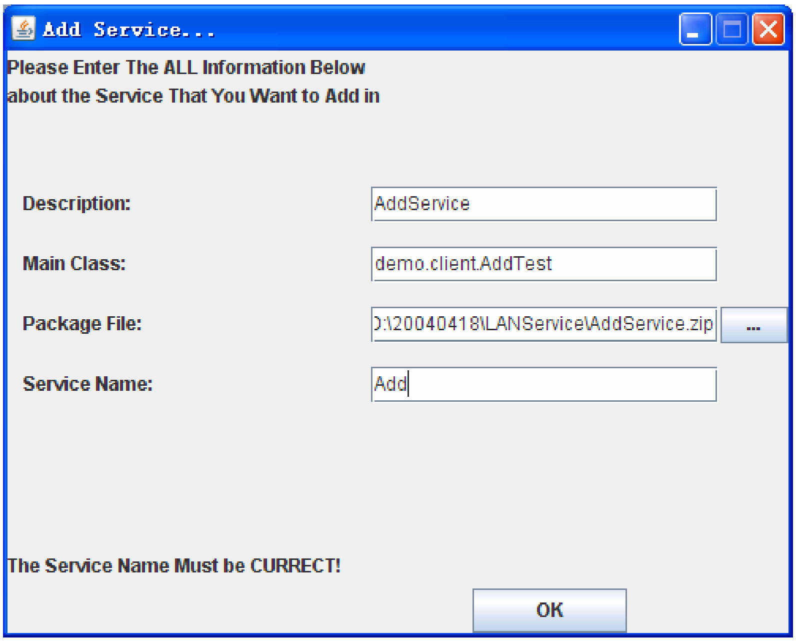


Figure 7.5 Fill parameters into the panel

Apart from the Description, any other parameters should be filled in an accurate format, and they are Case sensitive. Figure 7.5 gives an example how to input the parameters. Some principles of how to develop the services will be introduced later.

The above figure gives an example to add a service named "Add", where its main function should be in demo. Client Add Test Class in client. And the service now is in the server D:\20040418\LANService\AddSerice.zip. The other users will see the name "AddService" when they are browsing the service.

4.） Delete a service：

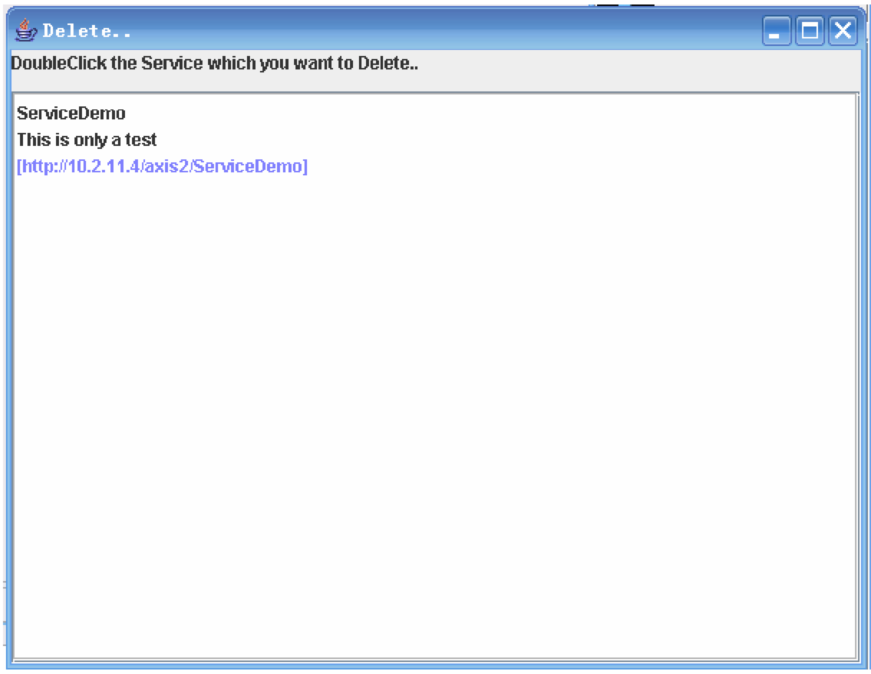


Figure 7.6 Panel for deleting services

If the user does not want to share a service any longer, he needs to use the function of deleting. In the main frame, after click on" DelSvc", the list of services appears(Figure 7.6), double click on the service which should be deleted, this service will be gone.

5.） To run a service in Service List.

Double click on the identifier of the user, which is "my_tomato/10.2.11.4" in this example, the service which is provided by this user will be listed in the Figure 7.7. To run the service, simply double click on the name of the service. A service parameter panel appears and after click on add, the request was sent to the service provider and when the service is called and the result of the calculation is returned, it will be shown in the jump out window(as in the Left of Figure 7.8).

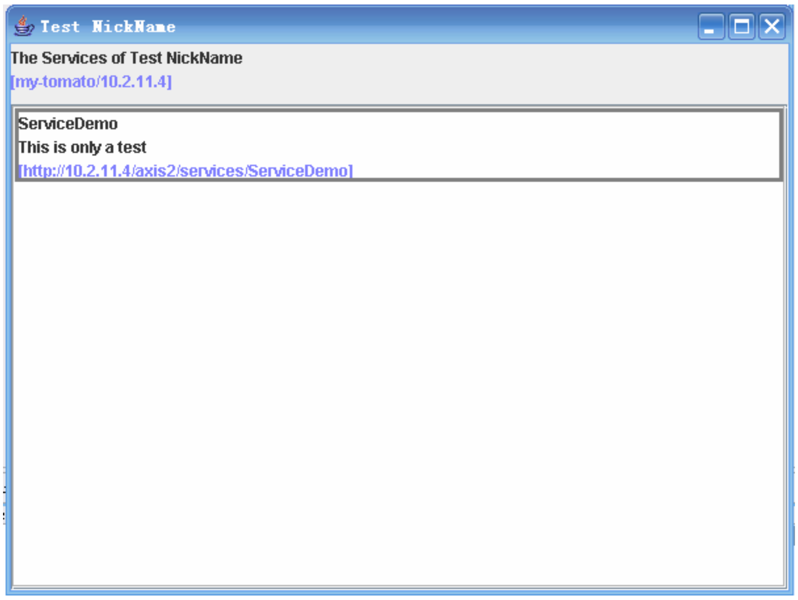


Figure 7.7 Panel of Service List

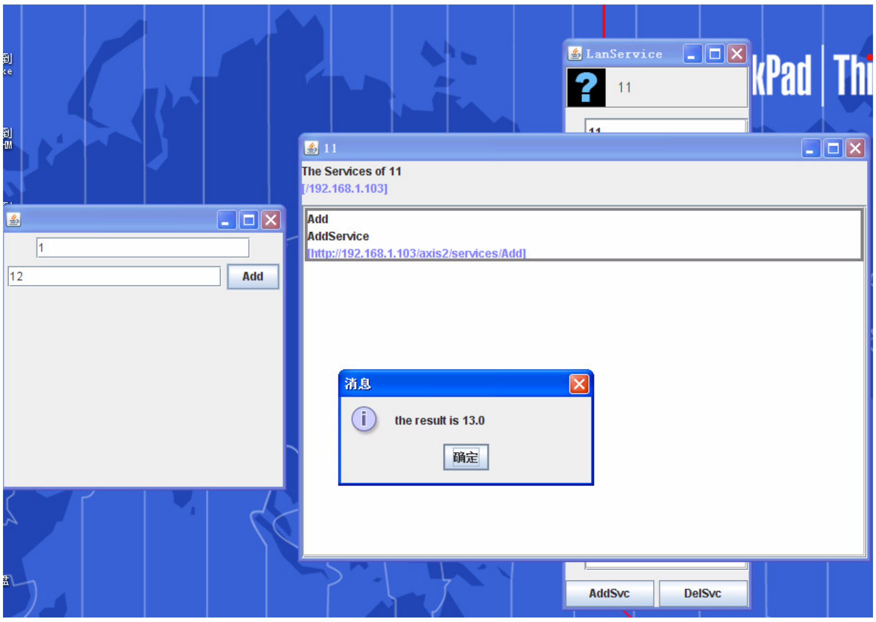


Figure 7.8 Call the service and get result

7.6 Concluding remarks

A CWE framework with upperware-middleware architecture is proposed. The CWE framework consists of four layers: resource layer, middleware layer, upperware layer and application layer. Within the framework, different middleware techniques are utilised such as Grid computing, Web services and Mobile agents, which offer the following features: seamless collaboration, virtuosity, autonomy, mobility and security. To achieve all the above is a novel and challenging research.

The wrapping methods and their benefits, as well as the issues related to the methods have been presented in this chapter. The wrapping methods provide useful means for developing and implementing a generic service wrapper for automatically migrating different legacy software systems to distributed collaborative resources in the CWE.

Finally a test example has been conducted to testify the theory; the implemented application can successfully search, locate, or invoke any grid services.

To make use of this approach the author need to apply it into a application area, the next chapter discusses how this approach helps the inventory management for manufacturer and retailer network.

Chapter 8

A Grid Service Based Collaborative Working Environment for Inventory Management

8.1. Background information of the company

In this Chapter, the J-Mortal Bike Company is taken as a case study to present the Grid service collaborative working environment developed by this research for inventory management. The company is one of the top three Mortal Bike manufacturers in China. Because of the commercial confidentiality reason, the author intentionally replaces the real name of the company with J-Company. This company sells products to about 2,700 county level areas amongst about 44821 town level areas in China; in addition, the products are also being sold worldwide, including America, Africa, Oceania, Europe etc. In 2006, around 1.4 million Mortal Bikes were sold by the company. However, imagine that there are tens of thousands or retailers existing and the company gets limited support from the business model and information system, so the company has to build a selling chain from the plant to the retailers to deal with the situation. Figure 8.1 shows an example of selling chain, linking retailer A in Shilin Town to the Plant. Products are delivered to five local distributors one by one and finally reach the retailers. Once the structure is fixed, it's not allowed to sell the product cross the distributors, for example, the plant can not sell the product to the retailer A directly because of the commercial moral.

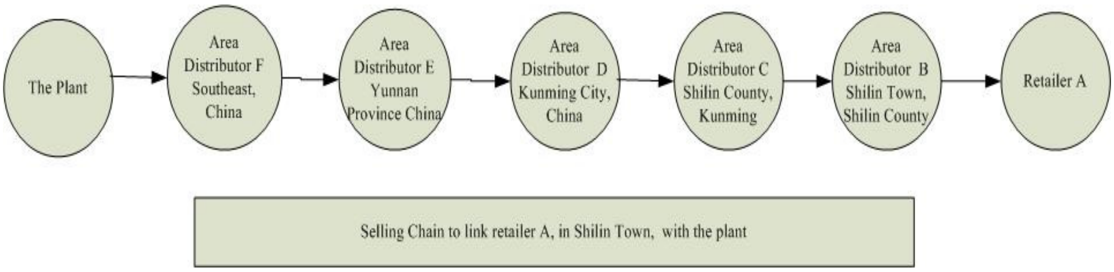


Figure 8.1 Selling Chain

Imagine that there is an order of 5 mortal bikes from a customer of Retailer A. Retailer A will order from the area Distributor B in Shilin County, the rest may be deduced by analogy. So a very similar chain to Selling Chain is formed, which is the order flow as Figure 8.2.

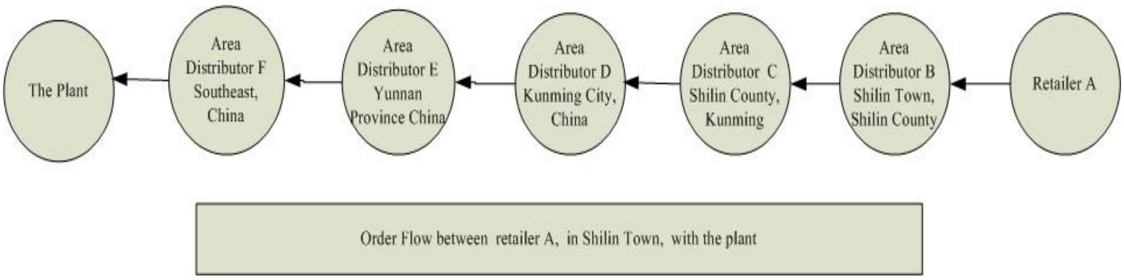


Figure 8.2 Order Flow

8.2. The problems and the bull whip effect.

This typical multi notes chain formed tens of years ago, however, too many nodes in the chain brings trouble, or even a disaster, to the company. Before talking about the problem, it is necessary to understand the role of the inventory.

Inventory has the function of balancing between production and demand. The retailer may not make an order to his upstream distributor until a request from customer is arrived; instead, he keeps a certain quantity of goods in stock so that he could provide better service to the customer- less waiting time. The same principle can be applied in each node in the selling chain. This means that each node keeps a number of goods in its inventory. While enough inventories provide a good service to customer, the redundant inventory became a threat towards every one in the chain.

This threat is called bullwhip effect. For example, retailer A, normally, could sell 5 Mortal Bikes in July from his experience; however, the uncertainty makes him feel that there is a possibility that he could sell 4 to 6 mortal bikes. So when he makes the order to the upstream distributor B, he requests 6 Mortal bikes because he does not want to lose any customer. When distributor B receives the order of 6 Mortal Bikes, the similar principle is applied in his decision, so he orders 7 mortal bikes from distributor C and so on so forth; when the factory finally receives the order from distributor F, it becomes 11 Mortal Bikes. So, in this selling chain, retailer A only sells actually 5 mortal bikes to the customer, but the factory has to produce 11 mortal bikes, more than twice of the sold mortal bikes. This problem is called bullwhip effect (Figure 8.3). It’s obvious that the multi nodes/levels situation in the selling chain is the main cause of the bullwhip. However, China is too big; it’s a very difficult task to reduce the nodes unless two aspects of the problem could be solved: new selling mechanism is created; new supporting Internet technology and systems emerges. To solve this big problem, the two aspects have to be considered together.

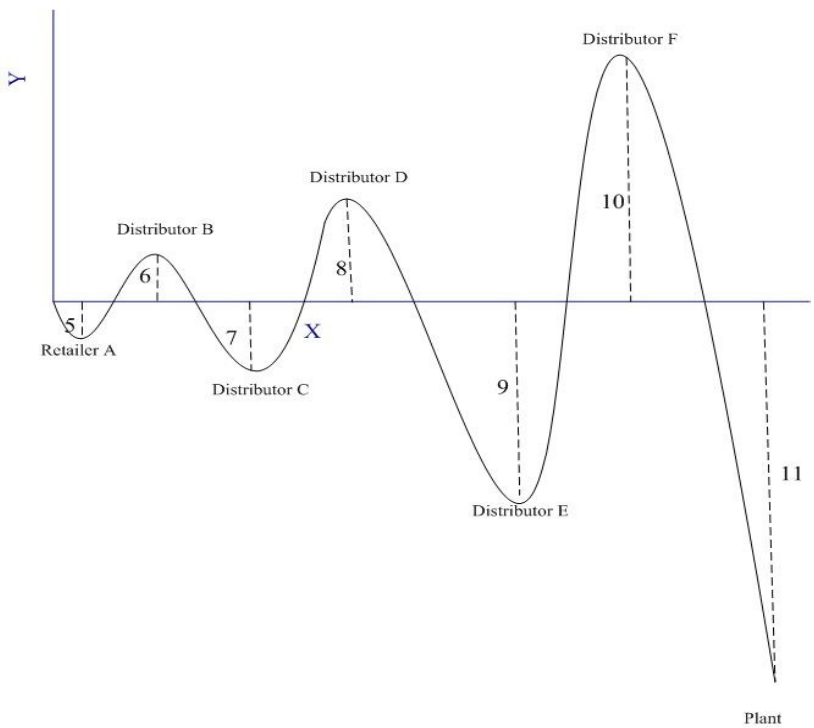


Figure 8.3 The bullwhip effect

8.3 Utilizing Internet technologies to implement the business models.

The Internet is changing the way we think and act in the world of manufacturing. Days are gone when plants operated in a vacuum with a set number of local suppliers and distributors. The Internet is making the manufacturing world truly global, connecting buyers and sellers all over the world. In 2006, B2B e-commerce has toped $2.2 trillion just only in EU while Internet is acting as a key driving force [83]. Web/Internet based systems are playing as a very important supporting medium between participants and the Internet. Various of Information systems are built and used in different aspect of business, Gregory E. Kersten [84] have developed a WWW-based negotiation system to help make decisions while users are distant away. Biller [85] has presented a dynamic pricing model for automotive industry based on Internet, etc. Supply chain management, which is one of the most important areas of business, has been given extremely attentions. Moreno Muffatto [86] has studied four cases before carries out a supply chain integrated business model based on the Internet and claims its "revolutionary", Somendra Pant [87] presents an implementation framework which was installed by two companies and proves that it is efficient. More and more researches which are Internet based have been done for supply chain management.

Supply Chain Management aims to reduce operating costs, lead time, and product inventory; it also aims to increase the speed of delivery, product availability, and customer satisfaction. Many companies have committed themselves to optimize the supply chain management system. The key to successfully managing a supply chain is to plan and to control the inventories and activities as an integrated single entity [88]. So many inventory management models were developed and have already been carried into practice; among them, the most widely used models are: Just In Time (JIT) and VMI (Vendor Managed Inventory). Different models are used to face different situations, for example, JIT inventory management systems most commonly are applied to heavy manufacturers concerns, whereby units are delivered within hours of when they are needed [89]. VMI aims to monitor retailers' inventories commonly related to commodities like apparels. JIT is a concept for producing a required volume of a required item at a required point in time, so JIT is considered to be an order driven model, which is still belong to a traditional supply chain. Different from JIT, VMI is a supply chain strategy where the vendor or supplier is given responsibility to manage the customer's stock [90].

From various of literatures about supply chain management, it can be concluded that Information is casting a significant role in supply chain ; Right and instant information can save and strengthen an enterprise while improper and late information could ruin it. Therefore, various of information systems come out to enhance the performance of supply chain management including enterprise resource planning (ERP) systems, warehouse management systems (WMS), transportation management systems (TMS) etc. However, most of these systems can only deal with work related to local management; they can not support cross enterprise or organization work. The information nowadays needs to be shared across organizations, for example, in inventory management, sometimes the retailers may have too many stocks for a product to be sold; What's more, they even have to be charged a high inventory holding cost; in the meantime, the manufacturer may not be aware of the retailers' perplexed situations and still keep producing the product which could hardly be sold, and this makes the situation even worse. Therefore both the retailers and manufacturers want to reduce the stock. However there is another scenario, if the product is out of stock while the manufacturer cannot produce it immediately, this leads to lost sale. Both situations are caused by insufficient information and should be avoided. Electronic Data Interchange (EDI) is employed to enable traditional information systems to work for multiple organizations. With the help of EDI, the local systems are linked [91]. However, EDI systems can not provide additional functions for logistics control across organizations. They can align with the structure of networked organizations, but lack the intelligence for co-ordination across the supply chain [92]. Furthermore, the overall business environment is becoming increasingly dynamic. Demand and supply for certain products can be very dynamic, supply chain operated in a network environment with dynamic information and decision-making models are called to accommodate these changes and uncertainties. There are some approaches to model and analyze the supply chain dynamics. In the area of inventory management, many researches were done as well to support dynamic and cross-organization inventory management [93]. Dr. Verwijmeren has presented the development of a distributed inventory management system, and claimed it's dynamic; however, the drawbacks of its object-oriented system architecture in distributed system are obvious [94].

The aforementioned models or systems mainly solve the specific problems encountered, for each of the research done, a solid system or a framework for a solid system is developed only, if the business environment changed, these systems can hardly be updated and follow, even if they are flexible and dynamic, they could not support future technology as well. Also, the existing systems are limited in a lower level of perspective in building information systems, they are only a specific problem solver and can not connect large number of enterprises; to most of them, any changes happen in future will lead to the rebuilding of the system or at least partly re-code of the system. In this chapter, a system for inventory management is presented which gets the support from the previous proposed Collaborative Working Environment Platform and provides a very strong basis to connect huge numbers of companies, stores, enterprises etc. This system also considers the extensibility of functions, the future upgrading of system, the integration with other systems, and the compatibility of future technologies.

8.4 Virtual Enterprise

In this application, the concept of Virtual Enterprise (VE) is employed to help J-Company deal with routine works. The notion of "virtual enterprises" has been around for quite some time and in some ways most IT efforts are concentrated on realizing some of the features of virtual enterprises, even without recognizing or labeling them as such. The meaning of the term "virtual enterprise" itself has undergone several versions or modifications. Some synonyms include "virtual corporation", "virtual organization", and, more recently, "extended enterprises" [95].

VE can be described as a Network of Enterprise from which temporary alignments are formed. This alignment combines the specific core capabilities of its members in order to rapidly exploit manufacturing opportunities associated with a specific product or service [96]. CWE is just the implementation of the VE concept, and CWE also provides all the facilities which are necessary to a VE. And this will be illustrated in the later section of this chapter.

In this research, the VE concept is employed and applied in the area of inventory management. When the terminology “Virtual Organization” or “VO” is used, it is referred to “Virtual Enterprise”.

From the aforementioned characters of business and VE, it could be concluded that, VE is a suitable solution to sort out the dynamic requirement of business. Some researches have been done to align VE with business: Tae-Young Kim [97] has presented a modeling framework for agile and interoperable virtual enterprise; this research shows the interoperations between and among VE partners. Hongmei Gou [98]. also developed a framework for business integration for business process modeling, analyzing and managing for virtual enterprise, Xiaoye Zhao etc. [99] have set up a "leader-member" type theory view of VE's inventory management and conducted needed analysis by UML (Unified Modeling Language). However, limited by their chosen technology and system architecture, these researches are either too general or not flexible and dynamic enough.

What’s more, most researches in Virtual Enterprise consider the Virtual Enterprise on producing a new product and seldom of them consider the utilization of VE in sale or distribution. This is probably because that it’s not easy for a system in manufacturer to reach all the sellers for a product, the number of sellers is large and is of huge dynamic. Too many of sellers which change their ideas everyday every moment on whether they want to sell a product or not, how can a traditional VE satisfy them on this? Too dynamic and too many partners bring out much trouble, so a new kind of technology and infrastructure are necessarily needed.

8.5 CWE Supported J-Company Virtual Organization

8.5.1 Introduction to the J-Company Virtual Organization

In this research, Virtual Organization System for J-Company and its retailers are introduced. This system is supported by CWE proposed in chapter 7 to connect all the nodes in the VO. Both retailers and the J-Company would benefit from the

Virtual Organization by sharing more instant information, reducing the stocks and making more accurate production plans and decisions.

In the J-Company Virtual Organization, the Vendor Managed Inventory (VMI) technology is adopted to support the business process, and it is found that Virtual Organization, in VMI mode, is very good at handling vast information. Decision Supported System for VMI has been given great attentions. Achabal [100] has presented such a system for apparel manufacturer, and proved its advantages for the improvement of customer services and inventory turnovers. Felice and Toni [101] gave a case study by comparing traditional replenishment system and VMI, and gave a clear idea on how much VMI could reduce the cost for supply chain. As the articles shows, the more accuracy the information is, the more efficiency the VMI can achieve. However, the bottleneck is the technology used by system, which greatly affected the accuracy of the information. Efficient Customer Response (ECR) highlighted the need to reach a critical mass to justify the technological investments supporting VMI. The functions of the system and the technology of the system used can greatly affect the efficiency and usage of the system.

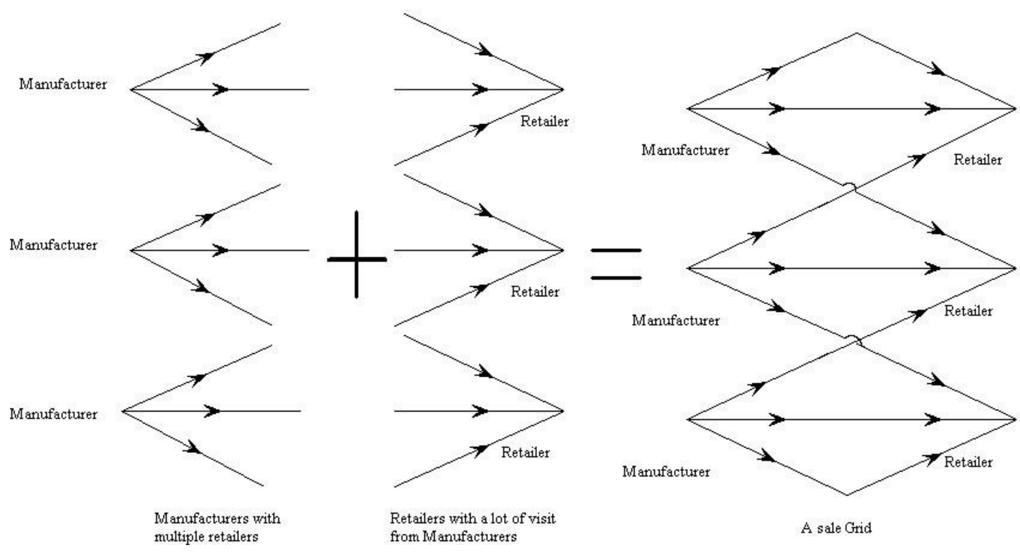


Figure 8.4 How the Virtual Organization forms

Figure 8.4 shows how the J-Company Virtual Organization forms during the interaction process and it must be noted that in the presented system one manufacturer may have more retailers than pictured and one retailer also may have products from many manufacturers. This results in a much more complicated virtual organization. If the “retailer” were a distributor, which has its own retailers, a multi layers system would be resulted.

8.5.2 The reason for choosing CWE to support the J-Company Virtual Organization.

From chapter 3, it is known that many technologies can be adopted to implement remote accesses such as DCOM, CORBA, or RMI. But these distributed computing systems are either tightly coupled or not flexible and reusable compared to web/Grid service [102]

In the author’s point of view, any ICT structure should at least have the following benefits:

- Increases profits by faster response to the market, better decision-making, and cheaper adoption of new strategies.

- Brings new capabilities by dynamic information repurposing and reconfiguration, adaptive service-oriented networks, and loosely coupled application connections and information.

- Reduces costs by faster data analysis and dissemination, better quality of service in partner exchanges, and

- Cheaper maintenance costs for IT systems integration [103].

In the proposed J-Company Virtual Organization, retailers can join or leave the VO with a short notice, depending on their capabilities and opportunities. The volatility of virtual organizations imposes strong requirements on their ICT supports. In order for the VO to be dynamic, the ICT infrastructure must be highly flexible [104]. Once a retailer gets an order from a customer and meanwhile, in the VMI mode of the system, current stock number of this product will be renewed and then captured by the manufacturer. The VO is not only to perform the monitoring task, but also handle exceptions. For instance, it is possible that one customer cancel the order for whatever reason or not being a retailer of the company; the system will reflect the changes to the manufacturer if the manufacturer agreed the change. Negotiation is important in this system, a retailer who wants to sell the mortal bike not only needs to register in the VO system, but also needs to formally discuss and reach agreement with the company to setup official relationship in reality. The system takes care of routine tasks, including negotiations, and involves the support of human decision making process in more complicated or unforeseen situations. Also, the platform needs to provide facility to talk with different current systems, no matter they were developed in Java, CORBA, or C＋＋.

From the above features, CWE is an ideal choice to implement this virtual organization. CWE mainly adopts Grid technology, but in the mean time, it provides wrapping method to accommodate other technologies. The main technology, Grid, pays more attention to the concept of Virtual Organization [105]. VO exists in a group of people who have similar requirements, for example, the university campus network can be built within a grid, and so is a local area sale network or an online game platform. So CWE utilizes Grid and can gather and connect geographically dispersed people in a virtual environment to communicate and share information with each other. Such a network should support large scale of users.

It is also an important reason to choose grid-enabled CWE as the implementation of the Virtual Organization that Grid and the business process have similar interaction process, which is a very complicated interaction process (Figure 8.4).

Below are also the reasons why traditional systems cannot support the VO as good as the proposed CWE system:

- Traditional systems cannot support a system with changing user interface or diverse client application, which can be easily solved by using Grid technology. In the CWE, current information systems could be wrapped and connected by exposing the interfaces as WSDL described services. Traditional middleware system cannot support a diverse visit from client and only provides users with an given client application, which means they cannot be adopted widely, because people always hate changes if they have already been used to the existing applications.

- Traditional technologies cannot support huge and dynamic numbers of users. In this system, inventories of retailers in a large area will be supervised and monitored; probably tens of thousands of retailers are involved. In that case, the huge computing power and intensive interactions would lead to the requirement of a system infrastructure with high efficiency and reliability. Old systems can only support limited connections of clients and limited power of computing.

Because of the plugin features of the system, the CWE system not only provides the default VMI model of business interaction, but also supports the traditional order-driven model (or Retailer managed Inventory, RMI), depends on the plugin modules in both the Clients and Servers. In theory, the CWE could reduce the number of levels of selling Chain to 2 levels (originally 7 levels), which means that the factory could directly supervise the change of stocks in retailers, though the number could be tens of thousands.

8.5.3 Description of the interaction

(a). Business interaction process

In retailers when one mortal bike is sold, the change of current stock is reflected by the decrease of related database value, which is used to record the quantity of the product stock. The database is wrapped into a service provided to the service consumer/requester. The service requester collects the current product information and so the manufacturer of the product would know instantly about the current product situations. If the current stock level of one product in retailer is lower than safe stock level and worth a delivery, the manufacture of the component would be automatically notified. However, this is only one case of the complicated system; of course, tens of thousands of retailers sell the same product come from the same manufacturer. Countless selling information from the retailers is collected and shown in the J-Company’s server system, those who have low stocks need replenishment; manufacturers need to locate all of them in real world. Based on the analysis of both the time and the cost, the production plan, the replenishment plan, the delivery route will be figured out by the decision support system in the server.

VMI is the default provided mode of the system. In reality, the VMI model may not be acceptable to some retailers because of various reasons (preference, operating concepts and other specific situations). Or some times, the retailers may want to change the models (also depend on specific situations). Because of this, traditional mode, which is order-driven based (e.g. JIT), is still supported by the system by plugin new application modules and services. So the retailers could switch between/among the two/more modes in the same system. If the retailers choose the Retailer Managed Inventory mode, they need to publish their orders and related information online so that the manufacturers could visit them. However, before the retailer switch the mode, negotiation should be carried out between retailers and manufacturers commercially about the feasibility of the action.

(b). General Data Procedure

The data procedure will also be described based on two modes: VMI mode and traditional order-driven mode.

i. In VMI mode

Manufacturers can recognize their products by reading the product ID (e.g. the barcode) and get related product stock information after the retailers deploy the database of stock as services. The data in the provided services can be updated in real time. Then the services need to be registered in Service Registry which belongs to the middleware component (middleware component will be introduced in next section), after that the retailers do not need to care about the delivery any more by leaving all such things to the manufacturers.

ii. For Retailer Managed Inventory mode

Different from VMI mode, the manufacturer does not need to check the product stock information but check the online orders from retailers. So, the retailers should publish their enquiries as services with information, e.g., how many products they need to order, which needs to follow a recognizable format.

8.6. System Design and implementation

The focus of this research is thus on providing a suitable infrastructure for inventory management in a dynamic ensemble of retailers and manufacturers. The ability to manage the inventory presumes the ability to exchange sales information in a proper extent. Such ability rests on the agreement on a platform neutral and sufficiently expressive language. The product information at retailer could be published by various ways such as the system provided default service, or the retailers' inter-operation information systems. The product could be identified by bar code, which is packed and described using XML as the common language.

8.6.1 System Components

The Virtual Organization has to satisfy a number of properties to provide a suitable infrastructure. Generally, there are four types of components: General services, retailer's component, manufacturer's component and Central services. Figure 8.5 shows the interaction among the constituents of the system.

(a) General services

General services are necessary and should exist in all distributed systems. In this system, the following issues are covered by general services:

- **Resource management**, which provides for a controlled creation, usage and destruction of the services;

- **Recovery management**, that determines which exceptions can be handled by the system;

- **A security model**, which states the threats that are dealt with by measures, such as the security protocol and authentication services.

General services provide hospitable surroundings for the VO from a security and data-safety perspective. Those services are reflected in the Middle ware layer of the CWE system.

(b) Manufacturer's components

One manufacturer might have multiple products, statuses of which should be monitored. In Figure 8.5, in order to make the process to be understood more easily, the interaction related to only one product is marked out (Different types of Mortal Bikes are considered as different products).

As shown in Figure 8.5, the functions of a system component residing in manufacturer are categorized internally and externally. External functions include checking the product registry information (Step I), searching the retailers' information (Step II) and binding the manufacturer with retailer (s) (Step III); internal functions include sales prediction, production plan, production scheduling and delivery scheduling functions. It should be mentioned that the internal functions are provided by the local information system of manufacturer, and are related to the future work of the research. Among the external functions, the checking function is used to make the system be aware of which registry the specific product is registered to; the finding function can help obtain the retailers' Network addresses. The binding function can facilitate the communication activities between the two parties which include sending request to the located retailers, retrieving the desired data, and even negotiation.

Once the two parties were bound, the interaction begins. Figure 8.6 shows the interaction between the two parties and the internal processes based on the two modes. The basic search, check and bind functions are also residing in middleware layer but could be controlled by Application layer when receive the requests from users.

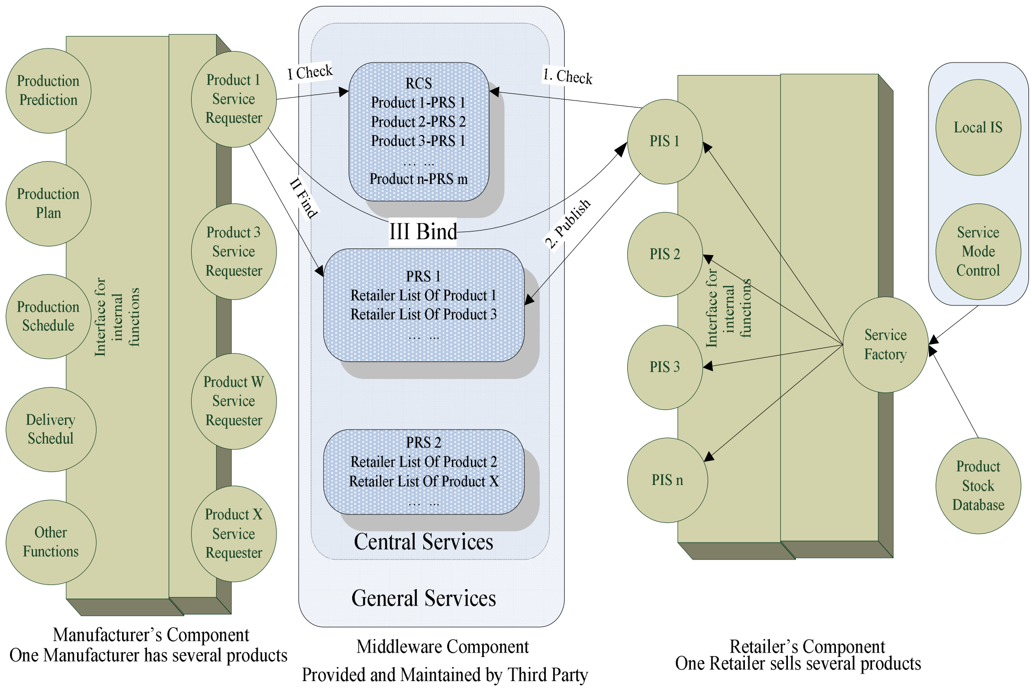


Figure 8.5 Interaction analysis

(c) Retailer's component

A retailer also has different products from different manufacturers. The retailer component is mainly used to produce and hold services, which have three functions: Checking on the register checking service; registering the product in the right Product Registry Services (PRS), and then being bound when there is a request from the manufacturer. The Product Information Service is produced by the Product Information Service Factory (PISF) which is an important function of the service holding component(at retailer side). The PISF produces one service for each type of product based on the chosen mode, after that PISF will align the services with the product database. If the business mode is changed, the service should be destroyed and regenerated to support the new mode; the service holding component is extendable and could host large amount of services for different products.

(d) Central services

There are still some central services which are indispensable, such as the aforementioned Registry Checking Service (RCS), Product Registry Services (PRS) and Service for Access Control List (ACLS). RCS is used for manufacturers and retailers to find the right PRS for further conducts. PRS is for the locating of products from manufacturer to retailers. ACLS takes the responsibility to secure the system and is invoked whenever a member joins or leaves the Virtual Organization. Some other information could also be provided such as retailers' information, manufacturer's information and production description, such kinds of information could be simply provided by a web portal.

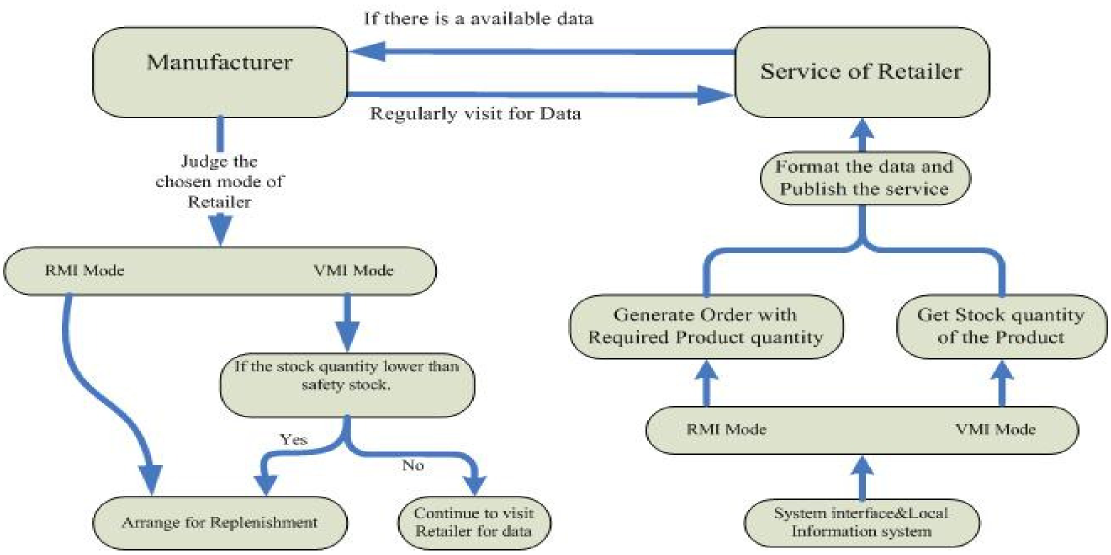


Figure 8.6 CWE collects inventory information

In this system, Manufacturer's component is acting as a service-consuming node while Retailer's is the service provider. Some system designs could probably make either the retailers or manufacturers to be both service provider and service consumer. The system supports one direction service request, the information is transferred from retailer to manufacturer (Figure 8.6), the reason is that one direction request makes the system much simple than two.

All the services are residing in the resource layer and the application in the top layer need to co-ordinate those services and achieve the aim of the missions.

(e) The reflection of the VO structure in the CWE layers

According to the above, the author uses the CWE structure to realize the Virtual Organization of J-Company. As shown in Figure 8.7, the company application can be inserted into the upperware layer of the CWE; the application will reach the bottom layer with the support of the Middleware facilities. The bottom layer composes all kinds of services needed by the VO. The mission of upperware and middleware is to coordinate all the services/functions in the bottom and could work for the Virtual Organization. The full Chart of the CWE for J-company is shown in Figure 8.7. Figure 8.8 shows the physical services forms in reality in Internet, these services describe the information extract from various of local information systems through the wrapping procedure.

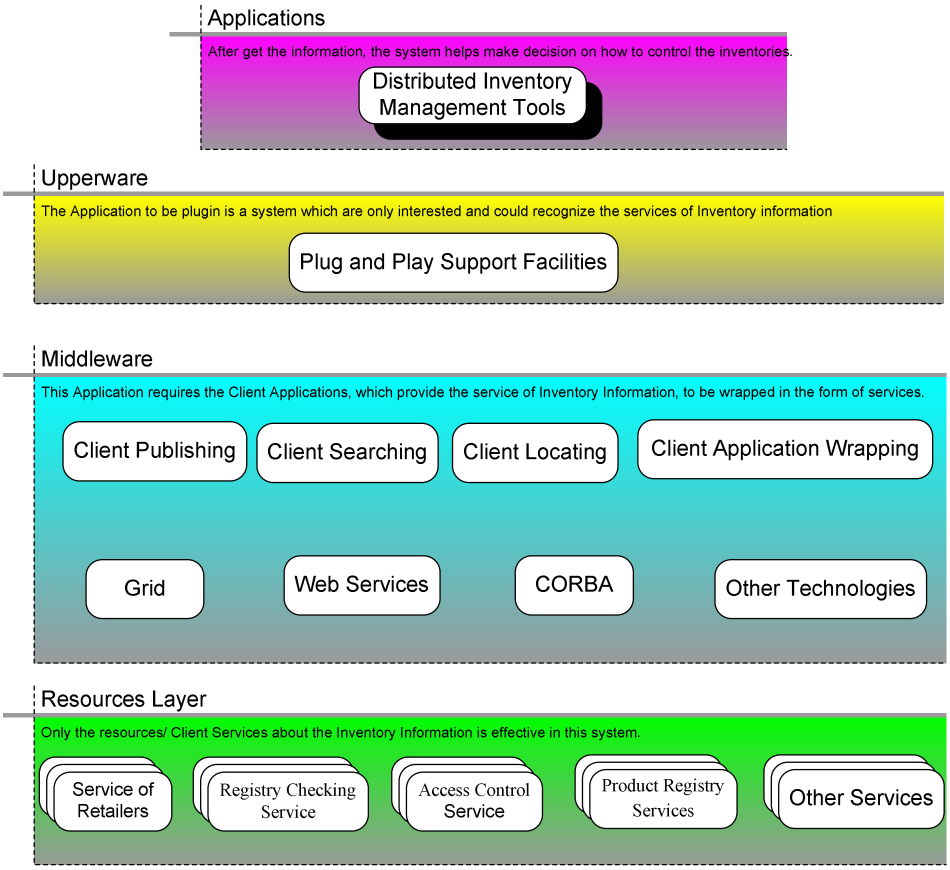


Figure 8.7 Realisation of the J-Mortal VO in the CWE system

8.6.2 Operations in the GBS-RMVO

(a) An enterprise joins the VE (VO)

In the previous section, we already discussed some of the infrastructures needed for the establishment and operation of a VO. The VO should offer a mechanism whereby a retailer or manufacturer can express its intention to join and, upon admission, get access to the necessary information, such as the information for retailers interested by manufacturers and the information of products interested by retailers. The VO-Web portal can provide this functionality. As a second step, the prospective manufacturer has to publish its enterprise information, product types and availabilities to the VO (in addition to the GCS providers, participant could also provide facilities for the VO) while the prospective retailers should provide its information including its scale, location, preferences etc. After that the prospective participant should install a retailer's/manufacturer's component and connect it to its back-office systems by implementing the translation methods (semantic alignment) to and from the VO ontology. When the necessary communication channel is established, the new member can be included in the member list of the VO, and then could be visited.

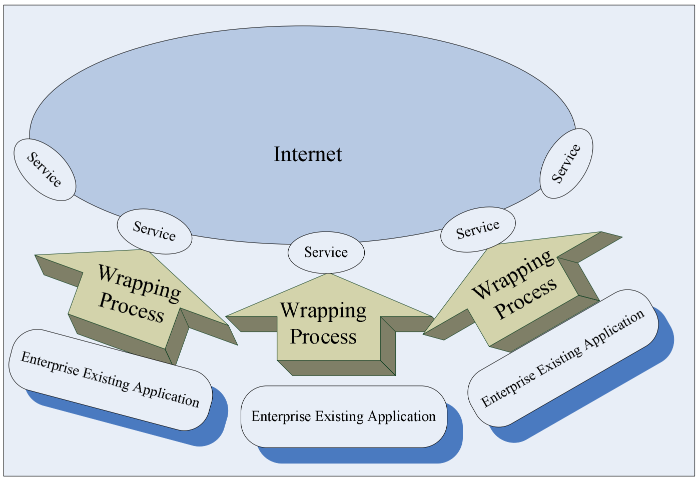


Figure 8.8 Client Provide Services in Resource Layer of CWE

(b) A member leaves the VO

When a member leaves the VO, the inverse processes of a new member joining and its product being added to the catalogue would be carried out. The system has to make its withdrawal known to the VO

(c) Retailer adds or removes new product

A retailer may want to sell a new product; in that case, he must look up the product information provided by manufacturers through web portal and decide the right one and note down its product ID (e.g. the barcode). After that, he can publish the status of the product in the service (stock quantity = 0) and register himself in PRS so that the manufacturer could locate it.

To remove a certain product, just destroy the PIS service after un-register it from the PRS, then the information of retailer is deleted from the PRS list.

8.7 An Example

This section describes how the system works and how it functions to aid the manufacture to make decisions. The data is collected from J-Company is on 22nd of June, 2007. The example is conducted in a laboratory environment.

8.7.1 The general description

In this case study, the data for ten types of mortal bikes were obtained from J-Company, which include: JC105, JC110, JC115, JE110, JE110-8, JE120, JE115, JE115-2, JE120-2, JE105. In the test, ten retailers registered their services in the registry. In reality, the company sells products to tens of thousands of Counties in China, and also sell products to America, Europe, Africa and so on, so when the company application checks the retailers' information on the registry, it will be directed to each of the services at the ten retailer sites. Core services such as the registration are provided by a set of computers on the network. In this test, the information contained in the services of retailers includes: the date, the number of products sold that day, the number of stocks left, and the address of the retailer. When the data is collected and gathered, the main application will assign each retailer with a unique ID based on their addresses. The reason to use an ID to represent a retailer is that the ID can be recognized by the computer so that the computer could help make decision to figure out the best route for replenishment to balance the customer service and the cost (ID contains the address information). Each ID is unique to represent a retailer and is coded following some rules which are explained in the later section.

8.7.2 System Deployment and functions

This section will introduce the system both in retailer and the manufacturer sides.

Retailer Side

Step 1. Login the system (Figure 8.9). In the main frame (Figure 8.10), click on the AddSVC button, Figure 8.11 appears. Fill the adding service panel as follows:

Description, VMI Retailer Service

MainClass, None (This service does not provide default client interface)

PackageFile, D:\CWEServices\VMIRetailer.zip

ServiceName: JCompanyRetailerService

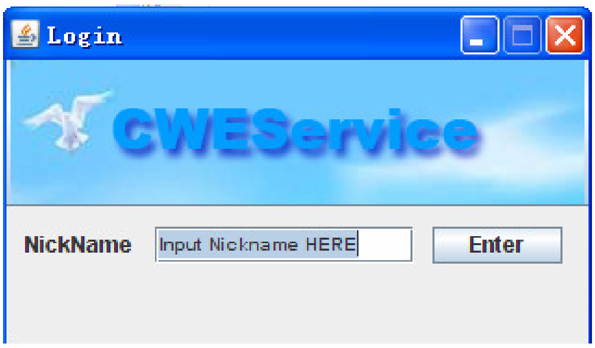


Figure 8.9 Login Frame

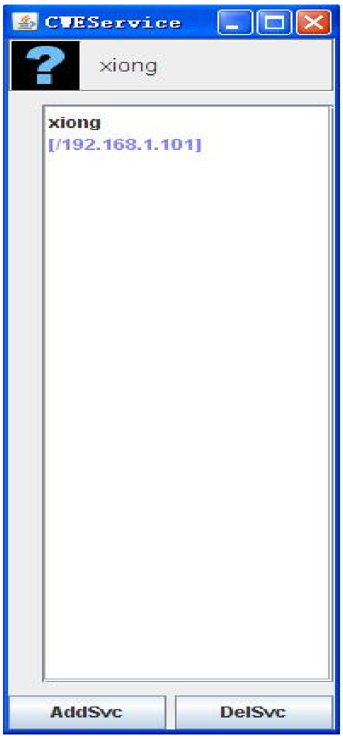


Figure 8.10 Main Panel

To add an application is similar to add a service. Application will be invoked by the provider himself while service is for other user to invoke.

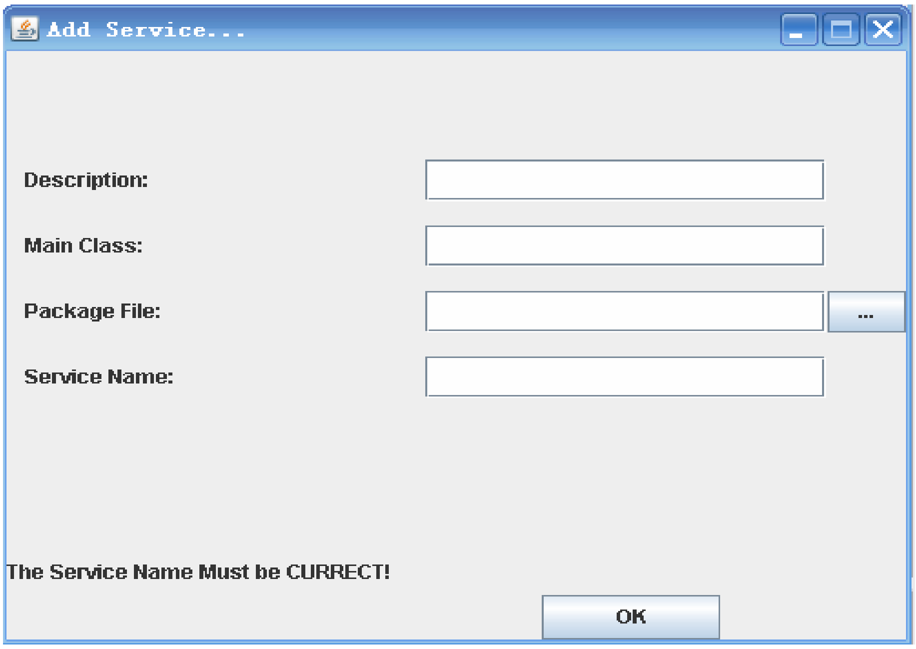


Figure 8.11 Adding Service

In the VMIRetailer.zip, following information is represented in XML format and published in the Network:

Date	22/06/2007	
Address	Niaoning Province, Da Lian City, Xigang District, No. 82 Caiyun Rd	
Product Type	Product Sold Today	Stock
JE110		
JE110-8		
JE120		
JE115		
JE115-2		
JE120-2		
JE105		

Table 8.1 Retailer's information.

The VMIRailer.zip can be operated manually, by providing a default controller to manage the document. Alternatively, it can be integrated into and operated by a local information system. The following application controller can input the information into the service and publish the services.

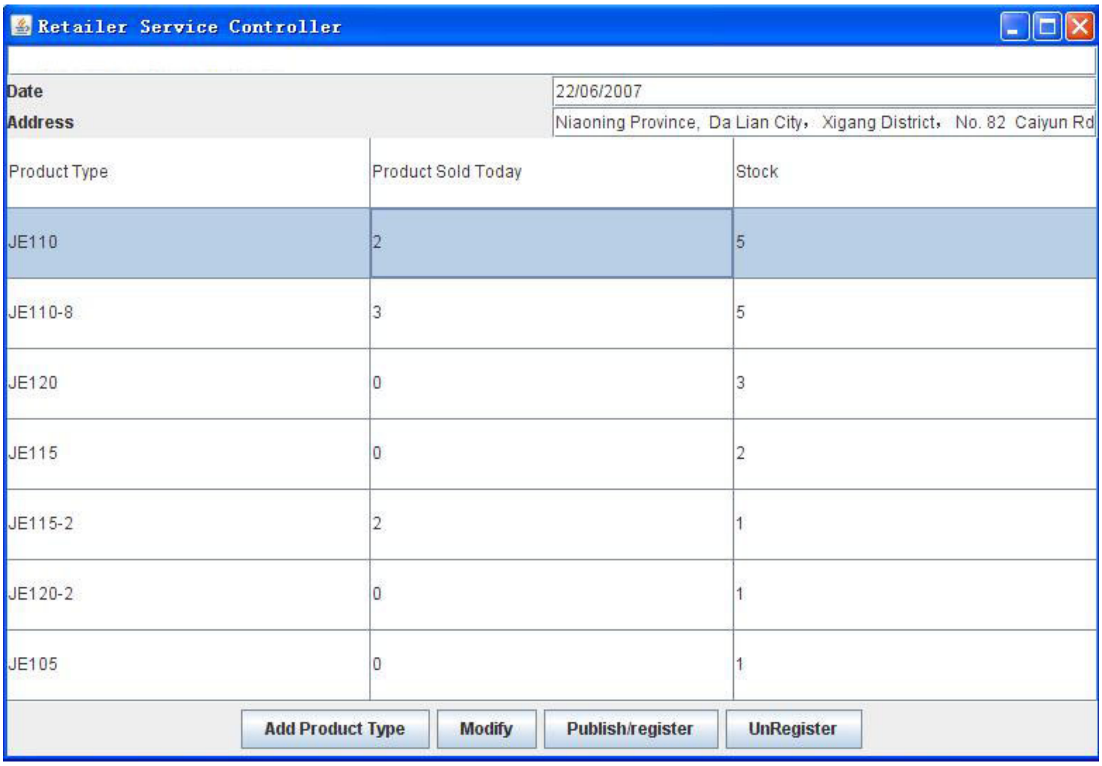


Figure 8.12 Retailer Service Controller

As shown in figure 8.12, this retailer sells 7 types of mortal bikes. The table reveals that on 22nd of June, two JE110 were sold and 5 left. Also, there are four buttons in the retailer service controller, which are operations to the services as described below:

- Add Product Type: If the retailer wants to sell another type of mortal bike, he needs to use this function. If he wants to remove a type of mortal bike, simply right click the type name and select “Delete this type”.

- Modify: Click on the field which needs to be modified, and click on" Modify", then input the right information.

- Publish/Register: When the information is modified, click on publish, and this change will go to the service, so that the company could recognize, if it is the first time to use the system. When click on publish, this will make the service registered in the registry.

- Un-register: If the retailer want to remove this service, he can click on the unregister button,which will remove the retailer's information from the service registry, and the company can not reach the service.

Company Side:

Step 1. Similar to the Retailer side, first open the CWE interface, then in the main frame (Figure 5), click on the AddSVC button, and fill the adding service panel as follows:

Description, VMI JCompany Application.

MainClass, None (is not a service)

PackageFile, D:\CWEServices\VMIJCompany.zip

ServiceName: JCompanyApp.

The application can be invoked by double click the application name under the server name in the main frame. Figure 8.13 is the application interface. Alternatively, a separate application can be made to do the same job.

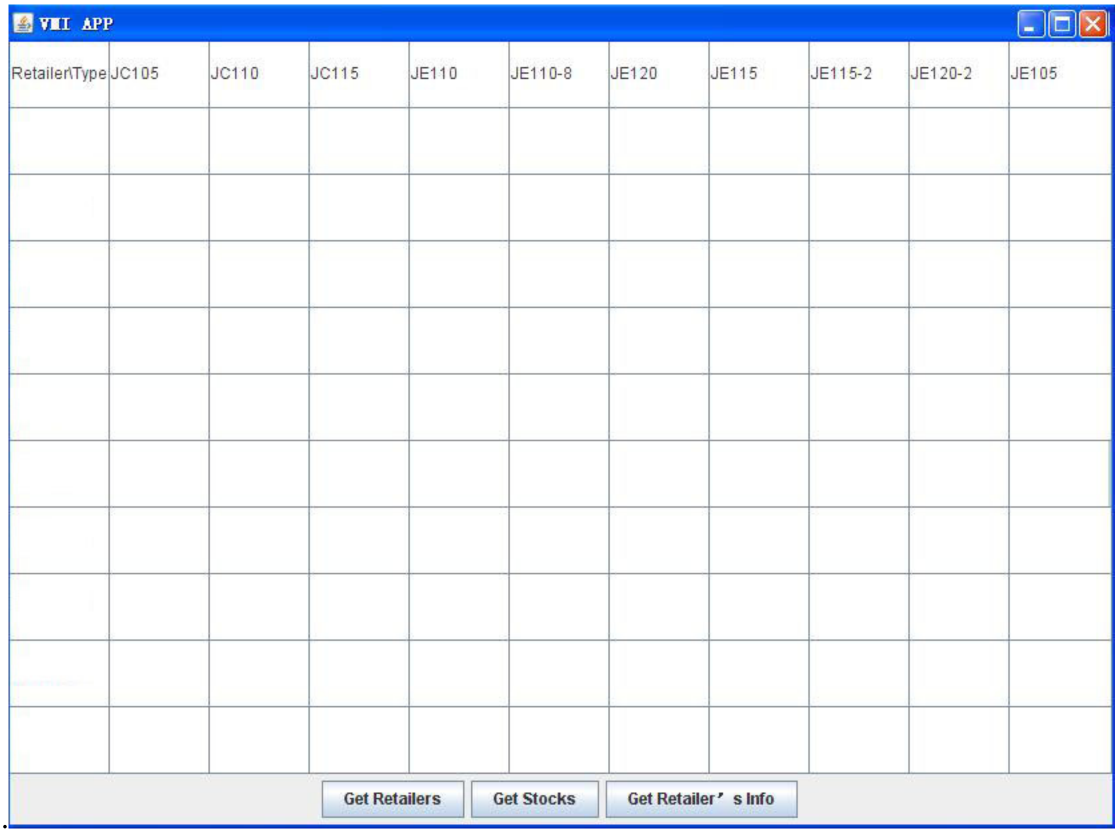


Figure 8.13 Application Interface

The function of this application is to look up the service registry and collect all the data then return them to the server. As shown in Figure 8.13, all the ten types of products are listed. In the bottom of the main interface, there are three buttons: “Get Retailers”, “Get Stocks” and “Get Retailer’s Info”.

When Click on the “Get Retailer” Button, the application will visit the registry to search the retailers; the retailers will be listed as shown in Figure 8.14: each retailer will be coded at this step.

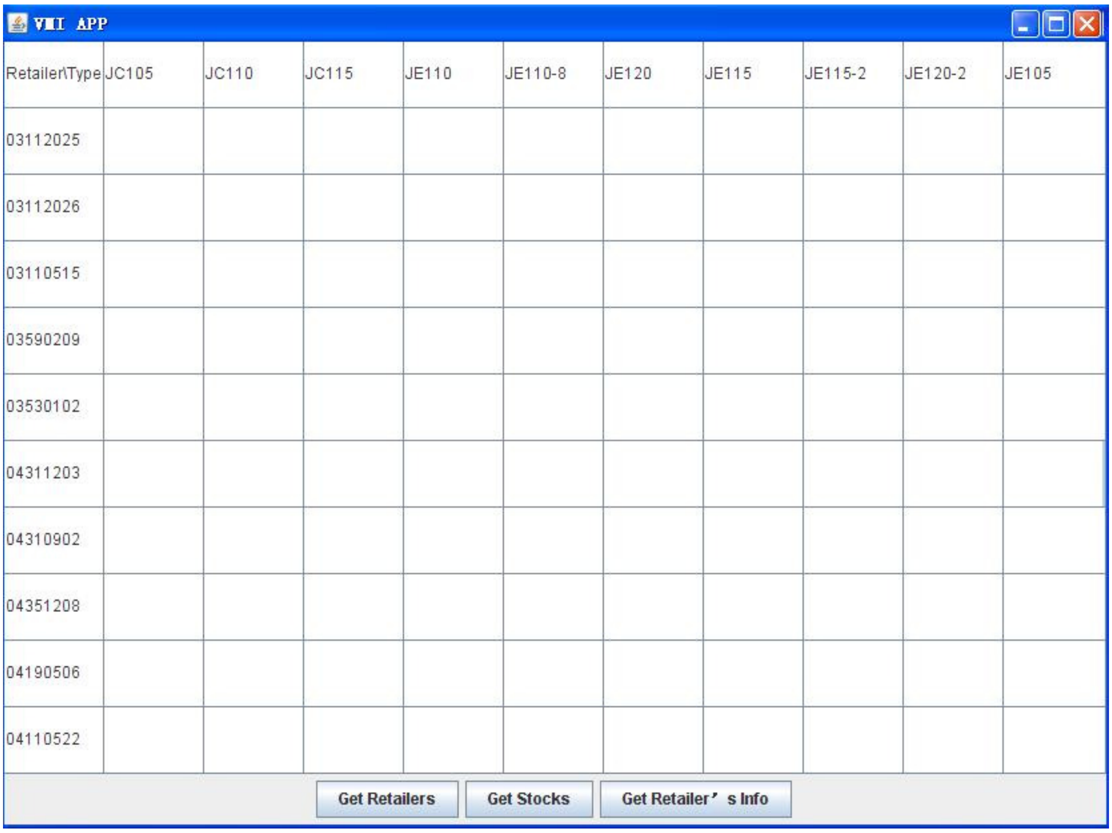
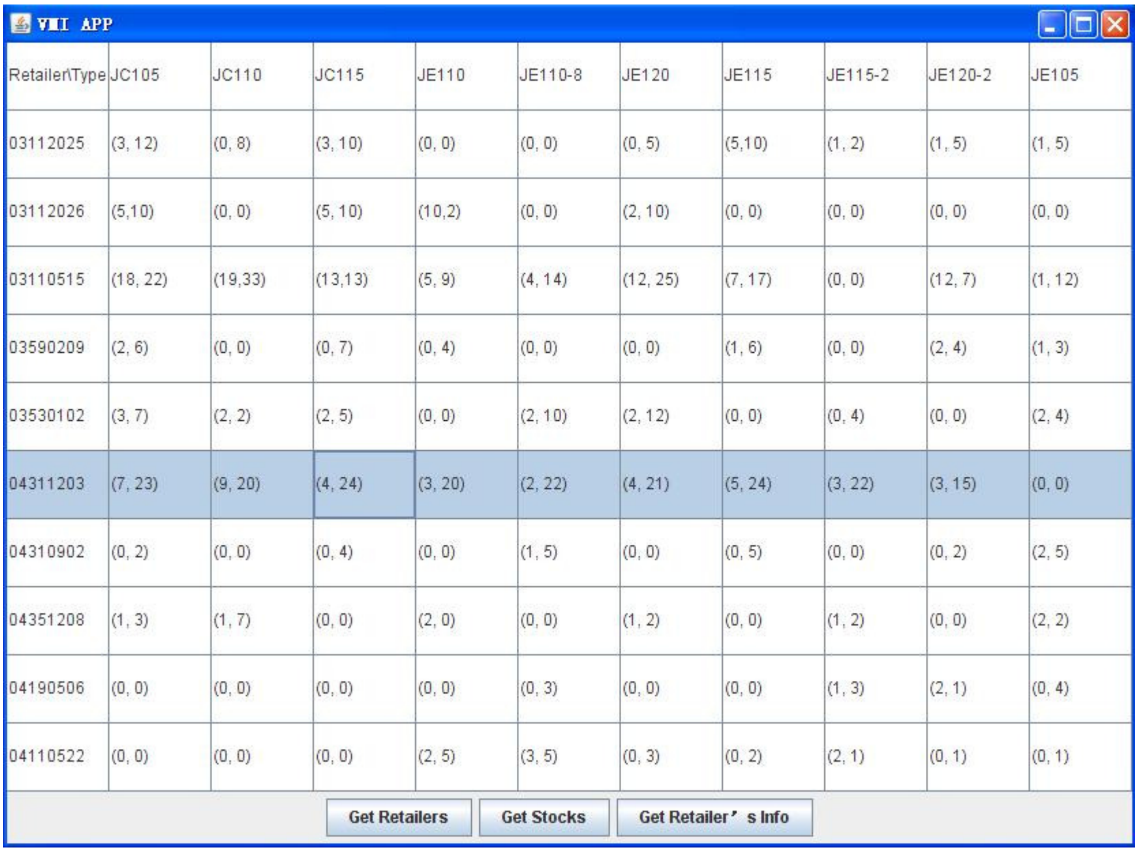


Figure 8.14 Retailer List

In Figure 8.14, ten retailers selling the company's product are found in the service registry. The system automatically codes each of the retailers and shows them as a number in the right of the panel rather than their real names.

With clicking on "Get Stocks" button, the application will visit each of the retailers on the network and get the stock information, and finally display them in the table. In Figure 8.15, the first row shows the type of mortal bikes, where the first column shows the list of retailers. In the grids, the first part in the brackets means the number of mortal bikes of that type sold on that day, while the second part in the brackets means the stock of this type of mortal bike left. For example, (3, 12) in the first grid under "JC105" means that, today, in the retailers numbered 03112025, there are three mortal bikes of type JC105 were sold, while there are still 12 mortal bikes of this type left. (0,0) in the first grid under"JE110" or "JE110-8" means that, this retailer currently does not sell this type of mortal bike. These data are collected from the retailers in different areas of the country. The aim of this approach is to reduce the stock as much as possible and to accelerate the inventory turnover, because, nowadays, the transportation fee for manufacturing becoming more and more cheap. It's important to know how many products were sold to help make decision on how many products to produce, and, also, it's important to know how many stocks left in inventory to help make decision when to replenish the retailer's inventory.



Retailer\Type	JC105	JC110	JC115	JE110	JE110-8	JE120	JE115	JE115-2	JE120-2	JE105
03112025	(3, 12)	(0, 8)	(3, 10)	(0, 0)	(0, 0)	(0, 5)	(5,10)	(1, 2)	(1, 5)	(1, 5)
03112026	(5,10)	(0, 0)	(5, 10)	(10,2)	(0, 0)	(2, 10)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
03110515	(18, 22)	(19,33)	(13,13)	(5, 9)	(4, 14)	(12, 25)	(7, 17)	(0, 0)	(12, 7)	(1, 12)
03590209	(2, 6)	(0, 0)	(0, 7)	(0, 4)	(0, 0)	(0, 0)	(1, 6)	(0, 0)	(2, 4)	(1, 3)
03530102	(3, 7)	(2, 2)	(2, 5)	(0, 0)	(2, 10)	(2, 12)	(0, 0)	(0, 4)	(0, 0)	(2, 4)
04311203	(7, 23)	(9, 20)	(4, 24)	(3, 20)	(2, 22)	(4, 21)	(5, 24)	(3, 22)	(3, 15)	(0, 0)
04310902	(0, 2)	(0, 0)	(0, 4)	(0, 0)	(1, 5)	(0, 0)	(0, 5)	(0, 0)	(0, 2)	(2, 5)
04351208	(1, 3)	(1, 7)	(0, 0)	(2, 0)	(0, 0)	(1, 2)	(0, 0)	(1, 2)	(0, 0)	(2, 2)
04190506	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 3)	(0, 0)	(0, 0)	(1, 3)	(2, 1)	(0, 4)
04110522	(0, 0)	(0, 0)	(0, 0)	(2, 5)	(3, 5)	(0, 3)	(0, 2)	(2, 1)	(0, 1)	(0, 1)

Figure 8.15 Retailers' inventory information

Click on "Get Retailers' info", the information of each retailer, mainly about the locations, is listed. There are some rules for the coding. Each Province, each city, and each area of the city are given a code. Figure 8.16 shows the detailer location information of the retailer.

.

Retailer\Type	Province	City	County	Address	Other
03112025	Hebei(031)	Shi Jiazhuang(1)	Zhao County(20)	No.25 Zhaoyuan Rd.	
03112026	Hebei(031)	Shi Jiazhuang(1)	Zhao County(20)	No.161, People Rd.	
03110515	Hebei(031)	Shi Jiazhuang(1)	Xinhua District(15)	No.6 Hezuo Rd.	
03590209	Shanxi(035)	Yun Cheng(9)	Yanhu District(02)	No. 192 Jiefang Rd.	
03530102	Shanxi(035)	Yang Quan(3)	Main District(01)	No.109, East St.	
04311203	Jilin(043)	Chang Chun(1)	Lvyuan District(12)	No. 3 Yinbin Rd.	
04310902	Jilin(043)	Chang Chun(1)	Nanguan District(09)	No.32 Da Jing Rd.	
04351208	Jilin(043)	Tong Hua(5)	DongChang District...	No.126 Baoan Rd.	
04190506	Niaoning(041)	Niao Yang(9)	WenSheng District(...	No. 28 ChangSou Rd.	
04110522	Niaoning(041)	Da Lian(1)	Xigang District(05)	No. 82 Caiyun Rd.	

Figure 8.16 Retailers' information

This case study is conducted in laboratory environment and is only tested with ten retailers. However, the system actually could support tens of hundreds of retailers all over the world. This system provides a loosely coupled structure and could tie all the retailers with the company in a complex network environment. In current situation, this system can not be widely deployed in real world immediately though the system gives a solution for the future of the company in ideal situation. The reasons are:

- Manufacture automation has not been so developed enough to adopt this approach. The manufacturing nowadays still need human intervenes. However this system could provide a helpful decision making.

- It can not be reality that all the tens of thousands clients adopting this solution due to the spread of Internet in China (not all the area of China covered with Internet).

This system provides a strong and powerful support to manufacturing decision making process. Because vast information is directly obtained from tens of thousands of selling terminals, the decision made by this information is accurate. However, nowadays manufacturing has not reach full automation without any human intervene, so this system can not exert its full capability. It can be imagined that, years later, this system can be more and more powerful with the development of manufacturing automation. The delivery time is shorter and shorter; one can get the product next day when he placed an order in website. In that case, it's more and more useful to monitor the real time selling status and making decision on when and how many products to produce and deliver.

8.8 Evaluation and limitations.

It can be argued that the test has been successful; however, the drawbacks can not be neglected. Firstly, the business model requests every retailers adopts VMI model, however some retailers may want to control the inventory themselves, and this flexibility for users is not provided. Secondly, for any business model, before the member joins or leaves the system, a negotiation process needs take place, at least present some terms and conditions. This research obvious did not consider this situation. Finally, for the retailers who are using existing information systems, though the wrapping method can allow them to keep the existing system, but they still need a professional programming team to develop and wrap their system.

8.9 Conclusion

In this research, the CWE system is applied to J-Mortal Bike Company and provides a technical solution to the Bullwhip effect on its selling chain. The solution is based on forming a virtual organization which includes the company and all its retailers. The CWE provides a feasible way to effectively manage the inventory in tens of thousands retailers and reduces the levels of the selling Chain. The system has embodied the power of the CWE. The four layers of the CWE can work coordinately to support the whole Virtual Organization of the company.

Benefits of the CWE supported Virtual Organization approach include:

- It has a dynamic operation feature. Every retailer who sells the same product, i.e., mortal bikes, can register or un-register their product on the PRS. In so ding, the company needs to send a request to the registry and get back the network location list of the retailers. The advantage is that the company does not need to be notified each time when a retailer joins the group or quits. They just need to do one thing, which is to send a request and retrieve the list of retailers with their information.

- The data retrieved by the system is also a reference for the scale of production capacity they should have. Manufacturers have a better view on the sale status of their products than retailers, and, hence, it is very important for them to reduce stocks and save cost and it is even better than JIT (Just in Time) production.

- Different from other VMI systems, the Grid-based VMI can deal with huge amount of information from unlimited number of users in the system, which results in more accurate data for production prediction.

- Due to characteristics of the grid service technology, retailers and manufacturers are in a virtual organization (VO), in which they can join or leave by a short notice. For the notable Grid features, the system is powerful, useful, and also extendable.

This system could be applied by other big companies by inserting different applications in the upperware layer and coordinate different services in the bottom/resource layer of the CWE.

Chapter 9
Discussion and Conclusions

. 9.1 Contribution to knowledge.

During the four years' research, the author has been involved in two EU projects: EU Asia-ICT and EU Asia-Link project. Two important research modules in the two projects were conducted by the author. One is Online collaborative CAD, and the other is online software sharing, which is also related to the re-use of legacy application. These become the first half of the author's PhD programme. In the second half, the author has been working hard on developing a collaborative working environment for manufacturing, which is the further development of the first half research achievement. This outcome of research contributes to the approaches to the future globalization of manufacturing and design.

This research mainly contributed to the construction of an online Collaborative Working Environment for design and manufacturing. The increasingly complex and changing global market, and the important roles of information technologies and systems necessitate a more advanced Collaborative Working Environment. The following highlight the main features and the contributions of this PhD project:

(a) Contribution to compatibility

This CWE, developed in this research utilizes open standards and provides the loose-coupled infrastructure so that it achieved the compatibility among different systems. This approach also archived the interoperation between heterogeneous applications. This contribution is important for the enterprises.

Existing computer systems/applications were always developed to fit one enterprise or organization. They are separated from each other and cannot coordinately work together. This forms a barrier when there is a requirement of collaboration between them. In some cases, those enterprises are almost independent to each other; they had their own preferences to choose any kind of tool/system/application to work with. However, many enterprises now need to join an enterprise group to survive fierce competition. Supply Chain Management, Virtual enterprise etc. became of the main strategy. Those strategies require the enterprise to intensively interact with other group members,, especially in information and manufacturing communities. There should be a way to enable the information system/Decision Support System and CAD/CAM/CAE applications in each enterprise to talk with each other. This contribution solved the compatibility problems in a service oriented environment.

(b) Contribution to reusability

An innovative three-services based wrapping method has been developed in this research to re-use old applications. The service-oriented wrapping method provides a new and effective solution and also, the advanced feature of service oriented program, interaction between services, is fully explored in this research.

It is always costly to build a new application. If an existing application still operates, the owner may reluctant to build a new one; especially for an application specifically built for an enterprise, which requires a lot of money, time and energy. A lot of surveys, research, investigations and tests have to be conducted to build an appropriate application. However, if some features of an old system are really important, is there a moderate method to balance the cost and utility? For example, a complex program of a very efficient algorithm may only support DOS; a very powerful tickets selling system, may not support network function etc. How to inherit the good functions of old applications and build upon them with new functions has become a hot topic.

(c) Contribution to universality

A solution to unify all the resources is presented. The author extends the three-services approach into a wide area. This approach extends the concept of wrapping and aims to unify all the computing resources. In the other hand this solution actually helps the popularity of grid; makes more traditional resources become available to grid technology.

In a network, though many entities exist such as computers, applications, resources, but the situation often happens that one entity cannot reach another or one is not recognized by the other. It is similar when a file is downloaded but the user cannot use it, as it cannot be open by any of the software in this computer. This is the same for the resources, as different entities provide different interfaces. They can be invoked and utilized only if the user/requester knows their interfaces, and this already hinders the urgent requirement of global resource sharing. This problem was solved well in this research.

(d) Contribution to upgradeability

To solve the upgradeability problem, this research provides a very flexible way to update the system so that not only the developer of the system, but also the user of the system could also update the system(only need to update the plugin module, and do not need to change the whole system). Upgradeability (for users) is a very important consideration of the system design and development. In this contribution, the plugin function, the loosely coupled modules and the well-coordinated four layers provide good solution to this challenge.

With the changing market, new requirements, new impact elements and new business models, all require very reliable and flexible systems with the ability to be frequently updated. Current updating methods include: online update, issue update package and new versions of software. However, those are usually for commercial software and cannot support specific software for certain enterprise. For example, if one enterprise wants to consider an update in a component which is out of date; in an old situation, they need to contact the original system producer to add this function. However, if this happens frequently, it may be very inconvenient and costly.

(e) Contribution to expandability

This CWE system can support accommodation of multi-discipline functions and modules flexibly by providing plugin facilities. The system enables expertise from different disciplines to work together. New modules and plugin applications may not be developed by the software developers, but by the enterprise technicians. The system is strong, reliable and flexible so that it could potentially be expanded to reach a great scalability. Expandability means more significant change to the system than upgradeability. In current and future practice, knowledge will keep expanding and multi-discipline knowledge will be necessary for decision making. Upgradeability and maintenance are not far less than enough for a future system.

Though many new applications/systems were developed with advanced features, the majority of the enterprises do not have time to catch up the development of technologies, if their original systems do not support future technology. Systems may not be powerful enough to meet the requirements of enterprises in the knowledge expanding era. The above five contributions are very important and are well resolved in this research.

9.2 Concluding remarks.

9.2.1 Online Collaborative Computer Aided Design

Online collaborative computer aided design involves multiple considerations from domain experts and their computerized tools to be integrated to work together for a common design task. Though some systems exist for that purposes, majority of them adopt traditional technology and system structure. These systems cannot be widely applied due to the limitations of the technology and system structure. To resolve the problem, this research applied Service Oriented technologies and SOA to provide solutions, which have the following features:

- It's easy to update/deploy each side of the system (Client/Server). The service oriented design does not requirement a simultaneous update or deployment to both of the two sides.

- It does not necessary for all the participating parties to start the collaborative work together. The designer who wants to share the design process with others can start his work but does not need to wait until all participants are ready. Other collaborate designers can join a design activity in the middle of the process after they got the authorization.

- This approach can penetrate firewalls because of the feature of SOAP. It is possible that the users are located away from each other. The communications among them via the Internet have to go through firewalls or proxies. In that case, the feature to go through firewall becomes very important. Technologies, such DCOM/CORBA, cannot satisfy this requirement. In that case, SOAP, adopting an HTTP format transmission, provides great convenience to this.

- This approach can be deployed in Linux or Unix system. With the support of platform independent Java technology and loosely coupled SOA infrastructure, the system supports users using different operation systems. Though MS Windows system takes up more than 90 % of PCs, for the manufacturing world, a relatively large proportion of users still prefer Linux and Unix as the security and reliability, making this feature important.

9.2.2 Online software sharing

There are two aspects in this research: a practical way of online software sharing and re-use of legacy applications.

For the first aspect, Asia-Link project required a software bank, which enabled the participants in the project to share software efficiently. This required an advanced software sharing system. The traditional way of sharing software was to give the source codes or binary file to the requester. The user could then get the whole of the software, though the software developer may make every possible effort to limit the illegal use of the software, sometime it is not as effective as expected. Once the user gets the full copy, it is possible to crack it. The new approach needs to restrict the potential for widespread coping of the software but still provide service to wide and huge amount of users.

Traditional software distribution methods also have problems when they need to be updated, once a bug is found, tens of thousands of copies need to be updated, which is obvious hard work.

The author's solution developed in this research not only works for the new software development, but also solves the re-use of legacy code problem. Though some current approaches have already utilized SOA systems and technologies to solve such problems, they did not exert its full capability. The author's three-services solution is powerful and can exert the advantages of web service such as interaction between services. The three-services approach has the following features:

- Change the interface of an application from a DOS based application to a Windows based application. It is costly to re-code large amount of old DOS applications. These applications do not have a friendly interface to the users; however, the functions and algorithms may not be out of date. In the author's approach, new interfaces could be given to legacy systems which are friendly and Windows based.

- Change applications from single user support to multi-user support. Many old applications support single user only; to provide better service to more users in one time, the application need to support multiple users. The three-services approach, which coordinates the AMS, APS, APFS to work with each other and can enable the multi-user support.

- Further to feature mentioned above, this approach can be deployed in a Grid system. In a grid system, there can be more than one computer to provide the service. If a large amount of users request the service, then single computer with multiple user support can not satisfy all the requests due to the capability of the computer. The three-services approach also support load balance in Grid environment which helps deploy services in multiple computers and support a great amount of requests.

9.2.3 Resource sharing in Grid environment

The three-services approach is also utilized in other resource sharing situation in Grid environment. In Chapter 6, the author argued that though Grid has attracted many researchers' attention and a large amount of grid applications have been deployed, there are still very little Grid resources exist because it is relatively a new technology. The application of the technology will be confined if the resources and uses of it are limited. However, it's very difficult to create a large number of new resources. Resources supported by traditional technologies still take overwhelming majority of all the existing resources and the situation may last over a long period.

To change this situation, the author applied the previous three-services research to fit a universal resource wrapping. Apart from software, any other resources such as online harddisk space, online-electronic equipment (in Chapter 3, Section 3.2, the application of virtual use of astronomic telescope), could be wrapped by this approach.

With the support of the three services, many traditional resources can be wrapped to fit the coming grid infrastructure. The benefits of this are two folds: First, this can increase the viability of old resources; if not, they may be washed out because they do not support future technology. Second, these wrapped resources greatly encourage the popularity of the advanced Grid technology. The more available resources, the more successful the technology is regarded.

In the end of Chapter 6, a case study is presented to support this research outcome, the wrapping of a legacy application for the gear design process. The DOS based, single-user application, was finally wrapped into a cross platform enabled, multi-user Grid service.

9.2.4 CWE framework development

Previous research made substantial progress in the development of the collaborative working environment framework. Utilizing the existing approaches, such as SOA interaction and resource sharing, the framework developed by this research inherits all the advantages of those approaches. The problems of compatibility, reusability, universality, upgradeability and expandability are solved properly.

The four-layer infrastructure provides a good solution to future distributed environment; features addressed in the infrastructure include seamless collaboration, virtuosity, autonomy and security. The four-layer framework puts the features into different layers and coordinates these layers to work as a whole piece. However, they are also loosely coupled. The change of elements in one layer will not affect the other, so the framework is flexible and can support different subjects or even multidiscipline work. The actual work for a specific task is done by the plugins; different plugins conduct different work, and new plugins can be inserted to conduct new work.

The important function of wrapping resources is re-emphasized in this framework as this framework is designed to integrate a wide range of resources. In Advanced Design and Manufacturing Engineering Centre, the research team has developed many important applications in design and manufacture. They can be much powerful if they are utilized in a whole system. In chapter 8, section 8.4, the author gives each application a brief introduction of each research. At the end of this chapter, the operation procedure of the implemented system is described. Though the infrastructure is complicated and the resources are heterogeneous, the interface is simple and friendly enough for the users to use.

9.2.5 Case study: a CWE for inventory management

A case study was conducted, which utilized the full capability of the Collaborative Working Environment. In this case study, the framework was utilized in inventory management on supply chain area. The system was assumed to connect the company with its tens of thousands retailers. The system demonstrated that it can solve a significant problem in supply chain: the bullwhip effect. Bullwhip effect is caused by traditional hierarchy of a supply chain, which could result in disaster to whoever is in the chain. In this solution, the concept of virtual organization is applied, which aims to gather a group of people who have some common ground. The tens of thousands of retailers also have at least one common ground: they sell the products from the same company/manufacturer. If they could share the sell information, or let the company to have macro control rather than micro control of themselves, the situation will be much better and all the participants in the VO will benefit from it. One of the business models is Vendor Managed Inventory (VMI). The inventory information directly reflected the selling situation, so if the company controls the production based on the gathered information from each of its retailers, more accurate decision will be made. Traditional technologies and traditional concepts enable the company to supervise limited number of its downstream units; this results the multi level selling, and then bullwhip effect. In this case study, this problem is solved and proved efficiency to implement the VMI model. The case study collected data from J-Mortal Bike Company; however, the application can be used for any complicated and large companies. When the industry reaches higher automation, the system can be even more powerful. The collected information can be used to produce a production plan and based on the plan; the machines can be automatically scheduled to produce. Then the distribution decisions are made to replenish the inventory of each retailers, such as where to deliver or how many to deliver.

9.2.6 Limitations of the research

As indicated in the earlier chapters, this research does have some un-neglectable limitations as all researches have. The SOF-CAD should have the real-time interaction feature but in fact, it has a latency, the current Web service protocol do not solve the latency issue well. Also for the SOF-CAD research, the development of the system did not consider well the concurrent control among the participants. If multiple designers are changing the same object in the drawing during a short perioed(1 seconds), the system can only recognize the latest modification from one of the users. Feedback from users suggested that the deploy of the system is complicated when setting up the environment for the system. For the software wrapping approach, certain difficult parameters will require a long period of time to calculate, but the users do not have clear idea on how the progress goes. For the same issue, Grid Parallel computing should be applied to short the calculation time. The software wrapping method has not provide a good solution for copyright and license problem. For the CWE system, the limitation is that the case study is under an assumption that every retailer is using information system, and adopting VMI method to manage their inventory, negotiation mechanism hasn't been provided which is necessary for both manufacturers and retailers. For all the above research, a more simple way to deploy the system is desired.

9.3 Future work

Future work includes solve some of the major limitations and develop closely related systems, these will therefore be discussed.

9.3.1 Solve the limitations

a). Software License:

As the author mentioned before, the license issue is a very important concern in software develop and sharing but it is not solved in this research. A software license comprises the permissions, rights and restrictions imposed on software (whether a component or a free-standing program). Use of software without a license could constitute infringement of the owner's exclusive rights under copyright or, occasionally, patent law and allow the owner to sue the infringer [106].

b). Negotiation system among business participants.

Majority of business activity will cause negotiation, though this CWE system support large and dynamic participants to join and leave the Virtual Enterprise, but usually, these activities should be carried out after a negotiation took place. This research limitation will be carried out in the future research.

c). Human-machine Interface.

Human machine interface(HMI) is also called user interface, in computing context, it means the interface provided to user so that the user could interact with the program/application. In this research, some interfaces were provided, but some others is not know to the user, for example, the setup of the SOF-CAD and the launch of the system are still under a DOS environment command line. The supports for the users' convenience are not well considered. The future research will concentrate more on the HMI.

9.3.2 Development of a more advanced automatic resource wrapper

Wrapper legacy application is a hot topic in engineering. Though the author has conducted the research on how to wrap popular resources, there are still more work to do. To cover more types of resources is a very challenging work, which could be considered in the future development of the CWE.

Automatic wrapper method needs to be researched further. An ideal situation is that the system could reach every kind of resources and automatically wrap them so that the system could accommodate almost every kind of resource in the network.

9.3.3 Development of a more advanced CAD

A more universal approach to support the communication among various software packages is required. This research needs to be further developed to strengthen the author's theory. The Standard for the Exchange of Product Model Date (STEP) is a comprehensive ISO standard (ISO 10303) that describes how to represent and exchange digital product information [107]. The future research will integrate STEP into this system so that it could provide a more sophisticated communication among different CAD applications when they operate the files/databases of different formats.

9.3.4 Integration of knowledge repositories into the CWE

The system is expected to generate more experiences and knowledge when it is widely applied, and hence the knowledge repository to collect and store knowledge has to be further considered.

Knowledge is a very important element, especially for a multi-discipline Virtual Community knowledge environment. For example, in the product design area, it is necessary to realize the conduct of the design process in a real world through the system which designers are using. As the increase of interactions among users, more knowledge is automatically generated, and the most important thing is how to record and store it. The knowledge repository is expected to:

- Integrate local and central product design knowledge repositories.
- Enable users to search for knowledge and create new knowledge.
- Enable the communication between question raisers and problem solvers and between Service Requester and Service Providers.

The operation functions should be provided by integrating various applications such as:

- Search for existing knowledge.
- Creating new knowledge in different ways including
  - i: Automatic creation.
  - ii. Manual creation.
  - iii. Group Knowledge creation.
- Knowledge repository transfer between difference repositories.

The types of knowledge should cover every aspect of product design from conceptual design to marketing such as: capturing the expertise from all over the world, evaluate of the prototype, predict the market, get to know the successful or failing cases of other users, etc.

References

[1] D. Makola, Y. W. Sim, C. Wang, L. Gilbert, S. A. W. Grange and G. B. Wills, 2006, A Service-Oriented Architecture for a Collaborative Orthopaedic Research Environment, Proceedings of 8th Annual Conference on WWW Applications, Bloemfontein, Free State Province, South Africa.

[2] S. Jeschke, C. Thomsen, and U. Sinha, 2006, Collaborative Working Environment for Virtual and Remote Experiments in Nanoscience and Nanotechnologies: T. Reeves & S. Yamashita, Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006, pp. 2055-2060

[3] R. D. Galliers, 1991, Choosing Appropriate Information Systems Research Approaches: A Revised Taxonomy. Information Systems Research.

[4] J. E. McGrath, J. Martin and R.A. Kulka, 1981, Dilemmatics: the Study of Research Choices and Dilemmas, Judgment Calls in Research, Newbury, Park, CA, Sage, pp. 69-102

[5] BIOPATTERN Network of Excellence, http://www.biopattern.org/, accessed on 28th of June, 2007

[6] K. Channabasavaiah and K. Holley, Migrating to a service-oriented architecture, http://www-128.ibm.com/developerworks/library/ws-migratesoa/, accessed on 28th of June, 2007

[7] Medical research methodology, http://www.biomedcentral.com/bmcmedresmethodol/, accessed on 28th of June, 2007

[8] Health research methodology, http://www.fhs.mcmaster.ca/grad/hrm/index.htm, accessed on 28th of June, 2007

[9] R. R. Kodali, What is service-Oriented Architecture? An introduction to SOA, http://www.javaworld.com/javaworld/jw-06-2005/jw-0613-soa.html, accessed on 28th of June, 2007

[10] D. Reilly, Introduction to Java RMI, http://www.javacoffeebreak.com/articles/javarmi/javarmi.html, accessed on 17th of April, 2007

[11] CORBA® BASICS, http://www.omg.org/gettingstarted/corbafaq.htm, accessed on 17th of April, 2007

[12] DCOM, http://en.wikipedia.org/wiki/DCOM, accessed on 10th of June, 2007

[13] Future for European Grids: GRIDs and Service Oriented Knowledge Utilities, http://www.semanticgrid.org/documents/ngg3/ngg3.html, accessed on 17th of April, 2007

[14] What is web service? http://www.webopedia.com/TERM/W/Web_services.html, accessed on 17th of April, 2007

[15] R. Marcus, 2006, Great Global Grid, Emerging Technology Strategies, Trafford

[16] The WS-Resource Framework, http://www.globus.org/wsrf/, accessed on 11th of August, 2006

[17] I. Foster, 2002, What is the grid, A three point checklist, GridToday

[18] IBM Solutions Grid for Business Partners: Helping IBM Business Partners to Grid-enable applications for the next phase of e-business on demand, http://www-304.ibm.com/jct09002c/isv/marketing/emerging/grid_wp.pdf, accessed on 18th of June, 2007

[19] Grida Project Introduction, http://grida3.crs4.it/grida3/doku.php, accessed on 17th of April, 2007

[20] Open Science Grid Home Page, www.opensciencegrid.org, accessed on 17th of April, 2007

[21] Teragrid Home Page, www.teragrid.org, accessed on 17th of April, 2007

[22] Climateprediction.net gateway, www.climateprediction.net, accessed on 17th of April, 2007

[23] AccessGrid Homepage, www.accessgrid.org, accessed on 17th of April, 2007

[24] New Collaborative Working Environments 2020, http://ec.europa.eu/information_society/activities/atwork/hot_news/publications/documents/new_collab_environments_2020.pdf, accessed on 15th of Feb., 2007

[25] Towards a middleware for collaborative working environments, 、 http://ec.europa.eu/information_society/activities/atwork/work_paradigms/experts_group/2004_05_04_expertsgroup_meeting/presentationofgroup/index_en.htm, accessed on 15th of Feb., 2007

[26] L. Bannon, and K. Schmidt, 1991, CSCW - four characters in search of a context: Studies in computer supported cooperative work - theory, practice and design, North Holland, New York, pp. 3-16

[27] P. H. Carstensen and K. Schmidt, 2002, Computer supported cooperative work: New challenges to systems design, Handbook of Human Factors, Tokyo

[28] J. S. Li, D.Z. Su and L. Henshall, May 2004, Development of a Web-enabled environment for collaborative design and manufacture, Proceeding of the 8th International Conference on Computer Supported Cooperative Work in Design, Issue 26-28, Vol. 2, pp. 540-545

[29] B. Christos, L. Petros, B. Antonis, T. Giorgos, November 1998, Web-Enabled Distance Education Environment, WebNet 98 World Conference of the WWW, Internet, and Intranet Proceedings, 3rd, Orlando, FL, pp. 2

[30] A. Gunasekaran, E. W. T. Ngai, R. E. McGaughey, 2005, Information technology and systems justification: A review for research and applications, European Journal of Operational Research, vol. 173, no. 3, pp. 957-983

[31] M. Ulieru, D. Norrie, R. Kremer, W. M. Shen, 2000, A multi-resolution collaborative architecture for web-centric global manufacturing, Information Sciences, Vol. 127, pp 3-21

[32] K. Z. Chena, X. Y. Feng, F. Wanga and X. A. Feng, 2007, A virtual manufacturing system for components made of a multiphase perfect material, Computer-Aided Design, Vol. 39, pp. 112–124

[33] B. Iyer, G. Shankaranarayanan and M. L. Lenard, 2005, Model management decision environment: a Web service prototype for spreadsheet models, Decision Support Systems, Vol. 40, pp. 283– 304

[34]. S.P. Lal and G.C. Onwubolu, Three tiered web-based manufacturing system, 2007， Robotics and Computer-Integrated Manufacturing, Volume 23, Issue 1, pp. 138-151

[35] Y. J. Yang and J. S. Zhang, 29.06.2005, Internet-based collaborative product development chain for networked product development, The International Journal of Advanced Manufacturing Technology, Springer London, Vol. 28, No. 7, pp. 845-853

[36] H. C. Chenga, and C. S. Fenb, 2006, A Web-based distributed problem-solving environment for engineering applications, Advances in Engineering Software, Issue 2, Vol. 37, pp. 112-128

[37] M. Younas, I. Awan and D. Duce, 2006, An efficient composition of Web services with active network support, Journal of Expert Systems with Applications, Issue 31, Vol. 31, pp. 859-869

[38]. J. Zhang, J.Gong, H. Lin, G. Wang, 2007, Design and development of Distributed Virtual Geographic Environment system based on web services, Information Science, Vol. 117, issue 19, pp. 3968-3980

[39] European Commission, 2006, Collaboration@Work: the 2006 report on new working environments and practices, ISBN 92-79-01411-0, European Communities, Luxembourg

[40] European Commission, 2007, ICT-Information and Communication Technologies Work Programme 2007-2008, ISBN 92-79-02828-6, European Communities, Luxembourg.

[41] D. Su, J. Li, Y. Xiong, and Y. Zheng, 2006, Collaborative Design and Manufacture Supported by Multiple Web/Internet Techniques: W Shen et al (ed.), Lecture Notes in Computer Science, ISBN 0302-9743, Springer Berlin/Heidelberg, Vol. 3865, pp. 483-492

[42] X. Chen, T. Luo, Y. He, W. Zhou and D. Sun, May 2005, Research on 3D Feature Modeling technology for Internet-driven Collaborative Design: 9th International Conference on Computer Supported Cooperative Work in Design, Coventry, UK, pp. 649-654.

[43] D. Su, and N. Amin, 2001, A CGI-based approach for remotely excusing a large program for integration of design and manufacturing over the Internet, International Journal of Computer Integrated Manufacturing, Vol. 14, No.1, pp. 55-65

[44] I. Mendikoa, M. Sorli, J. Barbero, and A. Carrillo, May 2005, Knowledge based distributed product design and manufacturing, 9th International Conference on Computer Supported Cooperative Work in Design, Coventry, UK, pp. 679-684

[45] I. Mendikoa, M. Sorli, J. Barbero, A. Carrillo and A. Gorostiza, January, 2006, Inventive approach for problems detection in collaborative design: proceedings, the International conference on Advanced Design and Manufacture, Harbin, China, pp. 424-428

[46] H. W. Wang and H. Zhang, 2006, Enabling Enterprise Resources Reusability and Interoperability through Web Services, IEEE International Conference on e-Business Engineering (ICEBE'06), pp. 655-662

[47] W. D. Lia, W. F. Lua, J. Y. H. Fuhb and Y. S. Wongb, August 2005, Collaborative computer-aided design—research and development status, Computer Aided Design, Vol. 37, Issue 9, pp. 931-940

[48] W. D. Li and Z. M. Qiu, 1 July 2006, State-of-the-art technologies and methodologies for collaborative product development systems, International Journal of Production Research, Vol. 44, No. 13, pp. 2525-2559

[49] Y. Wang and B. O. Nnaji, June 2006, Document-Driven Design for Distributed CAD Services in Service-Oriented Architecture, Journal of Computing and Information Science in Engineering, Vol. 6, Issue 2, pp. 127-138

[50] Y. Pan, D. Xu, C. Chen and Y. Zhang, 2004, Using Web Services Implementing Collaborative Design for CAD Systems: Services Computing, IEEE International Conference on (SCC'04), pp. 475-478

[51] O. Chira, C. Chira, T. Roche, D. Tormey and A. Brennan, December, 2006, An agent-based approach to knowledge management in distributed design, Journal of Intelligent Manufacturing, Vol. 17, No. 6, pp. 737-750

[52] Q. Hao, W. Shen, Z. Zhang, S. W. Park and J. K. Lee, Jan. 2006, Agent-based collaborative product design engineering: an industrial case study, Computer in Industry, Vol. 57, Issue 1,pp. 25-38

[53] D. Bovenzi, G. Canfora and A. R. Fasolino, 2003, Enabling Legacy System Accessibility by Web Heterogeneous Clients, Seventh European Conference on Software Maintenance and Reengineering (CSMR'03), pp. 73

[54] Y. Zou and K. Kontogiannis, May 2001, Towards A Web-Centric Legacy System Migration, International Conference on Software Engineering (ICSE) (4 Pages)

[55] G. Canfora, A. R. Fasolino and G. Frattolillo, 2006, Migrating Interactive Legacy Systems to Web Services, Proceedings of the Conference on Software Maintenance and Reengineering, pp. 24-36

[56] L. Erlikh, August 2002, Integrating Legacy System Using Web services, eAI Journal, pp.12-17

[57] G. Lewis, E. Morris and D. Smith, 2006, Analyzing the Reuse Potential of Migrating Legacy Components to a Service-Oriented Architecture, Software Maintenance and Reengineering, CSMR 2006

[58] Manufacturing industry online collaboration tool, http://www.ferret.com.au/articles/23/0c011d23.asp, accessed on 10th of Dec, 2004

[59] D. Z. Su and X. A. Chen, 2003, Network support for integrated design: Integrated Manufacturing Systems, MCB UP Limited, Vol. 14, No.6, pp. 537-546

[60] Introduction to XML, http://www.w3schools.com/xml/xml_whatis.asp, accessed on 10^{th} of Dec, 2004

[61] Simple Object Access Protocol, http://www.w3.org/TR/soap/, accessed on 10^{th} of Dec, 2004

[62] Web Services Activity, http://www.w3.org/2002/ws/, accessed on 10^{th} of Dec, 2004

[63] Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl, accessed on 10^{th} of Dec, 2004

[64] E. Article, January/February 2004, Gears and the Internet: Gear Technology – The Journal of Gear Manufacturing, pp. 25

[65] Chest Software Evaluation, http://www.eduserv.org.uk/chest/software/evaluations, accessed in March 2004

[66] S. Graham, S. Simeonov and T. Boubez, 2003, Building Web Service with Java: Making sense of XML, SOAP, WSDL, and UDDI, 2^{nd} Edition, Sams Publisher

[67] D. Chapell and T. Jewell, March 2002, Java Web Service: First Edition, O'Reilly Publisher

[68] J. Weise, August 2001, Public Key Infrastructure Overview: Sun PS^{SM} Global Security Practice Sun Blueprints™ Online

[69] N. Amin, 2002, Intelligent optimum design with the support of Internet techniques, PhD thesis, The Nottingham Trent University, UK

[70] D. Coward and Y. Yoshida, 24^{th} of November 2003, Java™ Servlet API Specification version 2.4, JSR 154 Sun Microsystems, Inc.

[71] P. Delisle and M. Roth, 24^{th} of November 2003, JavaServer Page™ Specification version 2.1: JSR 245 Sun Microsystems, Inc.

[72] D. Su (ed.), Internet Based Engineering: Applications and Case Studies, ISBN 1-84233-021-7, Nottingham

[73] I. Foster, C. Kesselman and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations,

http://www.globus.org/research/papers/anatomy.pdf, accessed on 20th of Dec, 2004

[74] K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, and S. Tuecke, Infrastructure to WS-Resource Framework: Refactoring and Extension: globus project whitepaper, http://www.globus.org/wsrf/specs/ogsi_to_wsrf_1.0.pdf, accessed on 10th of Dec, 2004

[75] K. Czajkowski, D. Ferguson and I. Foster, The WS-Resource Framework: globus project whitepaper, http://www.globus.org/wsrf/specs/ws-wsrf.pdf, accessed on 10th of Dec, 2004

[76] M. C. Brown, Merging Grid and Web services: IBM DeveloperWorks, http://www-128.ibm.com/developerworks/grid/library/gr-web/, accessed on 12th of Dec. 2004

[77] V. Silva, Transferring files with GridFTP: IBM DeveloperWorks, http://www-106.ibm.com/developerworks/grid/library/gr-ftp/, accessed on 5th of March 2005

[78] Virtual Learning Portal, The Nottingham Trent University, UK, http://www.elearn.ntu.ac.uk, accessed in March 2004

[79] E. So, M. Collins, R. Lee, R. Lawless and S. Reilly, P2P Search Engines, http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p8.html, accessed on 20th of June, 2007

[80] W. J. Peng, Y. Xiong, S. Y. Ji and D. Z. Su, May 2004, A Virtual Research Institute and Its Utilization in CAE for Worm Gear Drives and Gear Design Optimisation: 8th International Conference on Computer Supported Cooperative Work in Design, Xiamen, China, pp. 552-557

[81] Y. Xiong and D. Su, 2006a, A Web-Service Based Approach for Software Sharing: W Shen et al (ed.), Lecture Notes in Computer Science, 'Computer Supported Collaborative Work in Design II', ISBN 0302-9743, Springer Berlin/Heidelberg, Vol. 3865/2006, pp. 215-224

[82] Y. Xiong and D. Su, 2006b, A Grid Based System for a Manufacture/Retailer Virtual Organization: proceedings, International Conference on Advanced Design and Manufacturing

[83] Forrester, B2B e-commerce to hit €2.2 trillion in EU by 2006, http://www.out-law.com/page-2881, accessed on 25th of Jan., 2007

[84] Kersten, E. Gregory, Noronha and J. Sunil, 1999, www-based negotiation support: design, implementation and use, Decision Support System, Vol. 25, pp. 135-154

[85] S. Billery, L.A. Chanz, D. Simchi-Levix and J. Swann, April, 2005, Dynamic Pricing and the Direct-to-Customer Model in the Automotive Industry, Electronic Commerce Journal, pp.309-334

[86] M. Muffatto and A. Payaro, August 2004, Integration of web-based procurement and fulfillment: A comparison of case studies, International Journal of Information Management, pp. 295-311

[87] S. Pant, R. Sethi, and M. Bhandari, June 2003, Making sense of the e-supply chain landscape: an implementation framework, International Journal of Information Management, pp. 201-221

[88] T. C. Jones and D. W. Riley, 1987, Using inventory for competitive advantage through supply chain management, International Journal of Physical Distribution and Materials Management, pp. 94-104

[89] K. Takahashi, K. Morikara and N. Nakamura, 2004, Reactive JIT ordering system for changes in the mean and variance of demand, Internation Journal of Production Economics, pp. 181-196

[90] S. M. Disney and D. R. Towill, 2003, The effect of vendor managed inventory dynamics on the bullwhip effect in supply chains, International Journal of Production Economics, pp. 199-215.

[91] M. Anvari, 1992, Electronic data interchange and inventories, International Journal of Production Economics, pp. 135-143

[92] M. Verwijmeren, 2004, Software component architecture in supply chain management, Computers in Industry, pp. 165-178

[93] P. Fiala, Information sharing in supply chains, Omega International Journal of management science, pp. 419-423

[94] M.A.A.P. Verwijmeren, 2000, Exploiting distributed object technology to achieve networked inventory management, Computers in Industry, Vol. 41, pp. 239-250.

[95] S. Khoshafian, 2002, Web services and virtual enterprises: TECT, http://www.webservicesarchitect.com/, accessed on 30th of Dec., 2005

[96] S. Dowlatshahi, Q. Cao, The relationship among virtual enterprise, information technology, and business performance in agile manufacturing: An Industry perspective, European journal of Operational Research

[97] T. Y. Kim, S. Lee, K. Kim, and C. H. Kim, April 2006, A modelling framework for agile and interoperable virtual enterprises, Computer in industry, Vol. 57, Issue 3, pp.204-217.

[98] H. Gou, B. Huang, W. Liu and X. Li, 2003, A framework for virtual enterprise operation management, Computer in industry, pp. 333-352

[99] X. Zhao, X. Liu, and Y. Wang, 2005, Model of virtual enterprise inventory management system based on UML, Modular Machine Tool & Automatic Manufacturing Technique, No.11, pp. 89-91

[100] D. Achabal, H. Mcintyre, A. Smith and K. Kalyanam, A Decision Support System for Vendor Managedd Inventory, Journal of Retailing, Vol. 76, pp. 430-454

[101] A. Felice, D. Toni and E. Zamolo, 2005, From a traditional replenishment system to vendor-managed inventory: A case study from the household electrical appliances sector, International Journal of Production economics, pp. 63-79

[102] Q. Chen, H. Ghenniwa, and W. Shen, Web service-based wide area protection system design and simulation, Proceeding of the CSCWD2006 Conference

[103] J. T. Pollock and R. Hodgson, 2004, Adaptive information: improving business through semantic interoperability, grid computing, and enterprise integration, Wiley series in systems engineering management

[105] N. B. Szirbik, D. K. Hammer, J. B. M. Goossenaerts and A. T. M. Aerts, 1999, Mobile agent support for tracking products in virtual enterprises, Proceedings of the AGENTS'99 Conference on Working Papers of the Workshop on Agent-Based Decision-Support for Managing the Internet Enabled Supply Chain, Seattle, pp. 93–100

[106]. http://en.wikipedia.org/wiki/Software_license, accessed on 10th of December, 2007.

[107] http://www.steptools.com/library/standard/, accessed on 1st of July, 2007

Appendix
Function Specification
For the Developed Systems

A.1 Layout for Source Code Storage

The core source codes for the three systems developed are stored in three sub-folders in the root folder, which are: SOF-CAD Source Code Folder, Software Bank Source Code Folder, and CWE Source Code Folder. All the three sub-folders are under the Root Folder. The core source files of each system and their storage structure are illustrated below Figure A-1.:

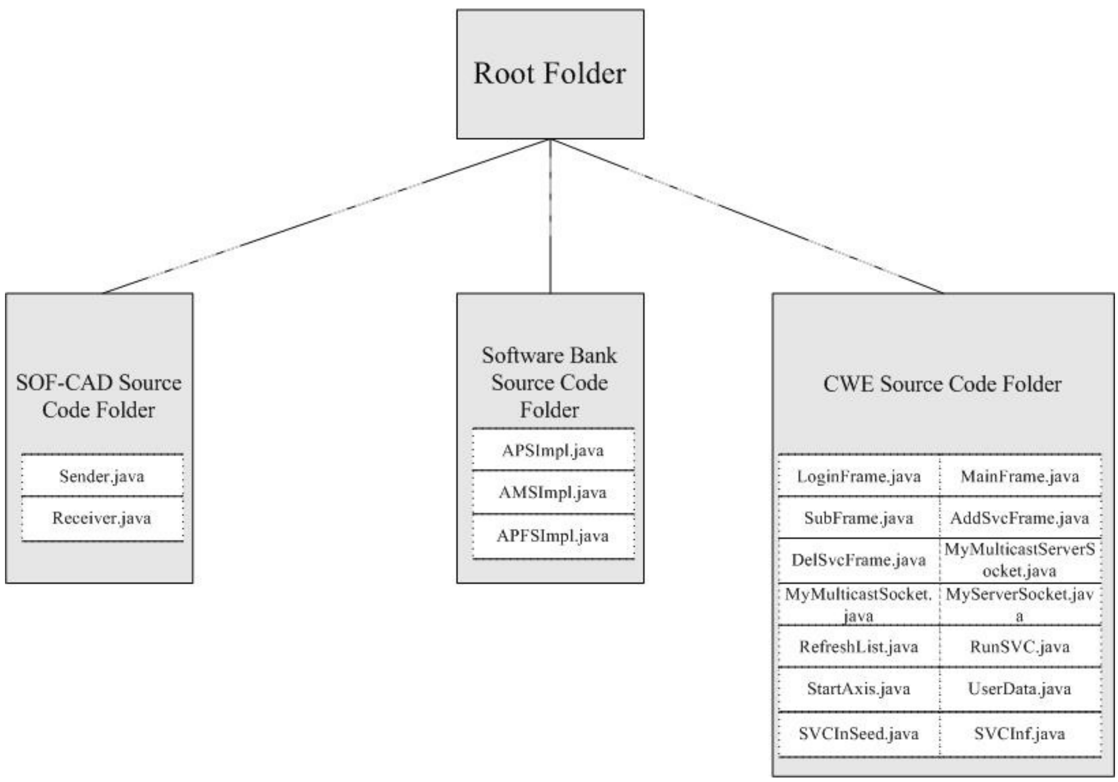


Figure A-1 File storage for the source code.

1. The SOF-CAD source folder contains two parts: sender.java as client side of the system, and receiver.java as server side of the system. The details of the two parts are presented in Section A.2.
2. Software Bank source folder contains three major parts of the system: APSImpl.java as the implementation of APS, AMSImpl.java as the implementation of AMS, and APFSImpl.java as the implementation of APFS. The core functions for each part are described in Section A.3
3. CWE source folder contains the core code for the software, and this system is more complicated, the fourteen core code documents are described in Section A.4.

A.2 Function specification for SOF-CAD

The main client functions are in Sender.java, and the main server functions are in Receiver.java

Client Functions:

Sender.java:

1. *void ensureExist()*, this function will check the designer's system on whether his database for design is available, if the database is not available, the operation will stop right here, other wise, the operation will move on.

2. *void ensureModified()*, this function will check the designer's database on whether it is modified. If the database for design is modified, the system will capture the change made by the user.

3. *byte[] bufRead()*, this function is used to read the changes and wrap the description into bytes format, so that it can be wrapped in SOAP message.

4.*void process()*, this function is used to invoke the functions of *ensureExist()*, *ensureModified()* and *bufRead()*.

Server Functions:

Receiver.java:

.

1. *byte[] initServe()*: this function will initiate the server by checking if the server is in right status.

2. *public String callFile(byte buf[]*: this function is used to transfer the information from the service requester to service provider.

3. *public byte[] sendFile()*: This function is used to transfer the information from the service provider to the service requester.

A.3 Function Specification for Software Bank

AMS, APS, APFS have different functions and they are described as follows:

In AMS side, the functions are described below:

1. *public boolean login(String in0, String in1),* this is the function for the security checking when the user application intend to access the service..

2. *public String createApplication(String appName, String userId, String pwd),* this is the function, when the request from user passed the login function, to create an application instance with the specified application name, username and password.

3. *private Properties loadProp(),* this is the function that can get the properties of the application.

4. *public void destoryApplication()*, this is the function to destroy the application instance. This function is used after the mission of the application instance is finished, it should be destroyed. AMS is only for management issue..

In APS side, the functions are described below:

1. *public void execute()*, this function is to execute the files on the server.
2. *public void setWorkingDIR(String in0),* this function will help set up the working folder for the request from user application, the folder is used to store the outcomes of each mission. The server usually serves a lot of request in the same time, however, for each request, different folder will be set up for it.
3. *public ResultVO getResult()*, this function will delivery the result from the server to the service requester.
4. *public float getStatus(),* by checking the outcomes from the server, this function can inform the user know the real time progress made by the calculation.

5. *public void destory()*, this is to destroy the mission and delete the files, in AMS implementation, the asp.destroy() fucntion actually goes to here.

In APFS side, there is one core functions as described below:

public String createAPS(), this function is used to generate APSs.

A.4 Function Specification for CWE.

There are fourteen important java classes in this system; the following is the description of the functions.

1. LoginFrame.java
This class(every Java document is a class) aims to draw the Login frame, and receive the nickname from the users. Through the ActionListerner function to launch the Mainframe class.

2. MainFrame.java
This class draws the main frame, and launches the Axis2 server. Go through the user list of the frame to listen to the actions. The MouseListener function launches the SubFrame class. The two buttons in the frame belong to Jbutton class and they will respectively launch the ActionListener and start the AddSvcFrame class and DelSvcFrame class.

3.SubFrame.java
This class is to create the service list frame, through event listener where the service list Jlist is registered(MouseListener) to start RunSvc class.
The main method in this class is:
void requestService(DefaultListModel ax, InetAddress userInetAddr)
This method request service list from userInetAddr, and then add it to the local list as where the Jlist is, this method use the"RequestService" to communicate, and get the service information from the destination, then create service information class MyService and add it to ax.

4. AddSvcFrame.java
This class is to launch the service frame, because Axis2 support plugin, so to add a new service, the user just needs to copy the server package and past into the specified folder.
The main functions includes
1) *void startAddSvcFrame()*

This method is used to launch the frame.
2). *void copytoDir_Client(ZipInputStream in, ZipEntry ze, long zipLength)*
This method is used to unzip the client files into the specified folder: axis2/repository/client
3). *void copytoDir_Service(ZipInputStream in, ZipEntry ze, long zipLength)*
This method is used to unzip the server files into the specified folder: axis2/repository/server.

5. DelSvcFrame.java
This class is to delete the service frame and delete the service.
The main functions includes:
1). Void startFrame()
This function is used to launch the frame.
2). Void requestService(DefaultListModel ax)
This function is used to request for the servicelist and put it into ax.

6.MyMulticastServerSocket.java
This Class is used to receive multicast packets and check the userlist.

7.MyMulticastSocket.java
This class sends multicast packets; inform the availability of user to any other users.

8. MyServerSocket.java
This class inherits the Tread class, launches the ServerSocket Class. It is used to process the socket connection during the running of the programme. Based on different protocols, it can transfer or receive correspondent information.
The main function for the class:
public void run()

This method is used to launch ServerSocket class and then waits for the coming connections. After the connection and based on its different parameters, it will process data transfer. After it received "RequestServiceList", it will then start to transfer the current service name, or transfer the client information.

9.RefreshList.java
This class inherits the Thread Class, the function is to refresh the Jlist. Because the data was stored in the List<UserData> myFriends has to be updated to the Jlist in MainFrame.
The main method is:
public void run()
This method repeatedly clears the list, and also add users who have received multicast packets.

10.RunSvc.java

This class is used to download the client and run main functions include:

1) public void run()

This method starts the thread and run the following two sub methods.

2). void downloadClient()

This method will download client and build the connection through socket.

3). public void runclient()

This method will run the client, the client is a JAR package, can be invoked by launching the thread methods.

11.StartAxis.java

This class launches the server, the main functions are:

1). public void run()

This method starts thread and run the sub-method startAxis2(String[] args) as below..

2). *void startAxis2(String[] args)*

This method launches Axis2 server, and in the mean time launches MyServerSocket class.

12.UserData.java

This class stores the user information, which includes: “user nickname, String username” and user address”InetAddress userInetaddr.

13.SvcInfSeed.java

This class is to store the information for single service, this includes: Service name: String svcName, Service Descrption: String description. This class also stores the information of client this includes client main function name String mainstart and Client name: String client_name.

14. SvcInf.java

This class is used to store the service list: LinkedList<SvcInfSeed>:

Main functions:

1)void addSeed(SvcInfSeeds).

Add a service information to the service list.

2). Void delSeed(String text)

Delete the service which names *text.*