Mobile Collaborative Working Environment of Product Design

Yongjun Zheng

A thesis submitted in partial fulfilment of the
requirements of Nottingham Trent University
for the degree of Doctor of Philosophy

This research programme was carried out at
Advanced Design and Manufacturing Engineering Centre,
School of Architecture, Design and the Built Environment,
Nottingham Trent University,
Burton Street, Nottingham,
NG1 4BU, UK

October 2009

Abstract

In response to the arrival of new Web/Internet environments, one of the most attractive challenges in current research is to exploit wireless computing technologies in collaborative product design, and hence to build a ubiquitous mobile information system to enable the collaborative product design within a mobile environment. However, the literature review reveals that although the progress of mobile technologies on wireless networks has largely changed the way people access the Internet; little has been achieved in mobile computing for collaborative product design. The reason is that, due to the distinct features of mobile devices and wireless networks (such as small display screen, limited bandwidth, unreliability of wireless networks, etc.), the methodologies and technologies used in stationary networks are not always applicable to mobile systems.

The aim of this research is to establish a Wireless Internet-based Collaborative Working Environment for product design through the combination of multiple technologies, by including: Web services, Parametric Design, the Semantic Web, Agent and Flex Technologies. In order to create, deploy, and manage the distributed resources, Web service is used to implement design resource integration in a platform-independent manner. In addition, Semantic Web Technology is used to create a general knowledge base. This approach includes two components: (1) ontology is used to represent abstract views of product data and (2) added semantic rules are also used to represent relationships among product data. Therefore, an ontology-based description model is thus proposed to facilitate expression and organisation of product information in order to manage and deploy the distributed design resources.

This research presents a mobile agent system for collaborative design, supporting the construction of mobile agents which can migrate and access the distributed design resources. After that, in order to resolve the Mobile Platform Compatibility problem, this research presents the most recent Flex technology to enable the mobile applications to be implemented on the different mobile devices; furthermore, with the support of combined Flex technology and Web service, remote users are able to invoke a large-scale computing program via mobile Web browser.

The combination of all these technologies provides the cornerstone and effective support in building scalable, extensible interactive mobile systems for collaborative design, as illustrated and demonstrated in this thesis.

Keywords: Collaborative Environment, Mobile network, Web service, Flex, Semantic, Ontology, Agent, Parametric Design, Product Design

Acknowledgement

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

Sincere thanks are due to my supervisors Mr, Paul Johnson and Mr, Leslie Arthur for their advice, comments, support and interests in my work. I am particularly grateful for Mr Arthur's kind support on my English.

I would like to thank Nottingham Trent University for providing the scholarship which made this research possible and enjoyable.

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. My special thanks are to all my colleagues and friends who made these years memorable for me. Without whom I would have been lost.

My greatest thanks are to my parents, my sister and my wife, for their immense love and supports.

Yongjun Zheng

28/08/2009

Publications

Journal Papers:

1) **YongJun Zheng**, Zhong MingRen, DaiZhong Su, Leslie Arthur, Semantic Similarity Based on Ontology Technology and its Application in Mobile Information Retrieval for Product Design, Key Engineering Materials Vols. 419-420 (2010) pp 741-744

2) Daizhong Su, Yu Xiong, Shuyan Ji and **Yongjun Zheng**, Framework for a Collaborative Working Environment, International Journal of Production Research, 46(9), 2008

3) 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:

4) Daizhong Su and **Yongjun Zheng**， Utilization of the Collaborative Working Environment for Online Computer Aided Mechanical Design， Keynote speech, 13th International Conference on Machine Design and Production (UMTIK 2008), 3-5 September 2008, Istanbul, Turkey.

5) **Yongjun Zheng**, Leslie Arthur, Cao Li, Daizhong Su, Collaborative Design within an Agent-based Mobile Environment, the 8th International Conference on Frontiers of Design and Manufacturing, September 23~26, 2008, Tianjin, China

6) Daizhong Su and **Yongjun Zheng**, Development of a Prototype Mobile Collaborative Environment for Product Design, Expanding the Knowledge Economy: Issues, Applications, Case Studies, 749-756, Amsterdam 2007

7) **Yongjun Zheng**, Paul Johnson, Daizhong Su, REVIEW OF WEB-BASED COLLABORATIVE PRODUCT DESIGN, International Conference on Advanced Design and Manufacture, 8th-10th January, 2006, Harbin, China.

8) Daizhong Su, Yu Xiong, Shuyan Ji, **Yongjun Zheng**, Jiansheng Li, EU ASIA IT&C AND ASIA-LINK PROGRAMMES SUPPORTED RESEARCH IN WEB-BASED COLLABORATIVE WORK ENVIRONMENT AND ITS FURTHER DEVELOPMENT, International Conference on Advanced Design and Manufacture, 8th-10th January, 2006, Harbin, China.

9) Daizhong Su, Jiansheng Li, Yu Xiong, **Yongjun Zheng**, Application of Internet Techniques into Online Collaborative Design and Manufacture, 9th International Conference on Computer Supported Cooperative Work in Design, 24-26 May， 2005, Coventry, UK.

Nomenclature

ACL	Agent Communication Language
ADMEC	Advanced Design and Manufacturing Engineering Centre
AIR	Adobe Integrated Runtime
AMS	Application manager service
APFS	Application proxy factory service
API	Application Program Interface
APS	Application proxy service
APT	Automatic Programmed Tools
C/S	Client/ Server
CAD	Computer Aided Design
CAM	Computer Aided Manufacture
CAPP	Computer aided process planning
CeCAD	PowerCAD CE for Pocket PC
CWE	Collaborative Working Environment
CORBA	Common Object Request Broker Architecture
CPAD	Collaborative product assembly design
CPD	Collaborative Product Design
CSCW	Computer Supported Cooperative Work
CSS	Cascading Style Sheets
CWE	Collaborative Working Environment
DARPA	Defence Advanced Research Projects Agency
DCOM	Distributed Component Object Model
E-tong	E-tong electromechanical technology Co.,Ltd
FIPA	Foundation for Intelligent Physical Agents
FP6	Sixth Framework Programme
GAFR	Genetic Algorithm for Features Recognition
GDEs	Geometric Datum Elements
GUI	User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol Overview
Hunan Power	Hunan Huadian Changsha Power Generation Co., Ltd,
ICT	Information and Communication Technologies
IDL	Interface Description Language
IPDD	Integrated Product Design and Development
ISA	Is-a
ISO	International Organization for Standardization
IST	Information Society Technologies

JADE	Java Agent DEvelopment Framework
JAVA ME	Java Platform, Micro Edition
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
MAS	Multi-Agent System
MCWE	Mobile Collaborative Working Environment
MIT	Institute of Technology
NC	Numerically Controlled
OLE	Object Linking and Embedding Automation
OS	Operation System
OWL	Web Ontology Language
PA	Parametric Agent
PDA	personal digital assistant
PDM	Product data management
PLF	Product Layout Feature
RA	Requestor Agent
RDF	Resource Description Framework
RIA,	Rich Internet Applications
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SA	Server Agent
SMEs	Small and medium enterprises
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SS	Semantic Similarity
STEP	STandard for the Exchange of Product Data
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
UNA	Unique Name Assumption
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VGA	Video Graphics Array
VRCE	Reality-Based Collaborative Environment
VRML	Virtual Reality Modelling Language
W3C	World-Wide Web Consortium
WSO -MCAD	Web Service Oriented Mobile Computer Aided Design
WECE	Web-enabled Collaborative Environment
WSDL	Web Services Description Language
XML	Extensible Markup Language

Content

Abstract I

Acknowledgements III

Publications IV

Nomenclature VI

Contents VIII

List of Figures & Tables XII

Chapter 1 Introduction 1

1.1 Background 1
1.2 Description of the Research 3
1.3 Aim and Objectives of this Research 7
1.4 Research Method 7
1.5 Outline of This Thesis 9

Chapter 2 Literature Review 12

2.1 Introduction 12
2.2 The Fundamental of Collaborative Product Design 13
2.3 Web-Based Collaborative Design 16
2.4 Agent Technology 19
2.5 Mobile Technology for Web Collaboration 22
2.6 Semantic Web Technology Used for Collaborative Design 24
2.7 Mobile Platform Compatibility: Flex Mobile Applications 29
2.7.1 Comparison of Three Mobile Development Technologies 30

2.7.2 Descriptions of Flex Technology 31
2.7.3 Advantages of Flex Mobile Applications 32
2.8 Conclusions 33

Chapter 3 The Framework of Mobile Collaborative Working Environment 36
3.1 Introduction 36
3.2 Service Oriented MCWE 41
3.2.1 Introduction of Web Service Technology 41
3.2.2 Comparison of Web Service and Other Distributed Technologies 43
3.2.3 View, Modify and Share CAD files via Mobile Device 44
3.2.4 Parametric Design 45
3.3 Semantic Web and Agent Technology for MCWE 46
3.4 Mobile Platform Compatibility: Flex Applications within MCWE 47
3.4.1 Mobile Platform Compatibility 47
3.4.2 Combination of Flex with Web Service 47
3.5 Summary 49

Chapter 4 Web Service Oriented Mobile Computer Aided Design (WSO-MCAD) 51
4.1 Introduction 51
4.2 Main Features of Web Services Utilized in WSO-MCAD 53
4.3 Development of Web Service for Online CAD within WSO-MCAD 53
4.3.1 Web Service Operation Procedure 53
4.3.2 System Components 54
4.3.3 Test tool and Discussion 61
4.4 Parametric Design 63
4.5 Experiment 65
4.5.1 Bolt design 65
4.5.2 Collaborative CAD Design via PDA 69
4.6 Conclusions 70

Chapter 5 The Application of Semantic Web Technology into Mobile Collaborative Design 72
5.1 Introduction 72
5.2 Ontology Construction and Modelling Technology 75

5.2.1 Build OWL Ontology 75
5.2.2 Ontology Modelling using Protégé 4 77
5.2.3 Ontology Modelling for Shaft Design 81
5.3 Method of Semantic Similarity 83
5.3.1 Semantic Similarity 83
5.3.2 Similarity Calculation 86
5.4 Through Jena API to Retrieve Semantic Data 87
5.5 Case Study 89
5.6 Conclusions 96

Chapter 6 The Mobile Agent for collaborative product Design 97
6.1 Introduction 97
6.2 Overview of Multi Agent System 98
6.2.1 Overview of Agent System 98
6.2.2 Agent Communication 100
6.3 Agent mobility 101
6.4 Experiment 102
6.5 Conclusions 106

Chapter 7 Mobile Platform Compatibility: Flex Applications within MCWE 107
7.1 Introduction 107
7.2 Development of the Method to Utilise Flex application for Communication with Web server 108
7.3 Using Flex Builder to Create Web Service-Based Flex Mobile Applications 110
7.3.1 WSDL Mapped to Flex Application 111
7.3.2 Flex application Invoke Operation within WSDL 111
7.3.3 Flex Application Request Mapping for Web Service 112
7.3.4 Security of Flex Application 115
7.4 Experiment 115
7.4.1 Mobile Platform Compatibility: Flex Applications across Windows PDA and Nokia PDA 116
7.4.2 Two Methods to Deploy Flex Application into Mobile Device 118
7.5 Conclusions 119

Chapter 8 MCWE Applications for Product Design 120
8.1 Case study 1: Product Data Sharing and Retrieving within MCWE 120

8.1.1 Introduction 120
8.1.2 Current Problems 122
8.1.3 Application of the Information System 123
8.1.4 Mobile System Design 127
8.1.5 Modelling of electromotor 128
8.1.6 System Design and implementation 133
8.2 Case study 2: Mobile Platform Compatibility, Remote Invocation of a Large-scale Computing Program 141
8.2.1 Standalone Design Application Package 142
8.2.2 Wireless Features Solution 142
8.2.3 Experiment 145
8.2.3.1 Login Function 145
8.2.3.2 Mobile Flex Application Communication with Web Server 146
8.2.3.3 Data Retrieval and display Result 150
8.3 Evaluation and Discussions 152

Chapter 9 Discussions and Conclusions 154
9.1 Discussing on Contributions 154
9.1.1 Contributions to Knowledge 155
9.1.2 Technical Contributions 156
9.2 Conclusions 157
9.2.1 Web Service Oriented Mobile Computer Aided Design 157
9.2.2 Application of Semantic Web Technology into Mobile Collaborative Design 158
9.2.3 Agent Technology for WSO-MCAD 158
9.2.4 Mobile Flex Application for WSO-MCAD 159
9.2.5 Case Study: Mobile Applications for Product Design 160
9.2.6 Limitations of the research 160
9.3 Future work 161

References 164

Appendix A Mobile Devices Implementation within MCWE 171

Appendix B The Sourcode of Main Functions 179

List of Figures & Tables

Figures	Title	Page
2.1	Steps of product development process	14
2.2	View product design data in different Mobile Devices	22
2.3	Semantic Web layered architecture	24
2.4	A screenshot of the SWoogle	27
3.1	Structure of the MCWE framework	37
3.2	MCWE: illustration of the framework	40
3.3	Component architecture for Web Service	42
3.4	Flex mobile application communicate with Web server	49
4.1	System modules and data transmission flow	56
4.2	Data process of Request Module	57
4.3	Conversion functions of VB.Net	58
4.4	Structure of Response Module	59
4.5	Process of Service Module	60
4.6	SoapUI Test Snapshot	62
4.7	Architecture of SolidWorks/CAD parametric design	64
4.8	Mobile bolt input interface	66
4.9	Parametric data of bolt and 2-D CAD picture	67
4.10	VB parametric product design code	68
4.11	SolidWorks Jpg file shown in PDA	68
4.12	Mobile CAD File transfer	70
5.1	Top-level classes and relations within Protégé 4	76
5.2	Classes and SubClasses within Protégé 4	78
5.3	Relations among A_Class_Hex_Bolt_M10 within Protégé 4	79
5.4	Individuals within Protégé 4	80

5.5	Specific dimensions of a shaft	81
5.6	Ontology model of Shaft	82
5.7	Abstraction of nodes hierarchy tree	84
5.8	Overall structure of the inference machinery	88
5.9	Jena ontology rule: products of same supplier	89
5.10	Example sequence of retrieving product data	95
6.1	Mobile agent mobility	101
6.2	Mobile search input interface	103
6.3	illustrations of the outcomes	103
6.4	Mobile Agent immigrate within WSO-MCAD	105
7.1	Flex application communicates with Web server	110
7.2	Mappings for service and port	111
7.3	Mapping for operation (Top: WSDL File, Bottom: MXML File)	112
7.4	Flex binding Web service	113
7.5	Mapping of multiple parameter operation for a Document-Literal service call	114
7.6	Sandbox file for security: crossdomain.xml file	115
7.7	Mobile Platform Compatibility: Flex Applications across Windows PDA and Nokia PDA	117
8.1	Air view of the power station	121
8.2	CAD Drawing of E-tong products	122
8.3	Relationship of four companies	123
8.4	Main function structure of WSO-MCAD	125
8.5	Architecture of Mobile Design function	127
8.6	CAD Files of Electromotor	133
8.7	Diagram of WSO-MCAD	134
8.8	2-D picture of wheel bracket	136
8.9	Bolt min diameter of property within Protégé (red colour rectangle)	136

8.10	Mechanical property levels of Protégé (red colour rectangle)	137
8.11	Mobile Interface of Bolt Search	140
8.12	Hybrid architecture of the system	144
8.13	Login MXML	145
8.14	Mobile Flex application-Input Parametric Data	148
8.15	Flex Mobile Application Mapping Web service: Document Oriented – Literal (Top: WSDL XML file, Below: MXML code)	149
8.16	Mobile Flex Application Retrieve XML data from Web server (red colour line)	150
8.17	Mobile Flex application-Result data and graphic	152
Tables	**Title**	**Page**
2.1	Sales market coverage of different mobile OS	30
2.2	Comparison of three mobile technologies	33
5.1	Semantic Similarity results	87

Chapter 1

Introduction

This chapter provides the introduction and the background of the research; it discusses the necessity of carrying out this research, summarises the discoveries made and achievements attained over the course of the investigation and presents the layout of the thesis.

1.1 Background

Many advanced computerised tools such as Computer Aided Design (CAD) and Computer Aided Manufacture (CAM) systems have been successfully used for over two decades and still play important roles in product design and manufacture today [1]. The CAD/CAM systems enable engineers to design, analyse, simulate, and test products by means of digital prototyping in the early stage of the product development process without having to actually manufacture any physical parts [1]. These tools have dramatically boosted companies' ability to produce higher quality products in a shorter time at a lower cost. But today, product development team members are often geographically dispersed, which requires collaboration among the different sites where the team members are located. As a result of such a product development paradigm, there is a great demand for Web/Internet software tools and techniques to effectively support the representation, collection and exchange of product information during the design process [2, 3].

In order to deploy Web/Internet software tools and techniques effectively support the representation, collection and exchange of product information during the design process, the European Union set up the 'Collaborative Working Environment (CWE)' as one of the strategic objectives for research funding under the Sixth Framework Programme Information Society Technologies (IST) programme in 2005, and sponsored 13 projects across a wide range of areas including an effort to catalyse rural development, research of distributed virtual manufacturing enterprises, virtual engineering for small and medium enterprises (SMEs), a digital factory for human-oriented production, strategic innovation, advanced robotic systems, human operators in emergency/disaster scenarios and a variety of others [4]. In the European Union's Information and Communication Technologies (ICT) theme for the 7th Frame Programme [5], it has been stated that 'the current Internet, **mobile**, fixed and broadcasting networks and related software **service infrastructures** all need to progress in order to enable another wave of growth in both the on-line economy and our society over the next 15 years.' 'The challenge is to deliver the next generation of ubiquitous and converged networks and service infrastructures for communication, computing and media.' The achievement of these goals will demand new collaborative working environments and associated technologies [6].

In response to the arrival of the new generation of Web/Internet environments, one of the most attractive challenges in current research is to exploit wireless computing technologies in the process of collaborative design, which may help to build a ubiquitous mobile product information system and support collaborative works within a mobile environment [7]. However, mobile devices (such as pocket computers, wireless handheld devices, mobile phones, etc) are currently being used more often as personal tools rather than design tools. New system aimed at providing a collaborative work environment for the sharing of design information, data and knowledge among distributed team members through mobile devices. With the aid of this system, the designer can collaborate with other team members online, a manufacturing engineer in the workshop can remotely access the product database stored in the Web server, process planning engineer can utilise and modify product data from any location and managers can access the product database when they are away from their offices; remaining in contact with their staff members by online communication, and being able to actually see drawings and relevant data, rather than by just hearing about them via telephone or email.

The aim of this research is to apply advanced Wireless Internet technologies to develop a Web-enabled environment for mobile users to enhance their capability by employing a wide range of information technologies in product design. The research results will include innovative solutions for collaborative design, support for nomadic work scenarios, and resource sharing and management; all that integrated into one unified platform usable in wireless environments.

1.2 Description of the Research

This research is initiated from the Advanced Design and Manufacturing Engineering Centre (ADMEC)'s previous Asia-Link and Asia IT&C projects; the two projects are supported by the European Commission and the work conducted includes the development of a Web-enabled environment [3, 8], and systems designed to facilitate online collaborative computer aided design and manufacture [9, 10], effective remote-execution of large size programs [11] and Web-enabled distributed product design [12]. This research focused on how to apply the Mobile Collaborative Working Environment (MCWE) within the field of Product Design.

Product design can be defined as the ideas generation, concept development, testing and manufacturing or implementation of a physical object or service. Product Designers conceptualise and evaluate ideas, making them tangible through products in a more systematic approach [13].

Collaborative product design is a broad term that is closely related to several technical and social disciplines such as engineering, computer science, human computer interaction, decision-making behaviour and the social sciences. In this research, the term “collaborative design” refers to a design activity where distributed applications will be shared and administered by an active main designer working towards the solution of a complex design problem over the Internet, as well as the sharing and retrieving of product data. From the viewpoint of product design, some collaborative design challenges for mobile system need to be resolved, such as:

- **View, Modify and Share CAD data via a mobile device:** Designers are able to work interactively for CAD functions in real time within Wireless Collaborative Working Environment, enable them to carry out tasks such as instantly modifying the drawings online.

- **Parametric Design utilisation on mobile devices:** Fast reaction to alternative design modification must be enabled; the system design should provide an integrated environment in which to implement the parameterised design mechanism. For each modification of the design input, the system must be designed to facilitate viewing of resultant data, geometrical shape creation and manufacturing checks.

- **Semantic Product Information System:** System should support the accessing and management for product data, provide an economic method of improving the information infrastructure within organisations, the mobile system could also provide more “semantic” answers rather than just send data back to the end user. A simple example used to motivate the Semantic Web Technology is the need to discover documents on the Web, not only from their textual content, as conventional search engines do, but also from a description. For example, in the response to the query “Bolt factory”, a search engine will respond with all the information not only about the factory, but also the CAD file, parametric data about Bolt; however these return data are too much for the remote user to view on the small mobile screen, Semantic Web Technology allows each product on the

Web to be described by who its supplier was, when it was purchased, and what data information it had; thus only those results with the appropriate data will be sent back.

- **Mobile Platform Compatibility Applications:** mobile applications are developed by Microsoft VB.Net, which could retrieve and share product data on the Windows Mobile Devices well. However, this mobile application could not be well supported by other Mobile Platforms. New system need to present and choose the latest mobile technology which could not only communicate with Web server, but be implemented across the different mobile Platforms.

Due to the distinct features of mobile devices and wireless networks (such as limited bandwidth, unreliability of wireless networks, etc.), the methodologies and technologies used in stationary networks are not always applicable to mobile systems; therefore, in order to deploy successful mobile applications within this distributed environment, the following major challenges must be overcome:

- Issues with network connectivity, which can have a large impact on the volume of data that may be downloaded by mobile devices; mobile users may not always be able to connect to the Internet. Sometimes, rapid access to information is required, thus, the issue of how to deal with various network conditions needs to be addressed when creating a mobile application.

- The physical constraints of mobile devices, such as screen size, display resolution, limited computational power and size of memory available, which can significantly affect the usability of mobile applications. It is a challenging task to deploy the design applications in mobile devices, because most graphics systems and CAD functionalities require powerful computer to perform their tasks.

The observed limitations and requirements of current product collaborative design system have led to the following formulated research question in this thesis: which tools, methods, and architectures are needed to develop a distributed system that can support collaborative product design over the Internet? This is a broad research question that can be further divided into a few major sub-questions, namely:

- How to View, Modify and Share CAD data via mobile device?

- How may the different product design resources communicate with each other? How can huge numbers of complex product design resources, which are distributed, heterogeneous, dynamic, and continuously evolving, be efficiently integrated?

- How to share and exchange data in different formats (databases, spreadsheets, Extensible Markup Language, etc.) within a distributed enterprise and its various partners, and return the related data back to the mobile users.

- How to develop the mobile platform-compatibility applications, how to remote execute a large-size computing program via mobile device without the necessity of rewriting the essential code.

The combination of all the above features to provide a robust mobile collaborative environment which can be applied in design and manufacture is a novel and challenging task. By having access to the relevant information from different resources, and having the tools for managing it, the major feature of this research will be to provide a well structured mobile collaborative environment that will however be a generic environment capable of extension to many problems faced by designers. The task in this project is to investigate and choose appropriate Internet and Web technologies to develop a collaborative environment for product design via a wireless network with the features of heterogeneity, scalability and interoperability.

1.3 Aim and Objectives of this Research

Aim:

- Establish a Wireless Internet-based Collaborative Working Environment for product design. This aim will be reached through a combination of multiple technologies, including Web Services Technology, Parametric Design, Semantic Web Technology, Agent Technology, Mobile Technology and Flex Technology.

Objectives:

- To review current relevant work and investigate enabling Web/Internet technologies and tools.
- To establish the infrastructure for a Mobile Web-based collaborative environment for product design.
- To increase reuse of existing resources by developing easier-to-use mechanisms for searching, managing and using information, knowledge resources and data, and to ensure that they will be accessible across wireless networks.
- To develop an approach to administrate remote CAD packages over the Wireless Internet by mobile devices.
- To develop an architecture to support collaborative work among multiple heterogeneous applications.
- Alongside the development of Web technology, find methods to successfully share and retrieve semantic product data in order to form a basis for seamless communication and thereby enable better integration of computing environments in collaborative product development.

1.4 Research Method

This research employs distributed computing technology to build a collaborative environment supporting wireless collaborative design, and an ontology-based (ontology technology will be described in chapter 2) description model is thus proposed to facilitate expression and organisation of product information in order to manage and deploy the distributed design resources. Integrated development environment, such as Flex builder 3 and Microsoft Visual Studio 2005, are employed to develop friendly graphical mobile client interface; Java programming tools are used to develop the Web server application, and Jena (described in chapter 5) to build Semantic Web applications, which provides a programmatic environment for RDF (Resource Description Framework), RDF Schema, OWL (Ontology Web Language), SPARQL and includes a rule-based inference engine.

- Web Service is used to implement design resource integration in a platform-independent manner. Existing design applications can be wrapped into distributed objects, irrespective of the language in which they are written, the platform on which they are running and on what hardware they are ported. Proper Web Service development tools need to be investigated in order to facilitate the necessary communications between Web Server applications and Mobile device applications, and between applications residing on different Mobile Operating Systems. Moreover, this Service Oriented mobile system is strong and reliable due to the design of the structure. The Service-Oriented Architecture (SOA), which is loosely coupled, makes faults much less likely to occur on the system in comparison with traditional systems.

- In addition, Semantic Web Technology is utilised to create a general knowledge base. This approach includes two components: (1) ontology is used to represent abstract views of product data and (2) added semantic rules are used to represent relationships between individuals. Therefore, to meet the demands of a mobile information system for collaborative product design, an ontology-based description model is thus proposed to facilitate expression and organisation of product data in order to manage and deploy the distributed design resources.

- Mobile agent system for collaborative design, which is based on Java technology, supporting the construction of mobile agents or mobile agent-oriented systems; an agent program can migrate or to make a copy (clone) itself across one or multiple network hosts.

- Flex applications, are small cross-platform programs that execute on the mobile client side of a Web connection and dynamically extend the functionality of a Web browser. Web Service is used as a mediation mechanism for the communication between a mobile client and an existing web application.

1.5 Outline of This Thesis

Chapter 2 Literature Review

This chapter will explain the benefits and drawbacks of current collaborative systems; additionally, current Web technologies, Semantic Web Technology, Agent Technology, Mobile Technology and Flex Technology relevant to collaborative design are introduced; leading to and the similarity between collaborative architectures and Web technologies are also discussed.

Chapter 3 Framework of Mobile Collaborative Working Environments

This chapter addresses the challenges of current research; describe the features of the Wireless Internet Based Collaborative Working Environment for product design and presents an overview of the combination of multiple technologies within MCWE to achieve the objectives of this research.

Chapter 4 Web Service Oriented Mobile Computer Aided Design (WSO - MCAD)

To meet the demand of online collaborative design and manufacture, a mobile collaborative work environment has been developed. The Web Service supported approach within the working environment for online CAD has been proposed, which enables the remote users to share and view their product design data.

Chapter 5 The Application of Semantic Web Technology into Mobile Collaborative Design

This chapter will present a mobile semantic product information retrieval system that supports collaborative work among remote users. With the development of the system, Semantic Web Technology has been introduced to provide semantic relationships and similarity of product data. The measure method of semantic similarity is proposed by taking advantage of the Is-a (ISA, described in chapter 5) relationship of concepts in ontology, which is shown by the experimental results that the method is efficient.

Chapter 6 The Mobile Agent for Collaborative Product Design

In order to make the remote users can share and retrieve information across different Web servers, this chapter introduce Agent technology, presents a multiagent-based system that adopts intelligent agents as a technology for tasks distribution and resource management in distributed systems.

Chapter 7 Mobile Platform Compatibility: The Flex Applications within MCWE

In order to resolve Mobile Platform Compatibility problem, this chapter is going to present the Flex mobile application, which could be implemented across different mobile platforms; and along with Web service, which could expose its applications to a set of loosely coupled services for the purpose of providing a fast solution.

Chapter 8 MCWE Applications for Product Design

In this chapter, case studies are going to be discussed; all the information utilized in the case studies is from the industry companies. The case studies are the experiment to prove this mobile collaborative design system is valid and applicable into theses factories in the future.

Chapter 9 Conclusions and Future Work

This chapter gives the conclusions from the research work with the descriptions of contribution to knowledge; further research works need to be improved for the mobile collaborative design system.

Chapter 2

Literature Review

Today mobile technology has been widely applied in various areas; however, little efforts have been made in utilising the technology for collabrative product design. This chapter presents a review of current collaborative systems, Agent Technology, Semantic Web Technology, Flex Technology and the latest Mobile Technology that are relevant to collaborative design. This literature review provides a clearly idea that, coupled with the latest mobile technologies and the Web technologies, mobile system could efficiently be used to support collaborative product design for remote users, and the last section highlights the necessity and innovative features, which will be incorporated into this research.

2.1 Introduction

Development teams in recent years involved in product design are often geographically distributed, requiring a real collaboration among the companies involved in the definition of a new product. Hiring low-cost talent was popular in developing widely dispersed design teams recently; for example, Christopher Tice, senior vice president and general manager of Cadence Systems Design's Verification Acceleration Group, said "We're seeing the establishment of very large design and development teams in the lower-cost areas, particularly by the larger multinationals” [14]. As a result of this new product development paradigm, there is a great need for software tools to effectively support the representation, collection and exchange of product information during the design and manufacturing process[1].

Computer Supported Co-operative Work (CSCW) studies how people collaborate with each other and the role that technology can fulfil to help this collaboration succeed. Since its origin in the 1980s, CSCW emerged as an interdisciplinary field that examines computer-assisted coordinated activities such as problem solving and communication carried out by a group of collaborating individuals [2]. Teams, not individuals, are the fundamental learning unit in modern organizations since teams can learn but the organization cannot learn. Computer networks have evolved to the extent that CSCW applications can be developed in many more research areas than ever before. These advances in communication technologies have also allowed the implementation of more complex CSCW tools, so today people can find a great amount of applications based on this idea.

In particular, industrial and design companies can benefit from research findings and methodologies from CSCW, in order to build and deploy successful design information systems based on collaboration. With the faster and higher demands of new and customized products, companies need to participate in global design chains and collaborate with each other and overseas partners to gain competitive advantages. Consequently, designers are increasingly faced with the challenges of integrating distributed multi-disciplinary product design and development teams made up of increasingly diverse sets of skills, varying design processes and different business measures. Product design, manufacturing and analysis have therefore a need for various levels of collaboration in a distributed environment. It is both technically and commercially imperative to develop new collaborative design tools or renovate traditional standalone CAD system by making it collaboration.

2.2 The Fundamental of Collaborative Product Design

Product design is the process by which the needs of the customer or the marketplace are transformed into a product which satisfies those needs. Five main specific criteria, all of which ultimately relate to profit, are commonly used to assess the performance of a product development effort: product quality, product cost, development time, development cost, and development capability [15]. Typically, the product development process consists of six basic steps, illustrated in Figure 2.1. All design starts with the identification of a need, ideas for new products or for ways, in which existing products may be improved, usually come from some combination of focus groups, customer feedback, market surveys, published studies, and company intuition.

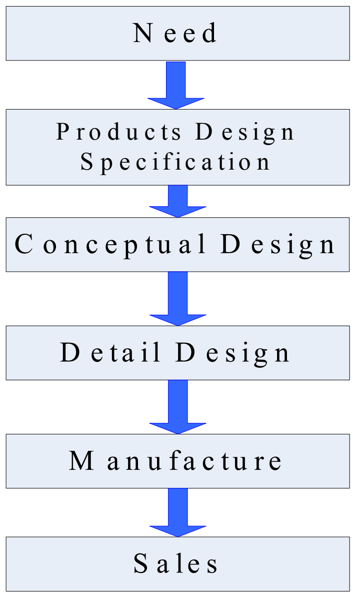


Figure 2.1 Steps of product development process

In the early design stages, being the most important technology in Product Design, CAD technology, which started the late 1950s, was developed to be a certain individual design tool. In the 60s, drawing a chart on the computer screen became feasible but needed to be developed, therefore people hoped to get rid of tediously aid of this technology. In the initial period of CAD software development, the CAD meaning was the chart board substitute merely, the complete content referred to Computer Aided Drawing (or Drafting) which were different from users frequently to discuss CAD contains today. The CAD technology continues to take the two-dimensional cartography as the essential target algorithm until the 1970s [16].

The design philosophy of existing CAD/CAM applications had been based on the computer environment developed in the 1970s and had been restricted to a single-location application. The development of CAD/CAM can be traced back to the Automatic Programmed Tools (APT) project started in the 1950s at the Massachusetts Institute of Technology (MIT) [17]. APT was used to represent the work piece geometry for numerically controlled (NC) machines in achieving high-precision operations.

In the 1980s, the manufacturing industry went through a fundamental change; the product supplies exceeded the market demand. Product quality, cost and request became more and more high, the product life cycle became shorter and shorter. Therefore in order to win the market, an enterprise must accelerate new product development, improve the quality, reduce the cost and provide the high-quality service. However, in these questions, rapidly developing the new product and entering the market as soon as possible became the most import thing for these companies [18, 19]. Therefore, by begun needed extensive research in the mentioned field, the research achievement was, "the parallel project". For example, in 1982, the American national defense senior research project bureau (Defence Advanced Research Projects Agency: DARPA) started to study the CSCW environment provided the ideal environment for real-time discussion of design ideas and the sharing of design data and test results. The virtual integrated product design and development (IPDD) system aimed to provide a comprehensive set of internet-working tools for efficient multigroup collaboration using the Web/Internet as an information-sharing platform [17].

Based on the previous research in IPDD system, Gadh [18] presented a new geometric representation called the Assembly representation and a prototypical system Collaborative product assembly design (CPAD), which implemented the collaborative paradigm and provided a networked environment for product assembly design that integrated collaboration, solid modelling and features based design. The system was based on the hierarchy of the top-down design process and therefore provided a mechanism to overcome specification conflicts early in the design stage and enabled concurrent design of the product.

Early development and achievements in computer supported concurrent engineering has been described above. After 1980s, recent surveys by Huang [20] indicated that the Web/Internet technology was playing a key role in the development of the Collaborative Product Design (CPD) system [21, 22]. Web and Computer technology had also been used into the CPD system, such as mobile agent, knowledge-based system and genetic algorithm [23-27]. These enabling technologies became the key feature and force in the development of distributed system.

2.3 Web-Based Collaborative Design

Along with the development of Web/Internet Technology, client/server architecture was based upon the first generation of Web technology, such as CGI, Servlet, etc. Monplaisir [28] described an approach to apply CSCW technology to the product design and development process. For the application of the system, IPDD system was observed as a distributed and iterative process that involved significant sharing of information early in the design processes had integrated applications and databases at different locations through the Web technologies, these were primarily closed methodology, i.e. only selected users were allowed to access and modify these Web-based distributed systems. To obtain the full benefits of object-to-object interactions, the collaborative design integrated with the second generation Web/Internet Technology, such as Distributed Component Object Model (DCOM), Remote Method Invocation (RMI), the Common Object Request Broker Architecture (CORBA), many new architecture of concerning CPD had been introduced.

In DCOM, Which is a Microsoft technology for communication among software components distributed across networked computers, the end users could develop new applications using the existing services provided by this technology [15].

In addition, Li [29] described a module-based multi-agent system, which was developed and deployed according to the CORBA standards to facilitate the coordination among designers within the range of enterprises and even virtual enterprises; Rosenman [30] also developed the framework based on CORBA in the field of mechanical design.

As a new import approach in software development, object-oriented analysis and programming had also been an important methodology. Though the Object Oriented approach, many authors had made a major revolution from traditional software development by which the promise of large scales code reuse did not become reality.

Due to the limited bandwidth of the Internet transfer of detailed CAD based assembly models was slow. In order to deliver and manipulate interactive 3D objects effectively in the Web, some concise formats, such as VRML, X3D (ISO standard XML-based file format for representing 3D computer graphics) and MPEG-4 (a patented collection of methods defining compression of audio and visual (AV) digital data), had been launched to represent the geometry of 3D CAD models as visualization-used triangular meshes and trimming lines [31, 32].

Fuh and Li [33] had performed some important works in Web-based visualization and 3Dconcise representations, 3D streaming technology and co-design systems and feature-/assembly-based representation elaborated. Kan [34] described an Internet-based virtual reality collaborative environment called virtual reality-based collaborative environment (VRCE) that was developed by VNet, free software, Java and Virtual Reality Modelling Language (VRML) that demonstrated the feasibility of collaborative design for small to medium size companies that focused on a narrow range of low cost products. The networking of VRCE is programmed in Java because of its powerful networking capabilities. The communication between client and server over networks in VRCE is based on VRML Interchange Protocol running over a TCP/IP. The VRCE server was built based on the VNet server, with modifications. It is a multi-threaded server that communicates with the client over networks based on VIP. The VRCE server listens for incoming connections and sets up a communication socket. It creates a new thread for each successful connection it receives. Inside the VRCE server, there are five engines that are responsible for different processes that facilitate virtual collaboration. These engines include the Graphical Synchronization Engine, Message Transferring Engine, Log Engine, Security Servicing Engine and Knowledge Engine. Eng [35] had also proposed a Collaborative CAD system, which extended an existing single-location CAD system to a multi-location CAD application so that two or more geographically dispersed users can work together on a 3D-CAD geometry co-editing and CAD-related tasks dynamically and collaboratively, the system was developed on Java platform for 3D-CAD data transmission.

With the latest development of Web/Internet technology, Su [36] developed a Web-enabled Collaborative Environment (WECE) which was a service oriented architecture (SOA) system based on Web services technologies, service oriented technology was known as third generation of Web technology. Xiong [37] also described Service Oriented Software Package Bank, provided sharing method, the term “service” referred to an abstract logic view of the real application defined in terms of what it did, typically carrying out the business logic of an application. Web had given a good interaction platform between human and computer programs, while Web Service was giving a better environment for interaction among program themselves. That’s why Web Service was chosen to be the intermediate technology of the CAD collaboration system [3, 6, 8 and 38].

Web service is an emerging Web technique, which has been attracting researchers’ attention, and it has shown great advantages for application in collaborative product design and manufacture.

- Web services allow the re-use of services and components within an infrastructure: Relying on enterprises or organizing existing technologies, Web Services can turn enterprise existing digital resources, packages, application systems into services. After that, the interfaces are described in WSDL (Web Services Description Language) files, which help the client to understand what the services are and how to access them. In this way, all the resources become the services that can be accessed via the Internet [39].

- Web services provide interoperability between various software applications running on disparate platforms: Web services have dynamic characters. Any authorized user could join the services without affect on other existing users. 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 are not originally developed [8, 40].

- Web services use open standards and protocols. Protocols and data formats are text-based where possible, making it easy for developers to comprehend [41].

- By utilizing HTTP (Hypertext Transfer Protocol), Web services can work through many common firewall security measures without requiring changes to the firewall filtering rules [39].

Despite the architecture of collaborative design mentioned, other enabling technologies have also been involved, such as Agent technology, Mobile Technology for Web Collaboration, and Semantic Web Technology Utilized for Collaborative

Design, etc.

2.4 Agent Technology

People use various processes to develop physical solutions to specific needs in Product Design. Products may make use of electronics but are not obliged to. They may be mass produced, custom-made, or customized. The interest of implementing mobile agent systems in mobile devices has increased during the recent years. Using the agent-based approach, a multi-agent-based system is constructed to integrate activities into customer product development process within a distributed, collaborative and concurrent environment [23].

Huang [20] introduced the agent concept for workflow management as a mechanism to facilitate the teamwork in a collaborative product development environment where remote Web-based Decision Support Systems were extensively used by team members who were geographically distributed. Wang [42] Provided a powerful solution for project managers and designers working on multiple design projects to share product information and knowledge which presented an agent-based distributed environment implemented with a number of emerging technologies including software agents, Web/Internet, Java, and XML, which was used for data management within the server as well as for message exchanges among software agents.

In the mobile world, people are paying more attention to agent technology today. The Mobile Agent was a more advanced model compared with traditionally distributed computing because it used the object-oriented technique to enable the Agent to have all characteristics belonging to a class [43]. Java, strongly supporting the programming of network and multi-threads, was a programming language independent of Platform and more suitable for developing the Mobile Agent. It built a new way for the Mobile Agent's development and explained the key problems related to the scalability and security of the system, which could provide new ways for the Mobile Agent's implementation [44].

Li [43] designed a kind of Java-based testing platform, which was designed and implemented to provide the core capabilities of mobile agent platforms, enabling mobile agents to be loaded from a class file, to migrate from one host to another, run-in a host, and offer reliable communication services. Grady and Hare [45] also described the design and implementation of Gulliver's Genie, an archetypal ubiquitous computing application which commissions a collaborative embedded agents approach. In the mobile world, Lee [46] presented a multi-agent framework which considered different contexts to support personalized services on wireless networks. In the proposed approach, client users, content providers, and service providers were all considered as software agents. Different application services were developed accordingly on a publicly available middleware platform, and the preliminary results showed that the multi-agent approach to personalization was promising, and efficient in the deployment of mobile services.

In the field of network transmission, there was also some important improvement. Peng [47] introduced a mobility management schema based on Windows CE.net (Windows OS for mobile device), in its research; embedded OS had been proposed to solve the drawbacks of mobile IP protocol. Mahmoud [48] also proposed an agent-based approach helping the end user to reduce information overload by the method to filter out unnecessary information and only showed relevant information; it provided novel ways for cellular service; a proof-of-concept prototype using Java RMI had been constructed as part of the Havanna mobile agent platform.

Agent technology has been recognized as a promising paradigm for next generation manufacturing systems [49], in order to enhance the system's capacity, the agent technique was utilised in the design of this mobile collaborative working environment.

2.5 Mobile Technology for Web Collaboration

The progress of mobile technologies on wireless network has influenced every aspect of human life, create new living, new working style, new strategy for country, and new economics [50]; the mobile Internet has unique advantages over the stationary Internet, due to the characteristic that users can connect to the mobile Internet wherever and whenever they want. For example, in Japan, the mobile Internet provider DoKoMo had launched its i-mode service as the first commercial service allowing broadband steaming of rich data into mobile phones [46]. In the product design field, the more and more latest mobile devices were introduced into the market, as shown in Figure 2.2, people could not only shared their design by the eDrawing file (Saved as Jpg in the Web Server) via iPhone (a), they also could view and modify the CAD DWG file in (b) [8], which installed the PowerCAD CE (CeCAD) for Pocket PC application as well as in (c), the mobile CAD application is ShortCAD.

Figure 2.2: View product design data in different Mobile Devices

Recent trends towards the convergence of wireless communications and Internet-based technologies have the potential to open new avenues of mobile collaboration, thereby minimising the impact of the physical dispersion of team members. Recently, a wide range of portable wireless devices have emerged based on personal area networking (e.g. Bluetooth), Local Area Networking (e.g. IEEE 802.11) and Wireless Wide Area Networking (e.g. GSM, GPRS, 3G). And advances in other Internet-based technologies such as the Semantic Web, Web Services, Agent Technologies, coupled with the improved wireless bandwidth and the ability to better capture context information, which can efficiently be leading to a new mobile collaborative design system.

But the physical constraints of mobile devices, such as screen size, display resolution, limited computational power and memory, can significantly affect the usability of mobile applications linked with a traditional database, if the mobile user wants to retrieve some results via a wireless network [46]. Concordant with the limitations of mobile networks, application services have to be re-designed in order to satisfy the needs of mobile users. Goren-Bar has conducted a study that service adaptation is strongly required for mobile devices [51]. For example, for the most popular Internet activity is the Web search; the growing number of wireless Web users seldom browses through many pages, but rather are interested in specific destinations. They would like to reach the page they require by following as few links and visiting fewer pages as possible. This is an important feature of mobile users' behaviour because they have to scroll through pages on small-screened mobile devices, on wireless networks with low bandwidth and high latency. Hence, the main task of wireless Internet is to allow mobile users to perform focused searches and satisfy their specific and immediate information needs. It is thus crucial to develop the methods to enable the mobile system could provide more related answers rather than simply send data back to the remote users. In order to satisfy that requirement, Semantic Web Technology will play a key role in this mobile information system.

2.6 Semantic Web Technology Used for Collaborative Design

The Semantic Web initiative of the World-Wide Web Consortium (W3C) has been active for a few years and it has attracted interest and scepticism in equal measure, a more flexible, integrated, automatic and self-adapting Web, providing a richer and more interactive experience for users [52]. The Semantic Web combines the descriptive languages RDF (Resource Description Framework) and OWL (Web Onto logy Language), with the data-centric, customizable XML (eXtensible Mark-up Language) to provide descriptions of the content of Web documents. These machine-interpretable descriptions allow more intelligent software systems to be written, automating the analysis and exploitation of web-based information. Figure 2.3 gives a typical representation of the diagram, which is first proposed by Tim Berners-Lee [53].

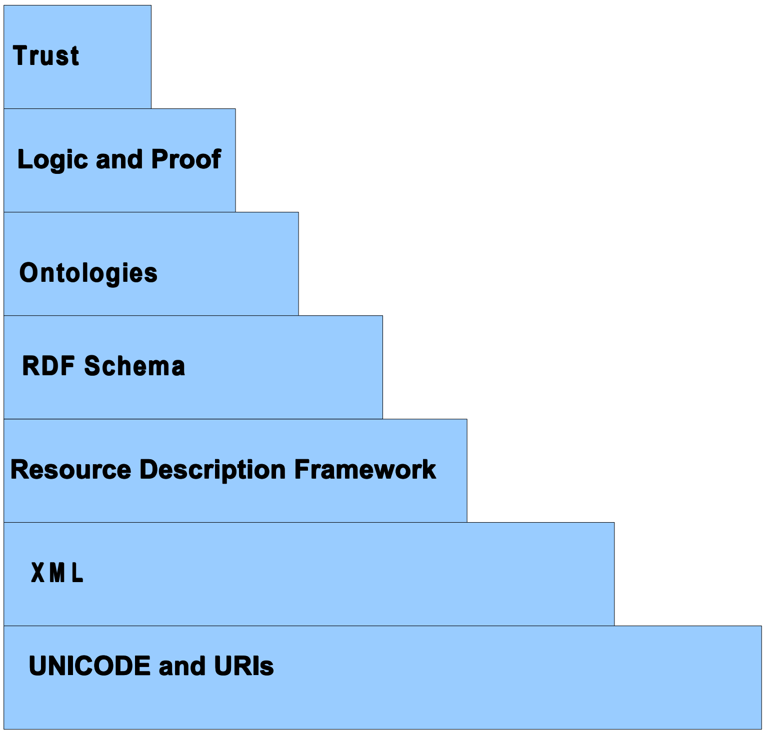


Figure 2.3: Semantic Web layered architecture

- **Unicode and URI (Uniform Resource Identifier):** Unicode is a computing industry standard what allow computers to identify and locate resources (such as pages on the Web), provide a baseline for representing characters, and identify resources. URI consists of a string of characters used to identify or name a resource on the Internet.

- **XML:** XML is a W3C Recommendation, which is a markup language much like HTML (Hypertext Markup Language), XML was designed to carry data, not to display data, XML tags are not predefined, and XML is designed to be self-descriptive.

- **Resource Description Framework:** RDF is the first layer of the Semantic Web proper. RDF is about metadata for Web resources, using URIs to identify Web-based resources and a graph model for describing relationships between resources. The RDF data model represents the properties of a resource and the values of these properties. The model is syntax independent but can be expressed in XML, and the specification uses XML as its syntax for encoding metadata.

- **RDF Schema:** It enables a system to describe basic features of an RDF as well as other RDF vocabularies. It is a simple type modelling language for describing classes of resources and properties between them in the basic RDF model.

- **Ontology:** A richer language for providing more complex constraints on the types of resources and their properties. In this research, applying Ontology Web Language (OWL) into the mobile system, to make the semantic data could be accessed via the Internet. Ontologies are used to capture knowledge about some domain of interest. An ontology describes the concepts in the domain and also the relationships that hold between those concepts. Different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium. OWL makes it possible to describe concepts. Complex concepts can therefore be built up in definitions out of simpler concepts. Furthermore, the logical model allows the use of a reasoner which can check whether or not all of the statements and definitions in the ontology are mutually consistent and can also recognise which concepts fit under which definitions. The reasoner can therefore help to maintain the hierarchy correctly. This is particularly useful when dealing with cases where classes can have more than one parent. OWL ontologies may be categorised into three species or sub-languages: OWL-Lite, OWL-DL and OWLFull. Using Protégé 4 (ontology edit and knowledge acquiring tool) to build the mobile system's ontology will be described in Chapter 5.

- **Logic and Proof:** an (automatic) reasoning system provided on top of the ontology structure to make new inferences. Thus, using this system, a software agent can make deductions as to whether a particular resource satisfies its requirements.

- **Trust:** The final layer of the stack addresses issues of trust that the Semantic Web can support.

Semantic Web Technology performs major implications in the development of new information management system recently [54-56]; people were taking a variety of approaches to develop tools to extend the current Web into a true Semantic Web, for example, Protégé by Stanford Medical Informatics, Ontolingua by Knowledge Systems Laboratory of Stanford University, and OntoSaurus by University of South California. These tools consider the construction process as system engineering, and focus on coding the concepts into the ontology, which computers can understand. These tools typically take an existing Web component users are familiar with, such as browsers, servers and search engines, and augment them with the power to process the semantic annotations associated with Web pages below [57-59].

Semantic Web Browsers, for example, extend the notion of the Web browser into the Semantic Web by allowing the RDF annotations of resources to be read and presented in a structured manner. A semantic Web server, such as Joseki [60] from HP Labs, was an HTTP engine that supported the SPARQL Protocol and the SPARQ L RDF Query language. SPARQL was developed by the W3C RDF Data Access Working Group. Users could perform some Online Demo to get the experimenting with SPARQL queries. The two Features of Joseki were: RDF Data from files and databases; HTTP (GET and POST) implementation of the SPARQL protocol

The most widely used tools on the Web are search engines, with Google being the most popular. A Semantic Web search engine, such as Swoogle, was a crawler-based indexing and retrieval system for the Semantic Web. It extracts metadata for each discovered document, and computes relations between documents. Swoogle can use ontologies to refine the search, and had harvested the existing ontologies and RDF data available on the Web. Swoogle provided services to human users through a browser interface and to software agents via web services. Several techniques were used to rank query results inspired by the Page Rank algorithm developed at Google but adapted to the semantics and use patterns found in semantic web documents.

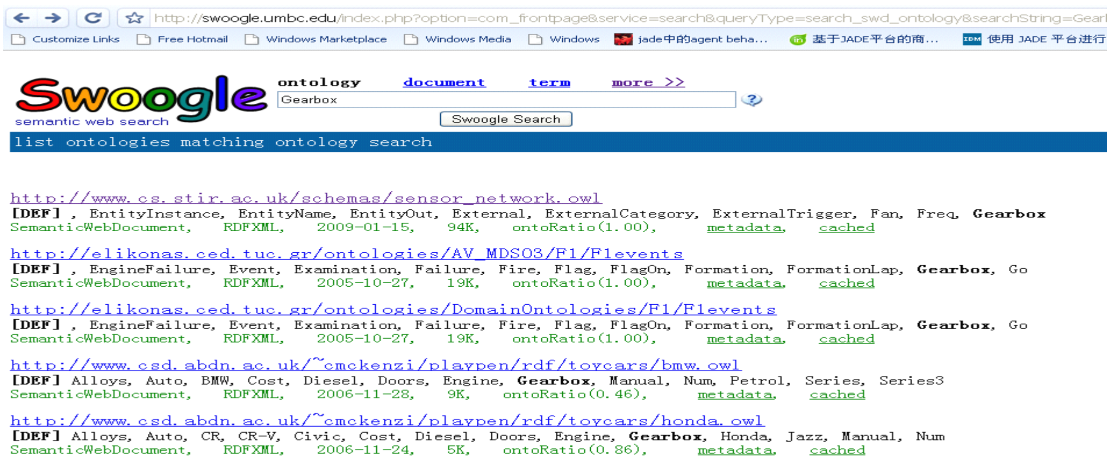


Figure 2.4: A screenshot of the SWoogle

And a practical approach at the mentioned field was the provision of 'Semantic Portals', including SEAL [61], Ontoweaver [62] and SWED [63], for example, SEAL realized semantic-based search and access facilities to information represented by semantic portals. Such portals typically provided knowledge about a specific domain and rely on ontologies to structure and exchange this knowledge. All the above information management tools provided a prototype of the basic infrastructure which will underlie the Web. They provided the user with an enhanced information management capacity, with the means to organise and structure the chaotic information on the Web. By providing the annotation in a machine readable format, user agent software can access and process that information automatically.

The main research issues applied the Semantic Web Technology into Industry Information Management currently is struggling to implement techniques that can reduce project duration and cost, whilst improving productivity and performance [55]. For example, LI [10] presented an ontology-based description model of part information using the three-layer framework, which provided an expression and organisation of part information in a computer aided process planning (CAPP) application system in order to make the part information database reusable and effective. To support the representation, integration and sharing of product knowledge, Hu [57] constructed a product knowledge representation model and two related nomenclatures of ontology - namely flow and function - which were developed on the basis of product knowledge representation techniques. Along with the development of Web technology, methods to successfully deploy, share and retrieve semantic product data will form the basis of this seamless communication and thereby enabled better integration of computing environments in collaborative product development.

Several efforts were also devoted to the development of techniques and approaches for the integration of heterogeneous data sources to provide global views on data provided from distinct organisations within a distributed environment, using ontologies to semantically organise the integrated knowledge about a particular domain. There were a number of difficulties in the process of ontology building, such as unclear building methods, non-unified ontology languages and the lack of tools. To solve these problems, Yang [59] presented an ontology building method using Jena. The method was composed of 5 parts: class description, property description, links between property and class, individual creation and ontology metadata adding.

The Semantic Web was an extension of the current web in which information was given a well-defined meaning, which facilitated enable computers and people to work in harmony [56]. It allowed the data to be defined and linked in the way that it can be used by machines not just for displayed purposes, but also for automation, integration and data re-use. The Semantic Web technologies provided intelligent access to heterogeneous distributed information, enabling software applications to mediate between user needs and information sources [54-56].

Semantic Web Technology enhances the capabilities of those tools which form a familiar part of the current Web; so that they can become useful information management tools in their own right. The Web is already an information source of choice for many learners and researchers. A more structured and directed approach to manage this information space, both within institutions and across the whole community, can make this information more useful, with less wasted effort, and more capacity to measure the quality of information. By making the annotation machine readable, it becomes accessible to automatic processing, carrying out many routine tasks which consume people’s time.

2.7 Mobile Platform Compatibility: Flex Mobile Applications

Mobile devices, utilized as the design tools within MCWE, their Operating System (OS) could be found in Table 2.1, as well as the market share Sales for their mobile devices [64].

Mobile Operation System	Develop organization	Market Share Sales
Symbian OS	Symbian Ltd	46.6%
Windows Mobile	Microsoft	13.6%
Linux	Open source	5.1%
iPhone OS from	Apple Inc	17.3%
BlackBerry OS from	RIM	15.2%
Android	Open Handset Alliance and licensed by Google	

Table 2.1 Sales market coverage of different mobile OS

2.7.1 Comparison of Three Mobile Development Technologies

Nowadays, there are three popular mobile development technologies, such as Flex/AIR, Microsoft .Net/Sliverlight and Java/Java Me, used to the mobile application implemented in different mobile devices. All the three technologies have very good and fast compilers, comprehensive and very powerful libraries and relatively fast execution speed. The comparisons of the technologies are shown below:

- Flex is a good software development platform especially for browser applets, if confronted with Microsoft .NET and Java. Adobe Flex is a software development kit released by Adobe Systems for the development and deployment of cross platform Rich Internet Applications based on the Adobe Flash platform. Flex applications can be written using Adobe Flex Builder or by using the freely available Flex compiler from Adobe company.

- Microsoft Silverlight is a Web application framework with a scope similar to Ad obe Flash, but its application could not be cross platform. There was a free software implementation named Moonlight were developed by Novell in cooperation with Microsoft, was available to bring compatible functionality to Li nux [65], but Microsoft Sliverlight application was still weak at the ability of platform compatibility.

- Java Platform, Micro Edition (Java ME) is the most ubiquitous application platform for mobile devices. Java technology can be used to create both client- and server-side applications. While the server-side implementation is widely accepted, the browser-based client-side applications have been less successful because the file sizes are large and the installation is cumbersome for mobile devices, especially for Windows Mobile PDA. And for Java Applets, which are generally mentioned when Java is used on the client side, that can run on a browser with Java support. But only the latest Mobile Brower are installed Java environment by default, for example, Mobile opera or Skyfire browser, most mobile devices need to install the mobile Java environment, and the deployment will cost more time for the mobile users [66,67].

2.7.2 Descriptions of Flex Technology

In February 2008, Adobe released the Flex 3 SDK under the open source Mozilla Public License. Adobe Flex Builder was used to build Flex applications, and Adobe Flash Player, was the runtime environment for Flex applications. Flex is a highly productive, free open source framework for building and maintaining expressive web applications that deploy consistently on all major browsers, desktops, and operating systems. A Flex application at its very basic level is Flash SWF file embedded in a generated HTML file. Since Flex outputs its data as a SWF file, this allows application to utilize for dynamic animations, sound and video handling and the Flash Drawing API.

MXML, an XML-based markup language, is the language developers use to define the layout, appearance, and behaviours of a Flex application. MXML and ActionScript are compiled together into a single SWF file that makes up mobile Flex application. One of the major reasons why Flex is becoming more and more popular is because it utilizes ActionScript 3.0, which goes beyond the scripting capabilities of previous versions of ActionScript. It is designed to facilitate the creation of highly complex applications with large data sets and object-oriented, reusable code bases. ActionScript 3.0 is a powerful, object-oriented programming language that signifies an important step in the evolution of the capabilities of the Flash Player runtime. The motivation driving ActionScript 3.0 is to create a language ideally suited for rapidly building rich Internet applications (RIA), which have become an essential part of the Web experience.

2.7.3 Advantages of Flex Mobile Applications

Many web designers and developers use Adobe Flash or Adobe Flex, which are part of the Adobe Flash Platform, to build RIAs. Flex is a cross-platform development framework for creating RIAs. Unlike page-based HTML applications, Flex applications provide a stateful client where significant changes to the view don't require loading a new page. Similarly, Flex and Flash Player provide many useful ways to send and load data to and from server-side components without requiring the client to reload the view.

Table 2.2 shows the comparison of three mobile development technologies, Flex is a good choice for mobile platform in author’s developing experience. Microsoft .Net/Silverlight is weak at cross-platform considering of current different platforms shown in Table 2.1. Java Me, it is weak at browser applets (Mobile), that means the mobile devices need to install the Java mobile application and JVM (Java Virtual Machine), but due to the limited size and lower speed processor of mobile devices, some mobile users prefer to download the client application when they are connecting with the Web server; the deployment of Flex mobile applications are much easier than Java Me application.

Technology	Platforms	Browser applets (Mobile)
Flex/AIR	Win/Mac/Linux	almost 100%
NET/Silverlight	Win, ~Mac	weak
Java/Java Me	Win/Mac/Linux	weak

Table 2.2 Comparison of three mobile technologies

Flex builder provides MXML, XML-based user interface markup language, for designing the layout of the mobile application interfaces [70]. In chapter 7, this thesis will describe Flex application combined with middleware technology, such as Web service, to exchange messages between SOAP (Simple Object Access Protocol) service endpoints (SOAP messaging in Flex is constrained to only use HTTP as the transport mechanism).

2.8 Conclusions

A framework for enabling collaborative designs should allow designers to access their favourite tools from a hypermedia workspace. Today's Web technology supports coordination through provision of a shared information space. However, to fully participate in a collaborative design, designers need to be able to not only exchange data [68, 69], but also to negotiate their design intentions regarding the design process. This negotiation requires the tools can be implemented on top of the Web infrastructure using the latest middleware technologies (such as Agent and Web Service). In addition, client side scripting, Flex Actionscript, often make significant contributions to the execution of design applications or tools. Furthermore, Semantic Web techniques, Object Linking and Embedding Automation (OLE) for Parametric Design are becoming enabling technologies in collaborative design, in an attempt to improve efficiency and intelligence in the design process.

From the literature review, it can be seen that the latest Web/Internet technologies could be used to establish an open architecture based on the existing Web infrastructure for communication, facilitating collaborative design activities or participating in a collaborative session via a Mobile Device. Most of the proposed systems are still in the proof-of-the-concept prototype development stage. It is clear that challenges in these areas will continue to provide rich opportunities for research; mobile collaborative environments based on the Internet are needed to obtain a more flexible integrated design environment with the following features:

- **Mobile design environment based on the Wireless Internet:** The mobile devices are used as tools for the users to communicate and collaborate with each other through the Web Server; this produces data for the product including CAD drawings, materials, manufacturing process, and supplier chain data which are effectively shared among the users.

- **Mobile Semantic information system:** This mobile system is used to support the exchanging, integration and sharing of product knowledge data over the wireless internet amongst remote users.

- **Multiple working models support and Mobile Platform Compatibility:** Over the Internet, Web-based design systems should facilitate multiple working models support, such as singular large scale computing program invocation, design resource sharing among multiple designers. Despite of the multiple working models, currently there also exist a number of different Mobile OS Devices, research need to present the last Mobile Technology, thus the new mobile application could be supported by the most popular mobile devices, such as HP Windows PDA, Nokia Linux PDA, Nokia Smart Phone, etc.

The combination of all the aforementioned features provides a robust tool for a Web-based collaborative environment, which is a novel and challenging task. Along with the emergence of high speed wireless networks, such as Wi-Fi, Bluetooth and 3G, and parallel advancements in Internet based technologies such as the Semantic Web, Web Services and Agent technologies, the tasks in this project aim to develop such a intelligent wireless collaborative environment with the above-mentioned features for product design has become a possibility.

Chapter 3

The Framework of Mobile Collaborative Working Environment

This chapter addresses the challenges of current research; describes the features of the Wireless Internet-based Collaborative Working Environment for product design and an overview of the combination of multiple technologies within MCWE used to achieve the objectives of the research.

3.1 Introduction

The widespread deployment and use of wireless data communication is generally recognised as being the next major advance in the information technology industry but it is not yet well supported by existing programming models and middleware architectures. Current research within Mobile Collaborative Working Environment is addressing topics such as the design of appropriate programming models for developing mobile applications, the transparent extension of existing industry-standard middleware architectures that take account of the distinctive characteristics of mobile collaborative environments for product design.

The structure of the framework is shown in Figure 3.1. The middleware interacts with the applications and resources, to provide services for collaboration. A distributed approach is proposed which combines various middleware techniques including Web services, Mobile agents to enable the following features:

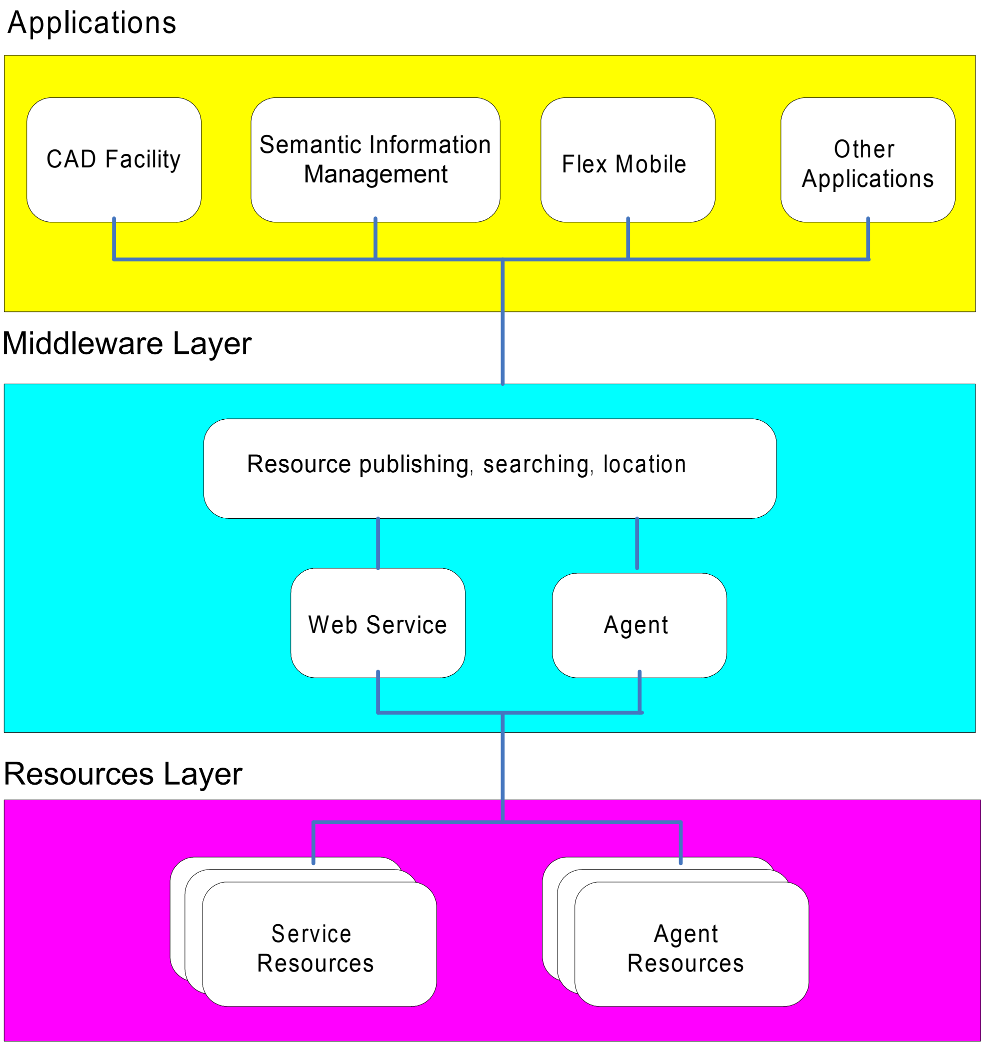


Figure 3.1 Structure of the MCWE framework

- **Seamless collaboration:** it supports resource sharing in dynamic and collaborative working environments and integration of distributed systems. It allows sharing control of the contact, facilitates cooperation among clients in a network; enables a distributed searching network, facilitates contact grouping, and permits the 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.

- **Mobility:** It provides a service with the ability to transfer between different hosts. Two types of mobility are available: remote execution and migration. For the remote execution, a service is transferred to a remote site and executes until task completion. For the migration, a migrating agent utilizes a number of hosts for the completion of the task, that is, the service may suspend its execution, move to a different host, or resume execution.

- **Security:** Information services provide information about available resources and status of resources, while security services provide single sign-on, authentication 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 within Mobile Collaborative Environment for Product Design [8]. This research provides the following three layer architecture of the mobile environment in Figure 3.1.

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, two kinds of resources are supported by default: Web Services and Agent resources. These resources independently exist in the system.

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 that provide a common programming abstraction across a distributed system and connect 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 two main enabling middleware techniques, such as Web services and Agent technology. 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 control the transmission of all feature model resources in the system.

The framework is built on top of service component-connection modules where new functionality can be simply inserted and used. It supports the re-use, 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 plugged into the supporting facilities to enable them work well within the collaborative working environment.

The applications:

The middleware provides facilities for applications to plug into this layer and so be loosely coupled. They will be adapted or produced to follow the standards specified in this layer. "Customizable" plug-in functions are utilized in the system to provide a flexible means to add functions into the system in the future to fulfil tasks for other collaborative work areas. Applications necessary to the design and manufacture are to be implemented by all of the partners using their own system, which will ensure the use of heterogeneous systems. More and more service could be added to the system no matter what collaborative work area 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.

The framework has been applied in the development of a mobile collaborative work environment for product design [3]. The total process of collaborative product design has been considered in Figure 3.2, including the following aspects:

- View, Modify and Share CAD files via mobile device;
- Parametric Design utilization on mobile applications;
- Exchange product data in different formats within a distributed enterprise and its various partners;
- Mobile platform compatibility; remote execution of a large-size computing program;

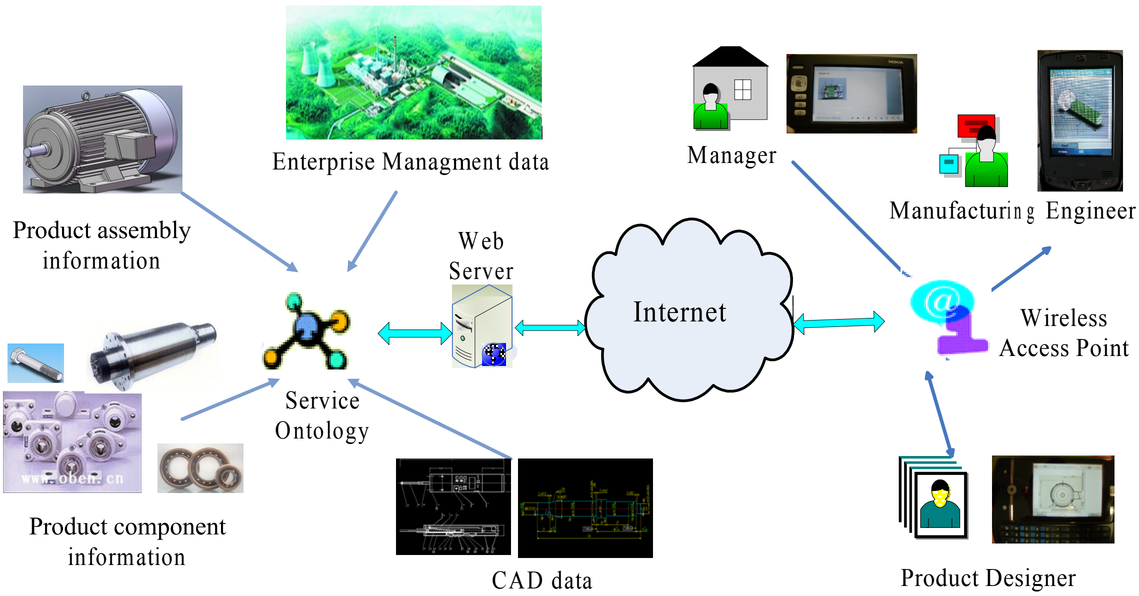


Figure 3.2 MCWE: illustration of the framework

3.2 Service Oriented MCWE

3.2.1 Introduction of Web Service Technology

Web Service is an emerging distributed middleware technology, regarded as a competing middleware technology to CORBA, which uses simple XML-based protocols and standards to allow applications or systems to exchange data across the Web [71]. Web service platform needs a set of agreements to realize the establishment of the distributed application program. The Web service platform must offer a standard for different kinds of systems communicating in different platforms, programming language and package model. In the traditional distributed system, the platform based on interface (annotation of translation: IDL (interface description language) in COM and COBAR). Similarly, Web service platform must offer one standard to let customer can receive enough message via the web.

The core components of Web Services standards are WSDL, SOAP (Simple Object Access Protocol), and UDDI (Universal Description, Discovery and Integration), which are all based on XML (eXtensible Markup Language). The component architecture for Web services is illustrated in Figure 3.3.

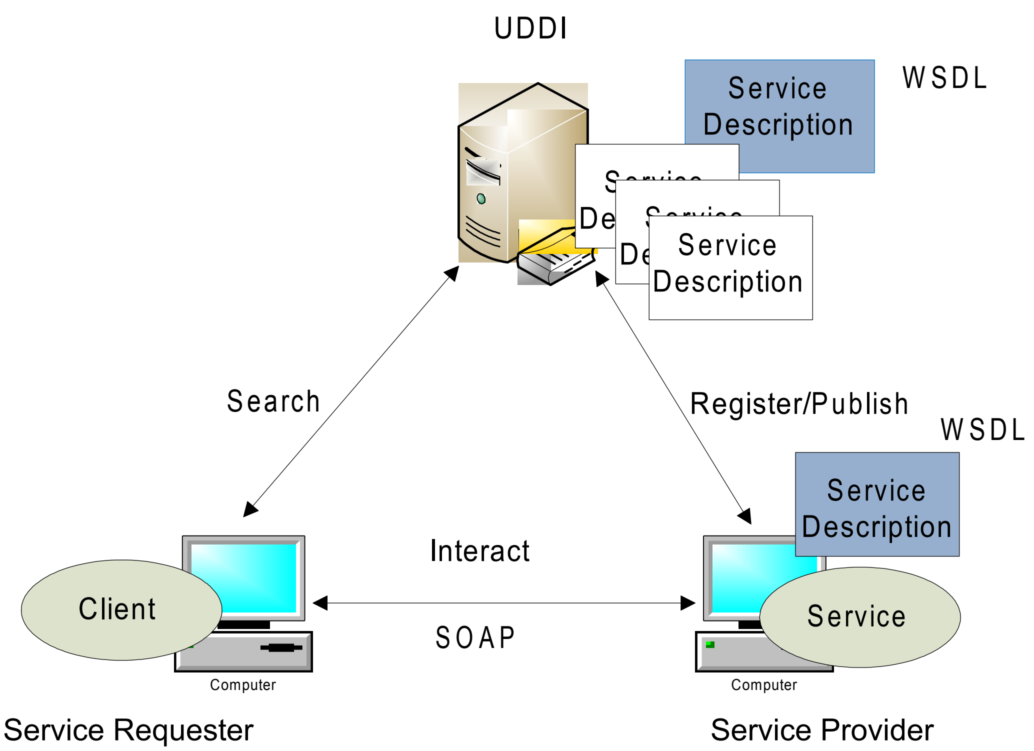


Figure 3.3 Component architecture for Web Service

XML is the basic form showing the data in Web service platform. XML has solved the problem that the data are expressed, but it has not defined a set of standard datum types, have not said how to expand the type of this set of data. The specifications (i.e., interface) of services can be described using WSDL, a metadata language that defines how service providers and service requesters communicate with Web Service applications. Because it is based on XML, WSDL can be read by computer as well as the people, which will be an important advantage.

SOAP is a communication protocol for exchanging information in a decentralised, distributed environment. It defines a mechanism to pass commands and parameters between clients and services. SOAP is independent of the platform, object model, and programming language being used. In addition to SOAP, there are other transport protocols such as JAX-RPC and XML-RPC.

A UDDI server provides a “meeting place” for Web Services; an information database for Web Services, which stores descriptions about service owners and the services they offer in a common XML format. Just as businesses list their products and services in a telephone directory, UDDI is used for providers to register and publish their services, so that requesters can then discover and invoke them.

3.2.2 Comparison of Web Service and Other Distributed Technologies

Some distribution tools have certain functions similar to Web service such as: DCOM, RMI, and CORBA, but,

- DCOM is platform dependent (Windows) and negating the goal of standards-based interoperability.
- RMI is Java technology-based, and thus does not easily play well with other languages.
- CORBA is limited however by its complicated, ad-hoc way of utilizing the power and flexibility of the Internet. It is difficult to develop and deploy CORBA for small projects, which may not justify its performance benefits because it requires considerable code development.

Compared with the above technologies, Web Service has the following characteristics which have been utilized in the mobile system：

- Web Service allows the reuse of services and components within an infrastructure: Relying on enterprises existing technologies, Web Service can turn enterprise existing digital resources, packages, application systems into services. After that, their interfaces are described in WSDL files, which help the client to understand what the services are and how to access them. In this way, all resources become services that can be accessed via the Internet.

- Web Service provides interoperability between various software applications running on disparate platforms: Web Service has dynamic characters. Any authorised user may join the service without affecting existing users. The structure of the service-oriented system is flexible enough to meet the constantly changing demands to update and append required additional functions. It can also integrate existing software (programs and packages) and avoids being required to perform the tedious task of re-writing the software into a format in which they were not originally developed.

- Web service uses open standards and protocols. Protocols and data formats are text-based where possible, making it easy for developers to comprehend them. By utilising HTTP, Web Service can work through many common firewall security measures without requiring changes to the firewall filtering rules.

The Web Service has given a good interaction platform between human and computer programs, while providing a better environment for interaction among the programs themselves. This is the reason why Web Service has been chosen to be the middleware technology of MCWE.

3.2.3 View, Modify and Share CAD files via Mobile Device

The Web-based design architecture developed in this research is mainly based on the concept of Web-server-centralised system approach. There exists at least one Web server, i.e., the collaborating server, which acts as a central unit and manages the interaction between users on the one side and application providers (e.g. servers) on the other. Design services such as computing, and analysis tools are provided by a number of object servers, which form the functional basis for the distributed network design system. Client procedures allow users to access available design applications and thus utilise the tool's functionality. In this architecture, Web Service plays a key role in connecting an individual site, where the CAD package is located and related services are available to the Internet. The mobile application carries out the tasks of dynamic data processing and links with Mobile CAD package and Web Service. The designer operates the CAD package to produce drawings and makes changes which are instantaneously sent, with the aid of the mobile application and Web Service, to the collaborative designers located in different sites. As shown in Figure 3.3, in order to load a CAD model as a Web service, the provider of the CAD model generates a describing file in WSDL format first and then registers the service with the WSDL file to the service registration server. This structure is used in distributed design application and presented in Chapter 4.

3.2.4 Parametric Design

With the development of CAD technology, parametric designs have become more popular in commercial CAD packages such as Pro/E, AutoCAD and SolidWorks. As a user of the mobile collaborative environment, the designer can change, receive and send the geometry model by revising the design parameters, because the data of the parameters and the basic graph of the geometry model provide all the information required for the designers, remote clients or collaborative designer/engineers. In chapter 4, SolidWorks is chosen as the CAD tool to demonstrate how parametric design may be conducted within a mobile collaborative environment. The method developed is generally applicable to other types of CAD packages.

3.3 Semantic Web and Agent Technology for MCWE

With the increased outsourcing and globalisation caused by heightened market competition, more powerful web-based product management systems are required to support the representation, exchanging, integration and sharing of product data over the internet. A product data representation framework is needed to support the exchanging, integration and sharing of product knowledge over the internet. A Semantic Similarity (SS) approach was introduced to enable the MCWE system to measure the strength of product concept similarity according to their SS values; this method could be used for providing an efficient and simple way to organise the search results displayed in mobile devices. The measure method of semantic similarity is proposed by taking advantage of the ISA relationship of concepts in ontology, which is shown by the experimental results to be an efficient method; these results are presented in chapter 5.

Semantic Web technologies help in the realisation of the vision of the Semantic Information System in the following ways:

- Semantic Web techniques, through the introduction of ontological reasoning, are suitable for flexibly discovering abilities in using information that were not specifically designed or intended for a particular use case.

- Separation of presentation and data, as ensured by the Semantic Web technologies, will enable the use of the same middleware tier for both mobile and fixed network clients.

- Semantic Web technologies can provide a standardised way of interpreting context, enabling both human and software agents to infer new context knowledge and consequently take intelligent actions.

But for the mobile users from different companies, remote users need to share and retrieve “Semantic” information among different Web servers; the solution is to introduce Agent technology within MCWE. The agent system for collaborative design, which is implemented based on Java technology, supports the construction of mobile agents or mobile agent-oriented systems with the dynamic loading class for various applications. An agent program could migrate or to make a copy (clone) itself across one or multiple network hosts in order to make a request directly, rather than communciate with the Web server over the network. The presented results in chapter 6 are based on a design example of an application operating in a mobile environment there need a common framework where different system can share their heuristic information on available resources.

3.4 Mobile Platform Compatibility: Flex Applications within MCWE

3.4.1 Mobile Platform Compatibility

With mobile web application expecting to have most of the characteristics of desktop applications, Rich Internet Applications are seeing a huge increase in popularity. Adobe Flex is the tool of choice for many web developers when it comes to building RIAs. Flex compiles to SWF files which are browser embedded Flash applets, the comparison with Microsoft Silverlight and Java Me has been discussed in chapter 2, Flex mobile application is a good choice to be deployed across different mobile platforms.

3.4.2 Combination of Flex with Web Service

Compared with Java technology, Flex itself is not client/server oriented, nor is writing client/server applications in Flex easier than Java. Introducing Web Service to the Flex environment means that Flex applications are no longer restricted to simple interaction with the user, but are instead capable of taking part in complex interactions with backend services. A combination of the Flex application with the Web service standard for application integration presents a good solution for creating application components capable of accessing multiple, shared backend services located across the Internet.

Web Service reciprocates by enabling the Flex application to interface with different programming languages and extends the Flex application by providing a framework for distributed object communications. Web Service provides a distributed object infrastructure over the Internet, and guarantees platform-independent functionality between vendors, which is essential for Internet applications, where different sites' objects, implementations, need to be able to communicate with each other.

The combination of Flex and Web Service provides a robust platform for Internet based applications, which consists of three tiers: the User tier, Web server tier, and back-end database tier, as shown in Figure 3.4.

- User tier — valid customers of the environment are allowed to visit the Web server tier over the Internet. On the user side, Microsoft Visual Studio and Flex builder is a popular integrated development environment for creating the mobile graphic interface.

- Web server tier — within the system, the CAD or other application program is located on the Server side. Users can input data, monitor the process of the execution, and retrieve resultant data through the interface. During the execution, the application program reads input data from and writes resultant data to database files through JDBC (Java Database Connectivity) API (Application Program Interface). The Server is a combination of Tomcat and Axis, which are the Web server and service providers.

- Back-end database tier — this tier is a database server that gathers and stores historical and real-time data received in the process of design and manufacture. The databases can be of different types such as Access, SQL, etc.

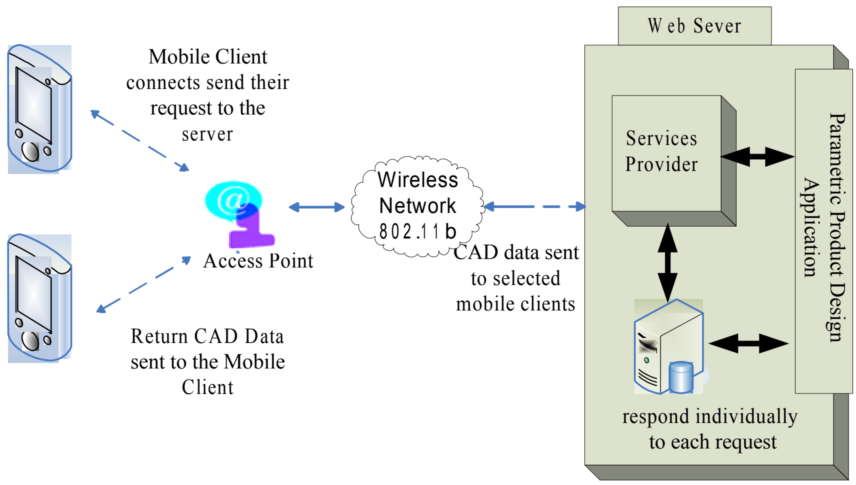


Figure 3.4 Flex mobile application communciate with Web server

This server-centralised structure could be used for remote invocation of a singular large-scale program, as presented in details in Chapter 8.

3.5 Summary

A Mobile Service-Oriented Framework for Computer Aid Design will be developed based on ADMEC previous research. The framework enables geographically dispersed team members to efficientl share and retrieves their product information at anytime and anywhere. The technique of Web Services has been utilized for the users to provide and to request the services, and processing the Mobile CAD application, such as PowerCAD or ShortCAD.

A Semantic Information System to support the exchanging, integration and sharing of product data over the internet was introduced. A Semantic Similarity approach is also introduced to make the MCWE system measure the strength of product concept similarity by their SS values; this method could be used for providing an efficient and simple way to organise the search results displayed in the mobile devices.
Mobile Flex SWF files are presented to enhance the platform-independent functions in distributed systems. Flex application is not only capable of simple interaction with the user, but also of taking part in complex interactions with backend services in a Service Oriented distributed system. A combination of SWF file with Web Service for application integration presents a good solution for application components capable of accessing multiple, shared backend services located across the Internet, that are complementary in order to ensure high source portability and robust platforms for Web-based applications.

Chapter 4

Web Service Oriented Mobile Computer Aided Design (WSO-MCAD)

Because of recent advances in wireless communication technologies, the world of mobile computing is flourishing with a variety of distributed applications being proved. This chapter presents an integrated architecture for mobile device based Computer Aided Design system that supports collaborative work among remote users.

4.1 Introduction

Product design is often geographically distributed, requiring collaboration and effective communication among the companies involved in the systematic definition of a new product [1, 72]. As a result of this new product development, some research findings have been continuously reported, indicating the substantial progress in this research area. For example, chenga [73] developed a prototype of Web-based distributed problem-solving environment to facilitate computer aided engineering technologies and to support networked collaboration, scientists around the world could interactively, visually and experimentally explore their daily design work through the proposed system. But the methodologies and technologies used in stationary networks are not always applicable to mobile systems; that limitation are described in details in chapter 1 and 2. Therefore, to deploy successful mobile applications in the distributed product design environment, two major problems need to be encountered [74].

- The first problem needed to be resolved was the network connectivity, which largely affects the time and size of data downloading, thus, how to deal with various network conditions also needed to be addressed in creating a mobile application.

- The second problem was concerned with mobile devices. The physical constraints of mobile devices, such as screen size, display resolution, limited computational power and memory, which can significantly affect the usability of mobile applications [48]. It is a challenging task to deploy the design applications in mobile devices, because most graphic system and CAD functionalities need powerful computers to perform their tasks. CAD parametric design is the way to resolve the problem; mobile designer can exchange, receive and send the product data via their mobile device, because most complicated CAD process will be operated in the Web server [38].

To resolve these problems, WSO-MCAD was developed based on ADMEC's previous research. ADMEC had developed an online collaborative CAD framework, but it did not support mobile devices. These mobile systems enables geographically dispersed team members to efficient share and retrieve their product data at anytime and anywhere within WSO-MCAD. The technique of Web Services has been utilized for the users to provide and to request the services, and processing the Mobile CAD application. CeCAD and ShortCAD, (Appendix A will describe the two mobile CAD packages) the two CAD application could be installed in the PDA, are used for the remote user to view or modify the CAD data [75, 76]. For example, if the designer or the manger needs to choose which type of bolt is really suitable to their requirement, firstly, they need to input the bolt type, bolt specifications and etc, all the Search, Analysis and Check process will be carried on in the Web server. After the mobile user input the Bolt type, for example, M12, and retrieve all the parametric data of bolt M12, therefore, mobile system could activate SolidWorks or AutoCAD package and return the Jpg or CAD Drawing file, such as DWG, DXF files, back to the mobile devices. If the user designs a product in the mobile device, this system could also send the mobile CAD file back to the server, users in the office could view or discuss the new design ideas with their colleagues (more details about Bolt Analysis and Check functions will be described in chapter 8).

4.2 Main Features of Web Services Utilized in WSO-MCAD

With Web service functions, Web applications can provide services for thousands of users at the same time; system can mix VB.Net, Java and various other languages together. Adopting unified XML, which is the most important feature of the Web Service application program, used for data description, making the service oriented architecture possible. Services can be described as Web entity and receive all information transmitted in SOAP in XML format and this type of message is easy to go through firewall. Web services easily allow software and services from different companies and locations to be combined easily to provide an integrated service, the change of one resource will reflect immediately to others that are sharing the same service. Therefore, the system is not only reliable, but also flexible and extensible, in order to meet the increasing new requirements.

4.3 Development of Web Service for Online CAD within WSO-MCAD

4.3.1. Web Service Operation Procedure

In WSO-MCAD, Web Service technique plays a key role of connecting an individual site, where the CAD package is located and related services are available to the Internet. The mobile application carries out the tasks of dynamic data processing and links with the Mobile CAD package and Web Services. The designer operates the CAD package to produce drawings and makes changes which are instantaneously sent, with the aid of mobile application and Web Service, to the collaborative designers located in different sites.

As shown in Figure 4.1, in order to load CAD model as a Web service, the provider of the CAD model generates a description file in WSDL format first and then registers the service with the WSDL file to the service registration server. WSDL file contains all the information of the service including the location 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, if the service is found, the registry server sends WSDL file to the requester. With the information provided in the WSDL file, 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 request, after that, data message would be transferred to the mobile client.

4.3.2 System Components

In this research, WSO-MCAD system consists of three modules:

- Request module: This is for a computer to send request to service via a network. Before that, this module needs to capture the parametric input data or the CAD drawing files and describe the request information with VB.Net application.

- 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. When the parameter data was 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 4.1 shows how the three modules work together in the procedure. It has to be noted that when the connection is established between mobile devices and computer server, all the mobile devices will be considered as clients. For example, in Figure 4.1, when mobile designer B sends parametric data or makes a change in the CAD drawing, the system sends the request using request module, that contains the information of the input data or drawing files, to computer A through middleware module. Computer A then calls its response module to action. In this scenario, Computer A is considered as a server and mobile designer B is a client. Another thing should be addressed is that; the request module and response module exist in the mobile client requester and service provider, they only deal with data processes and have nothing to do with data transmission through network.

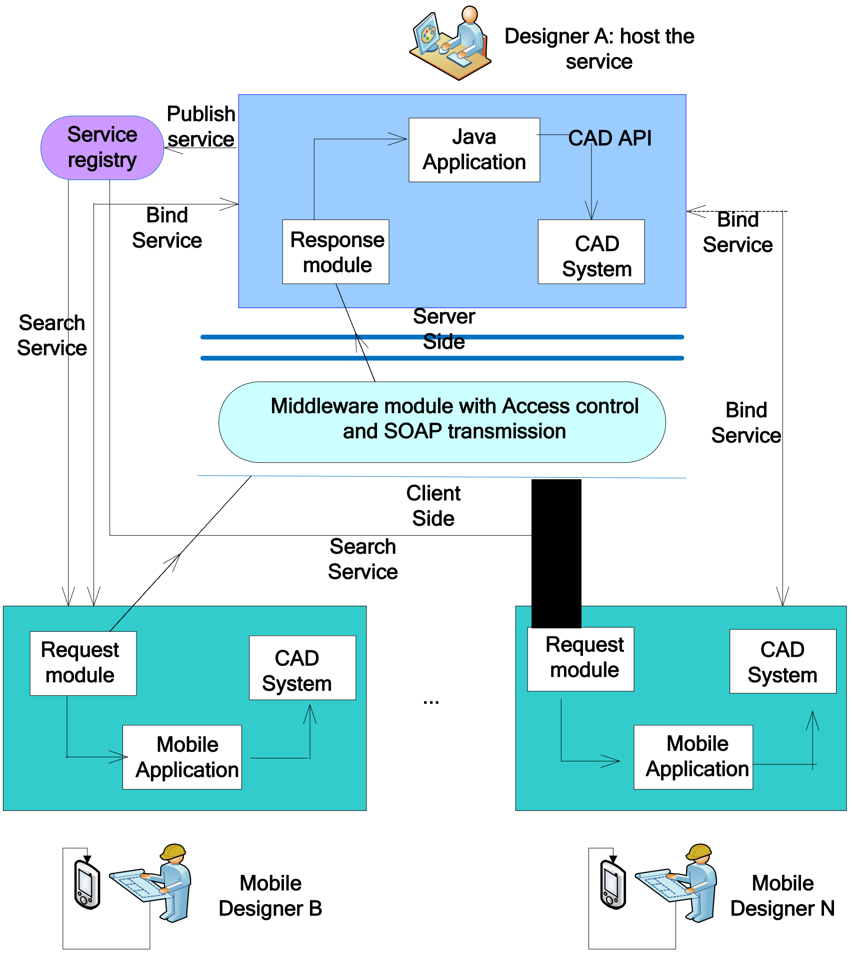


Figure 4.1 System modules and data transmission flow

Request module:

If the engineer or the manager wants to perform the parametric design or share the CAD drawing files with other people, but they are not in the office. They need firstly decide which CAD file format, that depend on which mobile CAD application installed in the mobile devices, to be sent or retrieve; system then need to call the mobile applications to sent the request to the Web server. The procedure can be divided into the following two sub-procedures: (1). Invoke outer program. (2) Send request information to the aimed computer Server (Figure 4.2).

(1) Invoke outer program

If the client wants to send the input parametric data, the mobile device needs to activate the VB.Net parametric application to get the input data from the user, or if the remote user wants to send the CAD drawing files to the aimed computer, CAD files need to be saved firstly by the AutoLisp program, which is supported by CeCAD; but for the people installed ShortCAD in PDA, they need to save the files as ShortDWG file format.

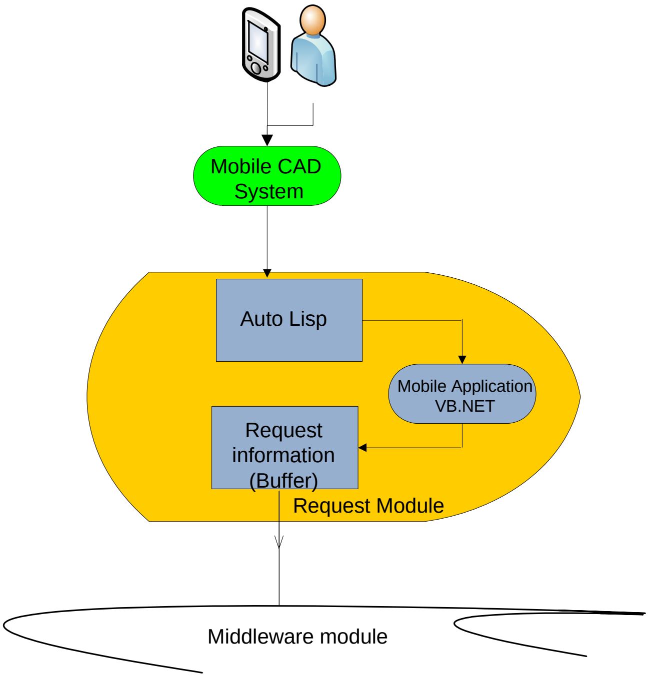


Figure 4.2 Data process of Request Module

(2) Obtain the input parametric data and read modified CAD drawing files

Microsoft VB.Net mobile application will get the input parametric data or read the CAD file, which the remote users want to share with other designers, all the reading CAD files will be converted into byte format. The VB.Net program is used to be packed into SOAP request and send the request to the middleware module.

Since Web Service developped by Java lanaguage, and the mobile application written in VB.net language, CAD file transfering from Web server to mobile device will lead to the type conflicted problem. For example, VB.Net used binary reader to read an array of SBytes to buffer, but Java used the Bytes input. In order to resolve this problem, author developed the new/extra functions to convert an array of Sbytes to an array of bytes or from bytes to SBytes, some source code of this function are shown in Figure 4.3.

```
  ...
For i = 0 To fssize.Length - 1
            If fssize(i) > 127 Then
                MySbyte(i) = Convert.ToSByte(fssize(i) - 256)
            Else
                MySbyte(i) = Convert.ToSByte(fssize(i))
              End If
 …
Dim ugif(recbyte.Length - 1) As Byte
        '   ReDim ugif(sgif.Length - 1)
        System.Buffer.BlockCopy(recbyte, 0, ugif, 0,
recbyte.Length)
        '   For i As Integer = 0 To recbyte.Length - 1
        'ugif(i) = Convert.ToByte(recbyte(i))   '<<<< compiler
error
        '   Next

        Dim recfs As New IO.FileStream("d:\vb\Mreceive.dwg",
IO.FileMode.OpenOrCreate)
        'receive data and save that into the file
        Dim newfilestr As New IO.BinaryWriter(recfs) 'stream save
        'recfs.Write(recbyte, 0, recbyte.Length - 1)
        newfilestr.Write(ugif)
          recfs.Close()
  …
```

Figure 4.3 Conversion functions of VB.Net

Response module

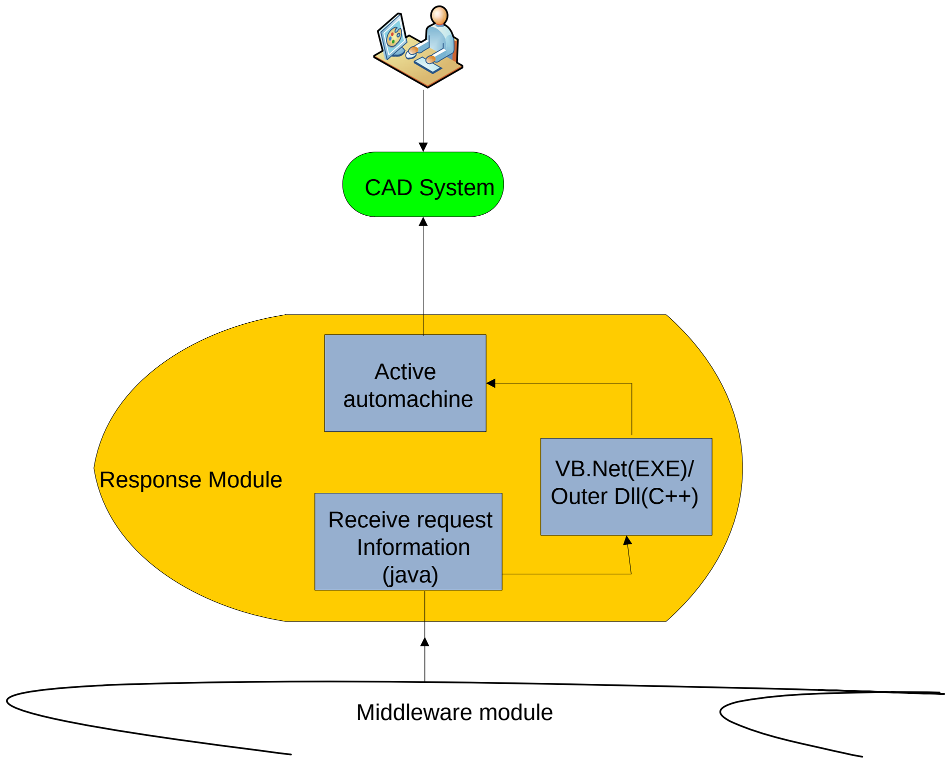


Figure 4.4 Structure of Response Module

This module is located in the service provider, which is mainly deal with the received request from the different locations. Similar to the request module, two processes here should be addressed:

When information for a request is received, the Java code from provider will decide to store the mobile CAD files into special fold or call the other EXE file, which will take the parametric data from the SOAP, and activate the Server CAD package to draw the CAD files in the Web Server; after the CAD package are active to process the parametric design, the output files will be sent back to the mobile client, more details about parametric design will be described in section 4.4.

Middleware module

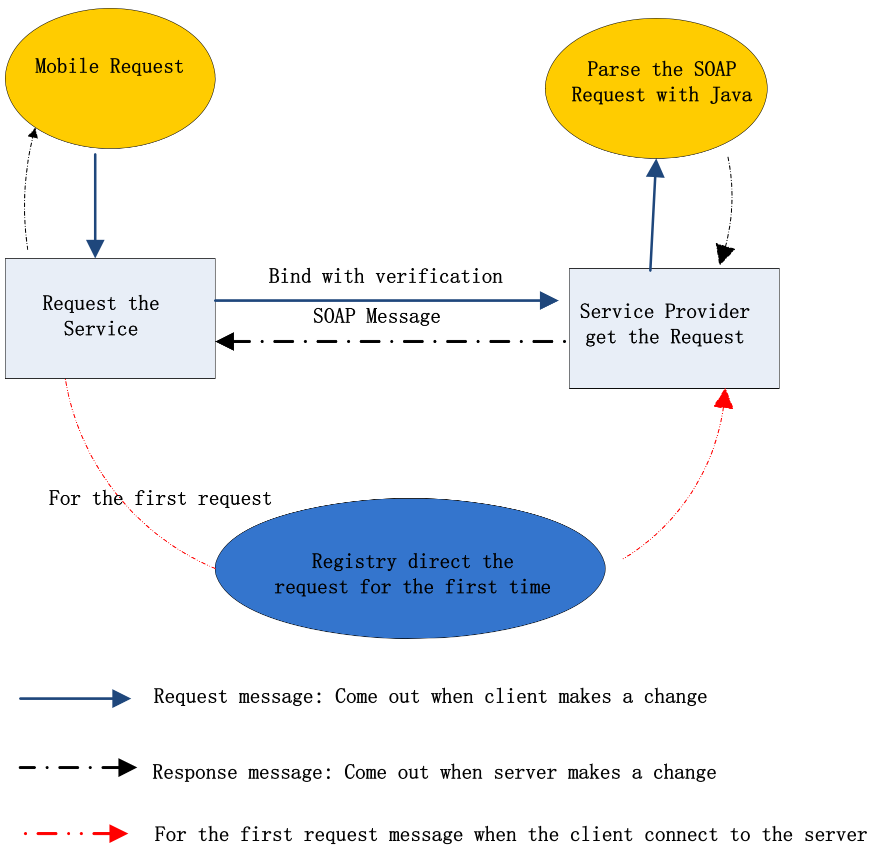


Figure 4.5 Process of Service Module

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

Security is taken into consideration within the area of this module. There is a table that is used to store the ACL (Access Control List) in the service catalogue. 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. In the future version, PKI (Public-Key Infrastructure) will be used to do access control [31].

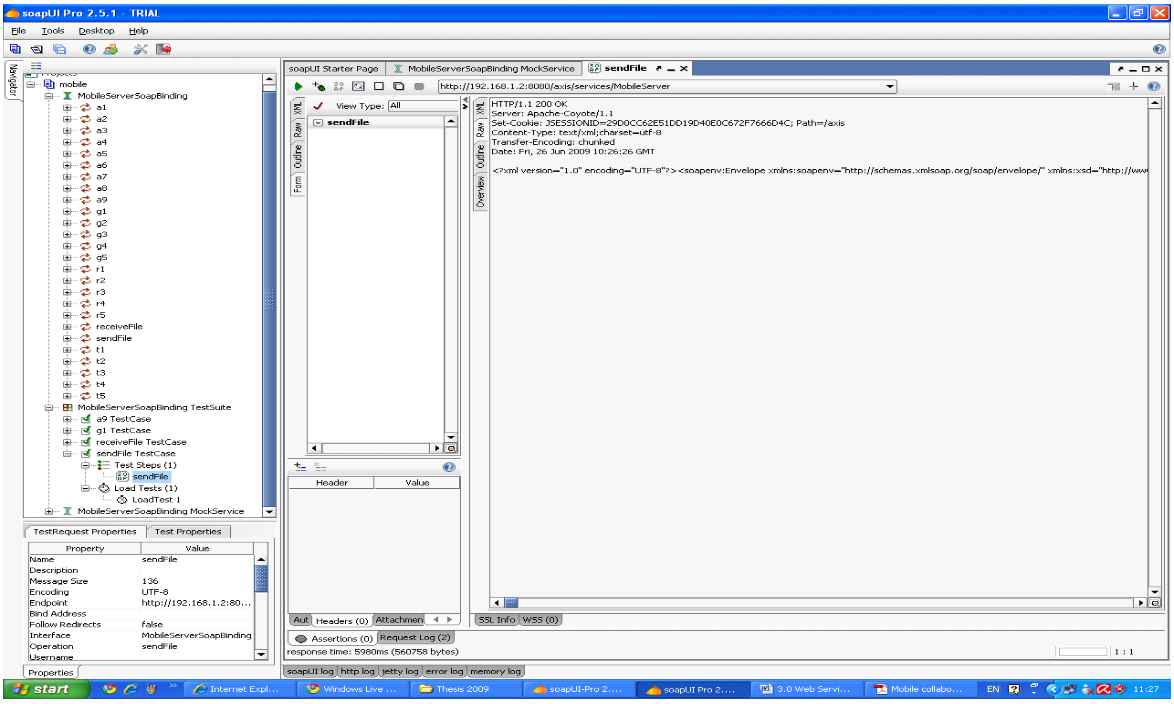
4.3.3 Test tool and Discussions

SoapUI, which is the leading tool for Web Service Testing [77], is chosen to test the WSO-MCAD; this tool can be used for a variety of purposes:

- **Unit testing:** validate that each Web Service operation functions as stated.

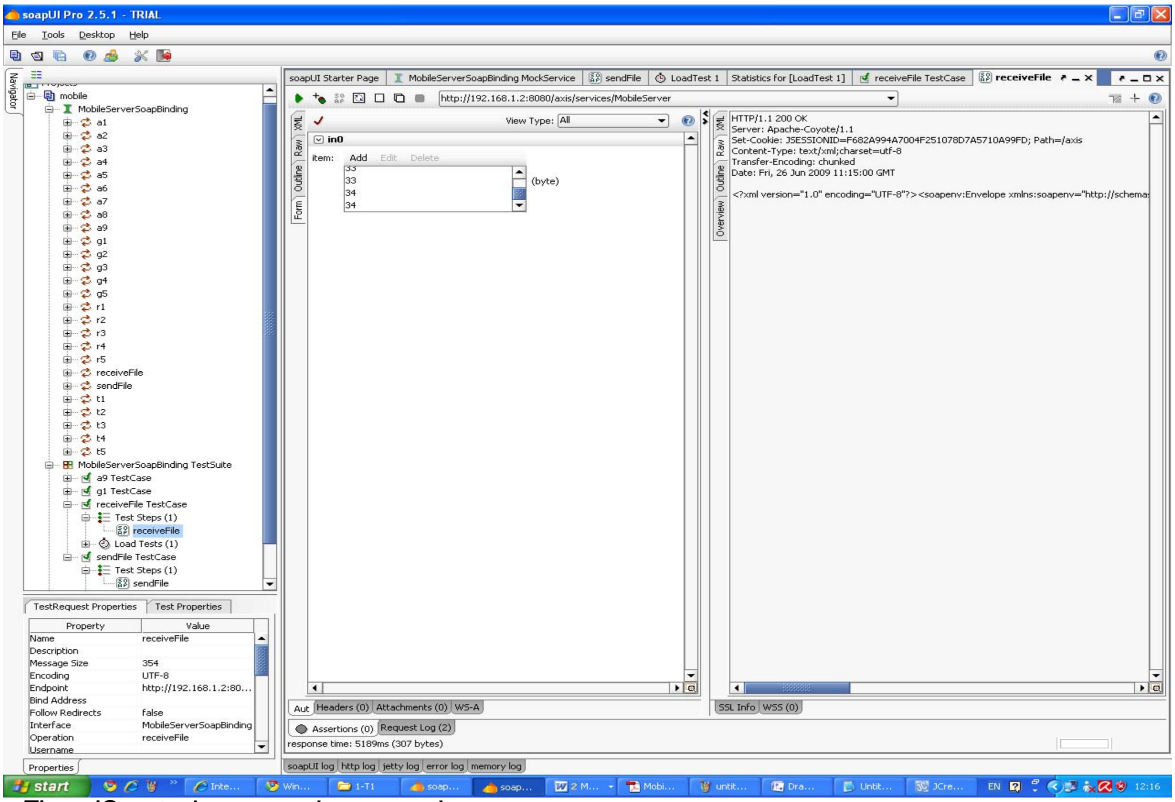
- **Compliance testing:** validate that the Web Service returns results compliant with its definition.

Figure 4.6 (a) is the test result when the mobile client sends the CAD drawing file to the Web server, SoapUI send different bytes of files to the Web server, checks the response from the Web Server; Figure 4.6 (b) shows the simulation about multiple users sending requests to the Web server. All the test results will be stored in a local XML file. This test tool is powerful; provide minimum, maximum and average size of transferring files as well as the speed of transferring and error information. This research currently focuses on the speed and error information generated by SoapUI (red line in the figure).



Test Step,min,max,avg,last,cnt,tps,bytes,bps,err
sendFile,5421,16149,6012.78,5751,414,0.73,232153812,410970,0
TestCase:,5421,16149,6012.78,5751,414,0.73,232153812,410970,0

(a)



ThreadCount,min,max,avg,last,cnt,tps,by
tes,bps,err
5,5049,5310,5188,5050,5,0,1535,301,0
5,5049,5310,5171,5093,9,0,2763,251,0

(b)

Figure 4.6 SoapUI Test Snapshot

In the test, all the three modules (section 4.3.2) of WSO-MCAD work 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 affect the cooperation amongst the three modules. Multiple threads applied in Web Service and activatate each module to enable the function which sends parameters while listening to others.

There maybe some exceptions, for example, it is possible that when a user sends a request, there are no service providers available, in this situation, the system will send an alert to the remote user.

4.4 Parametric Design

Today, parametric designs are more popular in commercial CAD packages (i.e., Pro/E, AutoCAD and SolidWorks). The Mobile designer can change, receive and send the geometry model by revising the design parameters, because all the information required for designers are provided by the data of the parameters and the basic graph of the geometry model.

Two methods have been used for the parametric design on CAD softwares:

- One method is to draw geometric entity of a part first, then to customize rational variables for controlling size of the entity in modelling. Modifying the beforehand setting variables with VB/VC++ program developed is the easy way to update the part entity. The main advantage of this method is its strong universal performance and convenience, but it is powerless for product of complicated surface [38].

- Another method is to fulfil whole course of geometric entity, applications based on this method create CAD files and adding/editing all the parametric data automatically, only the designers have good programming skills and be familiar with the design packages, could develop this kind of applications by themselves, that maybe be a complicated and difficult design task for most developers [42].

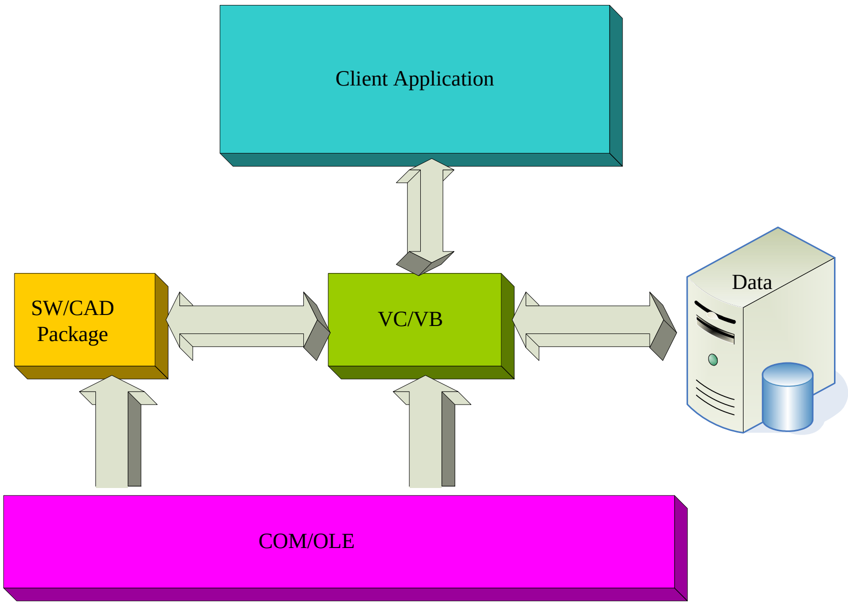


Figure 4.7 Architecture of SolidWorks/CAD parametric design

In the test, SolidWorks is chosen as the CAD tool to demonstrate how the parametric design is conducted within the mobile collaborative environment. The method developed is generally applicable to other types of CAD packages. Producing a design using SolidWorks, system will typically open a template worksheet, set several values, recalculate and then record some outputs, if the user wants to save the worksheet as the original filename or under a new filename. These actions form the basis for an automation interface. A SolidWorks API is developed in order to enable the automation interface and provide the capability to automate the inputs/outputs of a worksheet without needing to programmatically replicate the mathematical logic of the worksheet. Operation flowchart is shown in Figure 4.7.

4.5 Experiment

4.5.1 Bolt design:

A three dimensional Solidworks part drawing starts with a two dimensional sketch on one of the three original planes or a user create one if necessary. The sketch represents the section that includes the most details of the objective, based on which the features are extruded, cut or revolved. A complicated 3D drawing usually can't be accomplished by a single time of sketching and featuring, the process has to be repeated before it is finished. Solidworks is also capable of featuring curved surfaces.

The mobile interface is an implementation of the secondary development of SolidWorks which meets the Chinese standard and customer request. In this research, the dimension drive was applied as a foundation of this user interface. Dimension drive, in the condition that the configuration of model topology not changed, defines the dimensions of model as a variable and the relationships between each individual dimension. (Analysis and check function for Bolt will be presented in chapter 8)

Microsoft VB.Net application is deployed as the client input interface in the mobile device.

1. Choosing the bolt type and bolt specifications firstly in Figure 4.8 (a). Because the different bolt type and specification will decide the bolt head height, bolt size and bolt head width, which is shown in (b), after the user choose other values, such as, thread height type, thread length and thread pitch, then user could get the thread height. As shown in (c). The thread height is decided by thread length, mobile system search the Web database and return the suitable result back to the mobile device (red line).

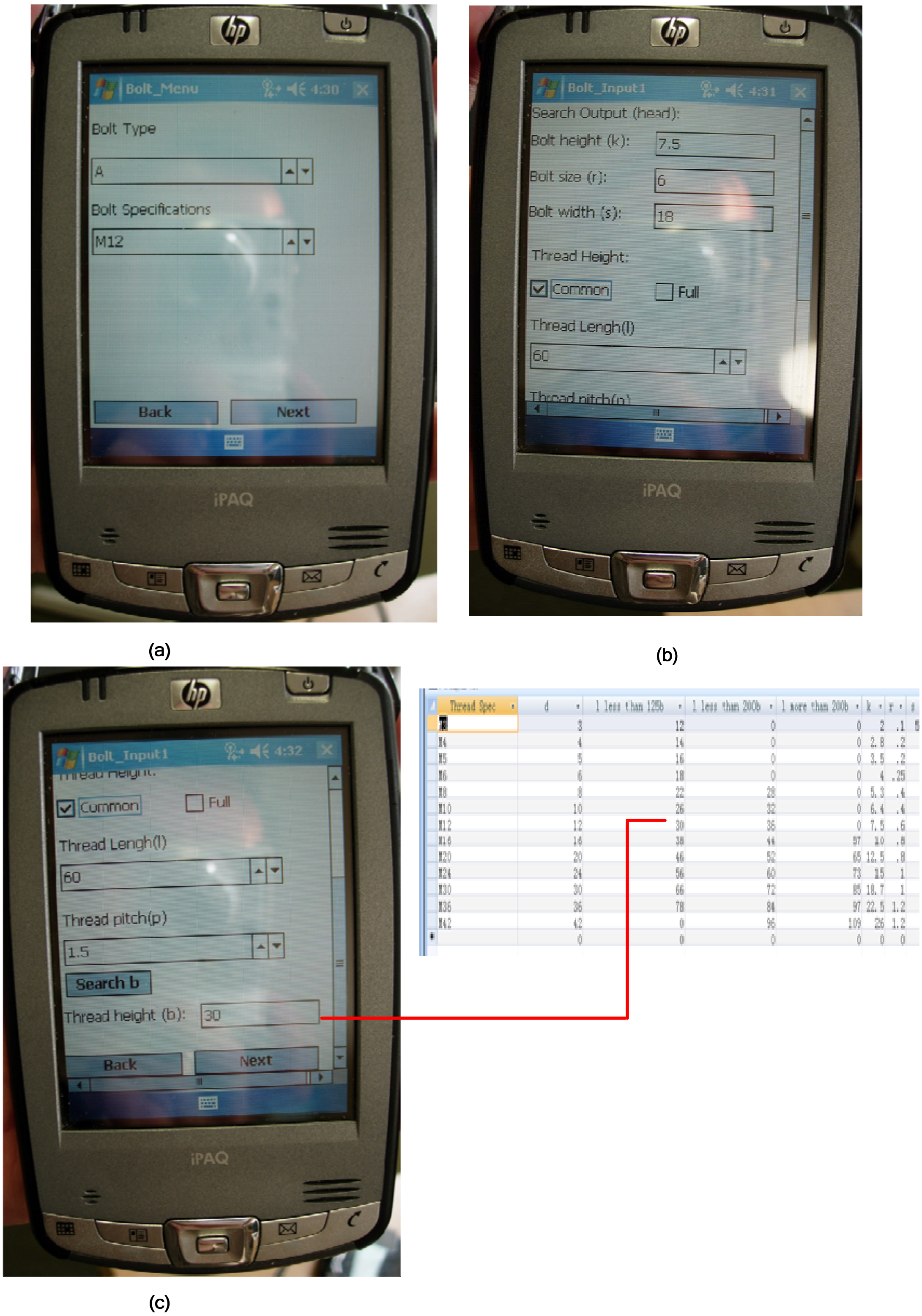
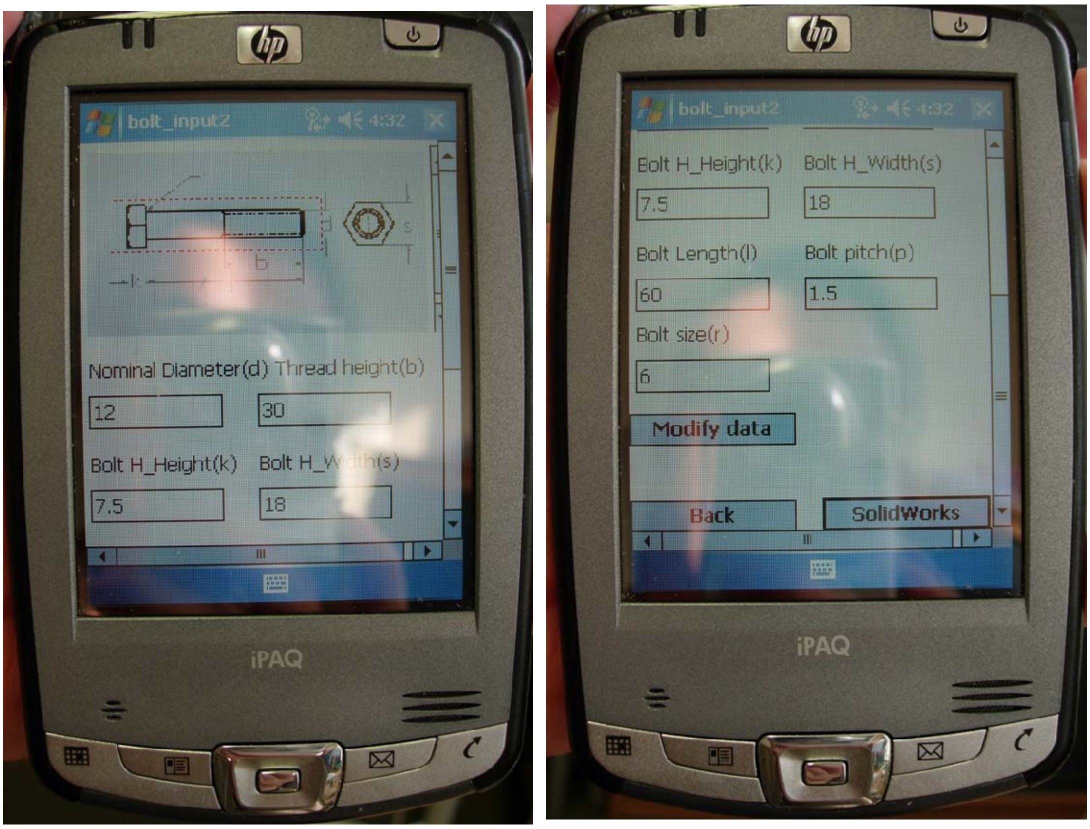


Figure 4.8 Mobile bolt input interface

2. After the mobile user gets the thread height, the mobile device will present the 2-D picture of bolt as well as the parametric data of bolt, such as: Nominal diameter (d); Thread height (b); bolt head height (k); bolt head width(s); bolt length (l); bolt pitch (p) and bolt size (r) in Figure 4.9. Mobile user also could change the parametric data of bolt via the mobile device again; all the parametric data will be utilized for the Bolt design for SolidWorks.



(a) (b)

Figure 4.9: Parametric data of bolt and 2-D CAD picture

After the remote user click the "SolidWorks" button, the mobile device will invoke the Web service to send the parametric data to the Web server; then Web Server activate SolidWorks package to create Jpg file, Figure 4.10 is the VB parametric design source code in Web server.

```
Option Explicit

Public Function ExecuteSQL(ByVal SQL As String, MsgString As String) As
ADODB.Recordset
   Dim conn As ADODB.Connection ?connection with database
   Dim rst As ADODB.Recordset
   Dim STokens() As String
   'Dim ConnectString As String
   On Error GoTo ExecuteSQL_Error
。。。

   Set swApp = CreateObject("sldworks.application") 惛ctivate SolidWorks
   swApp.UserControl = True
   □
   """"""Sketch SolidWorks function
   boolstatus = Part.Extension.SelectByID("D2@Sketch1@" + fileName + ".SLDPRT",
"DIMENSION", 0, 0, 0, False, 0, Nothing)

  …
  """""""Extrude SolidWorks function
   boolstatus = Part.Extension.SelectByID("D1@Extrude2@" + fileName +
".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
   Part.Parameter("D1@Extrude2").SystemValue = d1e2 / 1000 """"""""""1
   boolstatus = Part.Extension.SelectByID("D1@Extrude12@" + fileName +
".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
   Part.Parameter("D1@Extrude12").SystemValue = d1e12 / 1000 """"""""""2
```

Figure 4.10 VB parametric product design code

3. Parametric Design result will display the Jpg file in PDA; Jpg file is suitable for all the mobile devices.

Figure 4.11: SolidWorks Jpg file shown in PDA

Mobile system also could choose AutoCAD to be installed in the Web server, in this situation, parametric design will not only create Jpg file, but also DWG file could be provided by the Web server, thus Web server could send the DWG file back the mobile device, then the designer could view this CAD drawing file by the Windows OS devices.

4.5.2 Collaborative CAD Design via PDA

Another example is that, the task of the collaborative CAD conducted is to modify a shaft design online amongst three geographical dispersed designers A, B, C. That included computer with Pentium IV 2G processors, which was installed Tomcat 5.5.27 and Axis1.4, to provide Web server and Web services; VB.Net created the mobile application in two mobile devices, that are HP PDA and Acer PDA; Netgear Rooter provided the wireless Network, whose speed is 54Mb/s; ShortCAD as mobile CAD application run in each PDA, and ShortCAD for PC was installed in the Web server, system operation procedure is shown below:

Before the online collaboration, WSO-MCAD deployed the service and published it on the web. To initiate the online design, Designer B opened a drawing in SolidWorks on the Web Server computer, if the Mobile Designer A wanted to share its CAD drawing to the designer B, mobile VB.Net application will invoke the Web Service sendFile method, then the saved ShortDWG file will be sent to the Computer B, if Mobile Designer C want to retrieve the drawing file from Web server, Designer C invoke the Web Service method ReceiveFile method, then the Drawing file will be sent back to Designer C. Another thing need to mention that, in order to avoid the CAD file conflicted problem, if the PDA choose to install ShortCAD, the sharing CAD file had better be the ShortDWG format, which is the special CAD file for this mobile CAD package; but for CeCAD, because most CAD applications could view and modify DWG files, the file conflicted problem will be ignored.

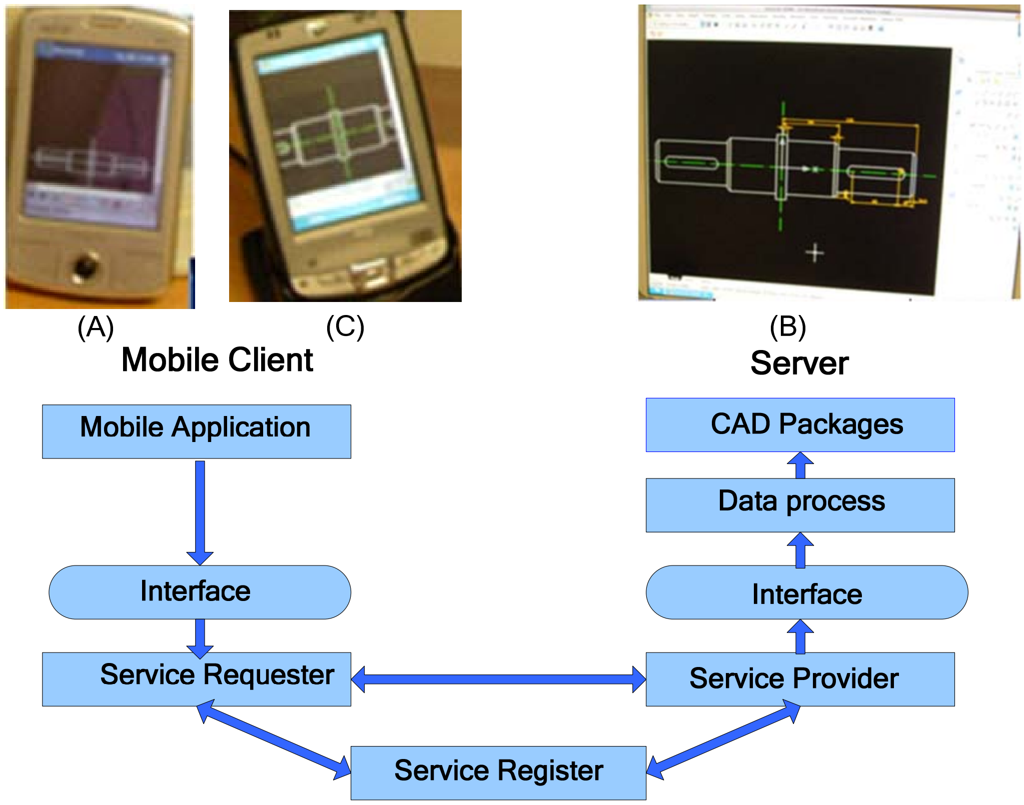


Figure 4.12 Mobile CAD File transfer

4.6 Conclusions

The above test indicates that WSO-MCAD system can meet the expectation in section 4.3. However, as a feedback from the users, the real-time interaction has latency limitation; the development of the system does not consider the concurrent control among the participants; and exchange different CAD files will lose some parametric data; all that shows that there is a distance from current research to the commercial software.

Another thing must be noted is the system is based on open source tools Tomcat and Axis, which could be downloaded everywhere without any problem of copyright. And because the author is a testing member of the ShortCAD Company, he could get the latest version of ShortCAD application, but users download the beta version which will lose some functions.

WSO-MCAD system is strong and reliable due to the design of SOA that already helped ADMEC on design and research in mobile collaboration enviorment. The service oriented architecture, which is loosely coupled, makes the system much less fault occurring in comparison with traditional systems. The feature also provides the possibility for the communication between AutoCAD and Pro/E, the development of WSO-MCAD system is the first step of the research on MCWE, where the Service-Oriented interaction plays an important role. The next chapter will consider the important aspect of collaborative design, share and exchange product data within wireless distributed system.

Chapter 5

The Application of Semantic Web Technology into Mobile Collaborative Design

Since the increased outsourcing and globalization of market competition demands, the designer needs more information when they are not in the office; this chapter will present a mobile semantic product information system that supports retrieveing and sharing product data among remote users. Semantic Web Technology has been utilized to provide semantic relationship and similarity of product data. The measure method of semantic similarity is proposed by taking advantage of the ISA relationship of concepts in ontology, which is shown by the experimental results that this method is efficient.

5.1 Introduction

Since the increased outsourcing and globalization of market competition, there is a demand for powerful web-based product information management system to support the representation, exchange, integration and sharing of product data over the internet. But, the traditional Product Data Management (PDM) system is difficult to describe the complex/underlying relation among the product data [21]; there are some disadvantages which cannot satisfy current requirements, such as:

- The search algorithm for traditional database is based on keywords searching, which cannot satisfy the high efficiency and complex requirements. Search results maybe lose some helpful information which is not indexed by the keywords. For the requirement of the mobile users, they need the return results more accurate because the small screen of the mobile device could not display a lot results and the connection fees could be expensive [7].

- Traditional database can not link all related information together to make the search result more appropriate to the user and easy to be managed. To overcome the limitations of traditional discovery models and improve discovery effectiveness, the emerging design guideline for novel discovery solutions is the adoption of semantic Web technology. Semantic Web technology permits explicit representation of interacting entities, e.g., services, resources and users, at a high level of abstraction while enabling automated reasoning about this representation, favouring interoperability and understanding between entities which have little or no prior knowledge about each other [57].

- The physical constraints of mobile devices, such as screen size, limited computational power and memory, which can significantly affect the usability of mobile applications link with traditional database, if the mobile user wants to retrieve some result via wireless network [46].

Therefore, in order to overcome the mentioned limitations and to deploy successful mobile applications in the distributed environment, following major tasks need to be focused on. Firstly, mobile client application should only ask for the minimum user input possible; secondly, the mobile system should resolve the user's privacy concerns and enable further services by taking the personal input data. For example, the user may ask the system to recommend "standard bolt" for a private requirement. In this case, the user may simply ask "find product bolt"; however, since "product bolt" in this context does not mean information about the supplier information of the bolt but rather material information for bolt, such as the CAD parametric data.

Conventional information system does not possess any intelligence to cooperate with database users. In order to meet the different requirement from users, system developers have to fully understand both the metadata and contents of the database.

Even the users are familiar to the information system; they also have to retry specific queries repeatedly with alternative values until the query result is satisfactory. In order to overcome the limitations of current information system, new Semantic Information System need to be developed with following features:

- An information system needs to understand the schema and semantics of the database, it will be able to return informative responses and help the user input more correlated queries.

- A query language is used to obtain information from a database; and more user-friendly and fault-tolerant query interfaces will be developed. When a query search condition does not match with the underlying database, users would rather receive approximate answers than null information by relaxing the condition.

Thus, the Semantic Web Technology performs major implications in the development of new “semantic” information management recently; within this research field, ontology plays a key role for realizing the Semantic Web, which provides a common, shared understanding of knowledge in an interest domain; capture and formalize knowledge by connecting human understanding of symbols with their machine process ability, and through the introduction of ontological reasoning, the approach are suitable for flexibly discovering abilities in using information, that were not specifically designed or intended for a particular use case [56].

Furthermore, semantic similarity (SS) approach is also introduced to make information system understand the schema and semantics of concepts and to measure the strength of similarity between data values [78, 79]. The value of SS can be used as a measure to determine the rank of each answer, which helps the users find useful information related to the input data. With the authors’ previous research in ontology construction and modelling in product design [6, 7], all the input data will be decided by their places in ontology tree and the return results will be assigned the value of semantic similarity; mobile semantic similarity information system utilizes the semantic similarity calculation method to provide a straightforward and efficient approach to rank the search results sorted by the value of SS. Another advantage of the knowledge system is that, the return result will not only show the information data property of the product, but people could find some underlying meaning; for example, if the user input “Bolt, the type is M12”, but the user maybe want to view the supplier’s information, or the other products which be provided by the same supplier. With the support from Ontology Reason Tools, such as Racer, Pellet etc, mobile system could resolve the problem by adding new ontology rules and do not needs to re-write the codes that make this system more robust and re-usable.

5.2 Ontology Construction and Modelling Technology

To meet the demands of the mobile information system for the collaborative product design, an ontology-based description model is proposed to present in order to manage and deploy the distributed design resources, this research used Protégé by Stanford University for constructing the ontology, but capturing the concepts and describing the relations between the concepts is still a difficult problem.

5.2.1 Build OWL Ontology

With the help from ADMEC members, author used the available documents in the first iteration of the building phase. A simple class hierarchy was built, but it was difficult to derive relations, constraints, and axioms from the documents; so after the document analysis, the focus was switched to the other knowledge sources, such as Email and Telephone with the Employers of the companies. The resulting ontology has 4 classes directly beneath the root of the subsumption hierarchy, such as Role, Document, Product and Requirement. The most general classes are illustrated in Figure 5.1 through a screen-shot from Protégé 4.0. (Companies' detail will be described in chapter 8)

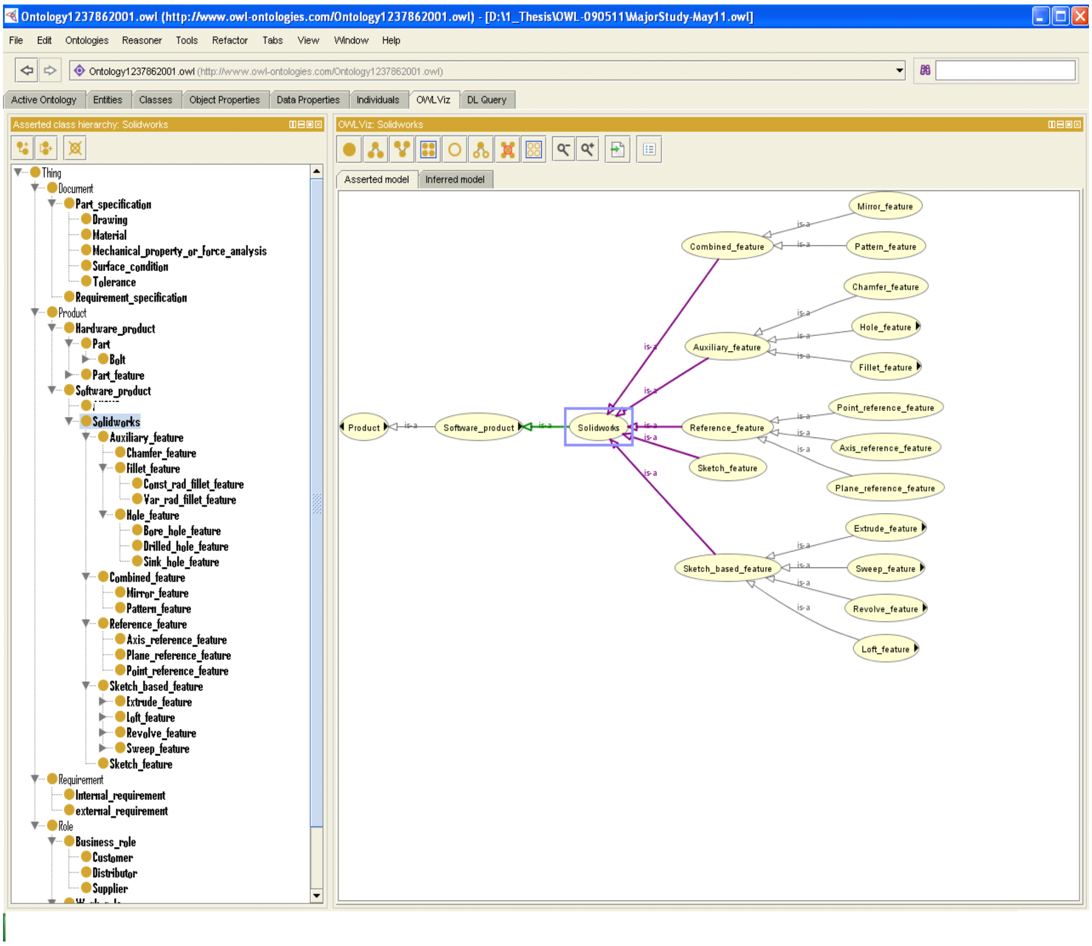


Figure 5.1 Top-level classes and relations within Protégé 4

The resulting ontology contains some major parts, as can be seen in Figure 5.1, the figure shows a small part of the ontology and some details are hidden to increase readability. Despite this, the division of the ontology into subject areas can be noted. For example, under the root class "Product", "part" is a (ISA) subclass of software, and "part" has the child-class "bolt", which denotes the product provided by the company; and if the user wants to analyse and check bolt quality, they could refer to the Root class "Requirement".

5.2.2 Ontology Modeling using Protégé 4

OWL ontologies have similar components to Protégé frame based ontologies. Below is the descripton of Building OWL Ontology using Protégé 4 for the mobile system.

- **OWL Classes** are interpreted as sets that contain individuals in Figure 5.2. They are described using formal (mathematical) descriptions that state precisely the requirements for membership of the class. For example, the class "Product" would contain all the individuals that are product in the domain of interest. Classes are organised into a superclass-subclass hierarchy, which is also known as taxonomy. Subclasses specialise their superclasses. For example consider the classes "Product" and "Bolt": "Bolt" might be a subclass of "Product" (so "Product"is the superclass of "Bolt"). That means that all "Bolts"are "Product"; all members of the class "Bolt"are members of the class "Product".

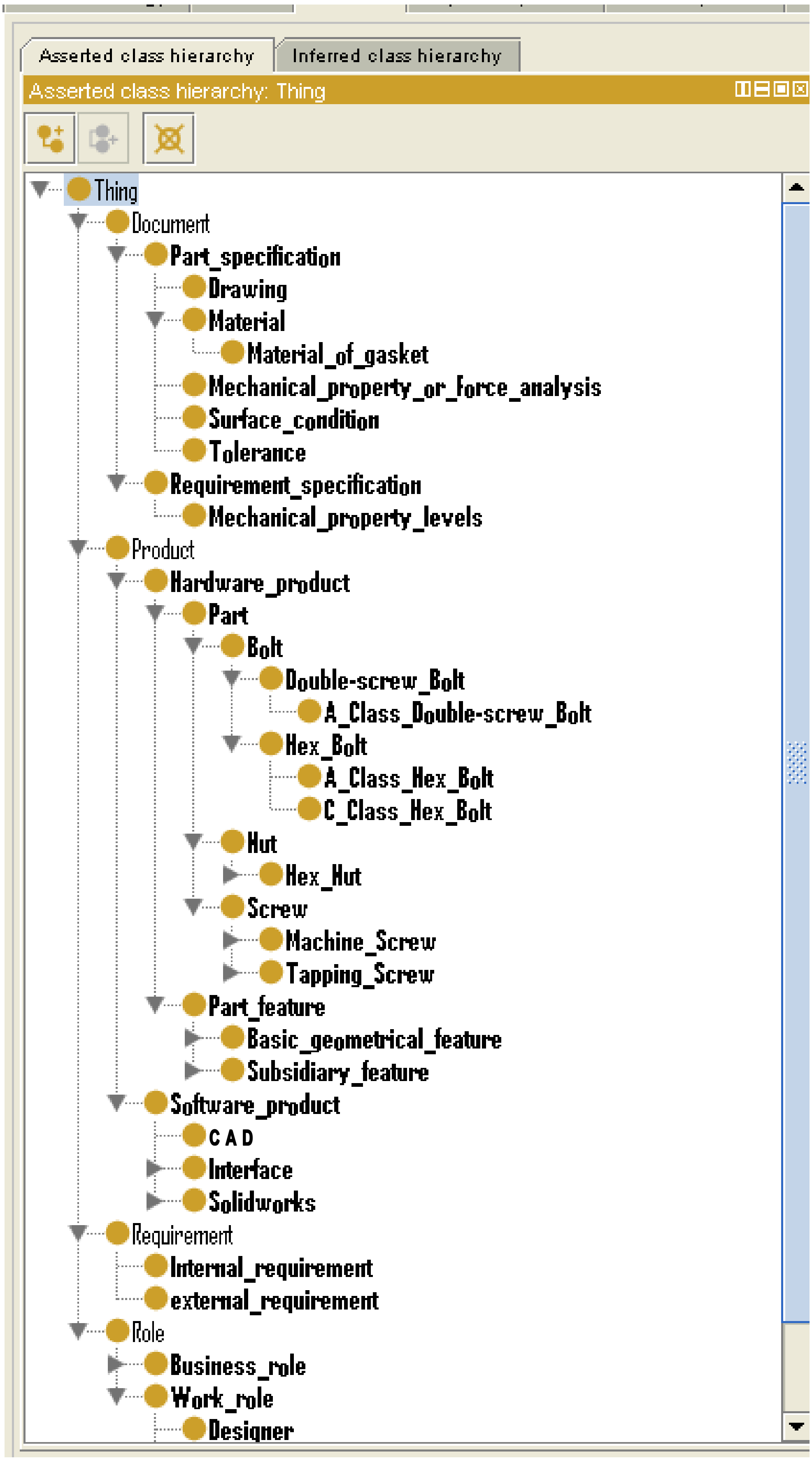


Figure 5.2 Classes and subclasses within Protégé 4

- **Properties**: They are also known as relations in Unified Modelling Language (UML) and other object oriented notions. For example, the property "Produce" might link the individual "Bolt Factory" to the individual "Bolt" ("Bolt Factory" Produce "Bolt"), or the property "Produced_By" might link the individual "Bolt" to the individual "Bolt Factory", Properties can have inverses. Figure 5.3 is the Property of "A_Class_Hex_Bolt_M10".

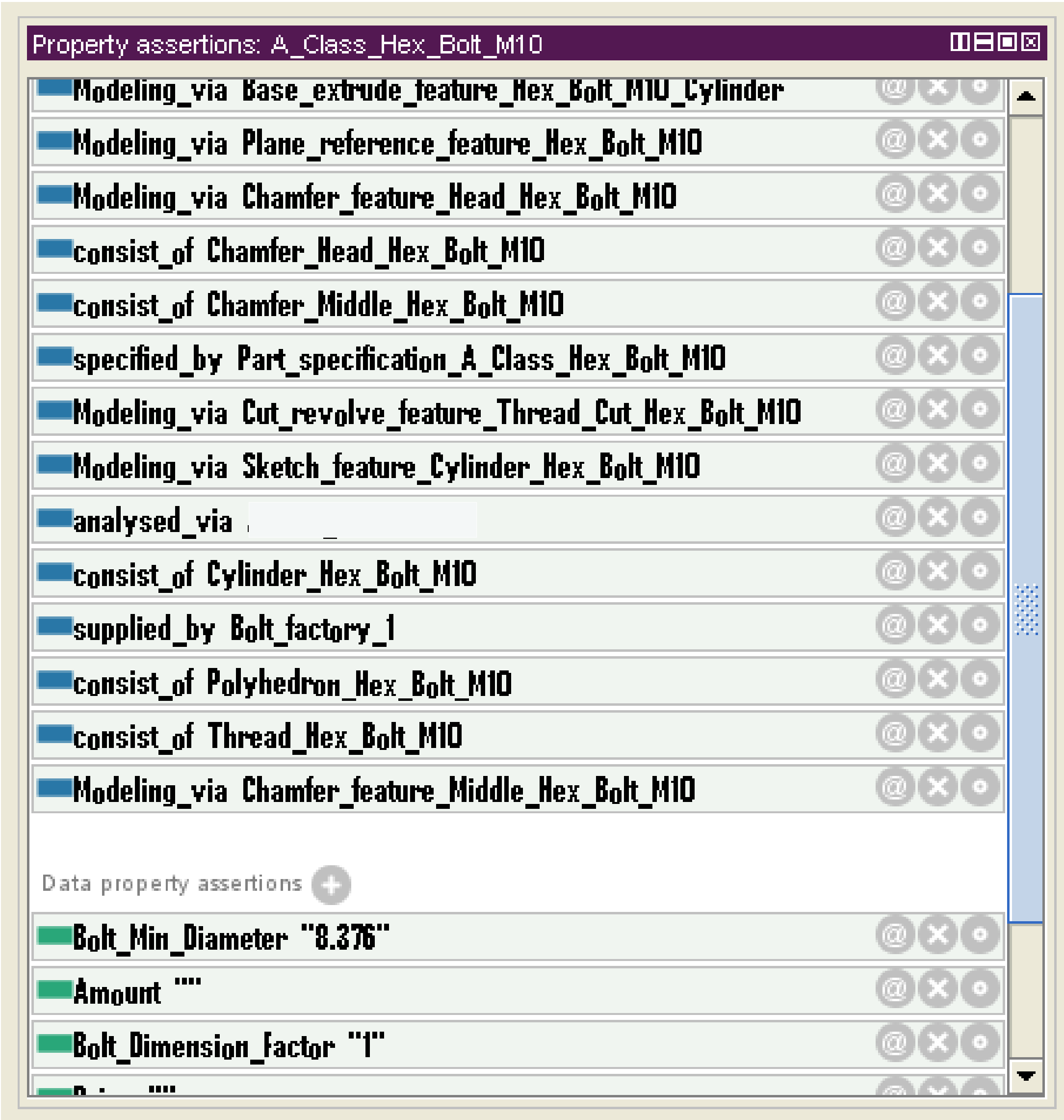


Figure 5.3 Relations among A_Class_Hex_Bolt_M10 within Protégé 4

- **Individuals,** represent objects in the domain. OWL does not use the Unique Name Assumption (UNA) that means two different names could actually refer to the same individual. Figure 5.4 shows a representation of the individuals in the domain.

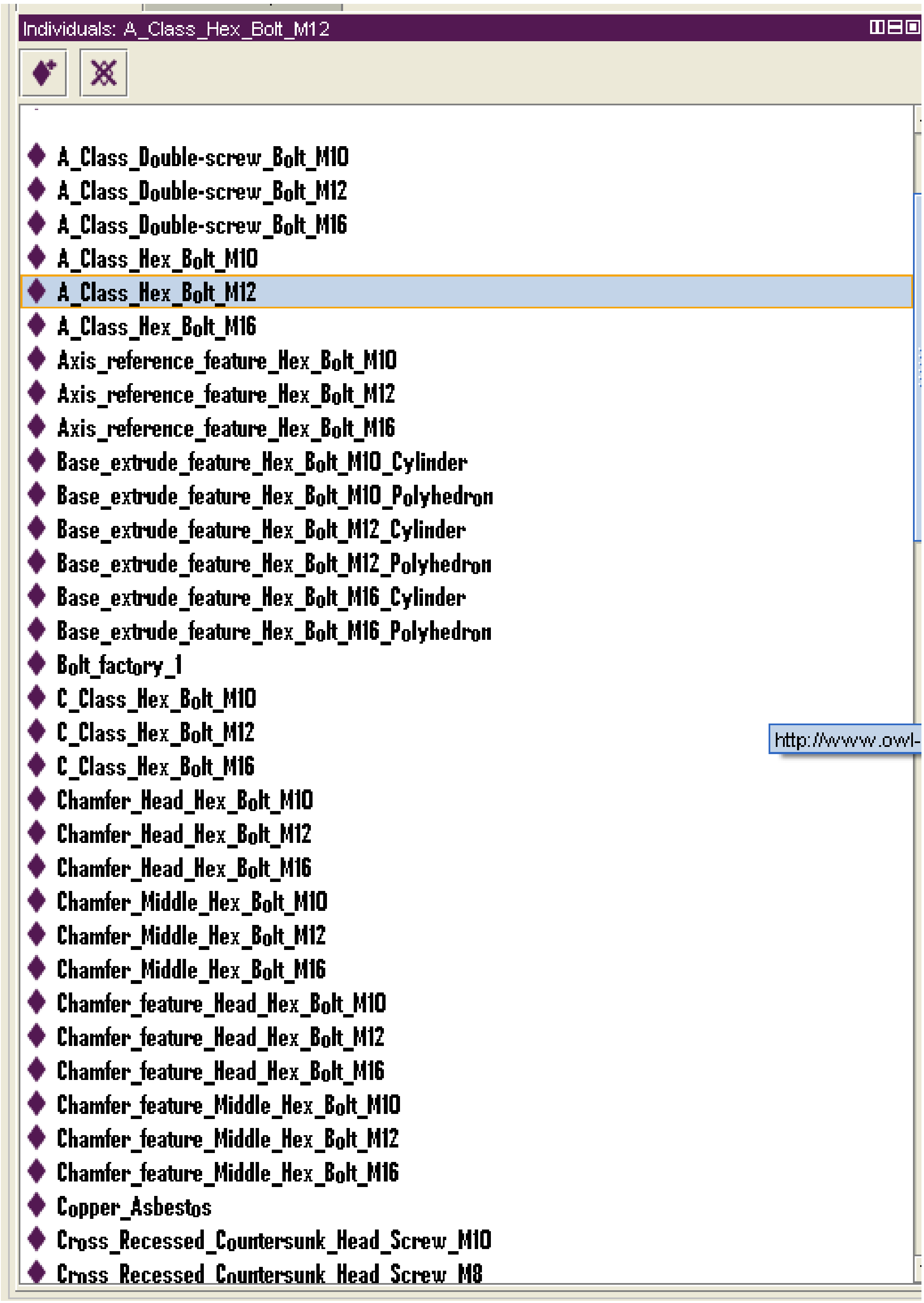


Figure 5.4 Individuals within Protégé 4

5.2.3 Ontology Modelling for Shaft Design

Based on ADMEC previous parametric design research in SolidWorks [6-8], the key concept for the new product design model in OWL is presented in the following way:

- Every concept in SolidWorks is a class. Thus, Cylinder, Keyway, Chamfer are defined as classes.
- Every Class has its instance. Thus, Cylinder has its own instance, for example, in GearBox design, bearing, cover and gear are all the instances of different classes.
- Every instance has its own property. Such as Base, Precision and Array property.
- Based on the OWL relationship, the system also has four relationships, such as part-of, kind-of, instance-of and attribute-of.

In the ontology modelling design, the author has divided the product into different components first, then described the classes and properties of each instance, and created the relations among them together with the domain expert. Below is to show the ontology utilized in the shaft design of Gearbox by OWL. Figure 5.5 is a shaft drawing with specific dimensions to be decided.

Figure 5.5 Specific dimensions of a shaft

The shaft has the following classes, properties and instances:

- Base Classes: Cylinder, Keyway, Chamfer and Fillet.
- Instance: the instance of the Classes, for example, the longest cylinder is the instance of object Cylinder.
    - Cylinder: has their property, such as length, radius, coordinate.
    - Keyway: has their property, such as length, coordinate.
    - Fillet: has their property, such as length, coordinate.
    - Chamfer: has their property, such as radius, coordinate.
    - Base Property: Base coordinate, the keyway centre line and the cylinder centre line.

(Steps of CAD drawing) Established the base coordinate system for the Shaft; took the centre cylinder base frame as a basic coordinate, built the partial coordinate system, which had the relations with each other. Figure 5.6 is the ontology model of the shaft in OWL.

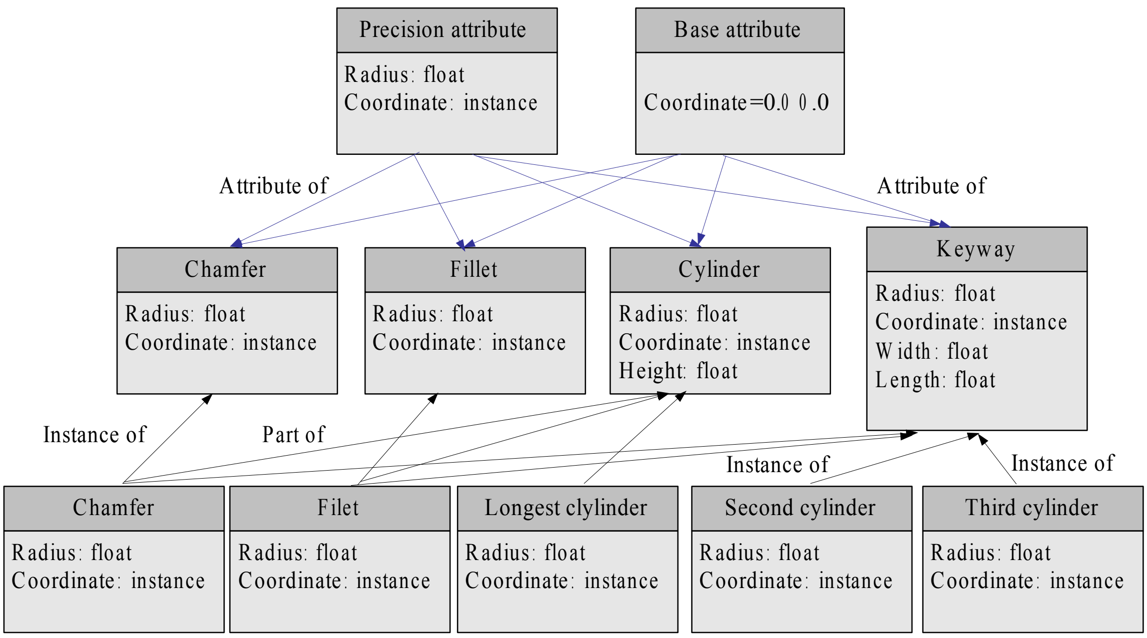


Figure 5.6 Ontology model of Shaft

After mobile system builded OWL Ontology using Protégé 4; Semantic Similarity method will be used to rank the search result and Jena could provide services for model representation, parsing, database persistence, querying and some visualization.

5.3 Method of Semantic Similarity

Semantic similarity approach is introduced in this section to make information system understand the schema and semantics of concepts and to measure the strength of similarity between product data.

5.3.1 Semantic Similarity

With the wide applications of ontology in the field of information retrieval and artificial intelligence, it brings new methods for semantic similarity computation [78, 80 and 81]. By taking advantage of the ISA relationship of concepts in ontology modelling, a new method for computing concept similarity was proposed. This method combined the semantic distance and statistical properties of the ontology. Li [80] presented a method of measuring similarity in general based on the statistic of large sample base, but it did not apply the Semantic Web technology for the input data, all the input data were not recognized as the semantic input and no close relation with the return result; and the algorithm did not care about the root concept or the super classes; in user's experience, if the concepts are separated by the root of ontology tree, these two nodes are farer away; for example, the distance across a root node, the similarity of the two values should be lower. In Figure 5.7, if search system finds HexM15 is the result, then system set Part (n0) to be the root nodes, all the distance value pass the node n0 will be added 1. By adding the mentioned new features and the correction of Li's formula, three principals of the calculation need to be focused on.

- Semantic Similarity is a value, generally in the range [0, 1], if two concepts has the same meaning, then the value should be 1, and if no path connectivity within the two concepts, the value is 0;

- The algorithm of the calculation should not be complex, the reason is not only for the huge data of current information system, but the return result should be sent back to the mobile device, the processing time have to be limited;

- The calculation of similarity should be adjustable, concept similarity is a very strong subjective, the concept of the application of different terms in different similarity could be modified, therefore the similar results could meet the needs of different systems.

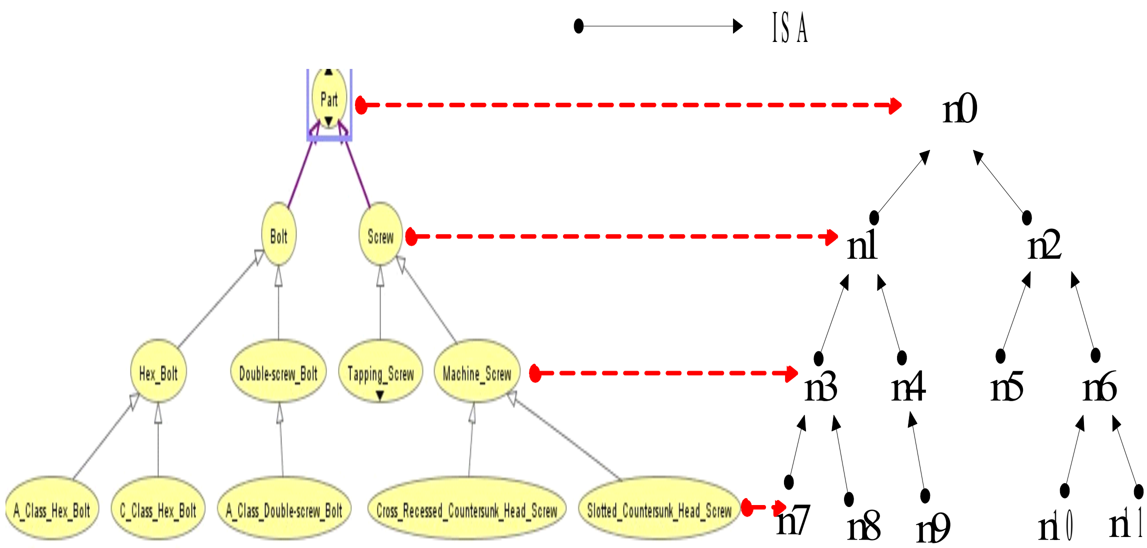


Figure 5.7 Abstraction of nodes hierarchy tree

According to the basic principles of ontology-based similarity calculation shown above, four factors should be considered:

- Semantic distance: the shortest path by the number of inter-edge between two concepts in the ontology graph, this thesis use Distant (A, B) to express the concept of A and B of the semantic distance. Semantic distance is an essential element to determine the similarity. In general, the bigger value of semantic distance means the lower of the similarity. This relationship needs to meet the following conditions: 1) A node with itself, the value are set to 1; 2) two nodes for the infinite distance, the similarity will be set to 0; 3) The longer the distance between the two nodes, the smaller of the similarity; 4) if the two nodes are cross root nodes, the distance should be added 1, in Figure 5.7, there exist four root nodes: Role, Document, Requirement and Product; thus, Distant (n2, n3) = 4, because n0 is the root node.

- Number of same parent nodes (include itself): count how many same parent nodes of two concepts, it can be translated into the number of public nodes, the more numbers of parent nodes, the less similarly between two concepts. System use a (x) represent x node back to root nodes. For example, in Figure 5.7, a (n3)=3, a (n7) = 4, a (n3)∩ (n7) =3.

- Level of depth: if the concepts have the same distance, and if the depth of the two nodes in the ontology tree increased, the value of the semantic similarly will increase. That is easy to understand, for example, Gearbox and Electric Motor are in the same level in this system, as well as Bolt and Shaft; but Bolt and Shaft should be more similarly in real world. For example, in Figure 5.7, node similarity of (n7, n3) is bigger than (n7, n1).

5.3.2 Similarity Calculation

Mobile system gets the following formula via taking into account the above four factors: $Sim'(x,y)=a(x)\cap a(y)/Distant(x,y)*(p*|h1-h2|+1)$

where $a(x)\cap a(y)$ is the number of parent nodes, distant(x,y) is the semantic distance of two nodes, $|h1-h2|$means the depth level of two nodes and p means the adjust factors. Adjustment factor is based on the design needs, because the concept similarity is a very strong subjective, in this thesis, system used expressed adjustable factors. In Figure 5.7,

- when p = l the value Sim (n3, n1) = Sim (n3, n4);
- when p> l, Sim (n3,n1) <Sim (n3, n4);
- when p <l , Sim(n3,n1)> Sim (n3, n4).

In general, in this mobile system p = 1.After the author gets the formula, the last step is to set the value range from [0,1], therefore system get the final formula is that: $Sim(x,y)=1-1/\mu^{sim'(x,y)}$ $(\mu=10)$

Tab 5.1 is the Semantic Similarity results about Figure 5.7, For example, if the user input the key word for the node n3 (which is "Hex_bolt"), and the system set threshold for 0.95, then the results are (n3,n7, n8), all the three node/classes have their instances, and the return data will present the more closer results; but if the threshold is set to 0.8, the result should be (nl, n3, n4, n7, n8), that means some results maybe less related to the input concepts. Compared with the different SS value between input and results, the new semantic similarity calculation approach is utilized to provide a straightforward and efficient method for rank the query input and the return results sorted by the value of SS.

	n0	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10	n11
n0	1	0.6838	0.6838	0.3187	0.3187	0.3187	0.3187	0.1746	0.1746	0.1746	0.1746	0.1746
n1	0.6838	1	0.5358	0.9	0.9	0.2501	0.2501	0.5358	0.5358	0.5358	0.1423	0.1423
n2	0.6838	0.5358	1	0.2501	0.2501	0.9	0.9	0.1423	0.1423	0.1423	0.5358	0.5358
n3	0.3187	0.9	0.2501	1	0.9	0.3690	0.3690	0.9684	0.9684	0.5358	0.1746	0.1746
n4	0.3187	0.9	0.2501	0.9	1	0.3690	0.3690	0.5358	0.5358	0.9684	0.1746	0.1746
n5	0.3187	0.2501	0.9	0.3690	0.3690	1	0.9	0.1746	0.1746	0.1746	0.5358	0.5358
n6	0.3187	0.2501	0.9	0.3690	0.3690	0.9	1	0.1746	0.1746	0.1746	0.9684	0.9684
n7	0.1746	0.5358	0.1423	0.9684	0.5358	0.1746	0.1746	1	0.9684	0.6838	0.2803	0.2803
n8	0.1746	0.5358	0.1423	0.9684	0.5358	0.1746	0.1746	0.9684	1	0.6838	0.2803	0.2803
n9	0.1746	0.5358	0.1423	0.5358	0.9684	0.1746	0.1746	0.6838	0.6838	1	0.2803	0.2803
n10	0.1746	0.1423	0.5358	0.1746	0.1746	0.5358	0.9684	0.2803	0.2803	0.2803	1	0.9684

Tab 5.1 Semantic Similarity results

5.4 Through Jena API to Retrieve Semantic Data

Jena is an open source Semantic Web framework for Java. It provides an API to extract data from and write to RDF graphs. Mobile system uses Jena API to do the search and retrieve the data in the server side. In Jena, ontology is treated as a special type of RDF model, OntModel. This interface allows the ontology to be manipulated programmatically, with convenience methods to create classes, property restrictions, and so forth. An alternative approach is to treat the ontology as a regular RDF model, and simply add statements defining its semantic rules. Note that it's also possible to add ontological statements to an existing data model, or merge an ontology model with a data model using Model.union(). SPARQL, an RD F query language, support in Jena is currently available via a module called ARQ (A SPARQL Processor for Jena) [72]. Jena also provides several Reasoner types to work with different types of ontology, and this mobile system use OWLReasoner.

In the Figure 5.8, Jena2 inference subsystem is designed to allow a range of inference engines or reasoner to be plugged into Jena. These engines are used to derive additional RDF assertions which are entailed from some base RDF together with any optional ontology information and the axioms and rules associated with the reasoner. The primary use of this mechanism is to support the use of languages such as RDFS and OWL which allow additional facts to be inferred from instance data and class descriptions.

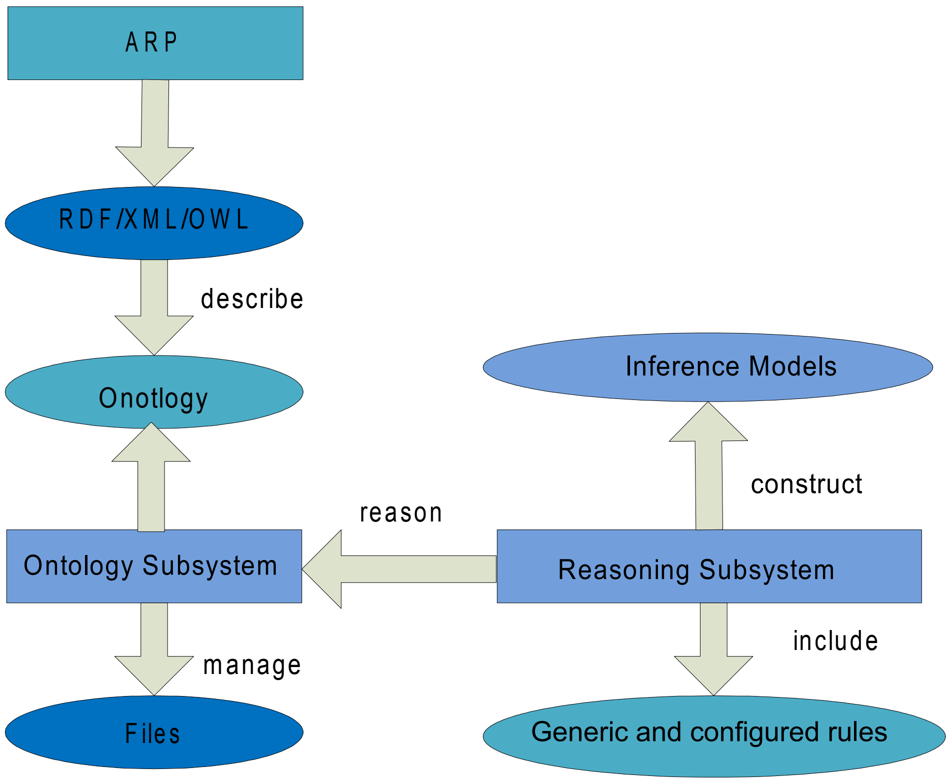


Figure 5.8 Overall structure of the inference machinery

Mobile system can create an instance of the OWL reasoner, specialized to the test schema and then apply that to the test data to obtain an inference model, as follows:

```
…
Reasoner reasoner = ReasonerRegistry.getOWLReasoner();
reasoner = reasoner.bindSchema(schema);
InfModel infmodel = ModelFactory.createInfModel(reasoner, data);
…
```

A typical example operation on this model would be to find out all the specific instances, this can be done using:

```
…
for (StmtIterator i = m.listStatements(s,p,o); i.hasNext(); ) {
      Statement stmt = i.nextStatement();
      System.out.println(" - " + PrintUtil.print(stmt));
…
```

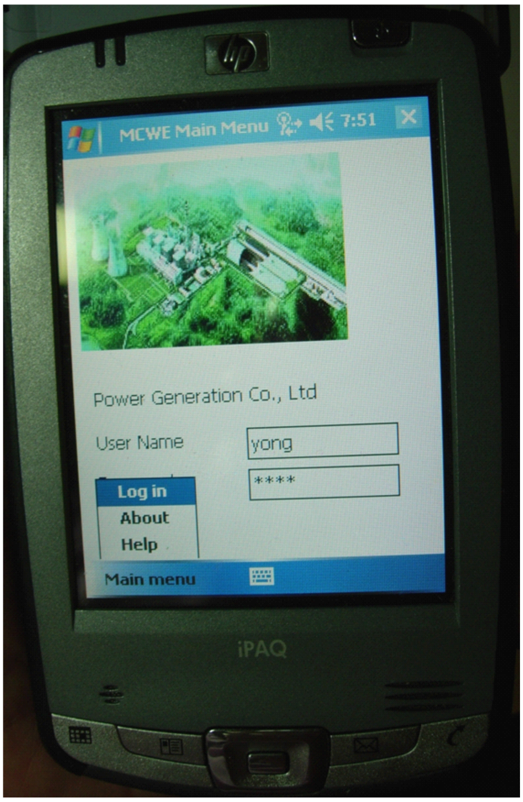
In the mobile system, if the mobile user wants to know the details about products of the same supplier, they could add the new ontology rule as shown below:

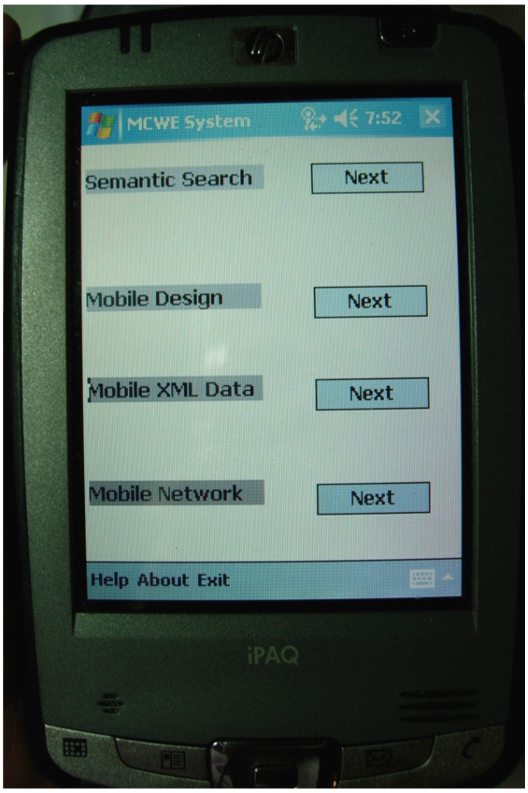
```
[bothSupplied: (?b http://www.domain2.com #Supplied ?a),
(?c http://www.domain2.com #Supplied ?a),notEqual(?b, ?c)->(?b http://
www.domain2.com #bothSupplied ?c)]
```

Figure 5.9 Jena ontology rule: products of same supplier

5.5 Case Study

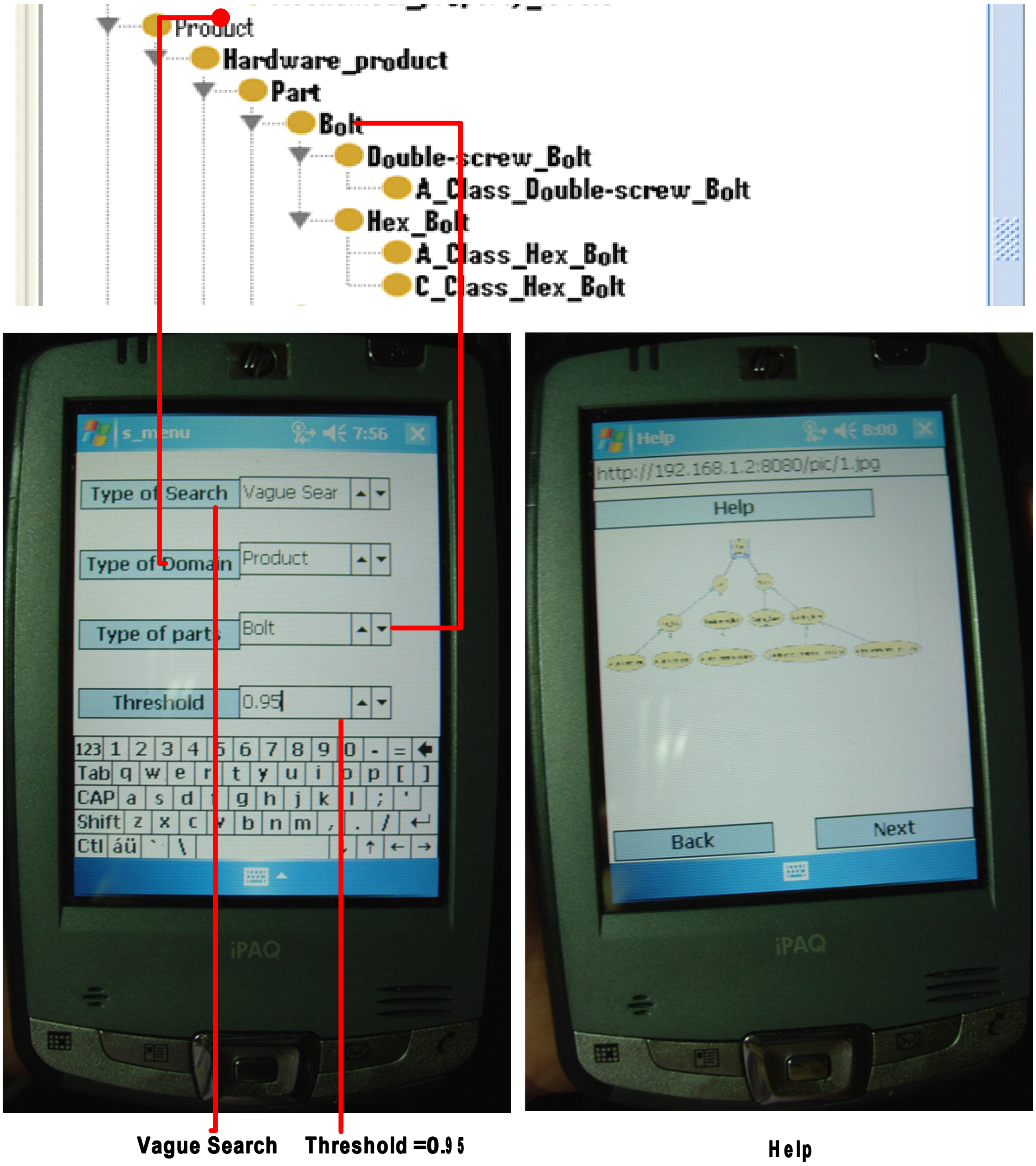
A semantic information system has been developed which accommodates semantic relationships and similarity of product data; and a case study in information retrieval is presented in this section which is for the Hunan Power Station, China. This Semantic Information System is based on the previous services platform that provides flexible node recommendations under given conditions in terms of user profile, context, and privacy concern preferences [7]. Information systems can automatically identify user input based on the ontology database in the Web server. A mobile user interface is implemented by VB.Net application. The user and service ontology are represented with OWL and Jena. To support the semantic similarity calculation between user's input and a specific service result, a Web service called SemanticService is implemented. Figure 5.10 Shows snapshots of an example of searching the Bolt information. In this example, the client inputs the relative words for pointing the Bolt; the related program function searches the ontology library on the Web and displays the results in the mobile device, the search procedure is shown below: (a) and (b) are the "login in" and "main module" interface of the mobile application:





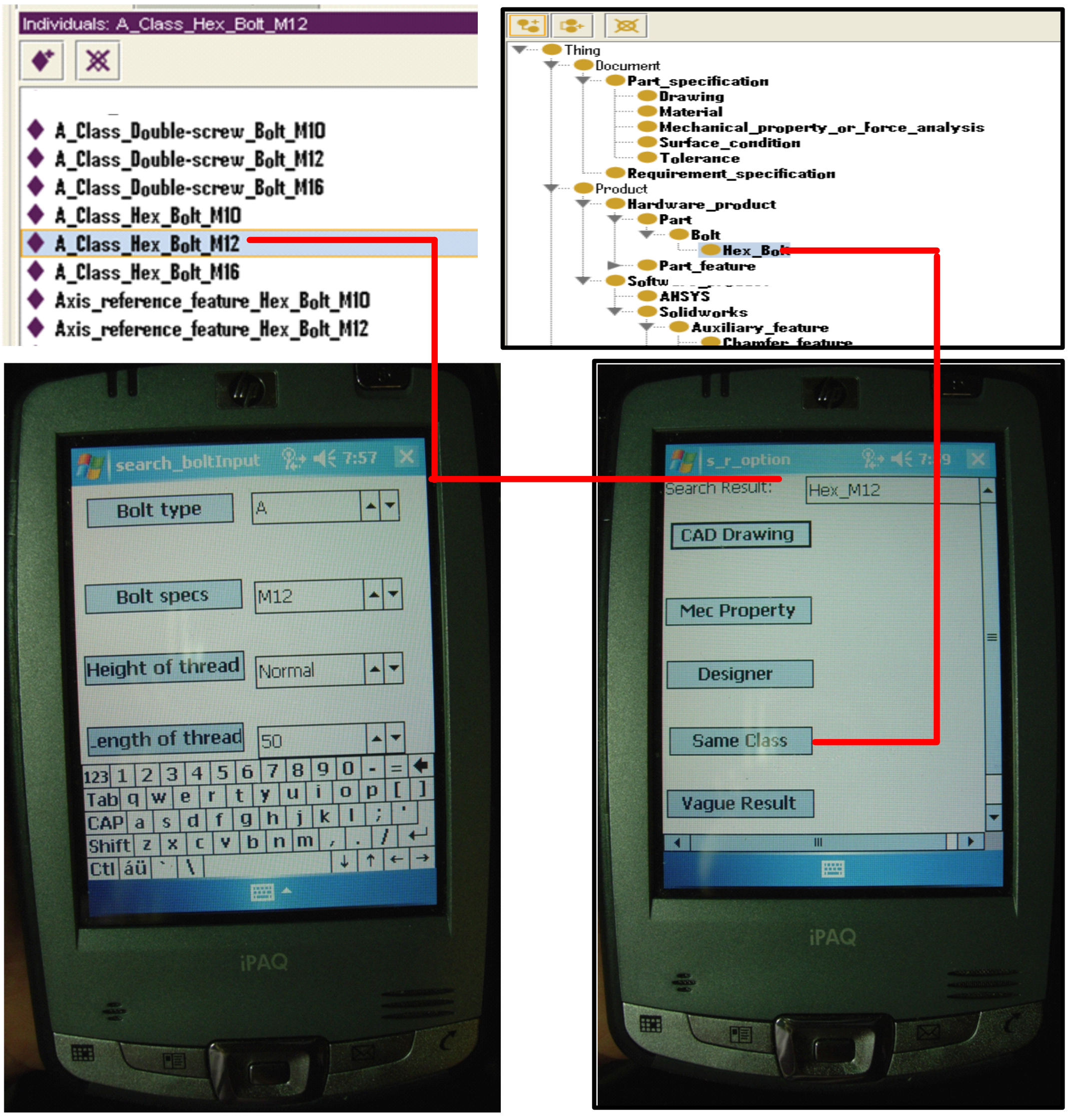
(a) (b)

In this example, a bolt M12 for the electromotor is required, and the user selects the options of Vague Search as shown in (c) and (d), all the input options are decided via the ontology tree in Figure 5.1.



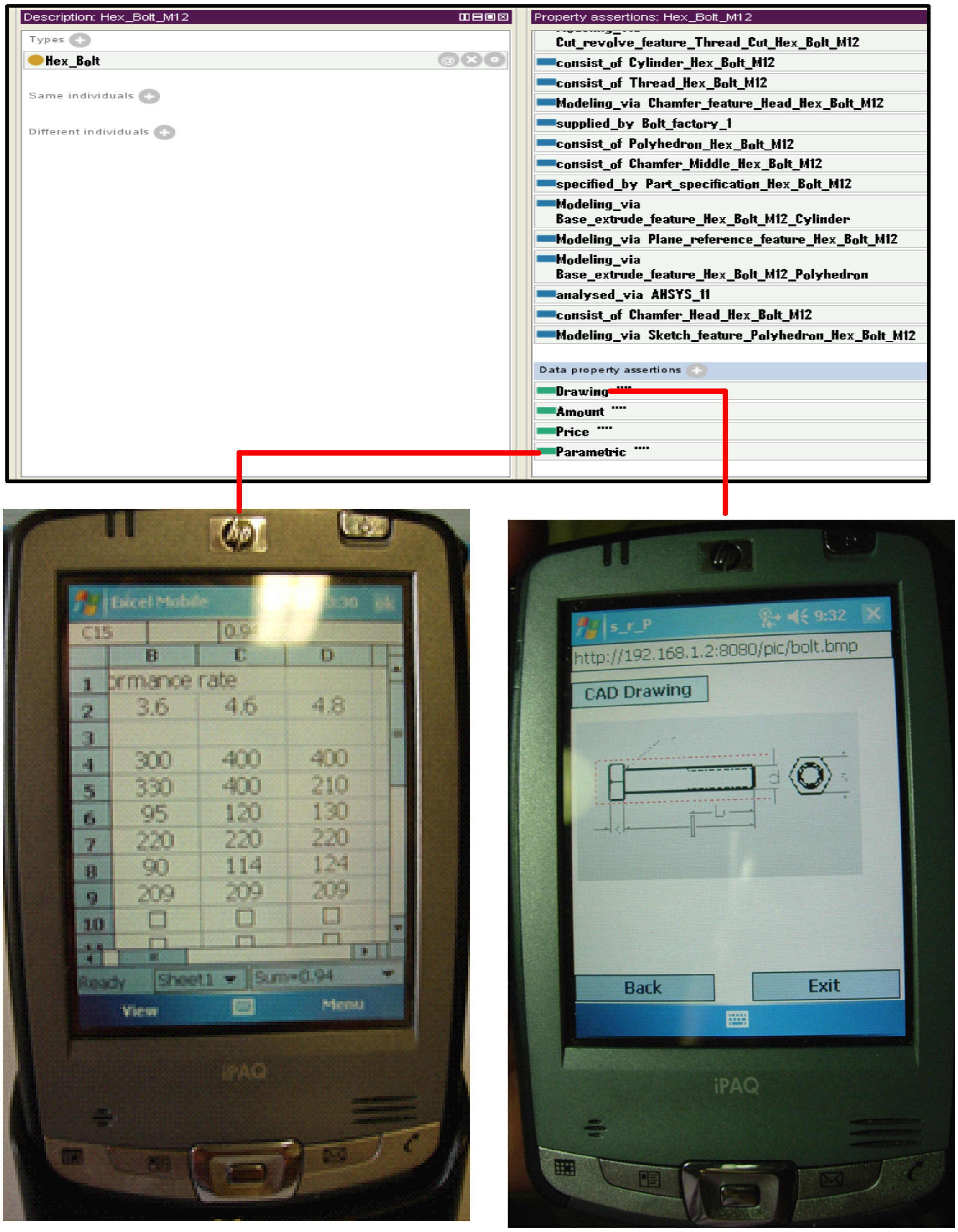
(c) (d)

After the mobile device sends the input information (e) to the Web server, Client will get the result back shown in (f).



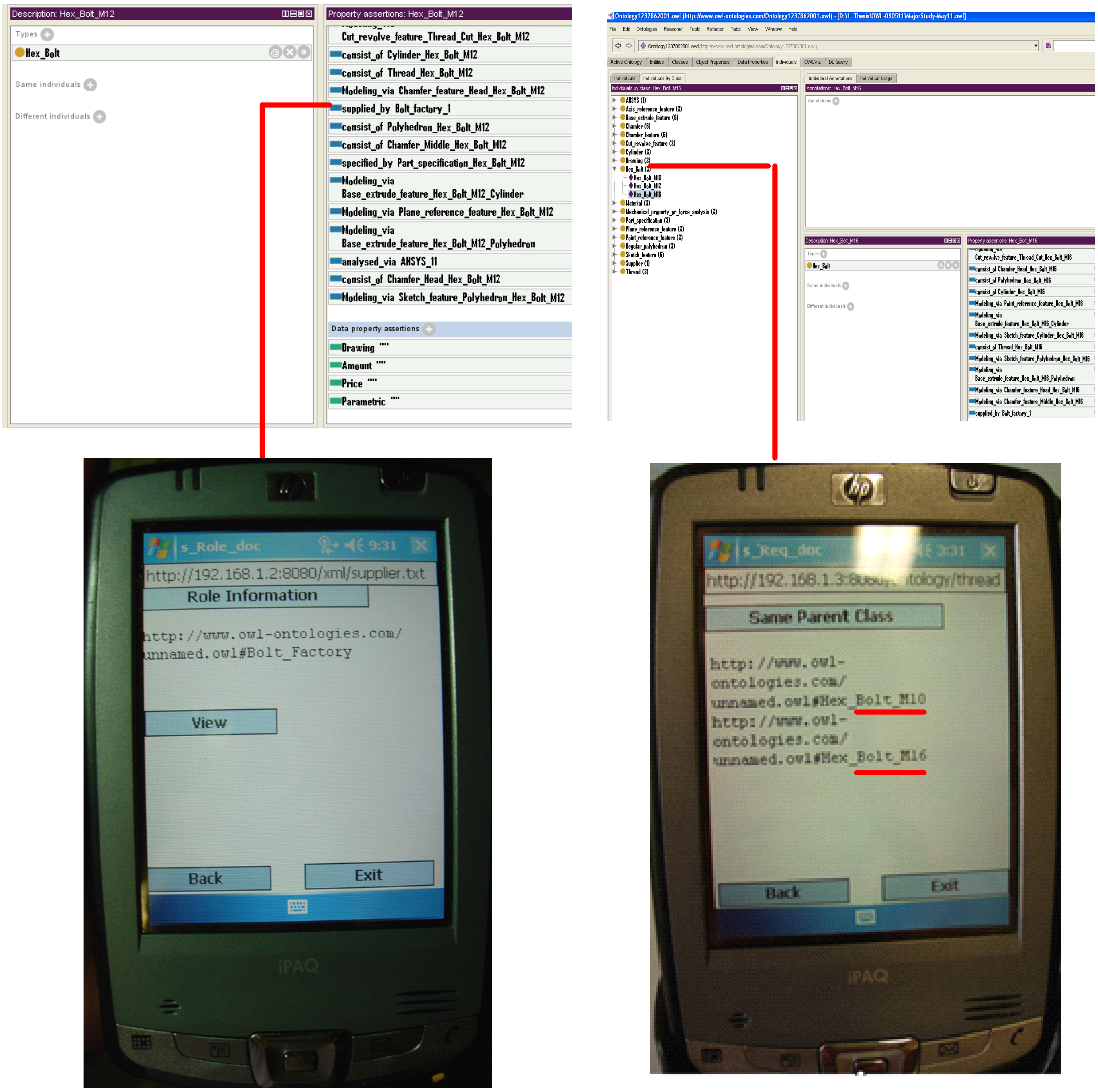
(e) (f)

If the user choose to view the HexM12, (h) and (i) are the parametric data and drawing information return by the Web Service, which will use Jena API to search the related Data property;



(h) (i)

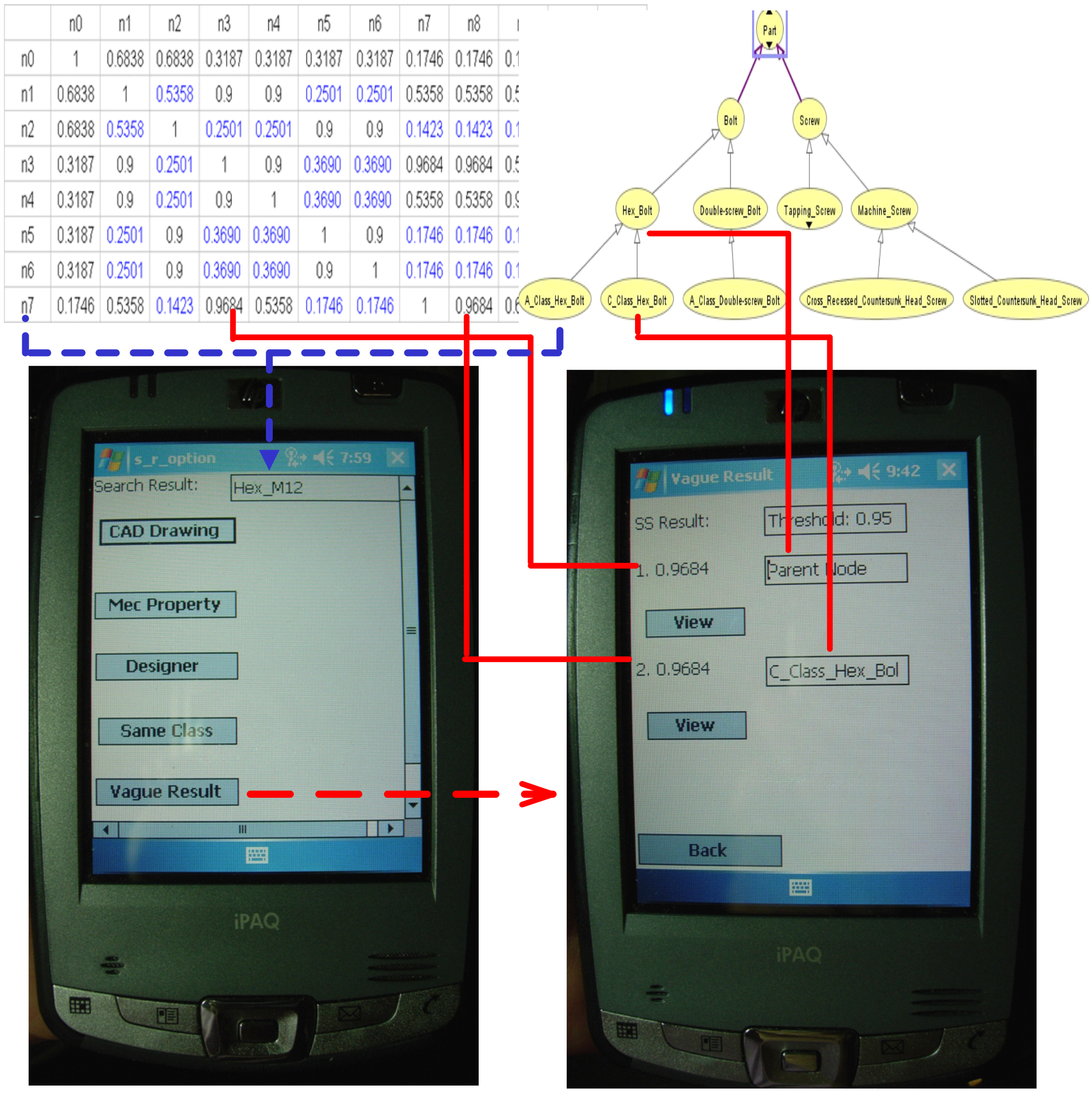
(j) is the designer information, which is the suppler name of Bolt M12, (k) is shown the other products which is from the same supplier, such is HexM10 and HexM16.



(j) (k)

The ontology rule code for retrieving other products of the same supplier could be referred to Figure 5.9 “Jena ontology rule: products of same supplier”.

If the mobile user sets the threshold 0.95, because in (f), the first result is A_Class_Hex_Bolt, that is the node n7 in the figure 5.7, and then system will check all SS values related to n7. As shown in table 5.1, system could find that the node n3 and n8 meet the conditions, because the two SS values between (n7, n3) and (n7, n8) are bigger than 0.95, that means the two nodes are the most related product data for this search result; After that, for example, if the mobile user wants to view the information of n8, the remote user could click the option link in (m).



(l) (m)

Figure 5.10 Example sequence of retrieving product data

If the mobile user do not choose Vague option, the mobile system will use Jena2 to express queries across diverse data sources, whether the data is stored natively as

OWL file or viewed as OWL file via middleware.

5.6 Conclusions

Semantic Web is an extension of current web in which information is given a well-defined meaning, which can better enable computers and people to work in harmony. It allows the data to be defined and linked in such a way that it can be used by machines not just for display purposes but also for automation, integration and data reuse. Semantic Web technologies provide intelligent access to heterogeneous distributed information, enabling software applications to mediate between user needs and information sources. This research presents semantic mobile information system and the measure method of semantic similarity, which is proposed by taking advantage of the ISA relationship of concepts in ontology, the experimental results proved that the method is efficient and reliable. Further job about this semantic information system is that, in order to re-use the basic ontology, system need to separate rules and constraints from the basic ontology. This ensures that the basic ontology can be used more widely, and makes the information applicable to special purposes more flexible.

Chapter 6

The Mobile Agent for Collaborative Product Design

In chapter 4 and 5, the thesis describes Mobile CAD Collaborative Design and Semantic Information System to support the exchanging, integration and sharing of product data over the internet. But it is difficult for the mobile users to share and retrieve product data across different Web servers, the reason could be the copyright of CAD packages, security of data or different Platform of Web servers. In order to resolve this problem, this chapter introduces Agent technology into WSO-MCAD and presents a multiagent-based system that adopts intelligent agents as a technology for tasks distribution and resource management in distributed systems.

6.1 Introduction

Product design is often geographically distributed, requiring collaboration and effective communication among the companies involved in the systematic definition of a new product [3]. WSO-MCAD needs a common technology where the remote users can share and retrieve information among distributed resources [7]. For example, if the Bolt Company provides a Web service which could process the input parametric data and return the CAD design information, but because of security, this Web service can only be invoked by the local server. Using mobile agents, which are able to travel from one node to another and accomplish their task, has become an important alternative to conventional approaches; the second advantage of Agent technology is that, this approach could resolve the network connectivity, which largely affects the time and size of data downloading. Due to mobile agent's migratory characteristics, it can help system to reduce network traffic, overcome network latency; and enhance the robustness and fault-tolerance capabilities of distributed applications [82].

An agent generally means a software entity that is situated in a certain environment and is capable of performing autonomous actions in the environment to meet its goal [83]. In contrast to the common stationary agents, mobile agents provide a better choice to support computation in distributed network environments, the embedded mobile agent could provide a transparent mobile service to operational system, can transport its state from one environment to another, with its data intact, and still be able to perform appropriately in the new environment.

Agent technology has been recognized as a promising paradigm for next generation manufacturing systems [84] and attracted researcher's great attention. As the mobile applications are generally operated in an open environment in which resources and repositories are distributed in different machines and the network connection is unreliable, it is essential to make sure that the infrastructure must comply with the established standards, especially for the multi-agent platform [85]. In this research, system choose JADE (Java Agent DEvelopment Framework) toolkit to develop applications, JADE is compliant with Foundation for Intelligent Physical Agents (FIPA) standard specifications so that agents developed on it can interoperate with other agents built with the same standard [86].

6.2 Overview of Multi Agent System

6.2.1 Overview of Agent System

With the development of ADMEC previous research in Collaborative Working Environments framework [6], agent technology is specified in the middleware layer.

Author used JADE as a platform and built the agents on it, the agent system mainly includes different types of agents for information collection, in this mobile collaborative system, different agents communicate with each other in a pre-specified language through a common channel to ensure the interoperability between agents.

- Parametric Agent (PA): Parametric Agent is created in the register centre, get the request information from Request Agent and send the service information back. PA transfer request information and send the according document to the requestor, PA could move into another host or remote Web server.

- Requestor Agent (RA): RA run in the client side and does not need to move in the network. RA has a XML documents which describe the user's demand.

- Server Agent: SA has a knowledge library, contains all necessary data of the product design; SA provides the data documents and sends them to the requestor via the PA. Its main goal is to receive message from PA, withdraw the design information and return the ACL (Agent Communication Language) message to PA back, after that, PA could send the messages to the end user.

- Register Agent: The register Agent is responsible for each PA and assigned a unique ID. SA preservation provides the product data documents, and submits it to the register centre; if CA needs the data, it can inquire the Register Agent, obtains the list of suitable SA.

In this multi-agent system, the information for each design data is encoded in XML format, SA is designed to reside in the server side to retrieve information from the repository in response to other agent' inquiries. With the XML specifications, it is dramatically easier to develop and deploy domain and mission specific internet applications. XML allows authors to specify their own document syntax, hypertext link semantics, and presentation styles. It also allows customization of domain specific tags of the markup language. Information providers can create new tags and elements with new attribute-value metadata to describe their services. The following sub sections will address the agent communication and migration in term for deploying personalized mobile applications in a multi-agent environment.

6.2.2 Agent Communication

A fundamental characteristic of multi-agent systems is that individual agents can communicate and interact with each other, which is accomplished through the exchange of messages, it is crucial that agents should agree on the format and semantics of the message. The Foundation for Intelligent Physical Agents [87] founded in 1996, is an international standardization organization promoting the development and specification of agent technologies. FIPA promotes agent-based technology and the interoperability of its standards with other technologies.

JADE follows FIPA standards, so that ideally JADE agents could interact with agents written in other languages and running on other platforms. There are many auxiliary parts to a message in addition to the content, for example: the intended recipients, the sender and the message type. It is essential for the message as a whole to respect a common format. In JADE, messages adhere strictly to the ACL standard which allows several possibilities for the encoding of the content [87].

6.3 Agent mobility

JADE platform supports agent mobility which shown in Figure 6.1, it exploits Java serialization API and dynamic class loading to move or clone JADE agent over different containers but within the Java environment. The support for mobility in JADE consists of a set of API classes and methods that allow an agent to perform the required actions by it or via the *Agent Management System*, and a mobility specific ontology (MobilityOntology), which is contained in the package *jade.domain.mobility*.

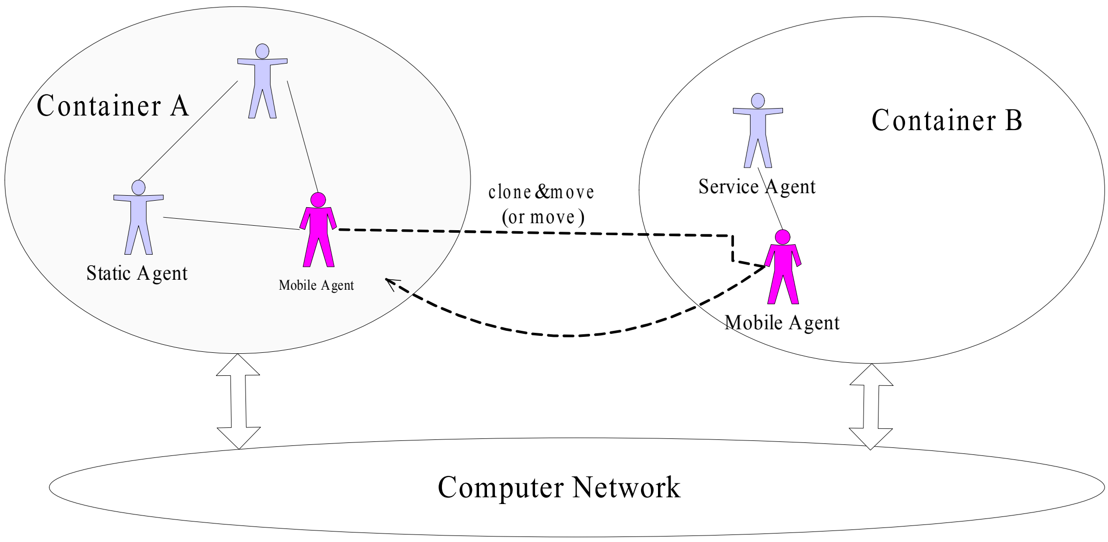


Figure 6.1: Mobile agent mobility

With these unique characteristics, mobile agent-based system provides a single framework and a convenient abstraction (i.e., the mobile agent) to build distributed applications more efficiently. Within the agent system, the mobile agent actually migrates to the server to make a request directly, rather than over the network. When the entire transaction is complete, the mobile agent will send the results back. Agent mobility is the ability for an agent program to migrate or to make a copy (clone) itself across one or multiple network hosts. At present, mobile computation relies on expensive and frail network connection, and, hence, those features become more and more valuable [46].

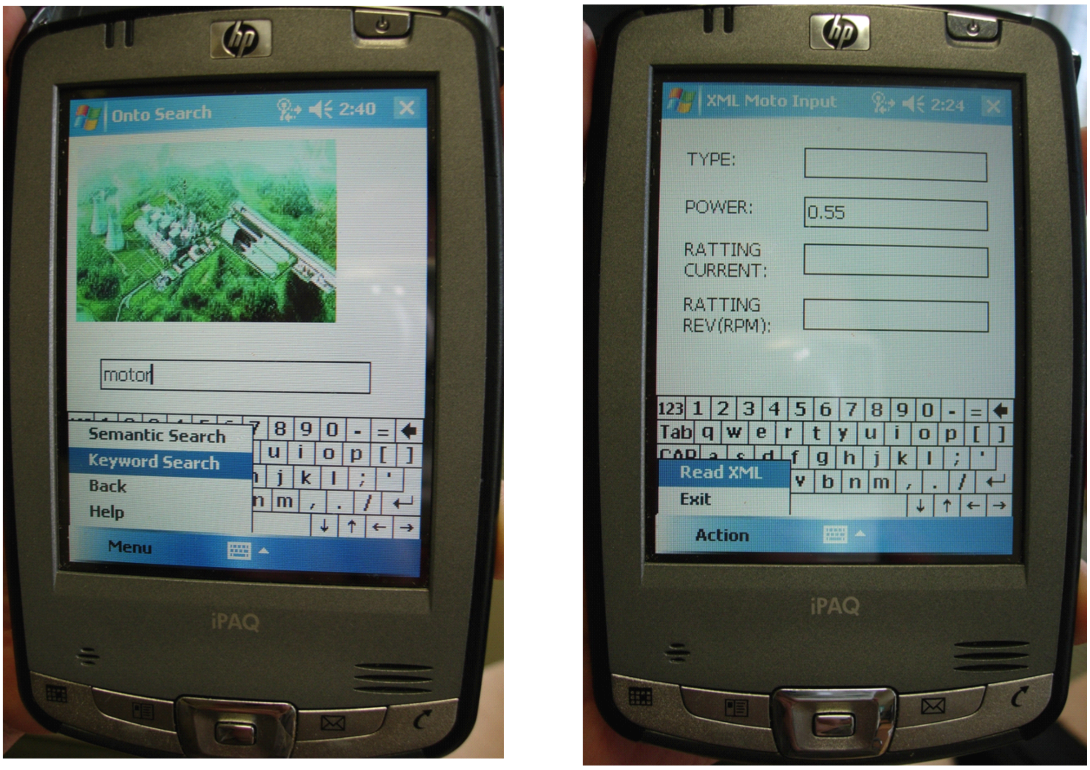
6.4 Experiment

A prototype agent system will be developed for the online collaborative design based on current research. With the SolidWorks 3D parametric features, mobile user can open the drawings of parts or assemblies, and make changes via the mobile device.

The administrator of Web server maintains the software residing on the server, and hence, there is no need to reissue updated software to users each time when a change is made. The underlying software is located on the company's server and can only be accessed /invoked via the local user/agents, a main JADE platform can be distributed across machines or different company's Web servers. CAD parametric design application will be invoked and the result will be sent back by the local agent, the Parametric Agent could be the local agent or immigrate from other company's Web server.

Figure 6.2 is mobile application interface, for example, if the electromotor data is required to be searched, mobile user input relative words for pointing the motor, system is going to search the relative information on the web and display them in the mobile device.

For example, if the mobile user connects to the Power Station's Web server, this server did not provide Electronic motor information, Mobile Agent from Power Station will immigrate to the Web server of electromotor factory; and the mobile Agent will send the result back. The search system is displayed as Figure 6.2 and 6.3. For instance, after the mobile user input the search information in Figure 6.2 (b), the system will automatically seek the relative information and illustrate them.



(a) **(b)**

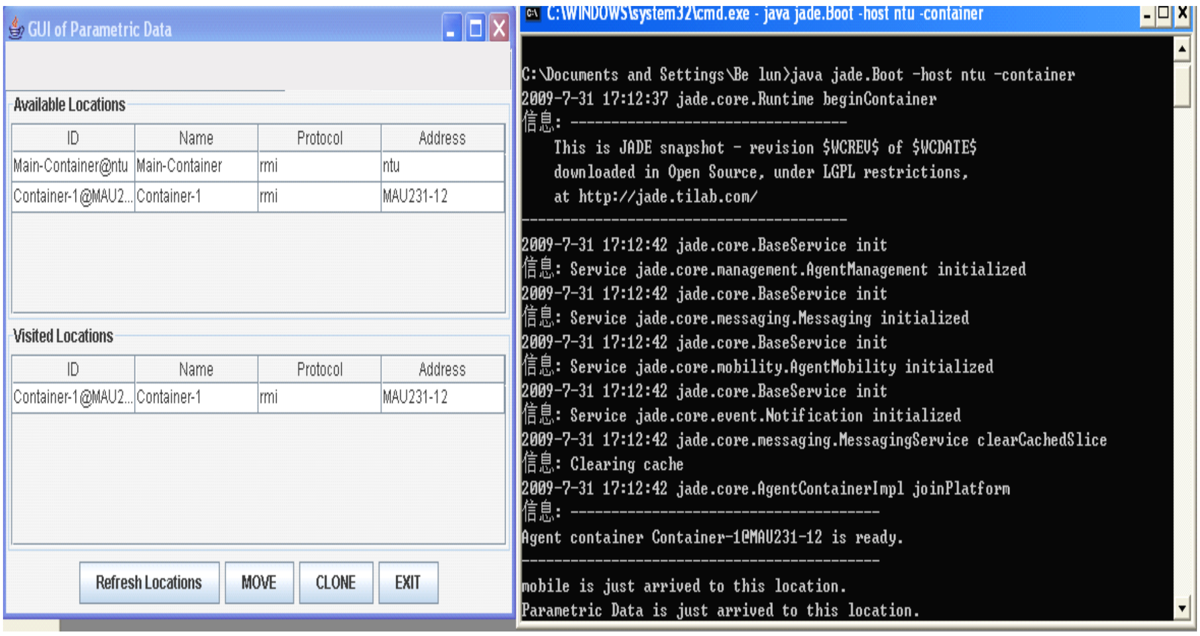
Figure 6.2 Mobile search input interface

Figure 6.3 illustrations of the outcomes

Simulation of Mobile Agent operation process is shown in Figure 6.4: there are two Web servers in two computers:

- One computer name is MAU231-12, the Agent container name is "container-1" , shown in Figure 6.4 (a) mobile user could send the input data to this Web server via Web service, but this Web server does not provide the related result, because of copyright license or security reason;

- Another computer name is NTU, there are Main container of Agent system in Figure 6.5 (b, c), and its Web server installed SolidWorks/CAD package, local agent could invoke SolidWorks/CAD and return CAD files or other useful product data back. Mobile Agent could immigrate from MAU231-12 to NUT, and obtain the product data back.



(a)

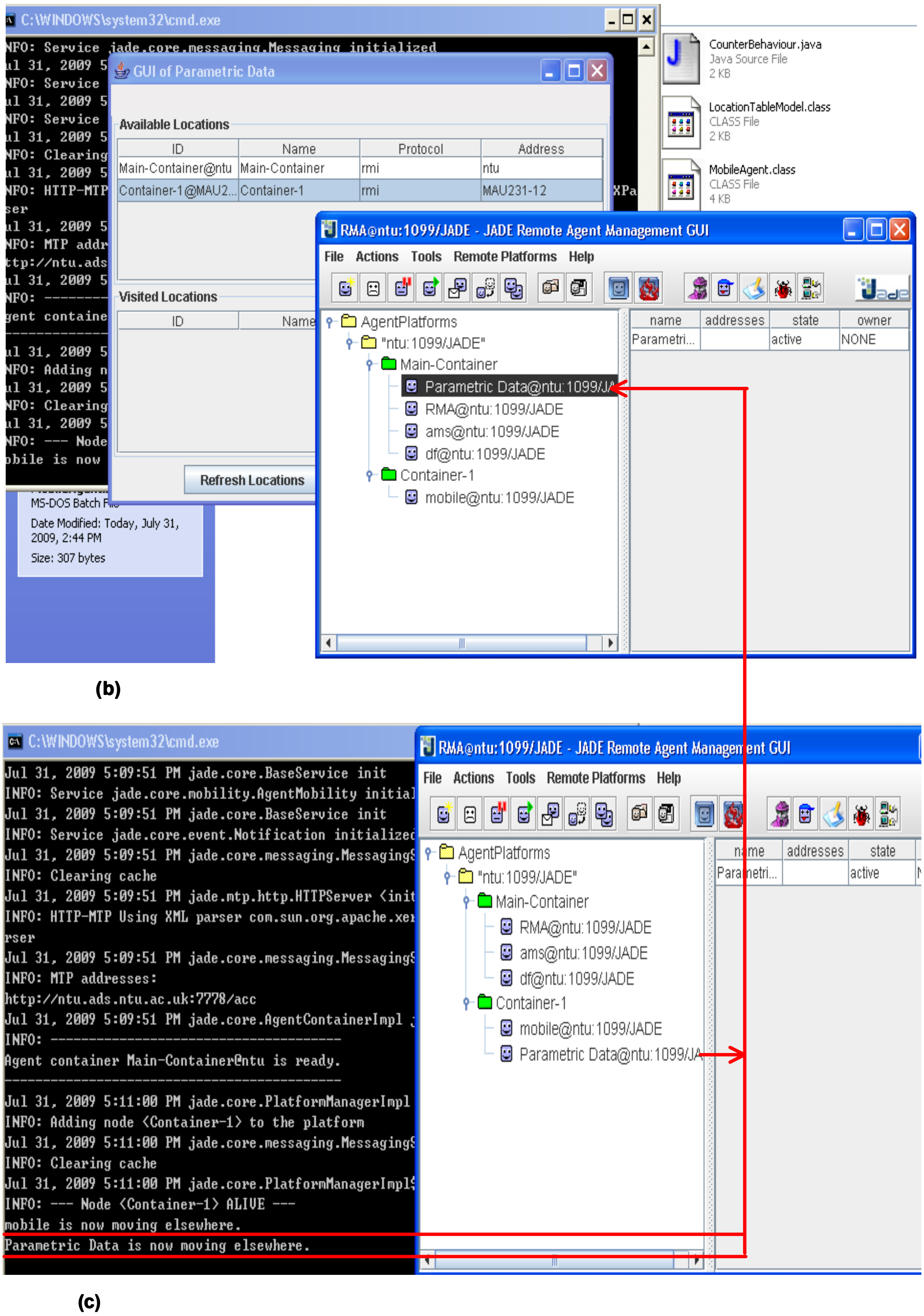


Figure 6.4 Mobile Agent immigrate within WSO-MCAD

Within the process, after the mobile client sends the parametric data to the Web server and activates the parametric agent in (a); but in this Web server (computer MAU231-12, container-1), there do not provide the Web service which performs the parametric design for CAD drawing; then the parametric agent could search in the JADE Agent platform and migrate to the main container, NTU, to carry on its task. (c) is the screen-shot before the migration of the agent, and (b) shows that the parametric agent has migrated from container-1 to the Main-Container, and the CAD package will be invoked to produce the CAD drawing files, therefore return data could be sent back to contaner-1 by Java Remote Method Invocation method.

6.5 Conclusions

Agent system for collaborative design, which is developed based on JADE, supports the construction of mobile agents or mobile agent-oriented systems with the dynamic loading class for various applications. The presented results are based on a design example of an application operating in a mobile environment there need a common framework where different systems can share their heuristic information on available resources. Future work about combining Agent technology with Semantic Web technology will be described in chapter 9.

Chapter 7

Mobile Platform Compatibility: The Flex Applications within the MCWE

With the work presented in the previous chapters, the mobile applications are developed using Microsoft VB.Net for the remote users to retrieve and share the product data by the Windows Mobile devices. However, many mobile devices have different Mobile Operation Systems and VB.Net application could not be well supported by these devices. This chapter is going to resolve the Mobile Platform Compatibility problem, Flex mobile application could be implemented across different mobile platforms; and Web service technology could expose Flex mobile applications to a set of loosely coupled services for the purpose of providing a fast solution.

7.1 Introduction

In order to develop and deploy mobile applications in different mobile platforms, this research chooses Flex builder 3 to develop the cross-platform applications, which can run across most of current popular Mobile OSs, such as Symbian OS, Windows Mobile, Mobile Linux, etc. The reason of the selection and the details of this Technology has been described in chapter 2 already. Mobile Rich Internet Applications into adobe Flex is the cross-browser and platform compatibility, Flex mobile application, which is designed to deliver graphics and animation over the Wireless Internet, has the following advantages:

- **Mobile Platform Compatibility:** Flex applications operate in all major Mobile browsers, leveraging latest mobile browsers technology and Adobe AIR on the Mobile Devices. Flex makes it easy to add rich, interactive functionality to existing applications [88]. A Mobile user could start the Flex application by Mobile Web browser. The Web page typically contains the SWF file created with

Flex builder. Alternately, the mobile device could download stand-alone executable applications from Web server, SWF file will be executed on the mobile user's devices in the Flash Player.

- **Network Delivery:** Flex application can be delivered over a network with limited and unpredictable bandwidth. The files are compressed to be small and support incremental rendering through streaming. Flex application uses Web service to send and load data to and from the server-side components without requiring the client to reload the SWF again, that method could reduce the network data transfer size and time between mobile user and the Web server, and avoid Network Latency problem.

- **File Independence:** Flex application can be displayed without any dependence on external resources, such as fonts.

- **Scalability:** Different computers have different monitors resolutions and bit depths. SWF Files could work well on limited hardware, while taking advantage of more expensive hardware when it is available.

As SOA technology are more popular utilized in the market by extending the established e-commerce systems to mobile devices, a core challenge is the choice of Flex mobile applications and Web service that adapts well to the existing architecture.

7.2 Development of the Method to Utilise Flex application for Communication with Web server

As shown in Figure 7.1, the utilisation of Web services enables the WSO-MCAD to provide a very loose coupling between an application that uses the Web service and the Web service itself. This flexibility may become increasingly important as the software is built by assembling individual components into a complete application.

- **Mobile Client:** A Mobile user could start the Flex application using Mobile Web browser. Flex application uses Web service to communicate with Web server without requiring the client to reload the SWF file again. By providing a cross-platform runtime environment for rich clients, ActionScript, included in the Flex application, has been used to offer the application dynamical behaviour. The main task of the ActionScript used in this research is to check the data before passing them to the server. If there is an invalid data, i.e., any of the data being wrong, missing or out of the permitted range, the ActionScript would prevent transfer of the data to the server and display a message box informing the user what the error is. The ActionScript code is embedded into the Flex document using the SCRIPT tag.

- **Web Server:** The client proxy receives the request from the mobile client, serializes the request into a SOAP request which is then forwarded to the remote Web service. The Remote Web Service receives the SOAP request, executes the method, and sends the results in the form of a SOAP response to the client proxy, which deSerializes the message and forwards the actual results to the client. There also exists Java or other programs that connects to the Resourses. This connection can be any communication link supported by the Web server.

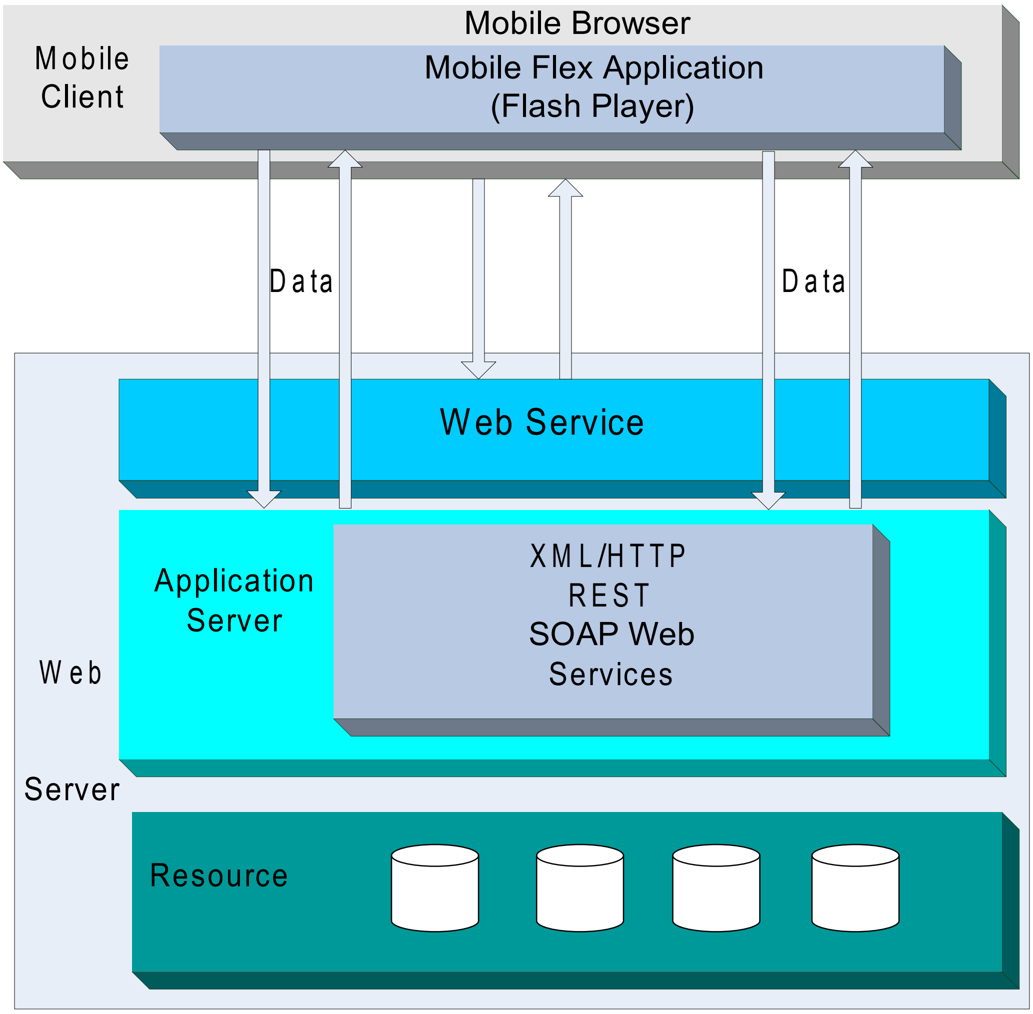


Figure 7.1 Flex application communicates with Web server

7.3 Using Flex Builder to Create Web Service-Based Mobile Applications

This section describes the details of the mappings between the WSDL and the Flex mobile applications. That includes the three main functions: WSDL mapped to Flex application; Flex application Invoke WSDL operation and Flex application request mapping for Web service.

—

7.3.1 WSDL Mapped to Flex Application

Declaring a Web service in MXML is to choose the Web services WSDL definition, system achieve this by setting the *wsdl* attribute of the <mx:WebService> element to the URL of the WSDL file. Mobile application selects the target service and port by setting the *service* and *port* attributes of the <mx:WebService> element. Figure 7.2 sho ws where to find the values of these attributes in the WSDL definition (words in red colour).

```
<mx:WebService id="HelloWorld" wsdl="http://192.168.1.2:
8080/spring_http/HelloWorld?wsdl" useProxy="false">
        <mx:operation name="sayHi">
            <mx:request>
                ...
            </mx:request>
        </mx:operation>
    </mx:WebService>
```

Figure 7.2 Mappings for service and port

These attributes are optional. For *service*, it defaults to the *name* of the first <wsdl:serv ice> (XML tags prefixed by *wsdl* are part of the WSDL namespace and are used to construct a WSDL file). For *port* it defaults to the *name* of the first <wsdl:port> contai ned in the selected <wsdl:service>. To avoid confusion, it's a best practice to set these attributes if the WSDL definition contains multiple <wsdl:service> and <wsdl:port> el ements.

7.3.2 Flex application Invoke Operation within WSDL

After selecting the service and port to use from the WSDL definition, a <mx:operation > element is added inside the<mx:WebService> element for each operation that will be invoked. The *name* attribute of each <mx:operation> element corresponds to the *name* of the <wsdl:operation> element that defines the Web services call. The selected <wsd l:operation>must be part of the target <wsdl:service> and <wsdl:port>. Figure 7.3 sho ws where can locate the value of the *name*attribute in the WSDL definition.

```
...
<wsdl:operation name="sayHi">
  < soap:operation soapAction=""
      style="document" />
  - <wsdl:input name="sayHi">
    < soap:body use="literal" />
      < /wsdl:input>
...
```

```
<mx:WebService id="HelloWorld"wsdl="http://
192.168.1.2:8080/
spring_http/HelloWorld?wsdl"useProxy="false">
        <mx:operation name="sayHi">
            <mx:request>
                ...
            </mx:request>
        </mx:operation>
    </mx:WebService>
```

Figure 7.3 Mapping for operation (Top: WSDL File, Bottom: MXML File)

Other attributes, such as *result* and *fault* attributes, are ActionScript Event Handlers; they contain ActionScript code that is executed, respectively, once a SOAP response or a SOAP fault arrives.

7.3.3 Flex Application Request Mapping for Web Service

After selecting the Web services operations, Flex's parameter binding mechanism is used to declare the parameters to these operations. As follows, each <mx:operation> el ement contains a <mx:request> element, which includes a mixture of XML and Flex bindings using '{}' in Figure 7.4.

```
<mx:WebService>
<mx:operation>
    <mx:request>
        <node1>{var1}</node1>
        <node2>{var2}</node2>
    </mx:request>
</mx:operation>
</mx:WebService>
```

Figure 7.4 Flex binding Web service

Common Mappings Method: Document Oriented - Literal

A WSDL document describes a Web service, WSDL binding describes how the service is bound to a messaging protocol, particularly the SOAP messaging protocol. A WSDL SOAP binding can be either a Remote Procedure Call (RPC) style binding or a document style binding. A SOAP binding can also have an encoded use or a literal use. This gives user four style/use models: RPC/encoded, RPC/literal, Document/encoded and Document/literal.

All Flex application of this mobile system will choose Document/literal model, because using RPC model just like users exchange message via telephone, but Document model rather than using the Email to transfer the message. And Document/literal has the following strengths:

- No type encoding information;
- System can finally validate SOAP message with any XML validator;
- Everything within the soap:body is defined in a schema;
- Document/literal is WS-I compliant.

—

The SOAP binding section in WSDL definition determines the structure of the body in SOAP message. The operation style is defined in the style attribute of the <soap:operat ion> element inside the binding. In a Document Oriented operation, XML documents are exchanged within the SOAP body without any additional wrapping elements. The following example uses a Web service with operations using the Document-Literal style in Figure 7.5.

```
- <xs:sequence>
  <xs:element minOccurs="0" name="arg0" type="xs:string" />
  <xs:element minOccurs="0" name="arg1" type="xs:string" />
  <xs:element minOccurs="0" name="arg2" type="xs:string" />
  <xs:element minOccurs="0" name="arg3" type="xs:string" />
  <xs:element minOccurs="0" name="arg4" type="xs:string" />
  </xs:sequence>
  </xs:complexType>
```

```
<mx:WebService id="HelloWorld"
        ...
        <mx:request>
            <arg0>{t1.text}</arg0>
            <arg1>{t0.text}</arg1>
            <arg2>{t2.text}</arg2>
            <arg3>{t3.text}</arg3>
            <arg4>{t4.text}</arg4>
        </mx:request>
        ...
</mx:WebService>
```

Figure 7.5 Mapping of multiple parameter operation for a Document-Literal service call

For Document-style operations, each <wsdl:message> element contains one <wsdl:part > that refers to a schema element. Because Flex treats all Web services calls as asynchronous, the client is not blocked to wait for the response from the Web service, rather it continues executing, and an event is active once the result message is received. The *result* attribute of the <mx:operation> tag refers to the ActionScript code executed once the result message is received.

7.3.4 Security of Flex Application

To control the client-server communication of mobile application to the server API, Flash sandbox is utilized to prevent those unauthorised clients from exchanging data with the foreign Web service. For example, the crossdomain.xml (Figure 7.6) policy file, which are deployed in the Web server, only allows the request from **www.ntu.ac.uk** to access the Web Service.

```
<allow-http-request-headers-from
domain="www.example.com" headers="SOAPAction"/>
So the complete file might look like the following :
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.adobe.com/xml/dtds/cross-
domain-policy.dtd">
<cross-domain-policy>
  <!-- This domain can accept the SOAPAction header from a SWF file from
www.example.com -->
  <allow-http-request-headers-from domain="www.ntu.ac.uk"
headers="SOAPAction"/>
</cross-domain-policy>
```

Figure 7.6 Sandbox file for security: crossdomain.xml file

7.4 Experiment

After the deployment of mobile client and server applications, there were the Software and Hardware speciation for the experiment:

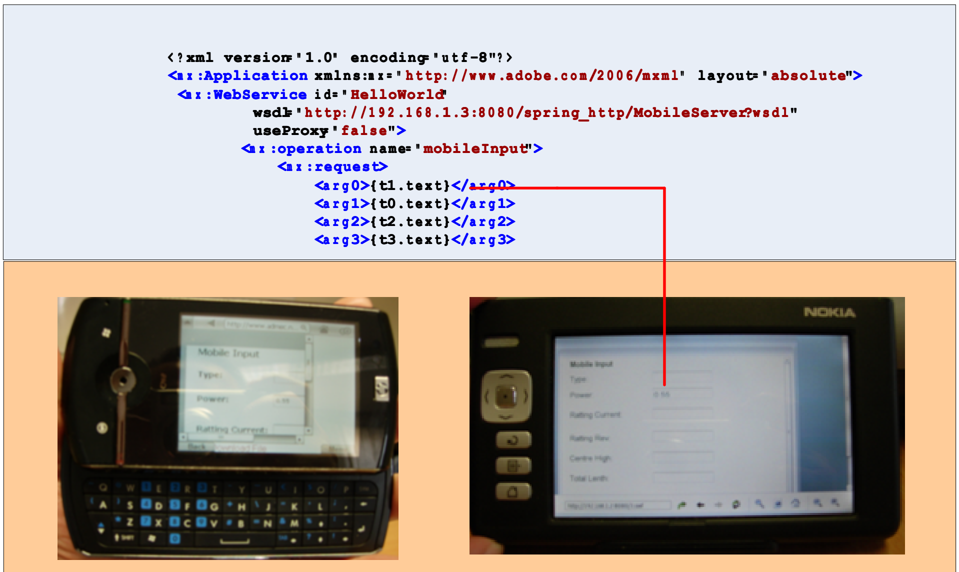
- Tomcat 5.6, Axis 1.4 and JDK5.0 to provide the Web server and create the Web Services.
- Flex 3 create mobile client SWF application; eclipse utilized to write the java Web Services.
- HP PDA: its OS is **Windows Mobile 6.1,** Qualcomm 7201A Processor 528 MHz, 128 MB SDRAM main memory for running applications, 256 MB flash ROM.

—

- Nokia PDA: its OS is **Maemo (operating system),** which is a modified version of Debian GNU/Linux; CPU running at 252 MHz, Memory, 64MiB of DDR RAM, and 128MiB of internal FLASH memory.

7.4.1 Mobile Platform Compatibility: Flex Applications across Windows PDA and Nokia PDA

There were two different Operation Systems installed in the PDAs, such as Windows Mobile 6.1 in HP PDA, and Mobile Linux in Nokia PDA. In Figure 7.7 the two mobile devices run the same Flex mobile application, system did not need to re-write the software for different mobile devices. Mobile user started the Flex application by Mobile Web browser. SWF file, which existed in the Web server, will be executed on the mobile devices in the Flash Player; Flex application used Web service to communicate with Web server and did not need to reload the SWF file again.



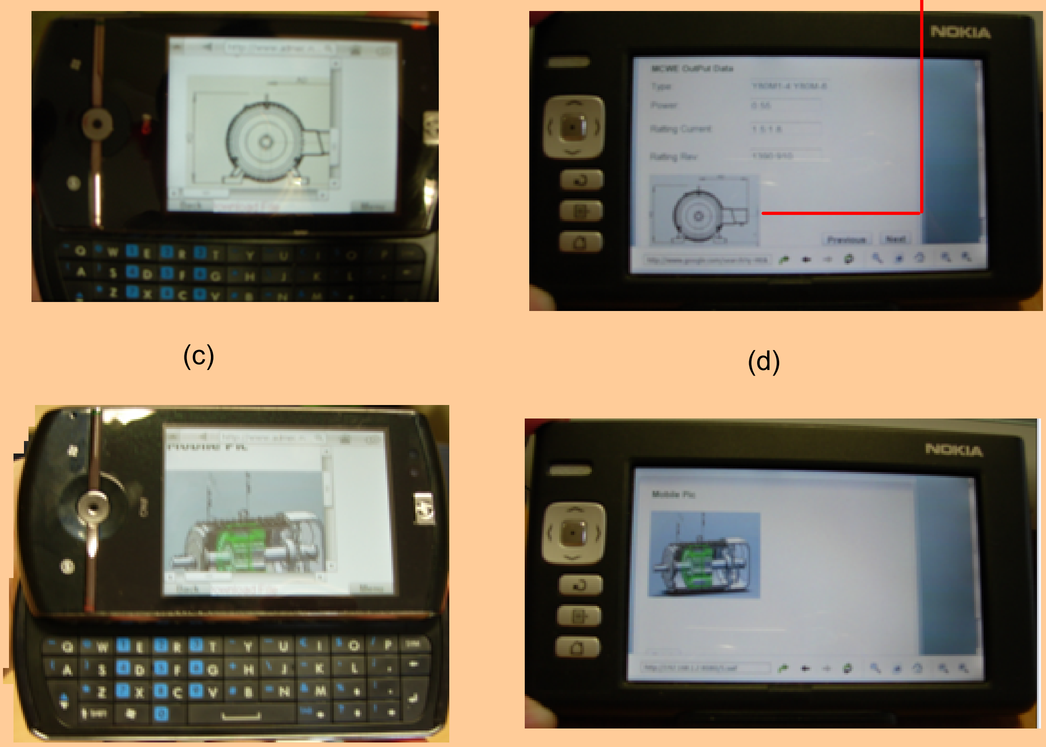
(a) HP PDA (b) Nokia PDA

This example shows that different Mobile OS devices run the same Flex mobile application; for instance, an electromotor with 0.55 KW power is required. After the mobile user input all the search information via Flex application, Flex application will send all the mobile input data to the Web server via invoking the Web service as shown in Figure 7.7 (b), therefore the Web server will automatically retrieve the relative information and send the results back to the mobile device as shown in (c-e).

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" click="loadpic()">
<mx:Script>
        <![CDATA[
                import flash.display.Loader; //import class
                import flash.net.URLRequest;
                import flash.events.Event;
                private var pictLdr:Loader = new Loader(); //declare Loader class
                private var pictURL:String = "http://www.admec.ntu.ac.uk/motor.jpg";
                private var pictURLReq:flash.net.URLRequest = new flash.net.URLRequest(pictURL);
                //    private var container:Sprite = new Sprite();
                var lc:LoaderContext  = new LoaderContext (true);
                var l:Loader = new Loader;
                private function loadpic():void //Load Picture
                {
                l.load(pictURLReq,lc);
                  //       pictLdr.load();
                           img.load(pictLdr.content);
                           stage.addChild(container);

                    l.contentLoaderInfo.addEventListener(Event.COMPLETE, imgLoaded);
                }
                          private function imgLoaded(e:Event):void
                {
              //     img.load(pictLdr.content);

              img.addChild(l.content);
```



(c) (d)

(e) (f)

Figure 7.7 Mobile Platform Compatibility: Flex Applications across Windows PDA and Nokia PDA

7.4.2 Two Methods to Deploy Flex Application into Mobile Device

There are two methods to deploy Flex Application into mobile device; one is depent on the latest mobile Web Browser Technology, another one is to install Adobe Flasher 9 plug-in:

- **Through Mobile Web Browser:** In the test, HP PDA was installed Skyfire Mobile Web browser to view the SWF file, but the speed of the downloading for SWF file is slow, because Skyfire uses proxy server to compress and optimize requested websites, which are later displayed on a mobile device. Through this method, there maybe exists a lot of data transaction between mobile device and Skyfire server; the mobile user will face the Network Latency problem, another disadvantage is that mobile devices must connect with Internet to access Skyfire Web server, sometimes, the connection fee would be expensive for the remote users.

- **Install Adobe Flasher 9 plug-in**: Another method could be used is to install Adobe Flasher 9 plug-in inside mobile device. From the user's experience, there are no Network Latency problem happened by this method; and the mobile user does not need to connect with the Internet, which could save the cost of the users. Compared with the two aforementioned methods, the author suggests that installing flash 9 plug-in into mobile device is the better way to view and implement Flex applications.

Mobile users could use Nokia PC Suite and/or Microsoft ActiveSync Tools to install Adobe Flash plug-in or the latest mobile Web browser; Flex mobile application could extend the capabilities of the mobile client with the richer Internet applications in real time [79].

7.5 Conclusions

Compared with other mobile program languages, the deployment and development of Flex mobile application have distinct advantages, the adoption of Flex is not difficult, it takes the user less effort to run the software; which could save the developer's coding time; make it easy to compile and debug the program. The limitation of the Flex mobile application is the system requirement of the mobile device, which need to install the latest Mobile Web Browser or flash 9 plug-in. In order to deploy the Flex application into the different Mobile Platforms much easier, Adobe Company are going to bring Flash Player 10 to Smartphone class devices to enable the latest Web browsing experience in 2009 October [89]. Multiple mobile devices, such as Google's Android, Nokia's Symbian OS, Windows Mobile, and the New Palm Web OS, will be among the first devices to support Web browsing with the newest Flash player.

Chapter 8
MCWE Applications for Product Design

This research centred on how product designer or engineer staff can share or access to vital information, services and different resources at any time and anywhere. In this chapter, one case study about information system is going to be discussed, all the test data are retrieved from the industry companies in order to prove the WSO-MCAD system is valid and the mobile system could be used into theses factories in the future; another case study present the combination of Web Services and Flex applications and the necessary modification of the gear design program make it possible to be invoked by the mobile devices.

8.1 Case study 1: MCWE Applications in Huanan Huadian Power Ltd

8.1.1 Introduction

In the case study, system utilize four related companies to be involved: Hunan Huadian Changsha Power Generation Co., Ltd, Hunan E-tong electromechanical technology Co.,Ltd and other two companies' information will be discrete, for reasons of confidentiality is an electromotor factory and a bolt factory.

Hunan Huadian Changsha Power Generation Co., Ltd, China

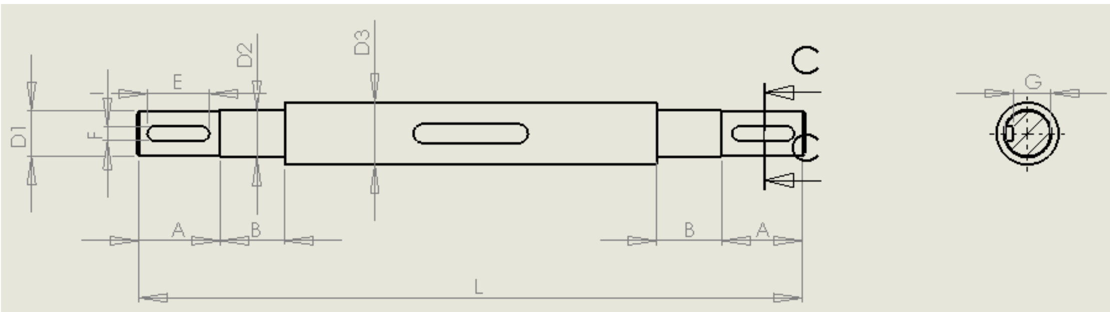
Hunan Huadian Changsha Power Generation Co., Ltd, (Hunan Power), was decided to be the main body of this case study by the author. Hunan Power is the biggest green power station in Hunan province. The total investment of the company was 15 billion Chinese Yuan (*12 Yuan equal to 1 British Pound in 10/2009*) and the occupied area was 430000 m^2. The annual payment of tax will be around 0.2 billion Yuan. The first stage was completed at October 2007 and the installed capacity was 1200 MW. It cost 5 billion Yuan in the first stage. It can support 1/10 electricity consumption of the whole Hunan province, China (67.78 million people, 2006).

Figure 8.1: Air view of the power station

E-tong electromechanical technology Co.,Ltd, China

E-tong electromechanical technology Co., Ltd (E-tong) is a supplier of Hunan Huadian Changsha Power Generation Co., Ltd. This company was built up in 1995; the registered capital of this company is 10 million Yuan. The main operation of company is machining and assembles the machine parts. The annual turnover is 20 million Yuan, and the annual payment of tax is around 6 million Yuan. The typical products of company are bearing with housing and shaft as the figures shown below.

(a) Dial indicator and Bearing with housing



(b) Shaft

Figure 8.2 CAD Drawing of E-tong products

8.1.2 Current Problems

The basic information of the companies has been described, the major problems of the power station could be pointed that:

- There are four or more maintenance every year, and the accident may happen occasionally in Power Station. For example, if the engineers are required to climb up the boiler to repair the machine there, in this situation, they can not take many things, such as laptop, paper drawings about the machine, which could facilitate an accident. Therefore mobile devices are the demand for their requirement.

- Power Station is a huge factory, so a huge number of material and components are going to be required during the production, how to control or manage the orders efficiently becomes a new challenge.

- Because of the magnetic field in the Power Station, some areas of the company are not suitable to locate the laptop/computer, in this situation, mobile devices are suitable and necessary for the designers or engineers to connect internet and obtain information via the Wireless network; or in some areas, if there is no local network available, people could access the internet via the mobile company's wireless network, even the accessing fees maybe expensive, but more flexible and mobility.

- There are close supply and demand relationships among the four companies; engineers or managers need to share and retrieve product data from the different company's Web servers.

8.1.3 Application of the Information System

First of all, the concept should be emphasized that WSO-MCAD are developed for all the four companies, but the test model and customer interface were made for individual company, in this case study, system focus on Power Station. Figure 8.3 shows a typical supply and demand relationship among these companies.

- Shaft factory, electromotor and bolt factory are all the suppliers of the power station.
- The electromotor factory is supplying electromotor or other products to the shaft factory and bolt factory.
- The shaft factory and bolt factory are the suppliers of the electromotor factory as well.

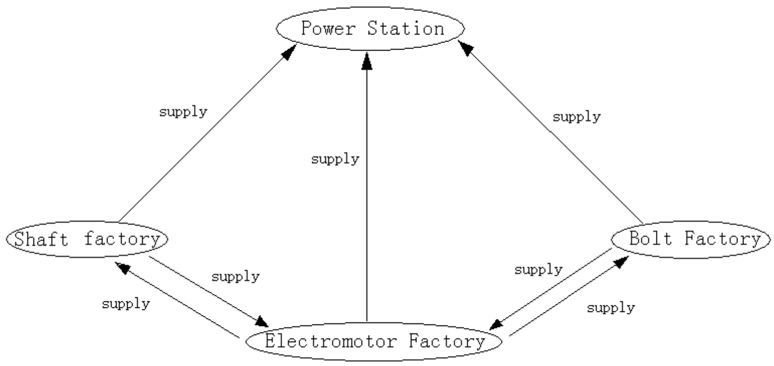


Figure 8.3 Relationship of four companies

The details relationships among the four companies are that: the electromotor factory is a supplier of the other three companies, because most mechanical factories need its product. The electromotor is made up of several parts such as: shaft, bolt, bearing, rotor, stator and etc. In the power station, when the shaft of electromotor was in bad condition, manager has the following options below:

- Option 1: Purchasing a shaft from the electromotor factory. Because it is the original manufacturer of the accessories of the electromotor; choosing this option is easier and sooner to get product, but probably more expensive.

- Option 2: Purchasing a shaft from the shaft factory. It is a manufactory for machining all kinds of shaft, and it will take longer but less costs.

The electromotor factory can produce all accessories, but if there is a huge orders from their customers, such as power station or bolt factory, that situation make the company unable to meet; therefore, there are also two options for the electromotor manager:

- Option 1: Assigning more employees or shifts into work.

- Option 2: Purchasing some of accessories from other factory such as the shaft factory and bolt factory in a lower price.

- Same situation will happen on the shaft factory and bolt factory.

Thus, in order to meet the requirement of the manager or engineer, it is the demand to efficient share and retrieve their product data information among the Web servers of the four companies at anytime and anywhere. The following sections will present the implementation of WSO-MCAD; Figure 8.4 is the task function structure of the mobile system.

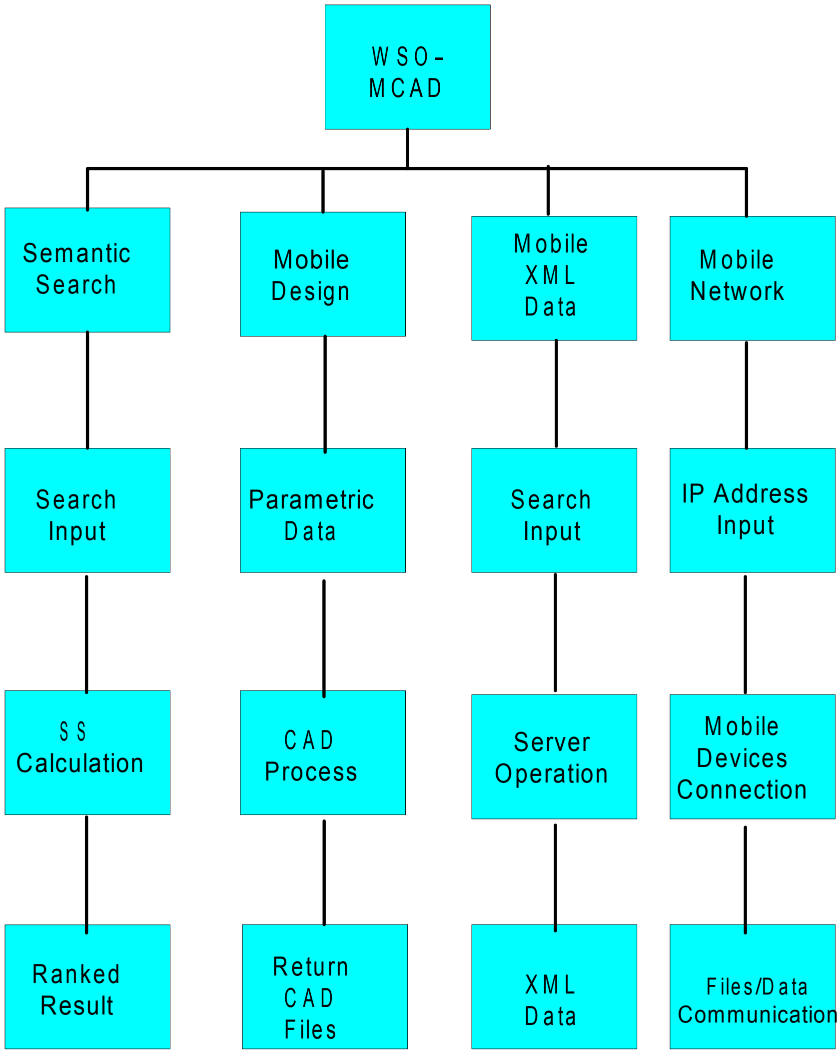


Figure 8.4 Function structure of mobile system

Mobile system has four functions:

- **Mobile XML Data Function** is to obtain product XML data from the Web server, that has been shown in chapter 6;

- **Mobile Network Function** is for mobile users exchange instant message via the wireless network, this technology utilized in this module are based on the TCP/IP (Internet Protocol Suite), the implementation of this function are similar as MSN.

- **Semantic Search Module Function**: manager could input the request into the mobile device when he is not in the office; WSO-MCAD will recognize the input data and sent back comprehensive results. Manager or the engineer will receive accurate and closely related product data information such as: the price information, delivery information, staff information and CAD drawings information to help choose the most suitable answer, if the engineer needs to obtain the DWG, DXF or ShortDWG file, they could go to Mobile Design Module to allow Web service to return CAD files and view it by Windows mobile devices. WSO-MCAD can automatically identify the user input based on the ontology database in the Web Server. A mobile user interface is implemented with VB.NET. The user and service ontology are represented with OWL and Jena. To support the semantic similarity calculation between user's input and a specific service result, a Web service called SemanticService is implemented, the case study concerning this function has been explained in chapter 5.

- **Mobile Design Function**: There are four or more maintenance every year, and the accident may happen occasionally. For example, if the engineer needs to choose the bolt which is really suitable to special situation, they need to input the force, work load, bolt style, etc. The final product will be constructed out from the modelling module; all the analysis and check processing will be carried on in the Web Server. And after the mobile user get the Bolt type, for example, "M12"; if he wants to know parametric data about such product, he could go back to the Ontology Search Module, to obtain more information about the M12 Bolt.

The following sections will describe the Mobile Design Function in details. For example, being a manager of Power Station, if there are any requests of the product, for example: The WSO-MCAD allows all the stakeholders involved in product design to collaborate over the Internet through out the design process; the architecture is shown in Figure 8.5. Mobile devices are used as tools for the users to communicate and collaborate with each other via the Web Server. Product data including CAD drawings and materials can be effectively shared between the users.

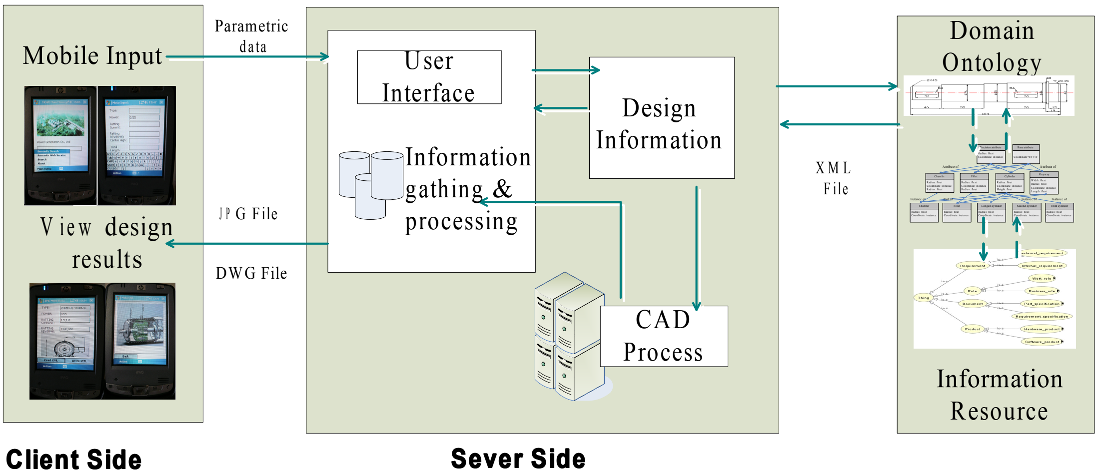


Figure 8.5 Architecture of Mobile Design function

8.1.4 Mobile System Design

From chapter 2, it's 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 the Web Service. In the author's point of view, any ICT structure should at least have the following benefits:

- Introduces new capabilities by dynamic information repurposing and re-configuration, adaptive service-oriented networks, and loosely coupled application connections and information.

- A reliable knowledge base can be established using inference power and Mobile devices are used as tools for the users to communicate/collaborate with each other via the Web Server.

- Economic maintenance costs for IT systems integration.

- Allows users to implement the invocation of software program from different mobile devices over the Wireless Internet.

From the aforementioned features, the WSO-MCAD is an ideal choice to be used with the wireless environment. In order to test the whole system, a model of electromotor has been constructed to be used within this mobile system, because,

- Most parts of the electromotor are standard; there are not so many complex calculations that make the design process easier to be tested.

- A relationship will be built up among the different companies via producing, purchasing and selling the standard parts, such as bolt, shaft, etc.

Because the electromotor is the standard product, the steps of modelling design are:

- Choosing a typical electromotor as fundamental model.

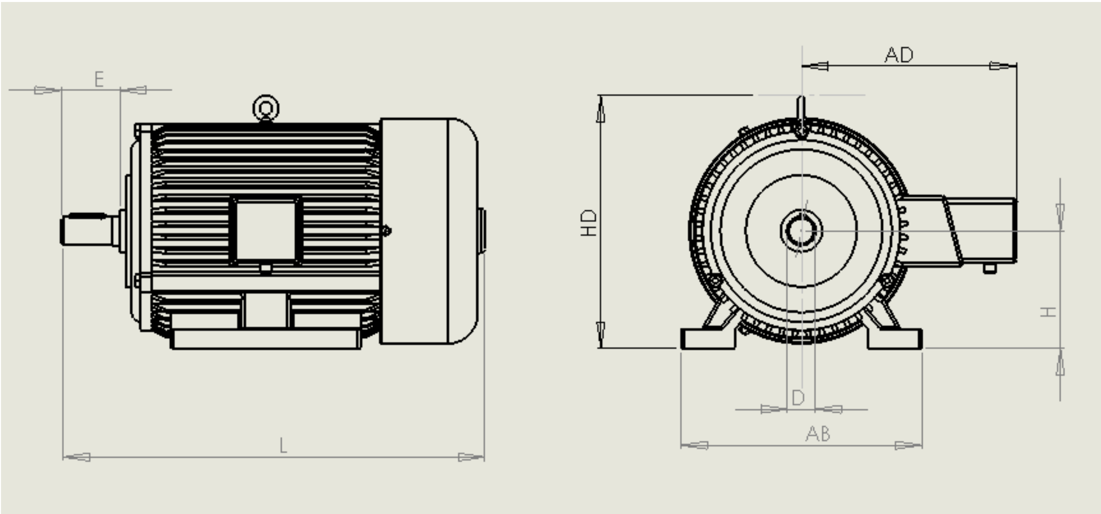
- After calculating and checking components of electromotor, constructed all models by SolidWorks 2007.

- Using secondary development of SolidWorks 2007 method, creating the series of all size of electromotor.

- Save all the product data into Web server.

8.1.5 Modelling of electromotor

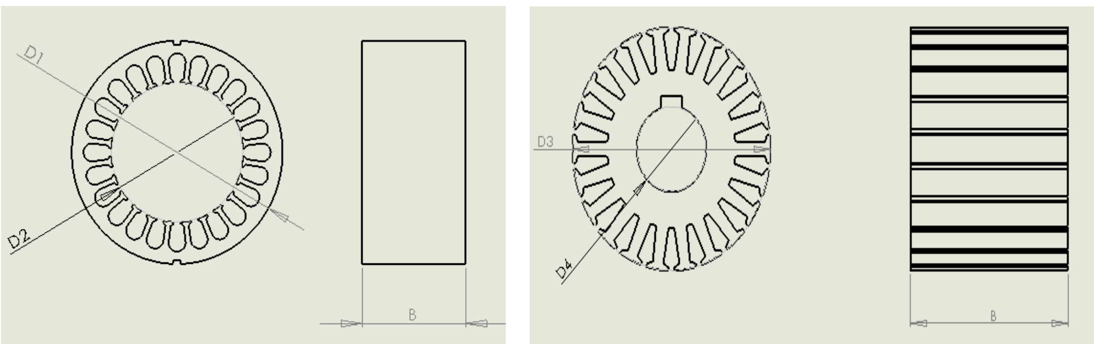
According to the information which was supplied by the companies, the type of Y80M1-4 was chosen to be the fundamental model of the electro- motor. The overall dimensions are:



TYPE	POWER (KW)	RATING CURRENT (A)	RATING REV (RPM)	D	E	H	AB	AD	HD	L
Y80M1-2	0.75	1.8	2830							
Y80M2-2	1.1	2.5								
Y80M1-4	0.55	1.5	1390							
Y80M2-4	0.75	2								
Y80M-6	0.55	1.8	910	19	40	80	165	150	175	290

(a)

Then the general dimensions will be considered into the entire modelling process. According the centre high and the design principles of electromotor, the stator and rotor were designed as follow:

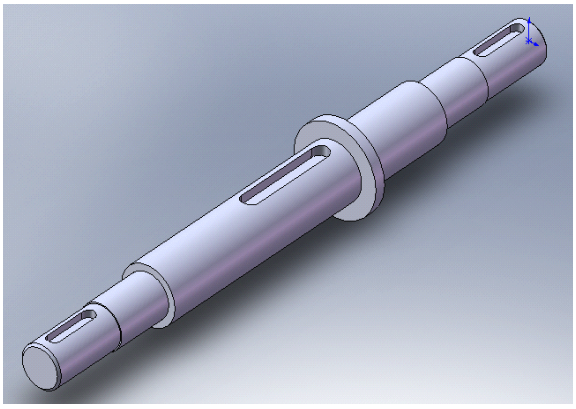
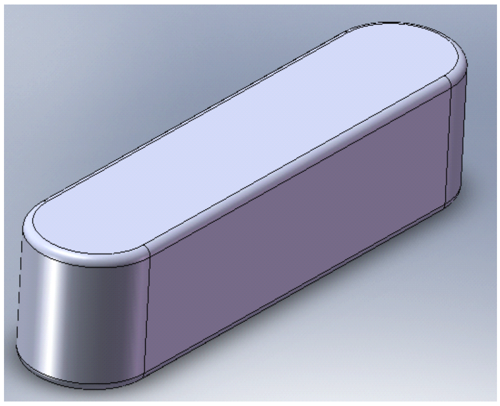


(b)

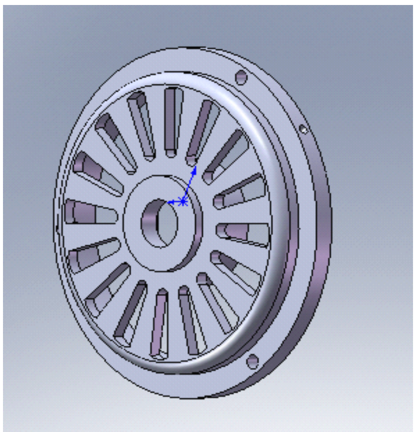
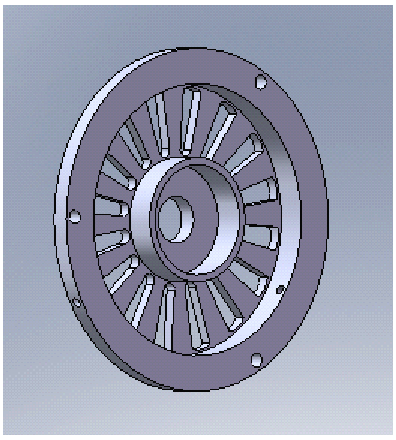
TYPE	D1	D2	D3	D4	B
Y80M1-2					
Y80M2-2					
Y80M1-4					
Y80M2-4					
Y80M-6	120	75	74.5	26	58.5

(c)

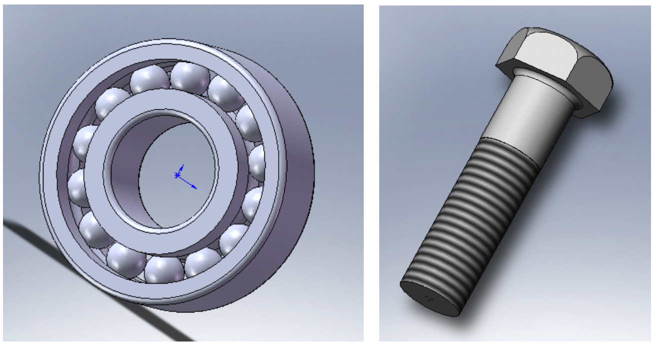
According the rotor and the overall dimensions, the shaft, bolt, shell, lid, bearing and key were decided as follow.

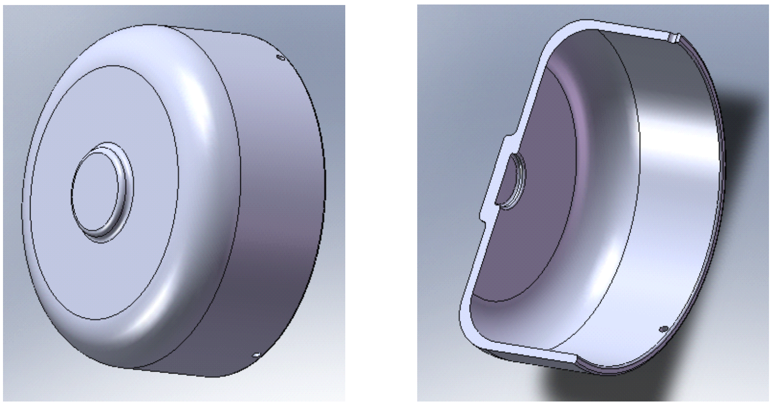
(d) Shaft

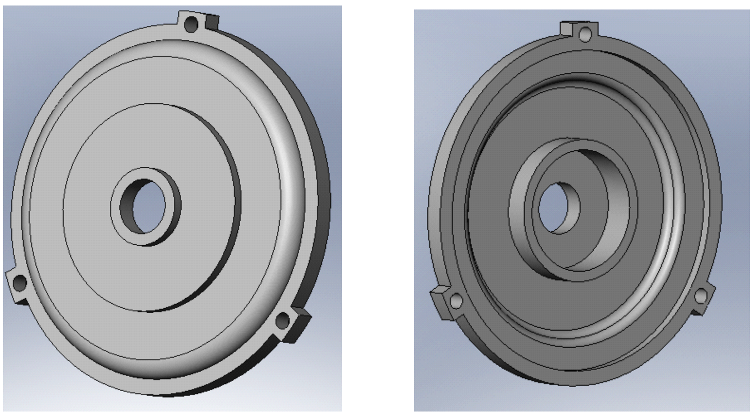
(e) Back cover



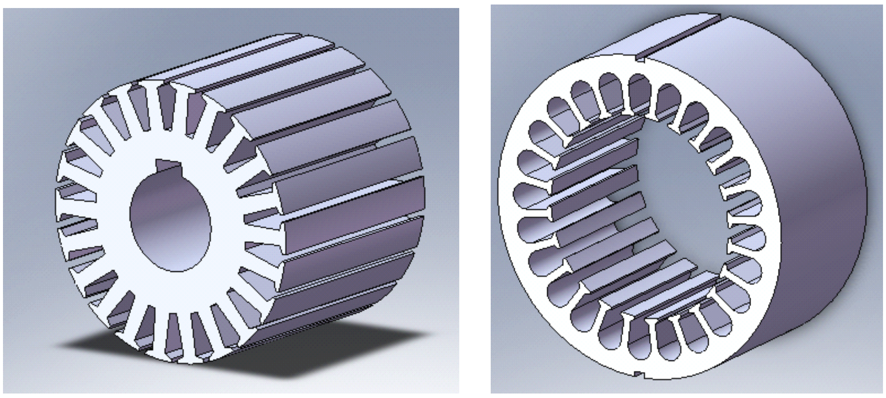
(f) Bearing, key and bolt



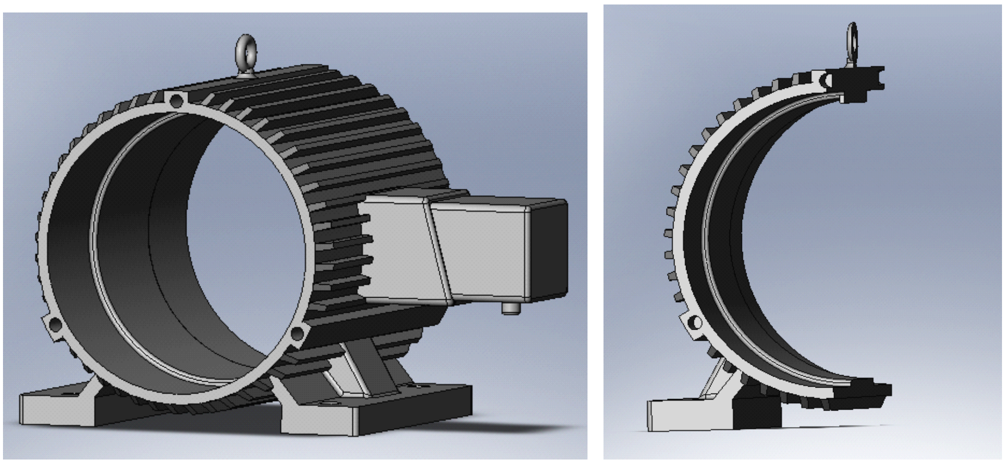
(g) Fan mantle



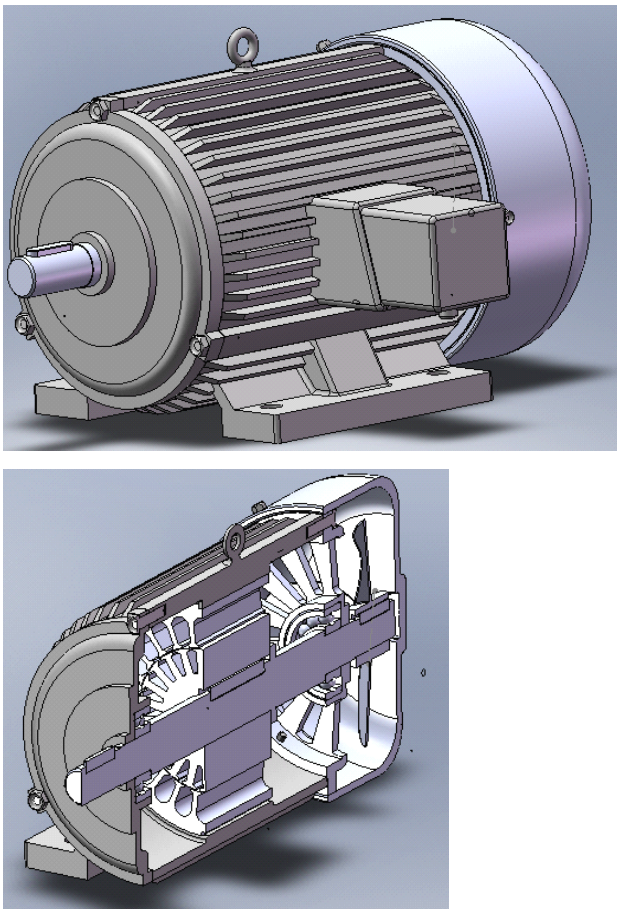
(h) Lid



(i) Rotor and stator



(j) Shell



(k) Final electromotor

Figure 8.6 CAD Files of Electromotor

8.1.6 System Design and implementation

Mobile Design Function:

Within this function, there are several modules of the system shown in Figure 8.7, such as:

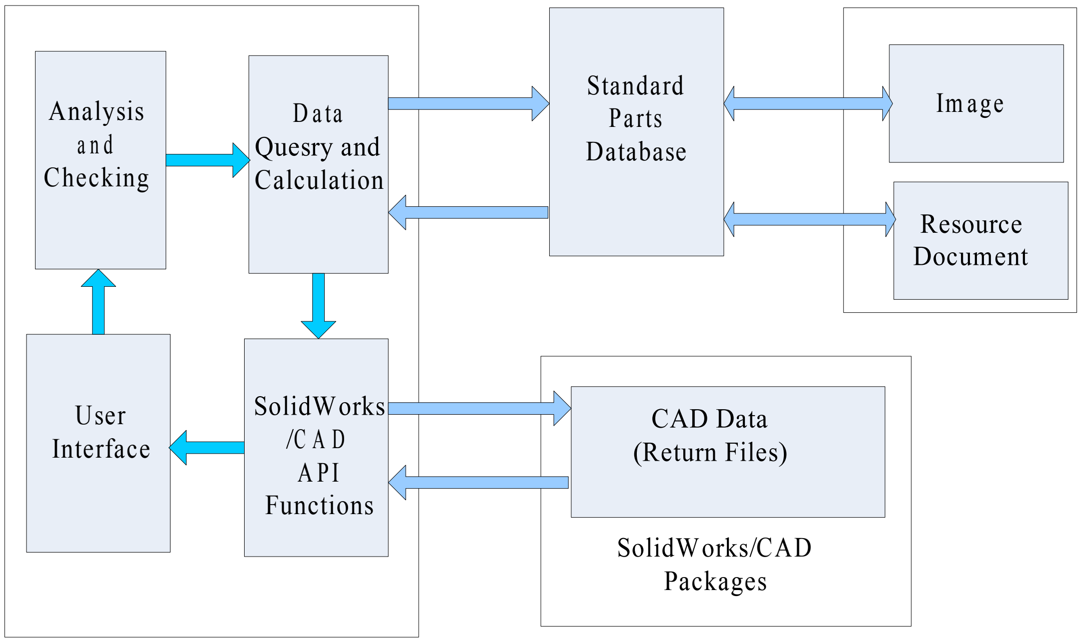


Figure 8.7 Diagram of WSO-MCAD

- The module of Data Query and Calculation: This module is the core of data processing. It will automatically judge the process order after receiving an appropriate operation from the customer. After choosing the specific part and parameter, the system is going to process this module which was connected with standard parts database. According to the request of customer, the module is going to fetch the certain track record, considering the specific instance, and finish the calculation.

- The module of Modelling: After creating Object variable and invoking the SolidWorks, the dimensions of part will be modified following the request of customer. After that, SolidWorks will rebuild and save the model in a new document, which is the final product.

- The module of Analysis and Check: this module is going to help the customer to choose the suitable bolt. After inputting the force, work load, bolt style and etc, the final product will be constructed out from the modelling module.

- The module of the Standard Parts Database: This module is the centre of data storage and management. The model information is represented as datasheet which is not the simple 2D dimensions sheet, but built up by the ontology data management system.

- The module of Image: Being an application of graphic user interface which is running under Windows, an appropriate image description is necessary. This module is independent from the main body of the system which could be invoked automatically during the system running.

- The module of resource document: Invoking this module is to avoid building up the model every time. According to the request of customer, the document will be modified and saved into a new document.

There is a need to compile and apply these modules to the system for the mobile design function. For example, as shown in Figure 8.8, there is a mobile electric generator supported by four wheel brackets. One of these bolts in the wheel brackets was curved. If do not change the wheel, it is dangerous and a potential safety hazard. In that situation, engineers will check the bolt is competent for work conditions.

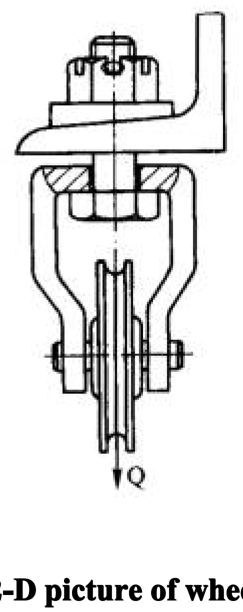


Figure 8.8 2-D picture of wheel bracket

Work conditions shows that one bolt fastens a wheel bracket to upper main part; a bolt suffers an axial load, 10kN, from the main part. The bolt is non-preloaded for making wheel bracket movable. Nominal size of this bolt is M10, and min diameter which is a property of the bolt is 8.376mm (Figure 8.9). Safety factor select 1.4 based on importance of this main part. Mechanical property level, 4.8, shows on the bolt; and the bolt yield stress is 320 Mpa (Figure 8.10).

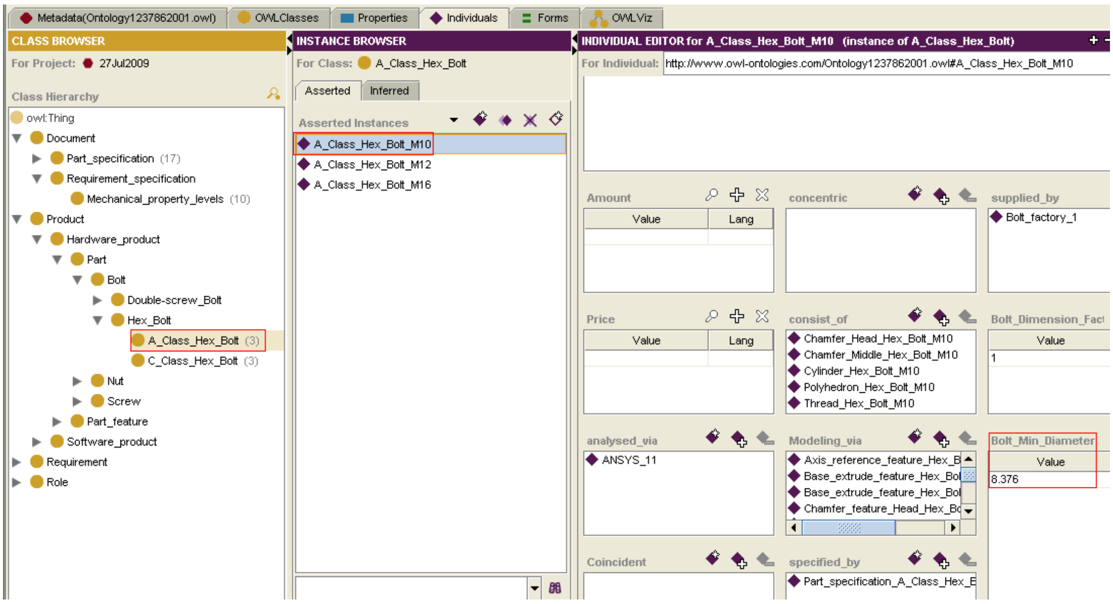


Figure 8.9 Bolt min diameter of property within Protégé (red colour rectangle)

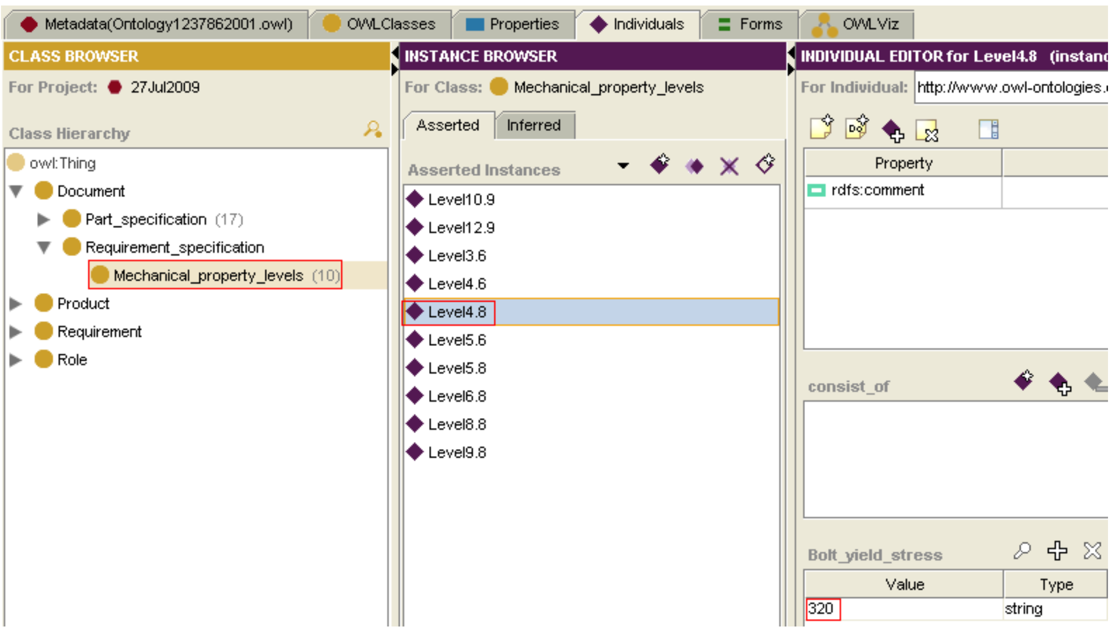
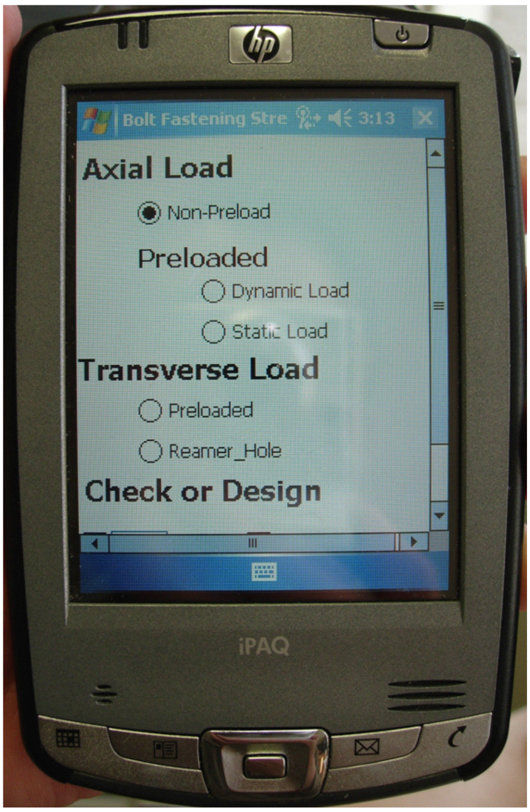
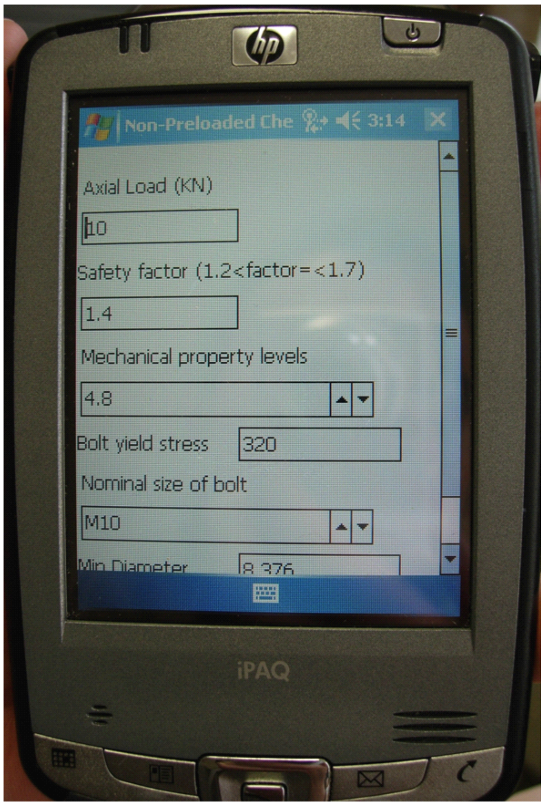


Figure 8.10 Mechanical property levels of Protégé (red colour rectangle)

Engineer needs to input these factors via a mobile application interface, when the computed stress is less than or equal to permissible stress, the bolt is satisfied for this condition. Otherwise, if the engineer wants to assemble these parts and select a suitable bolt, he can choose the design option, set all necessary factors and press calculate button, permissible stress is calculated at first as well as minium diameter, finally, all the available bolts will be shown in the mobile device.

Mobile application work example will be shown in Figure 8.11, this bolt is non-preloaded shown in (a), and the bolt suffers an axial load, 10kN, from the main part shown in (b). The bolt is to be checked it is competent for this working condition or not.





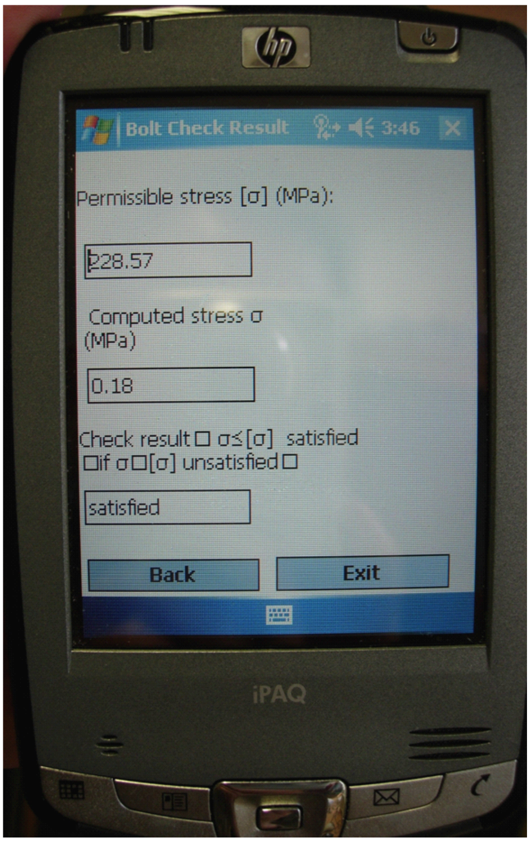
(a) (b)

After Mobile user has entered the parametric data in Figure 8.11 (b), such as:

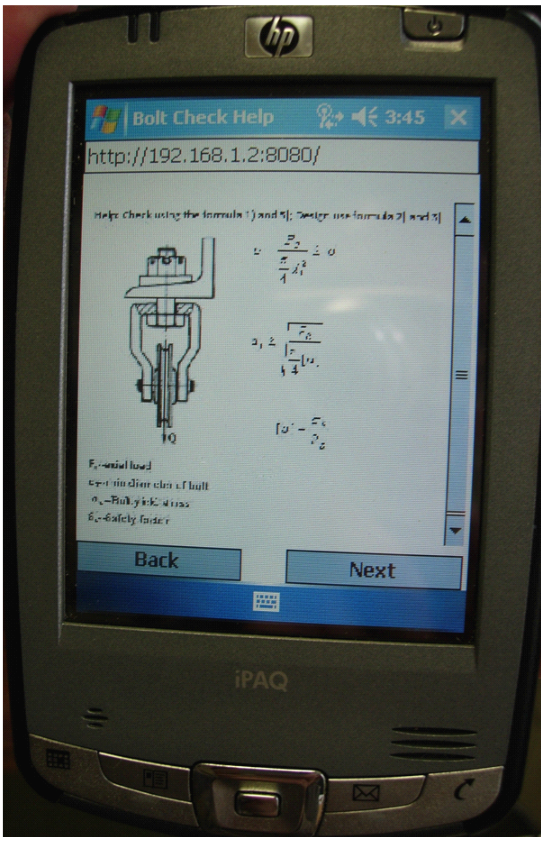
- Nominal size of this bolt is M10,
- Min diameter is 8.376mm
- Safety factor select 1.4 based on importance of this main part;
- Mechanical property level, 4.8, shows on the bolt and Corresponding, the bolt yield stress is 320 MPa.

Then result will be shown in Figure 8.11 (c)：

- Permissible stress [σ] = 228.57MPa
- Computed stress σ = 0.18 MPa
- Check result： σ ≤[σ] satisfied

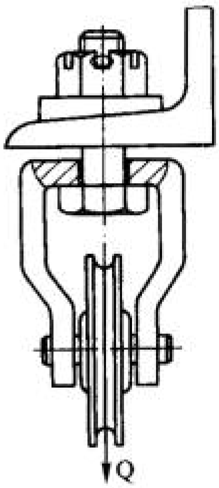


(c)



(d)

Figure 8.11 (d) is the help document of this engineer process, it is shown below:



$$\sigma = \frac{F_C}{\frac{\pi}{4} d_1^2} \leq [\sigma]$$

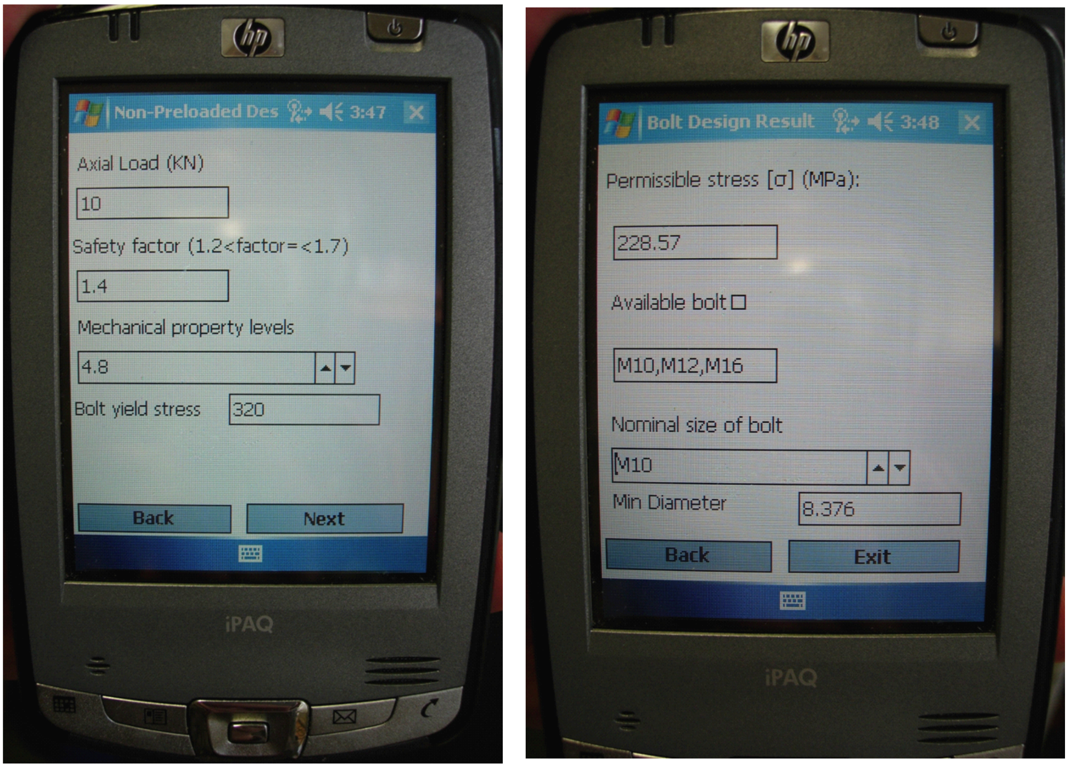
$$d_1 \geq \sqrt{\frac{F_C}{\frac{\pi}{4}[\sigma]}}$$

$$[\sigma] = \frac{\sigma_S}{S_S}$$

- Check: 1 and 3 formula;
- Design: 2 and 3 formula

F_c--axial load
d_1--min diameter of bolt
σ_s--Bolt yield stress
S_s--Safety factor

Otherwise, if the engineer wants to assemble these parts and select a suitable bolt. He can choose design option, set all necessary factors and press the calculate button, the result will show all the bolts which meet the requirements. For example, if the computed stress is less than or equal to permissible stress, the bolt will be satisfied for this condition. In design mode, permissible stress is calculated at first in Figure 8.11 (e), and min diameter is calculated, Figure 8.11 (f) is the result of the design option, all the three types of bolt: M10, M12 and M16 could be chosen by this situation.



(e) (f)

Figure 8.11 Mobile Interface of Bolt Search

8.2 Case study 2: Mobile Platform Compatibility, Remote Invocation of a Large-scale Computing Program

In ADMEC previous research, a Web server-centralised system was developed with combination of Java Servlets, Applets and CORBA, which enabled the existing large design program run remotely on the Internet, such that geographically dispersed designers can implement a design on a Web browser and obtain the multiple formats of result from any part of the world [90]. Today, it is the new challenge for current research to invoke the large-scale computing program via mobile devices. Based on the latest mobile technology and Web technology described in chapter 7, this section will show how the mobile Flex application invoke a large-scale computing program via a mobile device.

In product design, there are many computing programs, which are large-scale, time-consuming and not convenient in multiple programs working environment. Basically, it is ideal to be located and executed on the owner's computer with response to the client request to make sure thin-client requirement. And, because of recent advances in wireless communication technologies, one of the most attractive challenges in this today is to exploit wireless computing technologies that may help build a ubiquitous, mobile client/server system, thereby supporting collaborative works [46]. In the system, the time-consuming programs should not be interrupted during their execution; furthermore, in a collaborative environment, it is probably better to make such a large-scale and time-consuming program to be invocated singularly; otherwise the collaboration with other partners may mean waiting for too long a time to retrieve the results.

Flex is small programs that execute on the client side of a Web connection; they are developed to dynamically extend the functionality of a Web browser. The developed Flex application that is included in the response page could implement the display of resultant text data, graphics, and program execution progress bar. A user, i.e. a design engineer could conduct the gear design including inputting parameter data, activating calculation program, monitoring the progress of the program execution, and viewing resultant data and its analysis by the mobile device.

Web Service enables the Flex application to communicate with different programming languages and extends the Flex application by providing a framework for distributed object communications. Web Service provides a distributed object infrastructure over the Internet, and guarantees platform-independent functionality between vendors, which is essential for Internet applications, where different sites' objects and implementations, need to be able to communicate with each other.

8.2.1 Standalone Design Application Package

The Server package is used to optimise the design of external spur and helical gears with in volute tooth profile. The original GearOpt is a single user version without distributed features, which is developed by ADMEC [91]. As the demand of using this application for gear design is increasing, this research decided to wrap it and enabled it with wireless network features, then, the Service Oriented Architecture was employed.

As described in chapter 7, in order to make the Gear Design software to be remote implemented and across different Mobile platforms, the new mobile and Web technology will be explained in the following sections.

8.2.2 Wireless Features Solution

The gear design optimisation could not be conducted on the Wireless Internet unless the following problems have been resolved:

- How to remotely invocate a large sized software package over the Wireless Internet.
- How to pass the user's input data to the executing package and to send the results back to the mobile user.
- How to allow multiple mobile users to run the package at the same time.
- In addition, the copyright and security problems for the package also have to be considered.

To accomplish these tasks, in this system, the combined module of Java, C++, Flex and Web Services has been used. The complete structure of the system is shown in Figure 8.12. The gear optimisation package is located on the server side and a user on the client side wishes to conduct gear design by invocating remote design resources. The user only need to do is to interact with a Graphical User Interface (GUI) in a flex application from a Mobile Device. (Description about how legacy applications have been enabled for applying Web services technology has been described in Appendix A)

A registered user could access the main page for design, after authentication by entering their username and password within a welcome page. The user could give all the optimisation parameters through the interface in the Flex page via Mobile devices. The parameters captured by the mobile program in client are first packed in XML format, When the user clicks on the submission button, the request information includes the parametric data will be sent to the Web Service provider in SOAP message so that the service provider could receive the information and a C++ algorithm program located on the server will be invoked. A flag data representing the status of program execution is produced by the program and retrieved by the flex procedure that keeps updating the state of progress bar. Therefore, the user can view the execution progress state of the program. When the execution is completed and output files are created, the Web Servers will pass the results back to the client in XML format.

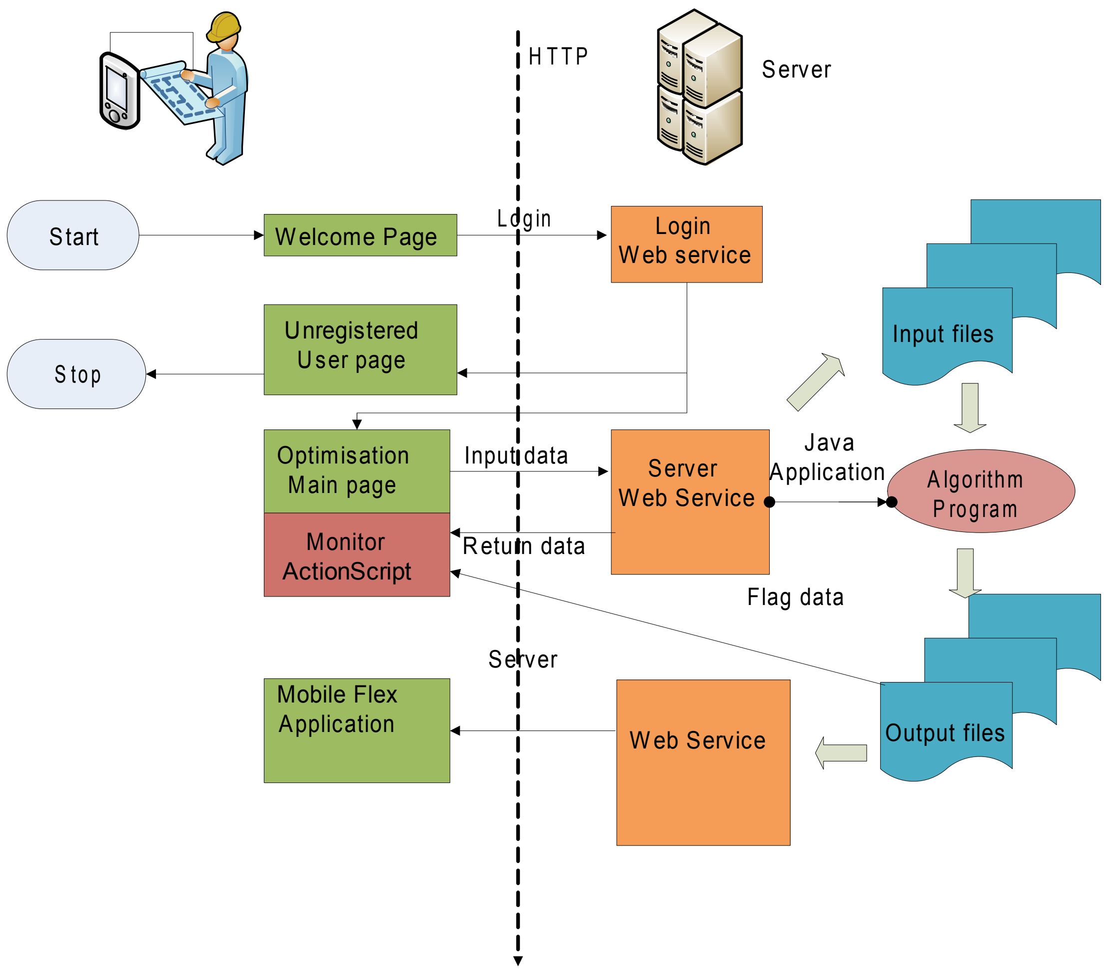


Figure 8.12 Hybrid architecture of the system

The gear optimisation package resides and runs on the server machine. This is considered to be a secure approach since the program is not downloaded to the user's machine and the client has no access to the source code. The existing gear optimisation package is of large size, time-consuming and platform-dependent. The execution of it on the server depends only on the power and platform and not relative to the client-side device. On the other hand, Flex applications are platform- independent, it is possible for users to run the package from any mobile device whether its OS is Mobile Unix, Windows Mobile or Nokia Symbas.

8.2.3 Experiment

Below is the Software and Hardware speciation in the test:

- Tomcat 5.6, Axis 1.4 and JDK5.0 to provide the Web server and create the Web Services.
- Flex builder 3 to create mobile client SWF application; eclipse utilized to write the java Web Services.
- Nokia PDA: its OS is Maemo (operating system), which is a modified version of D ebian GNU/Linux; Memory, 64MiB of DDR RAM, and 128MiB of internal FLASH memory; CPU running at 252 MHz.

8.2.3.1 Login Function:

Figure 8.13 is the login frame when the application in client is started. Login function of Flex application is designed to check the user identity, including username and password, to ensure that only a legal user is allowed to access the optimisation package from the different Mobile devices.

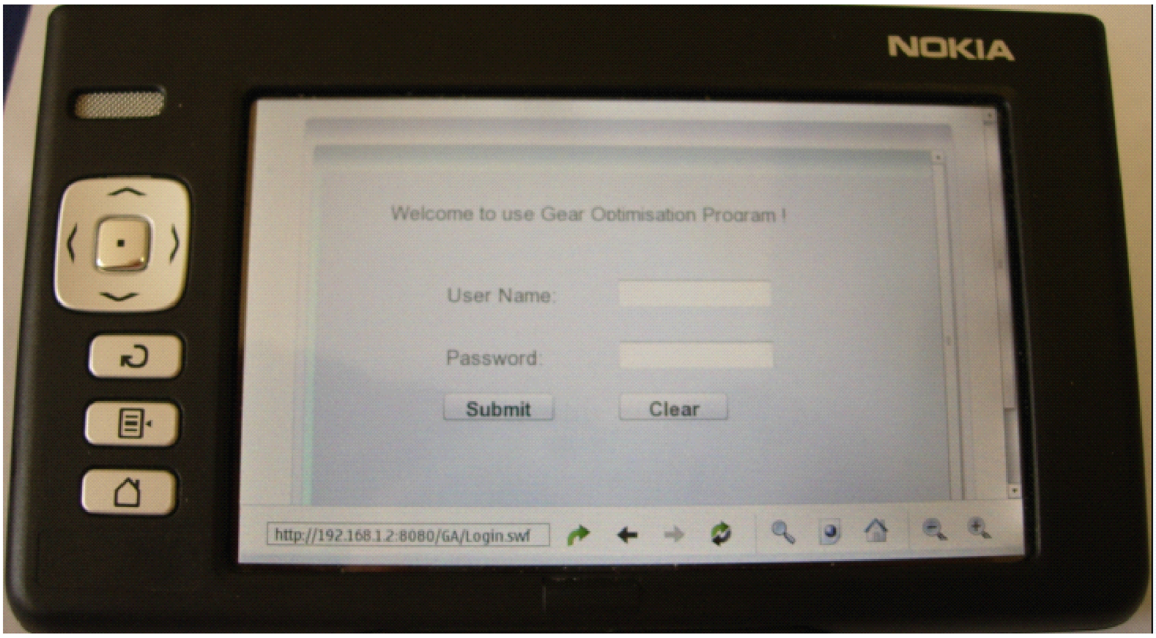
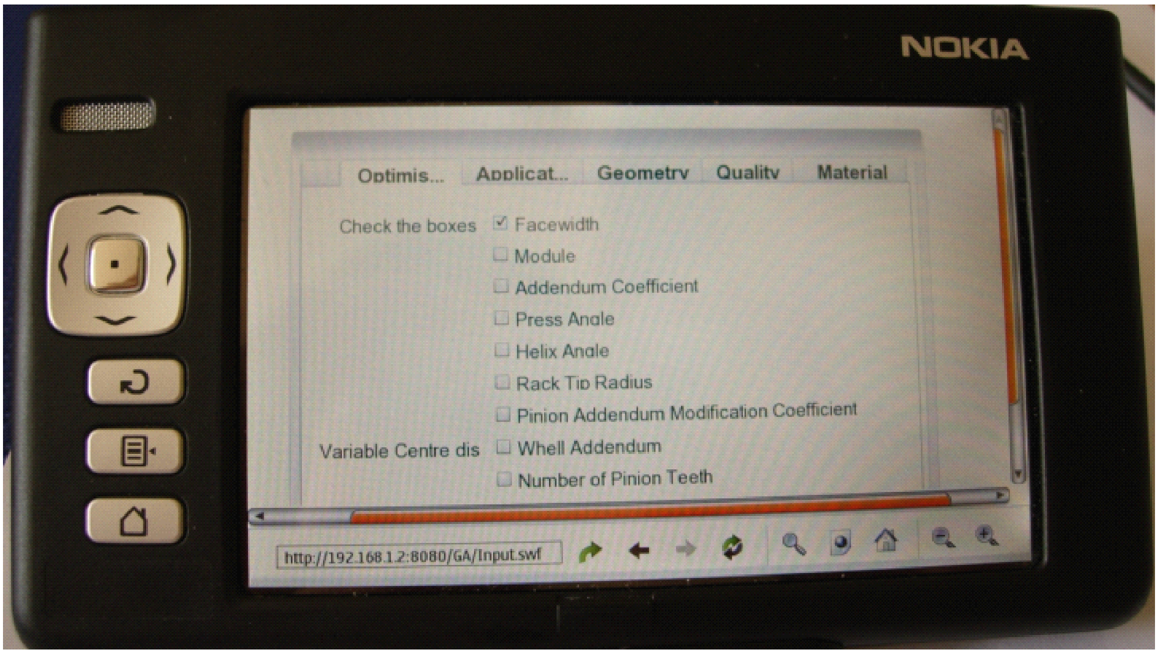


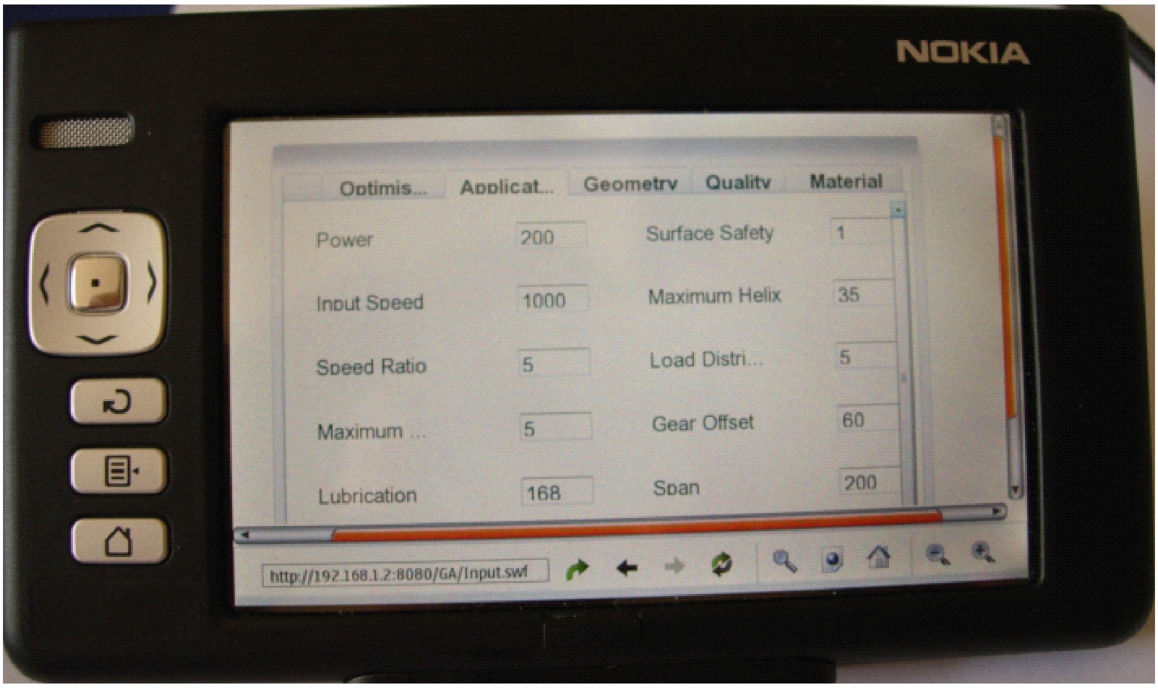
Figure 8.13 Login MXML

8.2.3.2 Mobile Flex Application Communication with Web Server

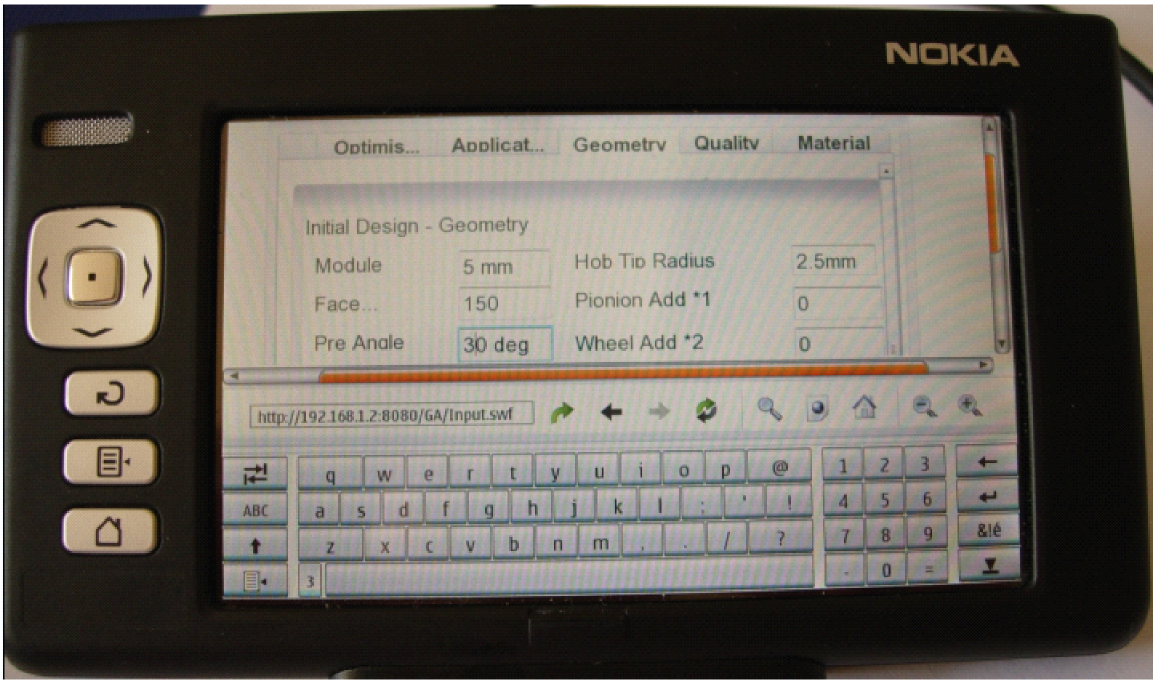
After login into the system, the main frame Figure 8.14 (a-e) comes out and the mobile user could input parameters in "Optimization Parameters" panel. The gear design optimisation model consists of six objectives, nine design variables, and twenty-four inequality constraints. The details and formula of the GA design variable could be referred to the Ji's PhD Thesis [66].



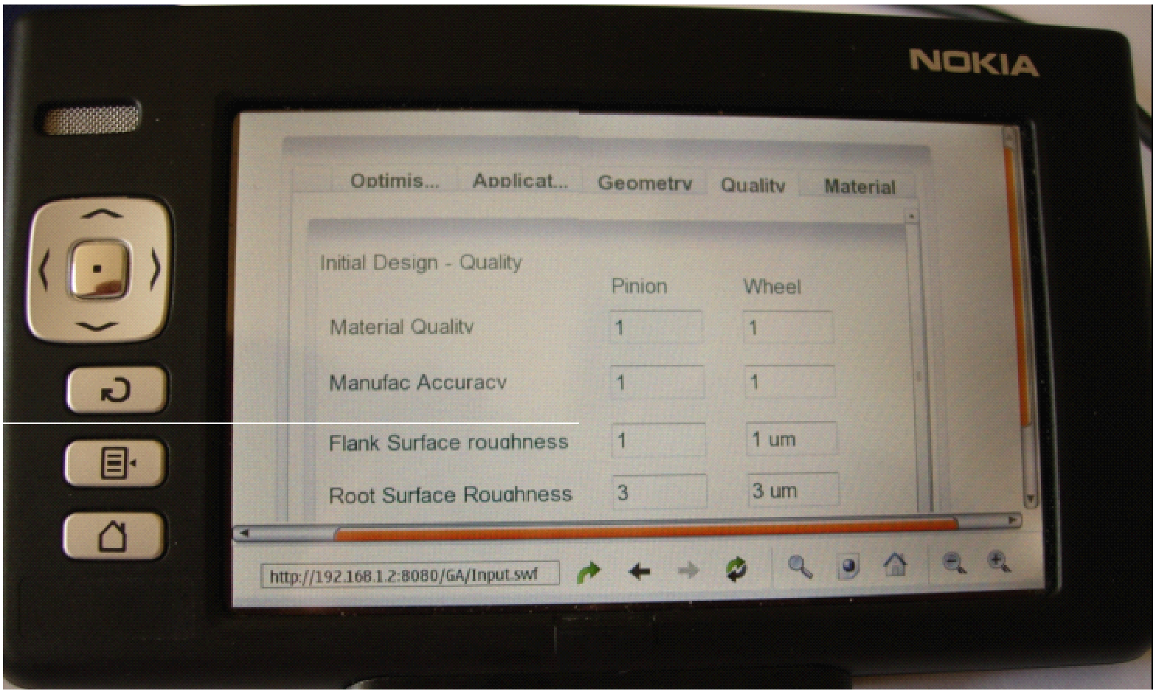
(a) Input MXML: Optimisation – Parameters



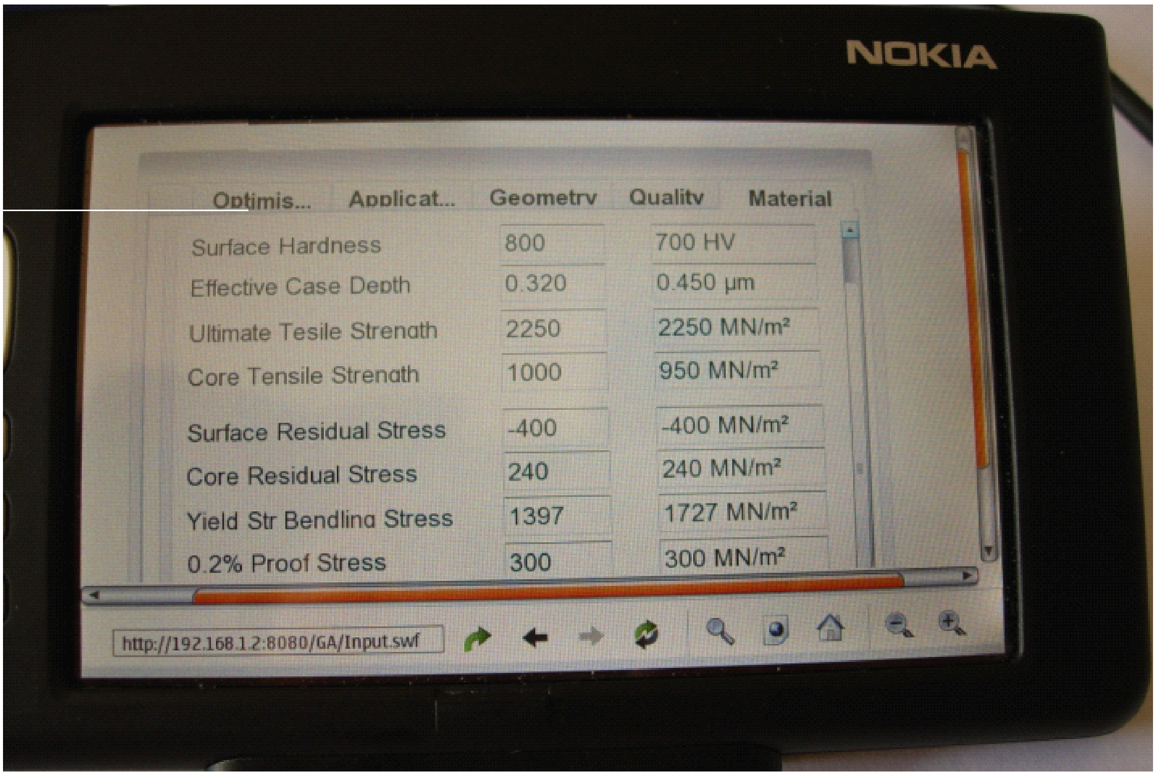
(b) Input MXML: Initial Design – Application



(c) Input MXML: Initial Design – Geometry



(d) Input MXML: Initial Design – Quality



(e) Input MXML: Initial Design - Material

Figure 8.14 Mobile Flex application-Input Parametric Data

The mobile requestor use the WSDL description to find out how to access the service, system can do this by setting the *wsdl* attribute of the `<mx:WebService>` element to the URL of the WSDL file. Then the mobile application selects the target service and port by setting the *service* and *port*attributes of the `<mx:WebService>` element. Figure 8.14 shows where to find the values of these attributes in the WSDL definition. Flex mobile application uses a Web service with operations using the Document-Literal style, the operation style is defined in the style attribute of the <soap:operation> eleme nt inside the binding (XML tags prefixed by soap are part of the SOAP Binding namespace and are used inside WSDL definitions). Its value can be document for a Document Oriented operation (Red line in Figure 8.15).

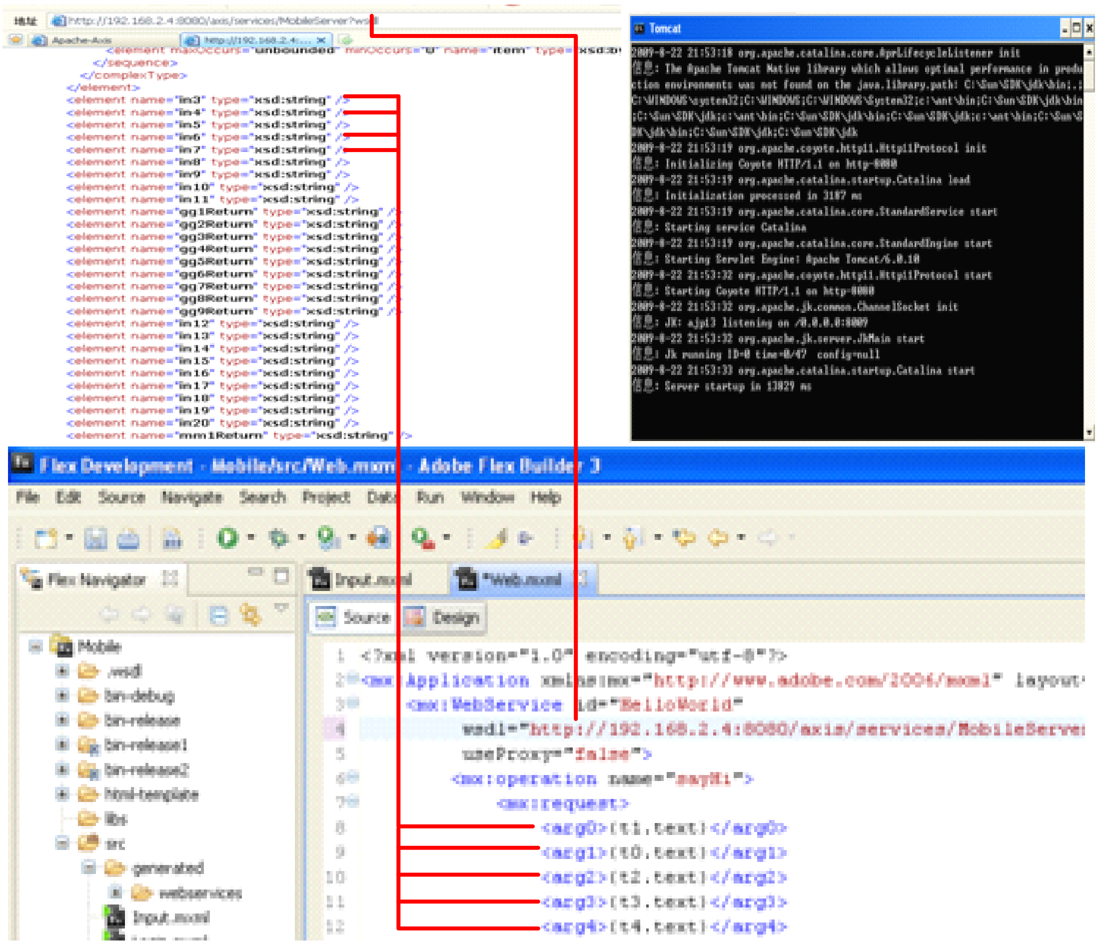


Figure 8.15 Flex Mobile Application Mapping Web service: Document Oriented – Literal (Top: WSDL XML file, below: MXML code)

8.2.3.3 Data Retrieval and display Result

All the data files that are produced by the GA program are saved on the server. The mobile application needs to access these files either for the progress bar or the graphs. Mobile application request data from a server in XML format based on HTTP GET params use mx:HTTPService (Figure 8.16). The URLLoader class downloads data from a URL as text, binary data, or URL-encoded variables. It is useful for downloading text files, XML, or other information to be used in a dynamic, data-driven application. An URLLoader object downloads all of the data from a URL before making it available to code in the applications.

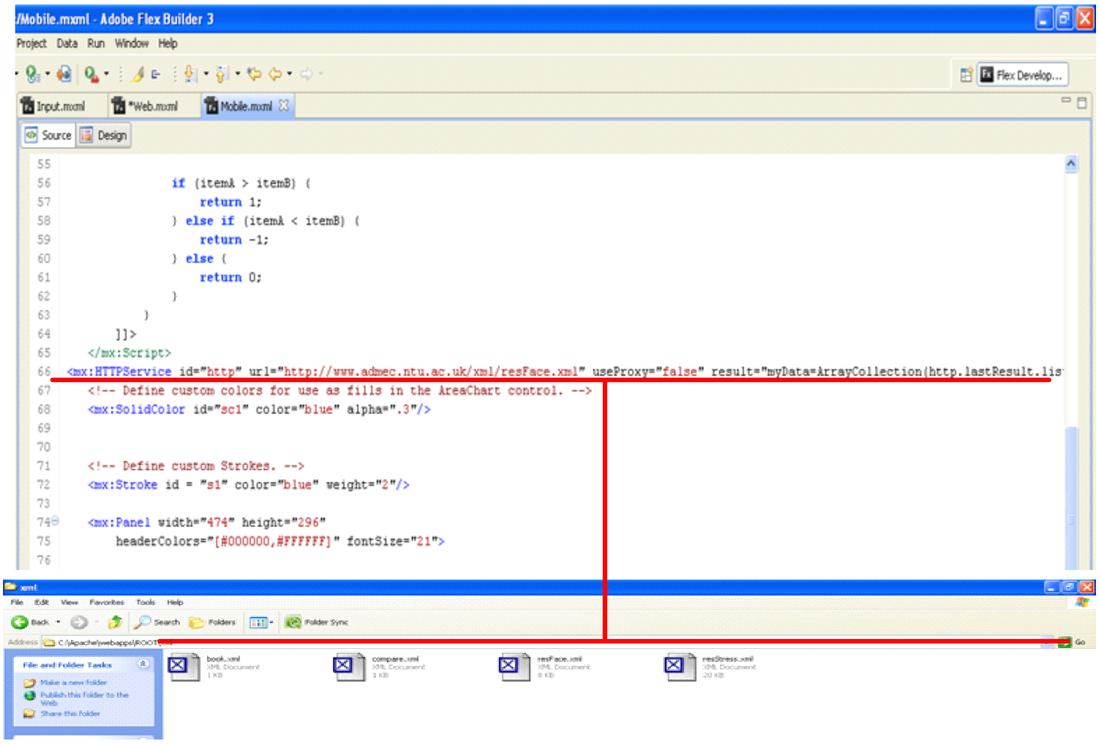
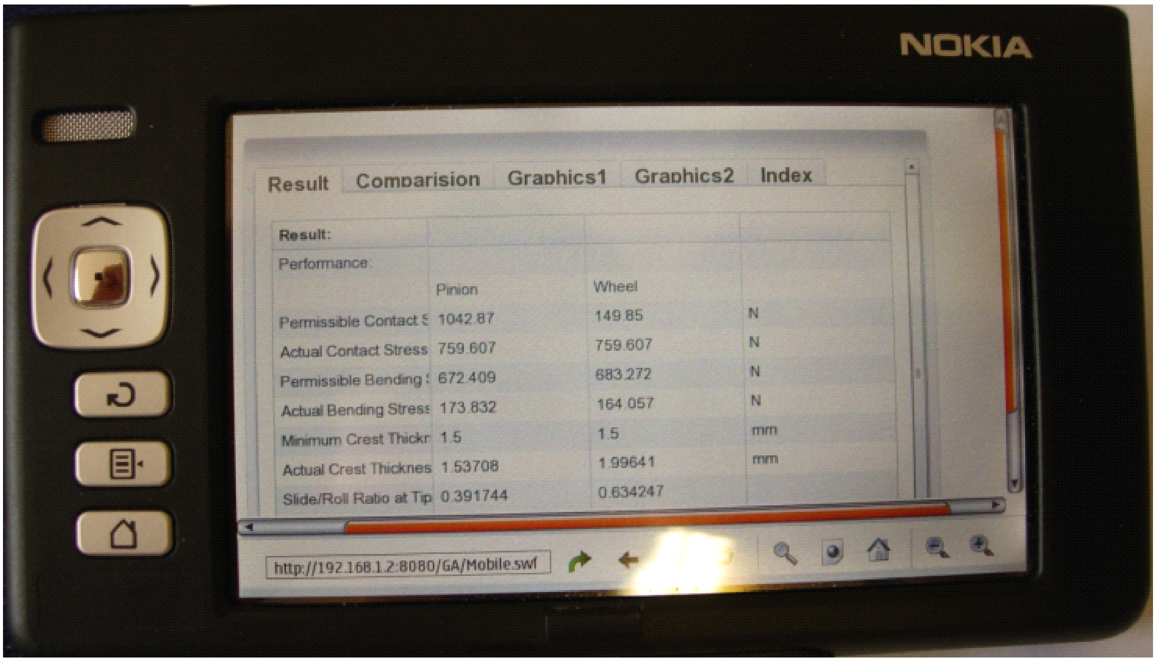
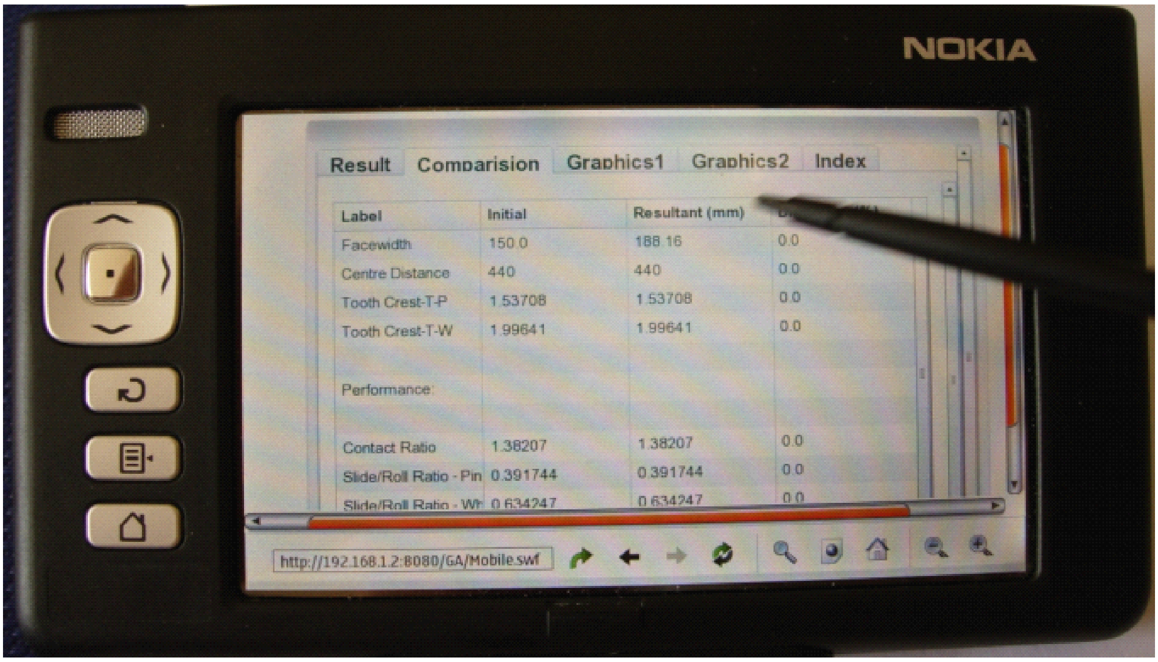


Figure 8.16 Mobile Flex Application Retrieve XML data from Web server (red colour line)

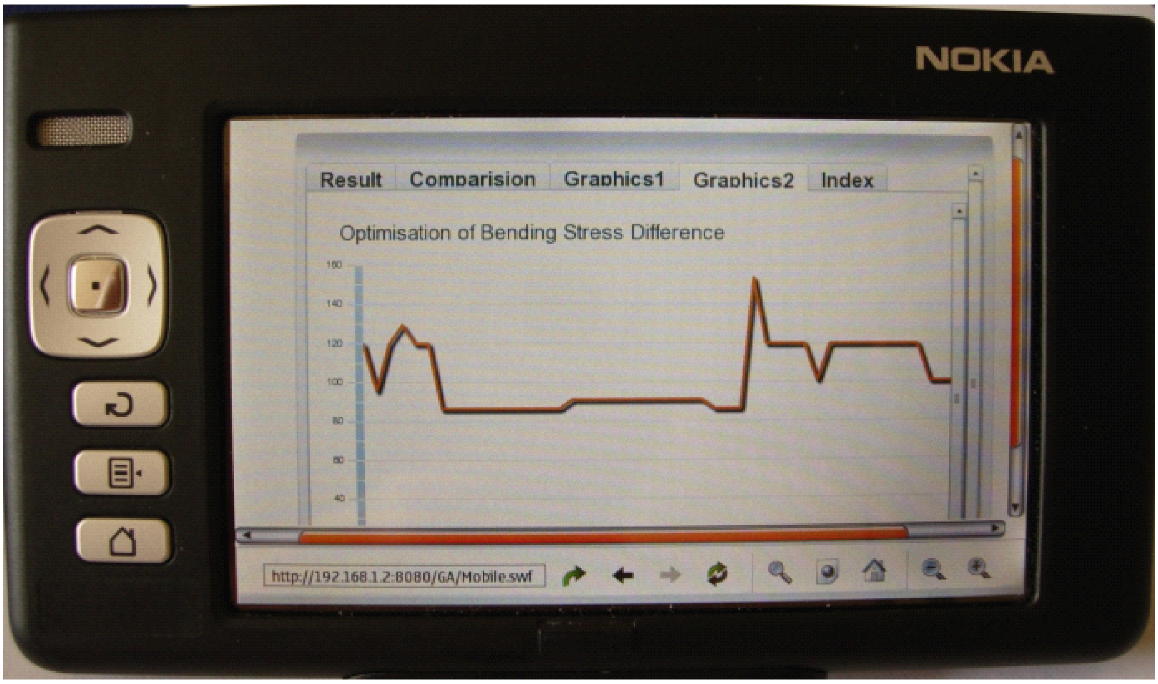
The mobile user also has the option to display the resultant graph of the optimisation. The performance of the designs is given to indicate the trend of the search and the levels. An example of the graph is shown in Figure 8.17 (c).



(a) Result MXML



(b) Comparison MXML



(c) Graphics of Difference

Figure 8.17 Mobile Flex application-Result data and graphic

8.3 Evaluation and Discussions

The Mobile system is applied to Power Station and provides a technical solution to allow people involved in product design to collaborate over the Wireless Environment throughout the design process. WSO-MCAD provides a feasible way to effectively deploy, search and retrieve the product data via mobile devices.

The utilization of Web Services improved the performance in response to the multi-user environment. The successful application of SOA provides a wider application region. As demonstrated in this chapter, various tasks can be implemented based on Web server by Web Services, such as sending different kinds of existing design resources onto the Internet and implementing the remote design of cross-region, communicating commercially with product design resources and enlarging the Internet service of a design.

The combination of Web Services and Flex applications and the necessary modification of the gear design program make it possible to master the execution of such a large-sized and time-consuming program, to provide the powerful processing of the results and to display the results graphically to the user via wireless network. Integrating Flex clients with server-side technologies to invoke a Large Size of Program, the additional benefit of the Web service connectivity model is its support for SOAP faults. In Web services, faults are the means by which a failed attempts to invoke a remote method does not result in a program error but, instead, returns an error to the client, allowing graceful handling of remote error conditions. In this way, if the remote service is unavailable for any reason, the user experience need not be compromised.

The test of the two case studies has been successful; however, the drawbacks can not be neglected. Firstly, evaluation about how well the ontology matches the knowledge in the domain is weak. Secondly, system need to construct a personal model for each individual user to predict his preference on the service activities, there is a need for a set of rules, decision trees, or a set of numerical values representing the corresponding weights of the activities. More discussions and future works will be described in Chapter 9.

Chapter 9
Discussions and Conclusions

9.1 Discussion on Contributions

The overall aim of this project was to construct a Wireless Web-based system to support effective collaborative product design for mobile users through the combination of multiple technologies, including Web Services Technology, Parametric Design, Semantic Web Technology, Agent Technology, Mobile Technology and Flex Technology. The Web Service has given a good interaction platform between human and computer programs, while providing a better environment for interaction among the programs themselves. In addition, Semantic Web Technology was utilized to create a general knowledge base. This research also introduced a mobile agent system for collaborative design, which is implemented and based on Java technology, the system supported the construction of mobile agents which can migrate and access the distributed design resources within the network. Discovering that mobile application could not be well supported by different mobile platforms, in order to resolve the Mobile Platform Compatibility problem, this research stated the latest Flex technology to enable the mobile application to be executed on different mobile devices; furthermore, with the support of combined Flex technology and Web service, remote users could also invoke a large-scale computing program via Mobile Web browser.

The combination of all the variations of technologies provides cornerstone and effective support in building scalable, extensible, and interactive mobile systems for collaborative product design, illustrated by case studies presented in this research. The outcome of this research contributes to the approaches to the future globalization of product design. The following sections will highlight the contributions and main features of this PhD project:

9.1.1 Contributions to Knowledge

The main contribution of this research is in constructing WSO-MCAD system for MCWE. The resultant Web-based system includes several structural and technical contributions. These contributions can be summarised as follows:

- **Compatibility:** This innovative three-layer Mobile Collaborative Working Environment framework for product design, developed in this research utilizes open standards and provides the loose-coupled infrastructure so that it achieved the compatibility among different mobile collaborative systems. This approach also archived the interoperation between heterogeneous applications. This contribution is important for the enterprises, because existing computer systems/applications are always developed to fit one enterprise or organization. They are separated from each other and cannot work together; and companies have different interests and priorities that form a barrier when there is a requirement of collaboration among them. Applying the WSO-MCAD three-layer framework into the system design could resolve the collaborative work problem better.

- **Reusability:** The service-oriented wrapping method provides a new and effective solution, the advanced feature of service oriented program, interaction between services, has been fully explored in this research. The key advantage of using Web services is the ability to create applications through the use of loosely coupled, reusable software components. This has fundamental implications in technologies and business applications. Software can be re-delivered and divided into as fluid streams of services as opposed to packaged products. It is possible to achieve automatic and dynamic interoperability between systems to accomplish business tasks. A SOA based system in this thesis was tested for a large computing program to be run remotely via the Internet such that the geographically dispersed designers can implement a design on a Web browser and obtain the multiple formats of results from any part of the world. A flag data that represents for the status of program execution and Flex application are used to implement the monitoring of the program. The Web service is used to provide the direct communication between GA legacy algorithm application program and Flex client program.

9.1.2 Technical Contributions

- **Parametric Product Design for Mobile System:** As a recent development of the CAD technology, parametric design is more popular in commercial CAD packages such as Pro/E, AutoCAD and SolidWorks. After applying the Semantic Web technology into collaborative design process, the system enables designer to get the OWL files for the product data, the designer can find the relations between the parametric data. The mobile designer can change, receive and send the geometry model by revising the design parametric data, because all the information required could be provided by mobile input and the basic graph of the geometry model are stored with semantic relationships.

- **Semantic Web Technology into Mobile Information System:** An appropriate semantic information system is used, which is one of the most important factors that can affect overall performance and the characteristics of the Semantic Information System; the emerging design guideline for novel discovery solutions is the adoption of semantic technology; which permits explicit representation of interacting entities, e.g., services, resources and users, at a high level of abstraction while enabling automated reasoning about this representation, thus favouring interoperability and mutual understanding between entities sharing little or no prior knowledge about each other.

- **Flex Applications within MCWE-Mobile Platform Compatibility:** Compared with Java Me and Microsoft Silverlight Technologies, Flex application is a good choice to be implemented in the mobile devices via the author's experience. Rich Internet Applications into adobe Flex is the cross browser and platform compatibility, Flex applications works in either Adobe Flash player or as desktop applications using Adobe Integrated Runtime (AIR). Combined with middleware technology, such as Web service, Flex uses the existing WSDL definitions by referencing a Uniform Resource Locator (URL) where the document is located, in order to exchange messages between SOAP service endpoints.

9.2 Conclusions

9.2.1 Web Service Oriented Mobile 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. However, the literature survey reveals that although the progress of mobile technologies on wireless networks has largely changed the way people access Internet information; a small number of researchers have been involved in mobile computing for the collaborative product design. To resolve the problem, this research applied Service Oriented technologies and SOA to provide solutions, 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 Pro/E, and the direct connection among software is not necessary with Web service support during collaboration.

SOA provides a good solution to future distributed environment; the features addressed in the infrastructure include seamless collaboration, virtuosity, autonomy and security. SOA 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.

Another thing must be noted is the system is based on open source tools like Tomcat and Axis, which could be downloaded without worrying about the copyright.

9.2.2 Application of Semantic Web Technology into Mobile Collaborative Design

Chapter 5 described a mobile information management system based on the Semantic Web Technology, this approach includes two components: (1) ontology is used to represent abstract views of the product data and (2) added semantic rules are used to represent relationships between individuals. Therefore, to meet the demands of a mobile information system for collaborative product design, an ontology-based description model is thus proposed to facilitate expression and organisation of product information in order to manage and deploy the distributed design resources. A reliable knowledge base can be established using inference power and Mobile devices are used as tools for the users to communicate/collaborate with each other via the Web Server. Product data including CAD drawings, materials, manufacturing process, supplier chain, etc. can be effectively shared between the users.

Furthermore, semantic similarity approach is introduced to make the processing system understand the schema and semantics of concepts and to measure the strength of similarity between the data values. The values of SS can be used as a measure to determine the rank of each answer, which will help the users to find more related information back.

9.2.3 Agent Technology for WSO-MCAD

As the mobile applications are generally operated in an open environment in which resources and repositories are distributed in different machines and the network connection is unreliable, this research presents an agent system for collaborative design, which is developed based on JADE, supports the construction of mobile agents or mobile agent-oriented systems with the dynamic loading class for various applications. This research based on intelligent agents gives the system a proactive behaviour. The results are based on a design example of an application operating in a mobile environment, there need a common framework where different system can share their heuristic information on available resources.

The multiagent-based system adopts intelligent agents as a technology for tasks distribution and resource management in distributed system. The system enables the mobile users from different companies to share and retrieve information among distributed resources within the agent system; mobile agent actually migrates to the server to make a request directly, rather than over the network. When the mobile agent needs data or access to a resource that the server provides, it doesn't communicate with the server over the network. Instead, the mobile agent migrates to the server's machine. When the entire transaction is complete, agent system will send the results back. Agent mobility is the ability for an agent program to migrate or to make a copy (clone) itself across one or multiple network hosts.

9.2.4 Mobile Flex Application for WSO-MCAD

This thesis describes Flex technology and its advantages compared with Microsoft Silverslight and Java Me. In order to develop and deploy mobile applications in different mobile platforms, the system chooses Flex builder 3 to develop the Platform compatibility applications, which can run across most of current popular Mobile OS, such as Symbian OS, Windows Mobile, Mobile Linux, etc. The RIAs into adobe Flex is the cross browser and platform compatibility, Flex applications runs in either Adobe Flash player or as desktop applications using Adobe Integrated Runtime.

This thesis also presents two methods to deploy Flex Application into the mobile device; one is depend on the latest mobile Web Browsers Technology and the another one is to install Adobe Flasher 9 plug-in. Mobile users could use Nokia PC Suite and/ or Microsoft ActiveSync Tools to install Adobe Flash plug-in or mobile Web browser; Flex mobile application could extend the capabilities of the mobile client with the richer Internet applications in real time.

9.2.5 Case Studies: Mobile Applications for Product Design

One case study was conducted, which utilized the full capability of the Mobile Collaborative Working Environment. In this case study, the framework was utilized in Power Station and provided a technical solution to allow people involved in product design to collaborate over the Wireless Environment throughout the design process. The product data retrieved from the companies were applied to be tested in WSO-MCAD, that also could be an experiment to prove the mobile system is valid and could be applied into theses factories in the future.

Mobile CAD software can only be installed in the Windows Mobile Devices; this research presents a second case study to describe Flex applications, which could be implemented across the different mobile platforms. The method of combination of Web Services and Flex applications has been used to master the execution of a large-sized and time-consuming program, to provide the powerful processing of the results and to display the results graphically to mobile user via wireless network.

9.2.6 Limitations of the Research

As indicated in the earlier chapters, the limitations of this research are.

- The WSO-MCAD should have the real-time interaction feature, but in fact, it has latency problem, current Web service protocol does not solve the latency issue efficiently;

- The WSO-MCAD did not consider well the concurrent control among the participants. And feedback from users suggested that the setting up of environment for the system to be operated is complicated.

- The CeCAD support DWG file well, and ShortCAD can not open DWG file, it support DXF and ShortDWG file; thus, the collabrotive work between the two mobile CAD applications will lead to file conflicted problem;

- Case study about the WSO-MCAD has been successful, but the evaluation about how well the ontology matches the knowledge in the domain is weak.

9.3 Future Work

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

- **Development of a more advanced WSO-MCAD:** A more universal approach to support the communication among various software packages is required. This research needs to be further investigated and developed to strengthen the author's theory. The Standard for the Exchange of Product Model Date (STEP) is a comprehensive ISO (International Organization for Standardization) standard (ISO 10303) that describes how to represent and exchange digital product information [92]. The future research will integrate STEP into this system so that it could provide sophisticated communication among different CAD applications, when they operate the files/databases using different formats.

- **Combined Agent Technology and Semantic Web service:** Semantic Web technologies can provide a standardised way of interpreting context, enabling both human and software agents to infer new context knowledge and consequently take intelligent actions. Agent technology has been recognized as a promising paradigm for the next generation manufacturing systems, a mobile agent could directly visits the server to communicate with other agents and to access the resources, avoiding the mass data transmission in the network and reducing the dependence on the network band width, thus reducing the time spent on communication. The mobile agents can encapsulate a task, dispatch it through the network, and then gets the results; in a situation that network is temporarily disconnected, it can wait for the request from the client. Web services are the main enabling middleware techniques in collaborative design of this research, which manage the complexity and heterogeneity inherent in distributed systems. But the future mobile system need focus on how to automatically compose existing Web services into more complex services ; such system need to present an overall structure of a Semantic Web-based Mobile Environment, Web Services will be defined and semantically deployed by OWL-S specifications, which provide a high level description of the services capabilities. All the intelligent process need to be carried out by the different Agent within the Web server, combined Agent and Semantic Web service is a good solution to design the new System Architecture.

- **Knowledge acquisition and Ontology evaluation:** Knowledge acquisition includes the elicitation, collection, analysis, modelling and validation of knowledge for knowledge engineering and knowledge management projects. Future work focus on developing semantic methods in ubiquitous computing environments, especially services for knowledge acquisition and knowledge sharing; current research in ontology evaluation is mainly concerned with the grammar rather than the content.

- **User context:** As is mentioned in section 8.3, regarding the work of personalization, the most important issue is to construct a personal model for each individual user to predict his preference on the ser vice activities. This model may be a set of rules, decision trees, or a set of numerical values representing the corresponding weights of the activities.

References

[1] R. Sudarsan, S. J. Fenves, R. D. Sriram and F. Wang, A product information modeling framework for product lifecycle management, Computer-Aided Design, 37(13), 1399-141, November 2005

[2] F. He, S. Han, A method and tool for human–human interaction and instant collaboration in CSCW-based CAD, Computers in Industry, Volume 57, Issues 8-9, December 2006, pp 740-751

[3] D. Su, Y. Xiong, S. Ji and Y. Zheng, Framework for a Collaborative Working Environment, International Journal of Production Research, 46(9), 2008

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

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

[6] D. Su, J. Li,Y. Xiong and Y. Zheng, Application of Internet Techniques into Online Collaborative Design and Manufacture, The 9th International Conference on Computer Supported Cooperative Work in Design Proceedings, 2005.

[7] Y. Zheng, L. Arthur, C. Li and D. Su: Proceedings of the 8th International Conference on Frontiers of Design and Manufacturing, September 23~26, 2008, Tianjin, China.

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

[9] X. Chen, T. Luo, Y. He, W. Zhou, and D. Sun, 2005, Research on 3D Feature Modeling technology for Internet-driven Collaborative Design, 9th International Conference on Computer Supported Cooperative Work in Design, 24-26 May 2005, Coventry, UK, pp 649-654.

[10] W. LI, Z. Feng, Research on Ontology-based Description Model of Part Information, Computer Engineering, 33(8), 250-252, 2007

[11] D. Su, S. Ji, N. Amin and J. B. Hull, Multi-user Internet environment for gear design optimisation, Integrated Manufacturing Systems, Vol. 14, No.6, 2003, pp 498-507, MCB UP Limited

[12] I. Mendikoa, M. Sorli, Barbero, J. Carrillo and A. Gorostiza, Inventive approach for problems detection in collaborative design, proceedings, the International conference on Advanced Design and Manufacture, 8 – 10

January, 2006, Harbin, China, pp 424-428.

[13] http://en.wikipedia.org/wiki/Product_design (Access on 29/07/09)

[14] http://electronicdesign.com/Articles/ArticleID/6819/6819.html (access on 11/08/2008)

[15] http://en.wikipedia.org/wiki/Distributed_Component_Object_Model (access on 22/10/2009)

[16] G. Boothroyd, Product design for manufacture and assembly, Computer-Aided Design, Volume 26, Issue 7, July 1994, Pages 505-520.

[17] Y.C. Kao and G. C. I. Lin, Computer Integrated Manufacturing Systems Vol.9.No.3.pp.149-160,1996.

[18] N. Shyamsundar and R. Gadh, Internet-based collaborative product design with assembly features and virtual design spaces, Computer-Aided Design 33 (2001) 637-651.

[19] S. C. Lu, A. B. Rebello, R. A. Miller, G. Kinzel and R. Yagel ,A simple visualization tool to support concurrent engineering design,Computer-Aided Design, Volume 29, Issue 10, October 1997, Pages 727-735.

[20] G. Q. Huang, S. W. Lee and K. L. Mak, Collaborative product definition on the Internet: a case study, Journal of Materials Processing Technology,139 (2003) 51-57.

[21] D. Xue, Y. Xu, Web-based distributed system and database modelling for concurrent design, Computer-Aided Design, 35(2003) 433-452

[22] Y. W. Bai, Z. N. Chen, H. Z. Bin and J. Hu, Collaborative design in product development based on product layout model, Robotics and Computer-Integrated Manufacturing 21 (2005) 55-65.

[23] D. Tang, An agent-based collaborative design system to facilitate active die-maker involvement in stamping part design, Computers in Industry 54 (2004) 253-271.

[24] G. Renner and A. Ekárt, Genetic algorithms in computer aided design, Computer-Aided Design, Volume 35, Issue 8, July 2003, Pages 709-726.

[25] Pralay Pal, A. M. Tigga and A. Kumar , Feature extraction from large CAD databases using genetic algorithm, Computer-Aided Design, Volume 37, Issue 5, 15 April 2005, Pages 545-558.

[26] K. S. Lee, K. Lee, Framework of an evolutionary design system in corporation design information and history, Computers in Industry 44(2001) 205-227.

[27] Y. D. Wang, W. Shen and H. Ghenniwa, a Web/agent-based multidisciplinary design optimization environment, Computers in Industry 52 (2003) 17-28.

[28] M. Leslie, An integrated CSCW architecture for integrated product/process design and development, Robotics and Computer-Integrated Manufacturing 15 (1999) 145-153.

[29] Y. Li, X. Shao, P. Li, and Q. Liu, Design and implementation of a process-oriented intelligent collaborative product design system, Computers in Industry 53 (2004) 205-229.

[30] M. Rosenman and F. Wang, A component agent based open CAD system for collaborative design, Automation in Construction 10 (2001) 383-397.

[31] U. Roy, S. S. Kodkani, Product modelling within the framework of the World Wide Web. IIE Trans 1999;31:667-77
[32] T. FEH, A. Roy, CyberCAD: a collaborative approach in 3D-CAD technology in a multimedia-supported environment, Computer Ind, 2003;52;127-45.
[33] W. D. Li, W. F. Lu, J. Y. H. Fuh and Y. S. Wong, Collaborative computer-aided design-research and development status, Computer-Aided Design 37 (2005) 931-940.
[34] H. Y. Kan, V. G. Duffy and C. J. Su, An Internet virtual reality collaborative environment for effective product design, Computers in Industry,45(2001) 197-213.
[35] F. Eng, H. Tay, A. Roy, CyberCAD: a collaborative approach in 3D-CAD technology in a multimedia-supported environment, Computers in Industry, 52 (2003) 127-145.
[36] D. Su, J. Li,Y. Xiong and Y. Zheng, Application of Internet Techniques into Online Collaborative Design and Manufacture, The 9th International Conference on Computer Supported Cooperative Work in Design Proceedings, 2005, pp 655-660.
[37] Y. Xiong, J. Liu, P. Fitzgerald and D. Su, Service Oriented Software Package Bank, The 9th International Conference on Computer Supported Cooperative Work in Design Proceedings,2005.
[38] D. Su and Y. Zheng, Development of a Prototype Mobile Collaborative Environment for Product Design, Expanding the Knowledge Economy: Issues, Applications, Case Studies, 749-756, Amsterdam 2007.
[39] H. Wang, Z. Huang, Y. Qu, J. Xie, Web services: problems and future directions, Web Semantics: Science, Services and Agents on the World Wide Web, Volume 1, Issue 3, April 2004, Pages 309-320
[40] M. Younas, I. Awan and D. Duce, An efficient composition of Web services with active network support, Journal of Expert Systems with Applications, Vol. 31, Issue 31, November 2006, Pages 859-869
[41] Y. Xiong and D. Su, A Web-Service Based Approach for Software Sharing', in: W Shen et al (ed.), Lecture Notes in Computer Science, Volume 3865/2006 ,Computer Supported Collaborative Work in Design II, ISBN 0302-9743, Springer Berlin/Heidelberg, pp 215-224
[42] M. Athanasopoulos, H. Ugail and G. G. Castro, Parametric design of aircraft geometry using partial differential equations, Advances in Engineering Software, Volume 40, Issue 7, July 2009, Pages 479-486
[43] H. li, J. WANG, Design and implementation of mobile agent testing platform, Computer Engineering and Desing, Vol 27, No.12, 2006.6
[44] J. Huang, S. Luo, W. Zhang, Structure of Mobile Agent based on Java, Railway computer application, Vol.13.No.2, 2004.04
[45] Grady and Hare, leading towards a new generation of applications and services, Information Sciences, 171 (2005) 335-353
[46] W. Lee, Deploying personalized mobile services in an agent-based environment, Expert Systems with Applications, 32, 194-1207, 2007.

[47] Y. Ling, P. Long, Design and Implementation of Embedded Mobile Agent, computer engineer and Design, 2006.21
[48] Q. H. Mahmoud, L. Yu, Havana agents for comparison shopping and location-aware advertising in wireless mobile environments, Electronic Commerce Research and Application 5(2006) 220-228.
[49] O. Hao, W. Shen, Z. Zhang, S. Park, and J. Lee, Agent-based collaborative product design engineering: An industrial case study, Computers in Industry, 57(1), Issue 1, 26-38, 2006
[50] S. Parka, W. Kimb and I. Ihm, Mobile collabrative medical display system, Computer methods and programs in biomedicine 89 (2008) 248–260
[51] D. G. Bar, Overcoming mobile device limitations through adaptive information retrieval. Applied Artificial Intelligence, 18, 513–532,2004)
[52] J. Jaehun, S. M. Lee, Adoption of the Semantic Web for overcoming technical limitations of knowledge management systems, Expert Systems with Applications, Volume 36, Issue 3, Part 2, April 2009, Pages 7318-7327
[53] http://www.w3.org/standards/semanticweb/ (access on 18/08/2008)
[54] L. Liao and K. Xu, Constructing intelligent and open mobile commerce using a semantic web approach, Journal of Information Science, 2005, 31(5), 407 – 419.
[55] S. C. Brandt, J. Morbach, M. Miatidis, M. Theißen, M. Jarke and W. Marquardt, An ontology-based approach to Knowledge management in design processes, Computers and Chemical Engineering, Volume 32, Issues 1-2, Janu ary 2008, pp 320-342
[56] X. Chang and J. Terpenny ,Ontology-based data integration and decision support for product e-Design, Robotics and Computer-Integrated Manufacturing, In Press, Corrected Proof, Available online 31 May 2009
[57] Y. Hu, S. Li, M. Guo, Ontology-Based Product Knowledge Representation, Journal of Computer-Aided Design& Computer Graphics, 15(12), 2003
[58] M. N. Huhns, M. P. Singh, Ontologies for agents, Internet Computing, IEEE(6), 81 83, 1997
[59] X. Yang, W. Min, MA Qiang, Research on Jena-based Ontology Building, Computer Engineering, 33(14), , 59-61, 2007
[60] http://www.joseki.org/ (access on 01/08/2009)
[61] http://www.aifb.uni-karlsruhe.de/about.html (access on 01/08/2009)
[62] http://projects.kmi.open.ac.uk/akt/ontoweaver/ (access on 01/08/2009)
[63] http://www.swed.co.uk/swed/index.html (access on 01/08/2009)
[64] http://www.canalys.com/pr/2008/r2008112.htm (Access on 30/07/09)
[65] http://en.wikipedia.org/wiki/Microsoft_Silverlight (access on 10/08/2009)
[66] S. Ji, Web-based Collaborative Environment for Integrated Design, PhD Thesis, 2006, Nottingham Trent Univesity
[67] http://en.wikipedia.org/wiki/J2ME_Wireless_Toolkit (access on 10/08/2009)
[68] D. Xu, T. Liu, A web-enabled PDM system in a collaborative design environment, Robotics and Computer Integrated Manufacturing 19 (2003) 315-328.

[69] Y. P. Zhang, C. C. Zhang and H. P. B. Wang, An Internet based STEP data exchange framework for virtual enterprises, Computers in Industry, 2000, 41(2000):51-63.
[70] http://www.adobe.com/products/flex/features/flex_builder/
[71] http://www.oasis-open.org/home/index.php (access on 30/07/2009)
[72] C. Y. Tsai, T. H. Sun, and J. X. Huang, A Web-based XML information sharing system for collaborative product development, International, Journal of Production Research; 8/1/2006, Vol. 44 Issue 15, pp 2955-2976,
[73] H. C. Chenga, and C.S. Fenb, A Web-based distributed problem-solving environment for engineering applications, Advances in Engineering Software, Vol. 37, Issue 2, February 2006, pp 112-128
[74] D. Zhang and B. Adipat, Challenges, methodologies, and issues in the usability testing of mobile applications, International Journal of Human–Computer Interaction, 18(3), 293–308. 2005
[75] http://www.shortcad.com/index.php?option=com_content&view=frontpage&Itemid=53&lang=en (access on 07/08/2009)
[76] http://www.graebert.co.uk/index.php?option=com_cad_uk&Itemid=57&id=Mobile (access on 10/08/2009)
[77] http://www.soapui.org/ (access on 16/08/2008)
[78] A. Budanitsky and G. Hirst, EvaluatingWordNet-based Measures of Lexical Semantic Relatedness, Computational Linguistics, Vol 32, No 1, 2006, pp.13-47.
[79] S. Li, J. Zhang, H. Xiong, Journal of Computer Science and Technology, Vol 17, No 6, 2002, pp.933-939.
[80] P. Li, T. Lan, W. Zuo, Computer Eng and Design, Vol 28, No1, 2007, pp.227-229.
[81] O. Kwon and M. K. Shin: Expert Systems with Applications, Vol 34, 2008, pp. 2966－2975.
[82] X. F. Zha, A knowledge intensive multi-agent framework for cooperative/collaborative design modelling and decision support of assemblies, Knowledge-Based System 15 (2002) 493-506.
[83] FIPA. The Foundation for Intelligent PhysicalAgents.http://www.fipa.org/, Accessed October 30th 2002
[84] W. Shen, Q. Hao, H. J. Yoon, and D. H. Norrie, Applications of agent-based systems in intelligent manufacturing: An updated review, Advanced Engineering Informatics, 2006, 20(4), 415-431.
[85] W. Lee, Deploying personalized mobile services in an agent-based environment, Expert Systems with Applications,1 194-1207,32, 2007
[86] http://jade.tilab.com/ (access on 07/07/2008)
[87] http://www.fipa.org/ (access on 01/08/2008)
[88] http://labs.adobe.com/technologies/flex/ (access on 11/08/2009)
[89] http://www.knowyourmobile.com/blog/274851/adobe_flash_10_beta.html (access on 10/08/2009)
[90] http://www.knowyourmobile.com/blog/274851/adobe_flash_10_beta.html

(access on 10/08/2009)

[91] S. Ji, D. Su, J. L. Henshall and J. B. Hull, Gear design optimisation using a Genetic Simulated Annealing Algorithm, poster proceedings, International Conference on Adaptive Computing in Design and Manufacture, 20 - 22 April 2004, Bristol, UK, pp 5-9

[92] G. L. Smith, Utilization of STEP AP 210 at The Boeing Company, Computer-Aided Design, Volume 34, Issue 14, 1 December 2002, pp 1055-1062

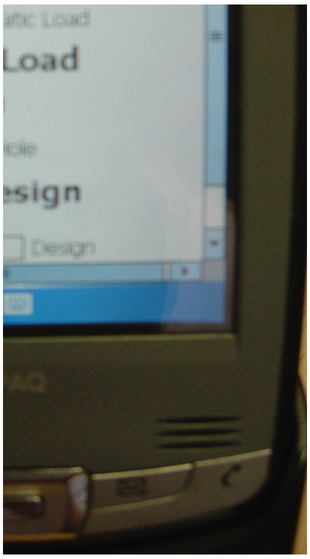
Appendix A:
Mobile Devices Implementation within MCWE

A.1: Different Mobile Operations Systems and Devices

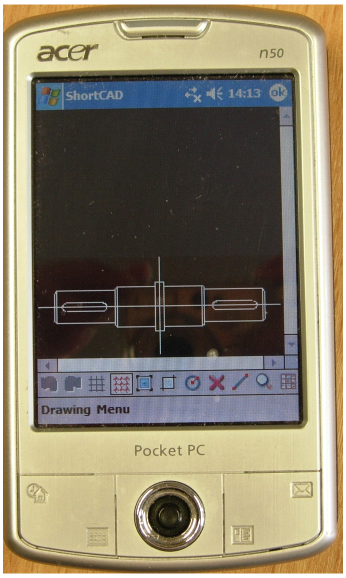
All of the Mobile Devices support Wi-Fi WLAN（Wi－Fi）denotes a set of Wireless LAN/WLAN standards developed by working group 11 of the IEEE LAN/MAN Standards Committee (IEEE 802).

1. **HP Smartphone/PDA:** The OS is Microsoft Windows Mobile 6.1, Qualcomm 7201A Processor 528 MHz; 128 MB SDRAM main memory for running applications, 256 MB flash ROM, Skyfire was installed in this device and such mobile Web browser are used to test Platform Compatibility for Flex application in chapter 7.

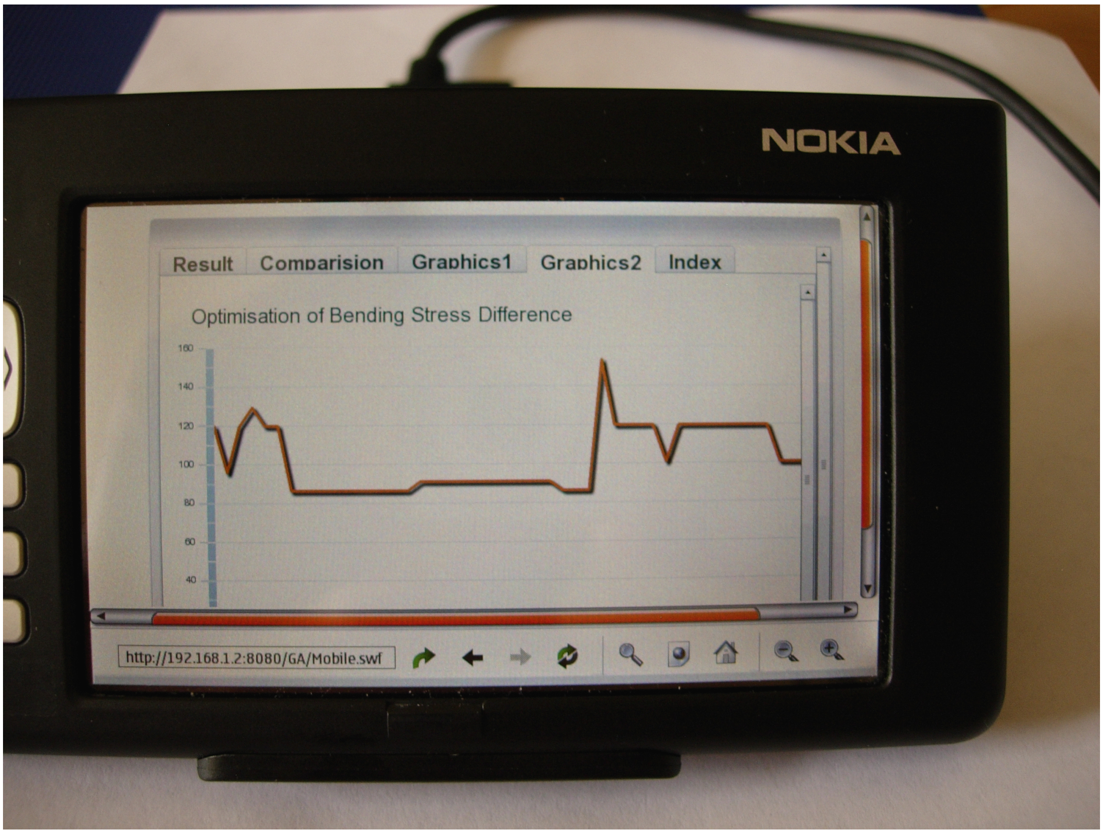
2. **HP PDA**: The OS is Windows Mobile 5.0; this device was used to test Microsoft VB.Net mobile client application for WSO-MCAD in chapter 4, 5, 6 and 8.



3. **Acer PDA:** The OS is Windows Mobile 5; this device was used to test the CAD file transferred between PDA and Web server in chapter 4.



4. **Nokia PDA:** The OS is Maemo (operating system), which is a modified version of Debian GNU/Linux; CPU running at 252 MHz, Memory, 64MiB of DDR RAM, and 128MiB of internal FLASH memory. The Nokia PDA, installed flash 9 plug-in, was used to test Flex mobile application in chapter 8.



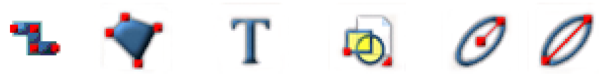
5. **Netgear Rooter:** Netgear rooter provides the wireless Network for test of mobile system, the transfer speed is 54Mb/s.

A.2: Mobile CAD Packages

A.2.1 ShortCAD

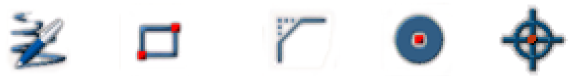
ShortCAD could be installed in Windows Mobile devices, when using ShortCAD, the designer can create and edit 2D vector-graphic drawings just as they do it in well known CAD applications on the PC. ShortCAD saves designer's drawings as .Short DWG files, but designer can also save them in .dxf format so that they can export them to other vector-graphic editors. ShortCAD provides drawing by means of the following graphic entities:

Line Polyline Circles Arcs



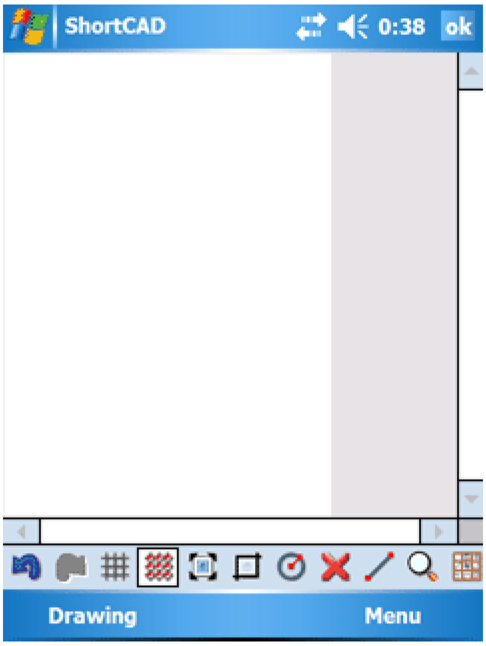
Traces Solids Editable Texts Inserted blocks Ellipses

More complex objects based on mentioned basic graphic entities available as well:



Free lines Rectangles Chamfers Donuts Points

Each command of ShortCAD is represented by separate icon on the Command Palette:



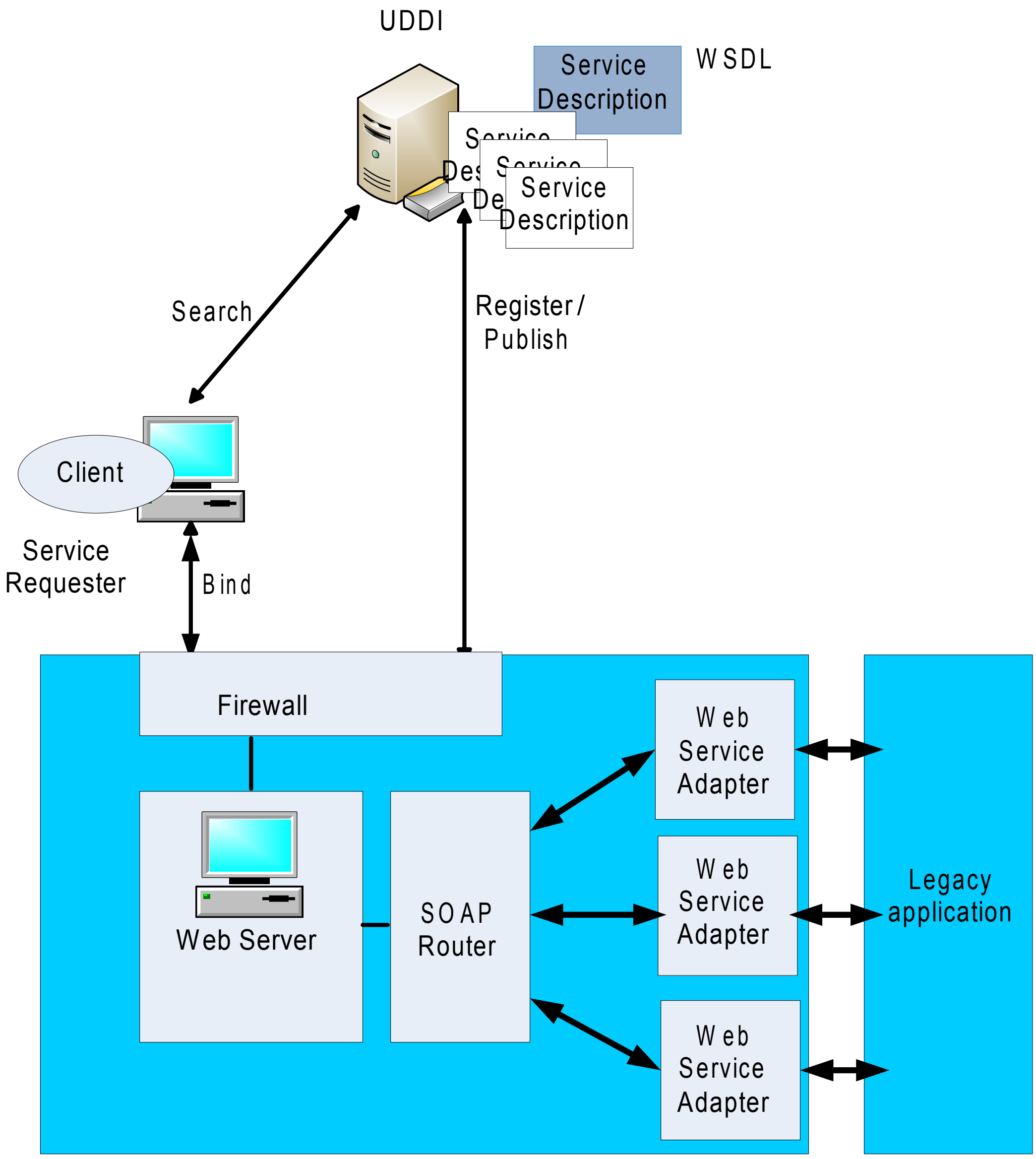
A.2.2: PowerCAD CE (CeCAD)

PowerCAD CE is the first mobile CAD application, which could be installed in Windows Mobile devices. Over 300 AutoCAD compatible drawing functions including lines, polylines, arcs, circles, rectangles, ellipses, offsets, and more. DWG, DXF, and DWF file export is also possible. With support for Lisp and C/C++ languages, PowerCAD CE Pro allows for the limitless extension or creation of tailor-made applications such as automated GPS/Interior Laser data collection, graphics-enabled sales automation systems, location based field services, or automated productivity extensions maximized for end-user simplicity and operation.

A.3 Legacy System Reusability

The cost of developing a system is an inevitable limiting factor. The re-use of legacy systems is a methodology for reducing the high cost of software development and maintenance. It is necessary to build an environment which can accommodate many existing design and manufacturing applications without the necessity of rewriting the essential code.

The Web services architecture is an ideal technology to incorporate enterprise legacy applications into this new area. This approach could be used to create a Web-browser-based user interface that interacts with an enterprise application in order to enable program-to-program communication. Figure below is the SOA approach which allows the legacy applications to be made available for the web.



The architecture covers three components of the Web service: service provider, service request, and service broker. The service requestor is the consumer of a Web service and is most likely a program running on the server of a business customer. The service provider program gets the information concerning available services from a UDDI repository, the service broker. The available Web services are described in the Web Services Description Language (WSDL) to support platform-neutral communications.

When the service requestor has selected a service, it will use the WSDL description to find out how to access the service. Once found, the WSDL description is used to generate a SOAP request message that is sent to the application server, which acts as the service provider. The Web service adapter has to be developed for each service to enable them for services access. It is typically a Java application program that connects to the backend server. This connection can be any communication link supported by the backend server.

Appendix B: The Sourcode of Main Functions

This appendix has five sections including the: Mobile VB.Net Code, Flex Mobile Code, Code of Web Server, Semantic Web Code and J2ME Mobile Code. The sourcecode is explained by the author with the each section.

C.1 The Mobile VB.Net Code

The Mobile VB.Net has four functions, these are the: Semantic Search, Mobile Design, Mobile XML data and Mobile Network, Figure C.1 is the main class diagram of the Microsoft VB.Net mobile client application.

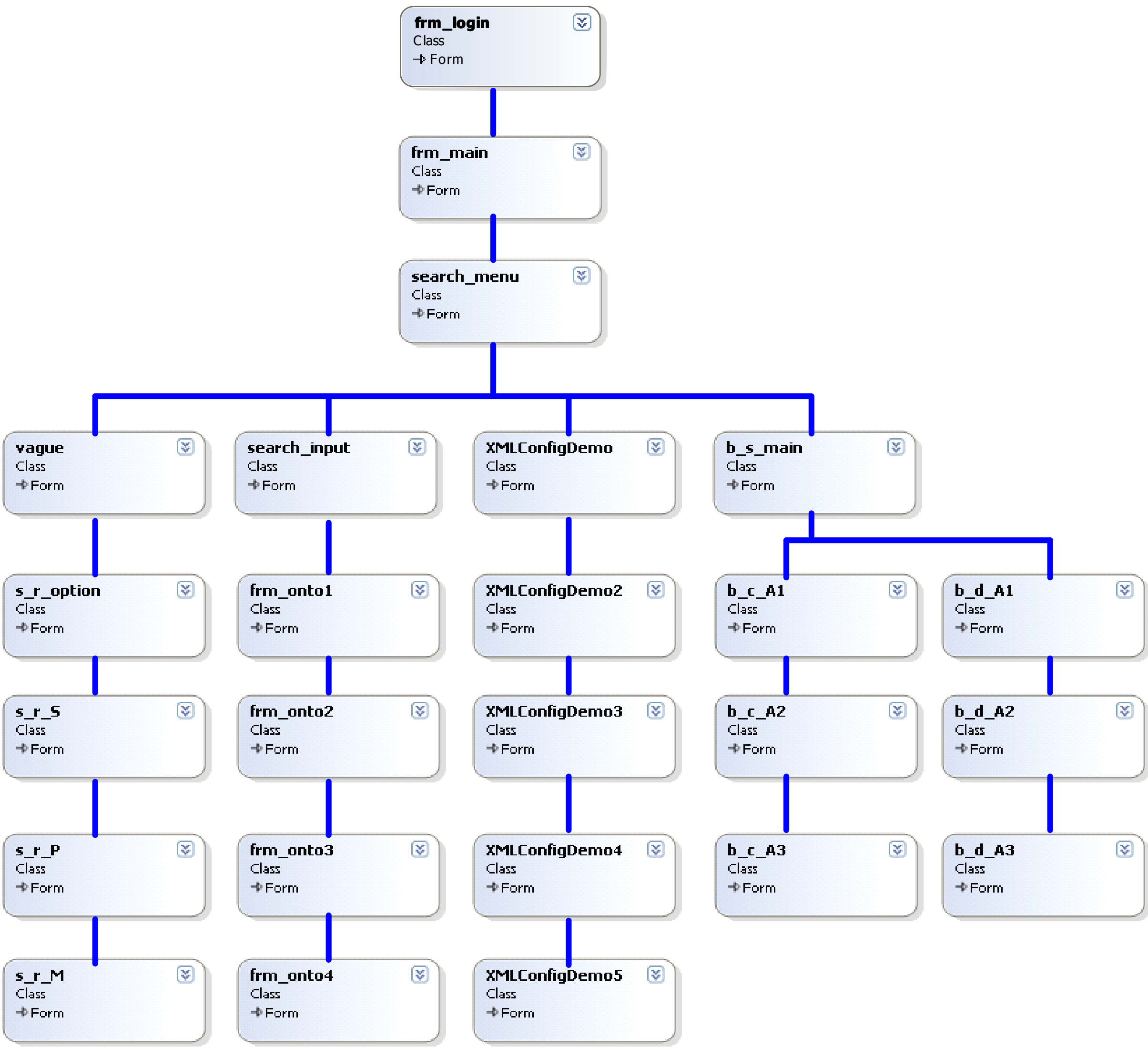


Figure C.1 Main Classes Digaram of VB.Net Code

C.1.1 The Mobile Design Function:

The Check Bolt class: The Engineer needs to input these factors via mobile application interface, when the computed stress is less than or equal to permissible stress, the bolt is satisfied for this condition. Application of this class could be found in page 137.

```
Public Class b_c_A1

    Private Sub d1_SelectedItemChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles d1.SelectedItemChanged
        If d1.Text = "3.6" Then             ' data from the OWL bolt property
       …
        End If
    End Sub

    Private Sub d2_SelectedItemChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles d2.SelectedItemChanged
        If d2.Text = "M10" Then         'data from OWL property
    …
    End Sub

    Private Sub ButtonBack_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ButtonBack.Click
        If Cursor.Current Is Cursors.WaitCursor Then
            Exit Sub
        End If
        Cursor.Current = Cursors.WaitCursor
        Dim frm_tmp As New b_s_main
        frm_tmp.Show()
        Cursor.Current = Cursors.Default
    End Sub

    Private Sub ButtonNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
ButtonNext.Click
        Dim ws As New MobileServer.MobileServerService()  'call Web service
        Dim i1, i2, i3, i4, i5, i6 As Double
        Dim r As Integer
        r = 1

        …

        ws.g1(g1)                                'send data to the Web server
```

```
        ws.g2(g2)
        ws.g3(g3)

        If Cursor.Current Is Cursors.WaitCursor Then
            Exit Sub
        End If
        Cursor.Current = Cursors.WaitCursor
        Dim frm_tmp As New b_c_A2
        frm_tmp.Show()
        Cursor.Current = Cursors.Default
    End Sub
End Class
```

On the other hand, if the engineer wishes to assemble these parts and select a suitable bolt, he can choose the design mode, set all of the necessary factors and press calculate button, the permissible stress and min diameter will be calculated, finaly all the available bolts meet the requirement will be shown in the mobile devices. The application of this class could be found in page 140.

```
Public Class b_d_A1

    Private Sub d1_SelectedItemChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        If d1.Text = "3.6" Then                    'data from OWL bolt property
            …
    End Sub

    Private Sub ButtonBack_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButtonBack.Click
        If Cursor.Current Is Cursors.WaitCursor Then
            Exit Sub
        End If
        Cursor.Current = Cursors.WaitCursor
        Dim frm_tmp As New b_s_main
        frm_tmp.Show()
        Cursor.Current = Cursors.Default
    End Sub

    Private Sub ButtonNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButtonNext.Click

   Dim ws As New MobileServer.MobileServerService()    'call Web service
        Dim i1, i2, i3, i4, i5, i6 As Double
        Dim r As Integer
```

```
        r = 1
          …
        ws.g1(g1)
        ws.g2(g2)
            ws.g3(g3)

        If Cursor.Current Is Cursors.WaitCursor Then
            Exit Sub
        End If
        Cursor.Current = Cursors.WaitCursor
        Dim frm_tmp As New b_d_A2
        frm_tmp.Show()
        Cursor.Current = Cursors.Default
    End Sub
End Class
```

C.1.2 The Semantic Search Function:

The Class Search_input: In this function, the client inputs the relative words for pointing the product; the search engineer then searches the relative ontology library on the Web and displays the results in the mobile device. Application of this function in page 90.

```
Public Class search_input
…
    Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        If Cursor.Current Is Cursors.WaitCursor Then
            Exit Sub
        End If
        Cursor.Current = Cursors.WaitCursor
        Dim frm_tmp As New search_menu
        frm_tmp.Show()
        Cursor.Current = Cursors.Default
    End Sub

    Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

      Dim ws As New MobileServer.MobileServerService()    'call Web service
        Dim i1, i2, i3, i4, i5, i6 As Double
        Dim r As Integer
```

```
        r = 1

        i1 = t1.Text
        i2 = t2.Text
        i3 = stress.Text
        i4 = m.Text
        i5= t5.Text
        i6=t6.Text

        Dim g1, g2, g3,g4,g5,g6 As String

        g1 = i1
        g2 = i2
        g3 = i3
        g4= i4
        g5= i5
        g6=i6

        ws.g1(g1)
        ws.g2(g2)
        ws.g3(g3)
        ws.g4(g4)
        ws.g5(g5)
        ws.g6(g6)

        If Cursor.Current Is Cursors.WaitCursor Then
            Exit Sub
        End If
        Cursor.Current = Cursors.WaitCursor
        Dim frm_tmp As New search_boltInput
        frm_tmp.Show()
        Cursor.Current = Cursors.Default
    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged

    End Sub
End Class
```

View Output: this class is to view the image file, which will be produced by the service provider.

```
Public Class s_r_P
```

```
    Dim address As String

    Private Sub ButtonC_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButtonC.Click
Dim ws As New MobileServer.MobileServerService()     'call Web service
        Dim sAddress As String
        sAddress = tbAddress.Text.Trim
''''''''''''''''''''''''''''''''''''add web service
        address = ws.gl(gl)                          ''''''''''''''''''''''add web service
        If String.IsNullOrEmpty(address) Then Return
        If address.Equals("about:blank") Then Return
        If Not address.StartsWith("http://") And _
            Not address.StartsWith("https://") Then
            address = sAddress & address
        End If

        Try
            Web1.Navigate(New Uri(address))
        Catch ex As System.UriFormatException
            Return
        End Try
    End Sub
    Private Sub Web1_Navigated(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.WebBrowserNavigatedEventArgs) Handles Web1.Navigated
        tbAddress.Text = Web1.Url.ToString()
    End Sub

    Private Sub ButtonNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButtonNext.Click
        If Cursor.Current Is Cursors.WaitCursor Then
            Exit Sub
        End If
        Cursor.Current = Cursors.WaitCursor
        Dim frm_tmp As New s_r_option
        frm_tmp.Show()
        Cursor.Current = Cursors.Default
    End Sub

    Private Sub ButtonBack_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButtonBack.Click
        If Cursor.Current Is Cursors.WaitCursor Then
            Exit Sub
        End If
```

```
        Cursor.Current = Cursors.WaitCursor
        Dim frm_tmp As New s_r_option
        frm_tmp.Show()
        Cursor.Current = Cursors.Default
    End Sub
```

C.1.3 The XML Data Function

This class is to retrieve the data via XML file, system use mobile agent to send the XML data back to the remote device. Application of this function in page 103.

```
Public Class XMLConfigDemo

    Private Sub btnLoad_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnLoad.Click
        ' return Pocket Pc application path
        Dim FilePath As String
        FilePath =
System.IO.Path.GetDirectoryName(Reflection.Assembly.GetExecutingAssembly().GetName().CodeB
ase.ToString())
        FilePath = FilePath & "\XMLResultFile.xml"
        tbName.Text = getXmlSettingValue(FilePath, "Ver", "Name", "")
        tbVersion.Text = getXmlSettingValue(FilePath, "Ver", "Version", "")
        tbBuildDate.Text = getXmlSettingValue(FilePath, "Ver", "Date", "")
        tbAuthor.Text = getXmlSettingValue(FilePath, "Startup", "Author", "")

          …
```

C.1.4 The Mobile Network

The Mobile Network is for mobile users to exchange instant message via the wireless the network, this technology used in this module are based on the TCP/IP, such module design and implementation are similar as those used MSN.

Client function code:

```
Function SendMessage(ByVal myMessage As String) As String
        Dim Client As Sockets.TcpClient
        Dim InBuffer(1000) As Byte
        Dim OutBuffer() As Byte
```

```
        Dim ReturnMessage As String

        ' connect server
        Client = New Sockets.TcpClient
        Try
            Client.Connect(txtIP.Text, txtPort.Text)
        Catch ex As Exception
            MsgBox("fail: " & ex.Message)
            Return "[no response]"
        End Try
        ''''''''''''''
        Dim fs As New IO.FileStream("\1.txt", IO.FileMode.OpenOrCreate, IO.FileAccess.Read)
'
        Dim fssize(fs.Length - 1) As Byte
        Dim strread As New IO.BinaryReader(fs) 'binary file
        'fs.Read(fssize, 0, fssize.Length - 1)
        strread.Read(fssize, 0, fssize.Length - 1)
        '  Client.GetStream().Write(fssize, 0, fssize.Length)
        '''''''''''''
        ' create message
        OutBuffer = System.Text.Encoding.Default.GetBytes(myMessage.ToCharArray)
        'send
        Client.GetStream().Write(OutBuffer, 0, OutBuffer.Length)
        ' waiting for response
        While Not Client.GetStream.DataAvailable()
            Application.DoEvents()
        End While
        ' read response
        ReturnMessage = ""
        While Client.GetStream.DataAvailable()
            Client.GetStream().Read(InBuffer, 0, InBuffer.Length)
            ReturnMessage     &=     System.Text.Encoding.Default.GetString(InBuffer,     0,
InBuffer.Length)
        End While
        ' close connection
        Client.Close()
        '
        Return ReturnMessage
    End Function
```

Server code:

```
Protected Sub ProcessRequest()
        Dim Buffer(100) As Byte
```

```
Dim Bytes As Integer
Dim CurrentThread As Threading.Thread
Dim CurSocket As Sockets.Socket
Dim InboundPacket As String
Dim OutboundPacket As String

' get current thread
CurrentThread = System.Threading.Thread.CurrentThread()

' accept socket
CurSocket = Listener.AcceptSocket

' listen to socket
While Not StopListener
    ' check data
    If CurSocket.Available > 0 Then
        Bytes = CurSocket.Receive(Buffer, Buffer.Length, 0)
        SyncLock CurrentThread
            InboundPacket = System.Text.Encoding.Default.GetString(Buffer, 0, Bytes)
        End SyncLock
        Exit While
    End If

    '
    Application.DoEvents()
    If Not CurSocket.Connected Then
        StopListener = True
    End If
End While

' deal with binary package
OutboundPacket = ProcessPacket(InboundPacket) ...
```

C.2 The Flex Mobile Code

C.2.1 The Flex application calling Web service: invoke Web service via Flex application

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application                 xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute">
    <mx:WebService id="HelloWorld"
```

```
        wsdl="http://localhost:8080/spring_http/HelloWorld?wsdl"
        useProxy="false">
       <mx:operation name="sayHi">
          <mx:request>
             <arg0>{t1.text}</arg0>
             <arg1>{t0.text}</arg1>
             <arg2>{t2.text}</arg2>
             <arg3>{t3.text}</arg3
```

C.2.2 Result SWF: Retrive XML data and create graphic in the Flex mobile application

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
   backgroundColor="#FFFFFF"
   backgroundAlpha="0"
   creationComplete="http.send()">

   <mx:Move id="hideEffect"
   xTo="-500" />
   <mx:Move id="showEffect"
   xFrom="500" />
<mx:XML id="tempXML"
   source="http://localhost:8080/xml/compare.xml" />

   <mx:XMLListCollection id="cuePointXMLList"
         source="{tempXML.CuePoint}" />


   <mx:Script>
      <![CDATA[
      import mx.collections.ArrayCollection;


 /////////
      import mx.rpc.events.ResultEvent;
                    import mx.collections.ArrayCollection;
                    [Bindable]
                    public var myData:ArrayCollection;

                    public                                              function
resultHandler(event:ResultEvent):void{
                              //       error.text  =  event.result  as
String;
```

```
                    myData          =          new
ArrayCollection(event.result.shuju.shuiguo);
                    }


            //////
            private   function   parametersLabelFunction(item:Object,
column:DataGridColumn):String {
                return item.Parameters.Parameter.length();
            }

            private  function  numericSortCompareFunction(objA:Object,
objB:Object):int {
                var itemA:Number = parseInt(objA.Time.text()) as Number;
                var itemB:Number = parseInt(objB.Time.text()) as Number;

                if (itemA > itemB) {
                    return 1;
                } else if (itemA < itemB) {
                    return -1;
                } else {
                    return 0;
                }
            }
        ]]>
    </mx:Script>
 <mx:HTTPService                                        id="http"
url="http://www.admec.ntu.ac.uk/xml/resFace.xml"     useProxy="false"
result="myData=ArrayCollection(http.lastResult.list.user)"/>
    <!-- Define custom colors for use as fills in the AreaChart
control. -->
    <mx:SolidColor id="sc1" color="blue" alpha=".3"/>


    <!-- Define custom Strokes. -->
    <mx:Stroke id = "s1" color="blue" weight="2"/>

    <mx:Panel width="474" height="296"
    headerColors="[#000000,#FFFFFF]" fontSize="21">

        <mx:TabNavigator id="viewstack2"
        selectedIndex="0"
        historyManagementEnabled="false"
        width="100%" height="100%">
```

```
            <mx:Form label="Result"
             hideEffect="{hideEffect}"          showEffect="{showEffect}"
fontSize="19">
            </mx:Form>

            <mx:Form label="Comparision"
             hideEffect="{hideEffect}"          showEffect="{showEffect}"
fontSize="36">
                <mx:DataGrid id="dataGrid0"
                        dataProvider="{cuePointXMLList}"
                        width="100%"
                        rowCount="{cuePointXMLList.length       +       1}"
height="199" fontSize="16">
                    <mx:columns>
                        <mx:DataGridColumn id="timeCol0"
                                dataField="Title"
                                headerText="Label"

sortCompareFunction="numericSortCompareFunction" />
                        <mx:DataGridColumn id="typeCol0"
                                dataField="Initial"
                                headerText="Initial" />
                        <mx:DataGridColumn id="nameCol0"
                                dataField="Resultant"
                                headerText="Resultant (mm)" />
                        <mx:DataGridColumn id="parametersCol0"
                               dataField="Difference"
                               headerText="Difference (%)"/>



                    </mx:columns>
                </mx:DataGrid>
            </mx:Form>

            <mx:Form label="Graphics1"
             hideEffect="{hideEffect}" showEffect="{showEffect}">
                <mx:Text text="Optimisation of Facewidth"/>
                <mx:LineChart      id="chart"      dataProvider="{myData}"
width="643" height="327">
        <mx:horizontalAxis>
           <mx:CategoryAxis categoryField="user"/>
        </mx:horizontalAxis>
        <mx:series>
```

```
          <mx:LineSeries yField="name" name="Apple"/>
          <!--<mx:LineSeries yField="name" name="Orange"/>-->
          <!--<mx:LineSeries yField="name" name="Banana"/>-->
       </mx:series>
    </mx:LineChart>
           </mx:Form>

           <mx:Form label="Graphics2"
            hideEffect="{hideEffect}" showEffect="{showEffect}">
               <mx:FormItem id="lab1" horizontalAlign="center">
                   <mx:Text    text="Optimisation    of    Bending    Stress
Difference"/>
               </mx:FormItem>
               <mx:LineChart     id="chart2"     dataProvider="{myData}"
width="628" height="326">
       <mx:horizontalAxis>
          <mx:CategoryAxis categoryField="user"/>
       </mx:horizontalAxis>
       <mx:series>
          <mx:LineSeries yField="name" name="Apple"/>
          <!--<mx:LineSeries yField="name" name="Orange"/>
          <mx:LineSeries yField="name" name="Banana"/>-->
       </mx:series>
    </mx:LineChart>
           </mx:Form>

    <mx:Form label="Index"
            hideEffect="{hideEffect}"         showEffect="{showEffect}"
fontSize="21">
               <mx:Text text="Thanks for your support!" />
           </mx:Form>


       </mx:TabNavigator>

    </mx:Panel>
</mx:Application>
```

C.3 The Code of Web Server

C.3.1 VB Parametric Design Source code

Parametric Design VB code will be implemented in the Web server; application of this function is in page 68.

```
Public Function ExecuteSQL(ByVal SQL As String, MsgString As String) As ADODB.Recordset
    Dim conn As ADODB.Connection
    Dim rst As ADODB.Recordset
    Dim STokens() As String
    'Dim ConnectString As String
    On Error GoTo ExecuteSQL_Error
    STokens = Split(SQL)
    Set conn = New ADODB.Connection
    conn.Open ConnectString
    If InStr("insert,delete,update", UCase$(STokens(0))) Then
        conn.Execute SQL
        MsgString = STokens(0) & "query successful"
    Else
        Set rst = New ADODB.Recordset
        rst.Open Trim$(SQL), conn, adOpenKeyset, adLockOptimistic
        Set ExecuteSQL = rst
        MsgString = "search" & rst.RecordCount & "count"
    End If
ExecuteSQL_Exit:
    Set rst = Nothing
    Set conn = Nothing
    Exit Function
ExecuteSQL_Error:
    MsgString = "search error：" & Err.Description
    Resume ExecuteSQL_Exit
End Function
Public Function ConnectString() As String
   ConnectString = "driver={Microsoft Access Driver (*.mdb)};pwd=;dbq=c:\result.mdb"
End Function
'" & App.Path & "\Res\Data.mdb

Pub_var:
Option Explicit
Public Fe As Double      'work load
Public Femin As Double   '
Public Fo As Double   '
Public Fr As Double   '
Public Fa As Double   '
Public LS_XH_Flg As Integer
Public isLS_XH As Boolean
Public strFeLX As String
```

```
Public Const PI = 3.1415926
Public Flg_Draw As Boolean
Public isSWStart As Boolean
Public Status As String
Public strMod As String
Public SavePath As String
Public swFileName As String
Public ExName As String
Public isSave As Boolean

''''''''''LS''''''''''''
Public LS_Atv As Boolean
Public LS_Flg As Integer
Public LS_SQL As String
Public LS_d As Single
Public LS_k As Single
Public LS_r As Single
Public LS_s As Single
Public LS_l As Single
Public LS_P As Single
Public LS_b As Single

''''''''''''LZ''''''''''''
Public LZ_Atv As Boolean
Public LZ_Flg As Integer
Public LZ_SQL As String
Public LZ_d As Single
Public LZ_x As Single
Public LZ_ds As Single
Public LZ_l As Single
Public LZ_bm As Single
Public LZ_b As Single
Public LZ_P As Single
''''''''''''''KC_LD''''''''''''
Public LD_KC_Atv As Boolean
Public LD_KC_Flg As Integer
Public LD_KC_SQL As String
Public LD_KC_d As Single
Public LD_KC_P As Single
Public LD_KC_b As Single
Public LD_KC_dk As Single
Public LD_KC_k As Single
Public LD_KC_n As Single
```

```
Public LD_KC_r As Single
Public LD_KC_t As Single
Public LD_KC_l As Single
''''''''''''''''NL_LD''''''''''''
Public LD_NL_Atv As Boolean
Public LD_NL_Flg As Integer
Public LD_NL_SQL As String
Public LD_NL_d As Single
Public LD_NL_P As Single
Public LD_NL_b As Single
Public LD_NL_dk As Single
Public LD_NL_k As Single
Public LD_NL_t As Single
Public LD_NL_s As Single
Public LD_NL_e As Single
Public LD_NL_r As Single
Public LD_NL_l As Single
Public LD_NL_lq As Single
'''''''''''''''''''''JD_LD'''''''''
Public LD_JD_Atv As Boolean
Public LD_JD_Flg As Integer
Public LD_JD_SQL As String
Public LD_JD_d As Single
Public LD_JD_P As Single
Public LD_JD_n As Single
Public LD_JD_t As Single
Public LD_JD_dt As Single
Public LD_JD_dp As Single
Public LD_JD_z As Single
Public LD_JD_l As Single
''''''''''''''''''''LM''''''''''''''''
Public LM_Flg As Integer
Public LM_SQL As String
Public LM_D As Single
Public LM_e As Single
Public LM_s As Single
Public LM_m As Single
Public LM_P As Single
'''''''''''''''''''PD_DQ'''''''''''
Public DQ_PD_Flg As Integer
Public DQ_PD_SQL As String
Public DQ_PD_d As Single
Public DQ_PD_d1 As Single
Public DQ_PD_d2 As Single
```

```
Public DQ_PD_h As Single
''''''''''''''''''TH_DQ'''''''''''''
Public DQ_TH_Flg As Integer
Public DQ_TH_SQL As String
Public DQ_TH_d As Single
Public DQ_TH_H As Single
Public DQ_TH_S As Single
Public DQ_TH_m As Single
Public DQ_TH_b As Single

SW_VAR:
Option Explicit

    Public swApp As Object
    Public Part As Object
    Public boolstatus As Boolean
    Public longstatus As Long, longwarnings As Long
    Public FeatureData As Object
    Public Feature As Object
    Public Component As Object
    Public fileName As String
    Public CurCFG As Object
    Public ConfName As String
    Public PartName As String
Private Sub Form_Load()
Dim Ds As Single
Dim Dj As Single
Dim X As Single
Dim LZ_Flg As Single
Dim LZ_bm As Single
Dim LZ_d As Single
Dim LZ_l    As Single
Dim LZ_b As Single
Dim LZ_P As Single
Dim LZ_SQL As String
Dim s1, s2, s3, e1, e2 As Single

''''''''''''''''''''''''''''''''''''''''''''''''''''''

LZ_SQL = "D_B_bolt"

LZ_d = 6
LZ_l = 16
LZ_P = 0.8
```

```
LZ_b = 10
LZ_bm = 5
''''''''''''''''''''''''''''''''''''''
        Dim txtSQL As String
        Dim MsgTxt As String
        Dim rst As ADODB.Recordset



        txtSQL = "select * from show_thread "
        Set rst = ExecuteSQL(txtSQL, MsgTxt)

        LZ_d = rst("d").Value
        LZ_l = rst("l").Value
        LZ_P = rst("P").Value
        LZ_b = rst("b").Value
        LZ_bm = rst("bm").Value

        rst.Close
        ''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''
LZ_Flg = 210
        If LZ_Flg = 210 Then
                Ds = LZ_d
                Dj = LZ_d / 10
        ElseIf LZ_Flg = 220 Then
                Ds = LZ_d * 0.85
                Dj = LZ_d / 100
        End If
        X = LZ_P * 1.5
  '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

      'LZ_SQL = "D_B_A"
    ' LZ_SQL = "D_B_B"
    '    LZ_SQL = "part5"
        LZ_SQL = "Assem1"
  '''''''''''get data from database''''''''''
        txtSQL = "select * from cover "
        Set rst = ExecuteSQL(txtSQL, MsgTxt)

        s1 = rst("s1").Value
        s2 = rst("s2").Value
        s3 = rst("s3").Value
        e1 = rst("e1").Value
        e2 = rst("e2").Value
```

```
    rst.Close

    Set swApp = CreateObject("sldworks.application")
    swApp.UserControl = True



    Part.EditRebuild

    Part.ShowNamedView2 "*equal shaft check", 7
    Part.ViewZoomtofit2

End Sub


Private Sub Form_Load()

Dim faceWidth, power, speed, Form As Single
Dim a, hh, hhh3, hhh4 As Single
Dim d2s1, d8s1, d5s1, d1s1, d1e2, d1s5, d2s5, d1e7 As Single
Dim d1s8, d2s8, d1s7, d2s7, d1s6, d4s6 As Single
Dim d2s4, d3s4, d1e12, d2s6, d3s6, d1s9, d2s9 As Single
Dim df, numberOfTeeth1, numberOfTeeth2 As Single
Dim MyVar

''''''''''''''''''''''''''''''''''''''''''

Dim txtSQL As String
Dim MsgTxt As String
Dim rst As ADODB.Recordset

LZ_SQL = "part5"

 """"""get part 5 from parameter""""""""""""""
 txtSQL = "select * from parameter"
 Set rst = ExecuteSQL(txtSQL, MsgTxt)

 module = rst("module").Value
 numberOfTeeth1 = rst("numberOfTeeth1").Value
 numberOfTeeth2 = rst("numberOfTeeth2").Value
 faceWidth = rst("facewidth1").Value
 power = rst("power").Value
 speed = rst("inputSpeed").Value
 rst.Close

 txtSQL = "select * from data"
```

```
Set rst = ExecuteSQL(txtSQL, MsgTxt)

hh = rst("hh").Value
hhh3 = rst("hhh3").Value
hhh4 = rst("hhh4").Value

a = Int(0.5 * module * (numberOfTeeth1 + numberOfTeeth2))

d2s1 = hh
d8s1 = hh
d5s1 = faceWidth + 24
d1s1 = 2 * a + 40
d1e2 = Int(2.5 * (0.025 * a + 1))
d1s5 = Int(4.5 * (0.025 * a + 1))
d2s5 = Int(a + 1.25 * (hhh3 + 10 + hhh4 + 10) + 5 + 5)
d1e7 = Int(1.5 * (0.025 * a + 1))

d1s8 = Int(1.25 * hhh4 + 10)
d2s8 = Int(1.25 * hhh3 + 10)
d1s7 = hhh4
d2s7 = hhh3
d1s6 = hhh4 + 5
d4s6 = Int(a)


txtSQL = "update part5 set d2s1=" & d2s1 & ",d8s1=" & d8s1 & ",d5s1=" & d5s1 & ",d1s1=" & d1s1 & ",d1e2=" & d1e2 &
",d1s5=" & d1s5 & ",d2s5=" & d2s5 & ",d1e7=" & d1e7 & " "

Set rst = ExecuteSQL(txtSQL, MsgTxt)

txtSQL = "update part5 set d1s8=" & d1s8 & ",d2s8=" & d2s8 & ",d1s7=" & d1s7 & ",d2s7=" & d2s7 & ",d1s6=" & d1s6 &
",d4s6=" & d4s6 & " "

Set rst = ExecuteSQL(txtSQL, MsgTxt)
''''''''''get data from database''''''''
   txtSQL = "select * from part5 "
   Set rst = ExecuteSQL(txtSQL, MsgTxt)

   d2s1 = rst("d2s1").Value
   d8s1 = rst("d8s1").Value
   d5s1 = rst("d5s1").Value
   d1s1 = rst("d1s1").Value
   d1e2 = rst("d1e2").Value
   d1s5 = rst("d1s5").Value
```

```
        d2s5 = rst("d2s5").Value
        d1e7 = rst("d1e7").Value
        d1s8 = rst("d1s8").Value
        d2s8 = rst("d2s8").Value
        d1s7 = rst("d1s7").Value
        d2s7 = rst("d2s7").Value
        d1s6 = rst("d1s6").Value
        d4s6 = rst("d4s6").Value

        rst.Close



    d2s4 = d1s5
    d3s4 = d2s5
    d1e12 = d1e7
    d2s6 = d1s8
    d3s6 = d2s8
    d1s9 = d1s7
    d2s9 = d2s7

    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


        Set swApp = CreateObject("sldworks.application")
        swApp.UserControl = True

        Set Part = swApp.NewDocument(App.Path + "\Res\" & LZ_SQL & ".SLDPRT", 0, 0, 0)
        fileName = Part.GetTitle
        Set CurCFG = Part.GetActiveConfiguration()
        ConfName = CurCFG.Name

        ''''''Sketch
        boolstatus = Part.Extension.SelectByID("D2@Sketch1@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D2@Sketch1").SystemValue = d2s1 / 1000 ''''''''''1:
        boolstatus = Part.Extension.SelectByID("D8@Sketch1@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D8@Sketch1").SystemValue = d8s1 / 1000 ''''''''''2:
        boolstatus = Part.Extension.SelectByID("D5@Sketch1@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D5@Sketch1").SystemValue = d5s1 / 1000 ''''''''''3
        boolstatus = Part.Extension.SelectByID("D1@Sketch1@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D1@Sketch1").SystemValue = d1s1 / 1000 ''''''''''4
        boolstatus = Part.Extension.SelectByID("D1@Sketch5@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0,
```

```
Nothing)
        Part.Parameter("D1@Sketch5").SystemValue = d1s5 / 1000 """""""5
        boolstatus = Part.Extension.SelectByID("D2@Sketch5@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D2@Sketch5").SystemValue = d2s5 / 1000 """""""6


        boolstatus = Part.Extension.SelectByID("D1@Sketch8@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D1@Sketch8").SystemValue = d1s8 / 1000 """""""11
        boolstatus = Part.Extension.SelectByID("D2@Sketch8@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D2@Sketch8").SystemValue = d2s8 / 1000 """""""22:
        boolstatus = Part.Extension.SelectByID("D1@Sketch7@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D1@Sketch7").SystemValue = d1s7 / 1000 """""""33
        boolstatus = Part.Extension.SelectByID("D2@Sketch7@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D2@Sketch7").SystemValue = d2s7 / 1000 """""""44
        boolstatus = Part.Extension.SelectByID("D1@Sketch6@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D1@Sketch6").SystemValue = d1s6 / 1000 """""""55
        boolstatus = Part.Extension.SelectByID("D4@Sketch6@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D4@Sketch6").SystemValue = d4s6 / 1000 """""""66



        boolstatus = Part.Extension.SelectByID("D2@Sketch4@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D2@Sketch4").SystemValue = d2s4 / 1000 """""""1:
        boolstatus = Part.Extension.SelectByID("D3@Sketch4@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D3@Sketch4").SystemValue = d3s4 / 1000 """""""2:
        boolstatus = Part.Extension.SelectByID("D2@Sketch6@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D2@Sketch6").SystemValue = d2s6 / 1000 """""""3
        boolstatus = Part.Extension.SelectByID("D3@Sketch6@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D3@Sketch6").SystemValue = d3s6 / 1000 """""""4
        boolstatus = Part.Extension.SelectByID("D1@Sketch9@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0, Nothing)
        Part.Parameter("D1@Sketch9").SystemValue = d1s9 / 1000 """""""5
        boolstatus = Part.Extension.SelectByID("D2@Sketch9@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0,
```

```
Nothing)
      Part.Parameter("D2@Sketch9").SystemValue = d2s9 / 1000 ''''''''6


     ''''''''Extrude
      boolstatus = Part.Extension.SelectByID("D1@Extrude2@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0,
Nothing)
      Part.Parameter("D1@Extrude2").SystemValue = d1e2 / 1000 ''''''''1
      boolstatus = Part.Extension.SelectByID("D1@Extrude12@" + fileName + ".SLDPRT", "DIMENSION", 0, 0, 0, False, 0,
Nothing)
      Part.Parameter("D1@Extrude12").SystemValue = d1e12 / 1000 ''''''''2
       Part.EditRebuild


      Part.ShowNamedView2 "*equal shaft check", 7
      Part.ViewZoomtofit2
   End Sub
```

C.4 Semantic Web Code:

The Jena API is used to search and retrieve the data in the Web server; in Jena, ontology is treated as a special type of RDF model, OntModel. This interface allows the ontology to be manipulated programmatically, with convenience methods to create classes, property restrictions, and so forth. Application of this function is in page 90.

Jena Search Result Class:

Search OWL file:

```
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.reasoner.*;
import com.hp.hpl.jena.vocabulary.*;
import com.hp.hpl.jena.reasoner.rulesys.*;
import com.hp.hpl.jena.util.*;
public class searchRerult {
      public getInfModel myGet = new getInfModel ();
      public InfModel model=myGet.getInfModel() ;
public    ResIterator search(String item,String value){

      Property searchProperty;
      ResIterator searchResult=null;
      getInfModel myGet = new getInfModel ();

      searchProperty=model.getProperty(myGet.itemToProperty(item));
```

```
    if(item.equals("author") ){
    Resource SearchValue;
    SearchValue=model.createResource(myGet.addNameSpace(value));
    searchResult=model.listSubjectsWithProperty(searchProperty,SearchValue);
    }else if (item.equals("keywords")||item.equals("title")){
    String SearchValue=value;
    searchResult=model.listSubjectsWithProperty(searchProperty,SearchValue);
    }

     return searchResult;
}
 public String getValue(Resource res,String item){
       String result=null;
       Property searchProperty;
       searchProperty=model.getProperty(myGet.itemToProperty(item));
       StmtIterator SearchResult=model.listStatements(res,searchProperty,(RDFNode)null);

       while(SearchResult.hasNext() ){
       result=SearchResult.nextStatement().getObject().toString() ;
       }
       return myGet.removeType(result);//SearchResult.toString();
      }

}
```

Get Model from OWL file:

```
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.reasoner.*;
import com.hp.hpl.jena.vocabulary.*;
import com.hp.hpl.jena.reasoner.rulesys.*;
import com.hp.hpl.jena.util.*;
import java.lang.*;
import java.util.*;
import java.io.*;
public class getInfModel {

   public InfModel getInfModel(){
   InfModel infModel;
      String file = "d:\\ontology\\pub.owl";
      Model data = ModelFactory.createDefaultModel();
      Model model = ModelFactory.createDefaultModel();
      InputStream in ;
```

```
    try{
        in = FileManager.get().open( file );
    data.read(in, "");
    }
    catch (Exception e ){
    }
     Resource configuration=model.createResource() ;
     configuration.addProperty(ReasonerVocabulary.PROPruleMode, "forward");
     configuration.addProperty(ReasonerVocabulary.PROPruleSet,    "d:\\rules\\pub.rules");
     Reasoner reasoner = GenericRuleReasonerFactory.theInstance().create(configuration);
     infModel=ModelFactory.createInfModel(reasoner, data);
return infModel;
}
    public String itemToProperty(String item){
String property=null;
    if (item.equals("title")){
            property="http://www.domain2.com#hasTitle";
        }
        else if(item.equals("keywords")){
             property="http://www.domain2.com#keywords";
        }
        else if(item.equals("author")){
             property="http://www.domain2.com#hasAuthor";
        }
        else if(item.equals("content") )     {
          property="http://www.domain2.com#content";
        }
        else if(item.equals("pulished") )     {
          property="http://www.domain2.com#isPublished";
        }
        else if(item.equals("Citing") )     {
                property="http://www.domain2.com#Citing";
            }else if(item.equals("bothCiting") )     {
             property="http://www.domain2.com#bothCiting";
                 }
          else if(item.equals("bothCited") )     {
          property="http://www.domain2.com#bothCited";
        }


return property;
}
public String addNameSpace(String value){
        String result=value;
```

```
            result="http://www.owl-ontologies.com/unnamed.owl#" + value;
            return result;
    }
        public String removeType(String sou){
            String result=sou;
            if(sou.indexOf("^^")>0){
            result=sou.substring(0,sou.indexOf("^^"));
            }
            return result;
}
    public String addPrefix(String sou){
    String result=null;
    result="http://www.owl-ontologies.com/unnamed.owl#"+sou;
    return result;
    }
}
```

Display.java:

Display the search result:

```
import com.hp.hpl.jena.rdf.model.InfModel;
import com.hp.hpl.jena.rdf.model.NodeIterator;
import com.hp.hpl.jena.rdf.model.Property;
import com.hp.hpl.jena.rdf.model.RDFNode;
import com.hp.hpl.jena.rdf.model.Resource;
import com.hp.hpl.jena.rdf.model.Statement;
import com.hp.hpl.jena.rdf.model.StmtIterator;

public class display
{
    public getInfModel myGet = new getInfModel();
    public InfModel model = this.myGet.getInfModel();

    public String getValue(String res, String item)
    {
        String result = null;
        Resource resource = this.model.createResource(this.myGet.addPrefix(res));

        Property searchProperty = this.model.getProperty(this.myGet.itemToProperty(item));
        StmtIterator SearchResult = this.model.listStatements(resource, searchProperty, null);

        while (SearchResult.hasNext())
            result = SearchResult.nextStatement().getObject().toString();
```

```
        return this.myGet.removeType(result); }

    public NodeIterator getIterator(String res, String item) {
        NodeIterator result = null;
        Resource resource = this.model.createResource(this.myGet.addPrefix(res));
        Property property = this.model.getProperty(this.myGet.itemToProperty(item));
        result = this.model.listObjectsOfProperty(resource, property);
        return result;
    }
}
```

C.5 The J2ME Mobile Code

Author had developed the J2ME mobile application for MCWE, but compared with the Flex mobile application, the deployment and development of Flex mobile application could be made easier and with greater flexibility.

```
/**
 * com.MobileServerSoap_Midlet.java
 * Sample Midlet v.1.0
 *
 * Generated on 07-6-7 15:41
 *Based on WSDL at
http://localhost:8080/axis/services/MobileServer?wsdl
 */

package com;

import java.io.IOException;
import java.io.InputStream;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;
import javax.microedition.lcdui.*;

public class MobileServerSoap_Midlet extends javax.microedition.midlet.MIDlet implements
javax.microedition.lcdui.CommandListener {
    javax.microedition.lcdui.Form mainForm =
        new javax.microedition.lcdui.Form("MCWE Client");
    javax.microedition.lcdui.TextField drawIn0Field =
        new javax.microedition.lcdui.TextField("draw In0:", "", 1,
```

```
javax.microedition.lcdui.TextField.ANY);
    javax.microedition.lcdui.TextField lIn1Field =
        new javax.microedition.lcdui.TextField("Length (l) :", "", 20,
javax.microedition.lcdui.TextField.ANY);
    javax.microedition.lcdui.TextField dIn2Field =
        new javax.microedition.lcdui.TextField("diameter (d) :", "", 20,
javax.microedition.lcdui.TextField.ANY);
    javax.microedition.lcdui.TextField bmIn3Field =
        new javax.microedition.lcdui.TextField("thread_h (bm) :", "", 20,
javax.microedition.lcdui.TextField.ANY);
    javax.microedition.lcdui.Command pCommand =
        new javax.microedition.lcdui.Command("get height (b)", javax.microedition.lcdui.Command.SCREEN,
1);
    javax.microedition.lcdui.Command bCommand =
        new javax.microedition.lcdui.Command("view image", javax.microedition.lcdui.Command.SCREEN, 1);
    javax.microedition.lcdui.Command drawCommand =
        new javax.microedition.lcdui.Command("design product", javax.microedition.lcdui.Command.SCREEN,
1);
    javax.microedition.lcdui.Command lCommand =
        new javax.microedition.lcdui.Command("input Data", javax.microedition.lcdui.Command.SCREEN, 1);
    javax.microedition.lcdui.Command dCommand =
        new javax.microedition.lcdui.Command("d", javax.microedition.lcdui.Command.SCREEN, 1);
    javax.microedition.lcdui.Command bmCommand =
        new javax.microedition.lcdui.Command("bm", javax.microedition.lcdui.Command.SCREEN, 1);
    javax.microedition.lcdui.StringItem resultField =
        new javax.microedition.lcdui.StringItem("", "");
    com.MobileServerSoap_Stub stub =
        new com.MobileServerSoap_Stub();



     String[] stringArray = {
                        "Facewidth","Module", "Addendum Coefficient", "Press Angle", "Helix
Angle","Rack Tip Radius"
                    };
               ChoiceGroup groupd = new ChoiceGroup("Power:", ChoiceGroup.POPUP, stringArray,null);
     String[] stringArray1 = {
                            "960","6", "8", "10", "12","16"
                        };
                  ChoiceGroup groupbm = new ChoiceGroup("Input Speed: ", ChoiceGroup.POPUP,
stringArray1,null);
     String[] stringArray2 = {
                            "5","6", "8", "10", "12","16"
                        };
                 ChoiceGroup groupl = new ChoiceGroup("Thread Length(l):     ", ChoiceGroup.POPUP,
```

```
stringArray2,null);


    public MobileServerSoap_Midlet() {
     //   mainForm.append(drawIn0Field);



      // mainForm.append(groupd);
      // mainForm.append(groupbm);
       //mainForm.append(groupl);

   //      mainForm.append(dIn2Field);
   //      mainForm.append(bmIn3Field);
      //    mainForm.append(lIn1Field);
      /********************************************************************/
      mainForm.append("Form Input: " +
                                  "Stage Two:Initial Design-Material");
                                  mainForm.append("");
      //    mainForm.append(groupd);
                        mainForm.append(
                             new TextField("Material Type:   ", null, 15, TextField.ANY));
                        mainForm.append(
                        new TextField("Hardness Process: ", "", 15, TextField.EMAILADDR));
                        mainForm.append(new TextField("Surface Hardness: ", "", 15, TextField.NUMERIC));
                        mainForm.append(
                                  new TextField("Effective Case Depth:", "", 15, TextField.DECIMAL));
                        mainForm.append(
                                  new     TextField("Ultimate     Tensile     Strength:",     "",     15,
TextField.PHONENUMBER));
                        mainForm.append(
                                  new TextField("Core Residual Stress", "", 15, TextField.ANY));
                        mainForm.append(new TextField("Yield Strength:", "", 15, TextField.ANY));
           /********************************************************************/
           mainForm.addCommand(lCommand);

         mainForm.addCommand(pCommand);


         mainForm.addCommand(drawCommand);
           mainForm.addCommand(bCommand);
  //        mainForm.addCommand(dCommand);
  //        mainForm.addCommand(bmCommand);
         mainForm.append(resultField);
//          stub.resultField = this.resultField;
```

```
        mainForm.setCommandListener(this);
    }

    public void startApp() {
        javax.microedition.lcdui.Display.getDisplay(this).setCurrent(mainForm);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }

    public void commandAction(javax.microedition.lcdui.Command c, javax.microedition.lcdui.Displayable
d) {
        try {
            System.out.println("Processing command action");
            if (c == pCommand) {
                resultField.setLabel("thread height (b) results");
                resultField.setText("working...");
                java.lang.String result ="10";//stub.p( );

                String resultString=""+result;
                resultField.setText(resultString);
//                   **********************
                                      //fImageForm.deleteAll();
                                                      Item imageFormItem = null;
                                                      try {
                                                           Image image = getImage();
                                                           if (image != null) {
                                                                  imageFormItem = new ImageItem(null,
image, Item.LAYOUT_CENTER, null);
                                                           }
                                                      } catch (SecurityException e) {
                                                              imageFormItem = new StringItem(null,
e.getMessage());
                                                      }
                                                      if (imageFormItem == null) {
                                                            imageFormItem = new StringItem(null, "Image
was not retrieved, see console log");
                                                      }
                                                      mainForm.append("\n");
                                                      mainForm.append(imageFormItem);
                                                      //fMidlet.switchDisplayable(fImageForm);
```

```
                                       //************************

            }
            if (c == bCommand) {
                    //                  ***********************
                                       mainForm.deleteAll();
                                                       Item imageFormItem = null;
                                                       try {
                                                            Image image = getImageF();
                                                            if (image != null) {
                                                                  imageFormItem = new ImageItem(null,
image, Item.LAYOUT_CENTER, null);
                                                            }
                                                       } catch (SecurityException e) {
                                                                imageFormItem = new StringItem(null,
e.getMessage());
                                                       }
                                                       if (imageFormItem == null) {
                                                            imageFormItem = new StringItem(null, "Image
was not retrieved, see console log");
                                                       }
                                                       mainForm.append("\n");
                                                       mainForm.append(imageFormItem);
                                                       //fMidlet.switchDisplayable(fImageForm);
                                       //************************
            }
            if (c == drawCommand) {
                resultField.setLabel("draw ");
                resultField.setText("working...");
                java.lang.String parm0= drawIn0Field.getString();
                int result =stub.draw(  parm0 );
                String resultString=""+result;
                resultField.setText("successful!");
            }
            if (c == lCommand) {
               // resultField.setLabel("l results");
               // resultField.setText("working...");
               /*
                java.lang.String parm0= lIn1Field.getString();
                int result =stub.l(  parm0 );

                     java.lang.String parm1= dIn2Field.getString();
                                         int result1 =stub.d(  parm1 );
```

```
                    java.lang.String parm2= bmIn3Field.getString();
                                        int result2 =stub.bm(  parm2 );
                                        */
                    //****************************************
                    java.lang.String parm0= getSelectedItemString(groupl);
                    int result =stub.l(  parm0 );

                    java.lang.String parm1= getSelectedItemString(groupd);
                    int result1 =stub.d(  parm1 );

                    java.lang.String parm2= getSelectedItemString(groupbm);
                    int result2 =stub.bm(  parm2 );



             //  String resultString=""+result;
             //  resultField.setText(resultString);
         }
         if (c == dCommand) {
             resultField.setLabel("d results");
             resultField.setText("working...");
             java.lang.String parm0= dIn2Field.getString();
             int result =stub.d(  parm0 );
             String resultString=""+result;
             resultField.setText(resultString);
         }
         if (c == bmCommand) {
             resultField.setLabel("bm results");
             resultField.setText("working...");
             java.lang.String parm0= bmIn3Field.getString();
             int result =stub.bm(  parm0 );
             String resultString=""+result;
             resultField.setText(resultString);
         }
     } catch (Exception e) {
         e.printStackTrace();
         resultField.setLabel("Error:");
         resultField.setText(e.toString());
     }
 }
  private Image getImage() {
             HttpConnection connection = null;
             InputStream inputStream = null;
             Image image = null;
             try {
```

```
        // as user if it is okay
        connection = (HttpConnection)
    ...
```