A Framework for Collaborative Working Environments

Daizhong Su*, Yu Xiong, Yongjun Zheng and Shuyan Ji
Advanced Design and Manufacture Engineering Centre
School of Architecture, Design and the Built Environment
Nottingham Trent University, UK

Abstract To utilize the advantages of existing and emerging Internet techniques and to meet the demands for a new generation of collaborative working environments, a framework with an upperware-middleware architecture is proposed, which consists of four layers: resource layer, middleware layer, upperware layer and application layer. The upperware contains intelligent agents and plug/play facilities; the former coordinates and controls multiple middleware techniques such as Grid computing, Web-services and mobile agents, while the latter are used for the applications, such as semantic CAD, to plug and loose couple into the system. The method of migrating legacy software using automatic wrapper generation technique is also presented. A prototype mobile environment for collaborative product design is presented to illustrate the utilization of the CWE framework in collaborative design and manufacture.

Keywords: Web application, Internet based collaboration, collaborative work environment, mobile agent, design and manufacture

1. Introduction

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

Researchers and industrialists have been paying great attention to the research in this area. For example, Yujun and Jinsong (2005) introduced a paradigm called Internet-based collaborative product development chain, which connects users in geographically different locations to carry out product development activities simultaneously and collaboratively using a series of tools, including product management tools, workflow management tools, and project management tools. Chenga and Fenb (2006) developed a prototype of Web-based distributed problem-solving environment to facilitate computer aided engineering technologies and to support networked collaboration such that scientists around the world could interactively, visually and experimentally explore their daily design work through the proposed system. Younas et al. (2006) presented a protocol in order to improve the efficiency of Web services composition, which was based on the peer-to-peer paradigm, to exploit the capabilities of underlying networks such as the processing carried out at the network nodes. Li and Lin (2007) proposed the architecture of collaborative design grid and its application based on LAN. Tsai et al. (2006) described a web-based information sharing system that used XML (eXtensible Markup Language), ontology, and access control technologies to deal with product design collaboration problems. More and more research findings have been continuously reported, indicating the substantial progress in this research area. However, the literature survey reveals that although middleware techniques such as CORBA, Web cervices, grid and agents have been utilised in the current CWEs and related applications, combination

* Corresponding Author. E-mail: daizhong.su@ntu.ac.uk

of such existing and emerging middleware to form a more powerful and flexible CWE is still a challenge research.

As an important research initiative, the European Union set up the 'Collaborative Working Environments' as one of the strategic objectives for research funding under the FP6 IST programme in 2005, and sponsored 13 projects in a wide range of areas including catalyzing rural development, distributed virtual manufacturing enterprises, virtual engineering for SMEs, digital factory for human-oriented production, strategic innovation, advanced robotic systems, human operators in emergency/disaster scenario and others (European Commission 2006). In the European Union's ICT theme of the 7th Frame Programme (European Commission 2007), it has been stated that 'the current Internet, mobile, fixed and broadcasting networks and related software service infrastructure need to be progress accordingly in order to enable another wave of growth in the on-line economy and society over the next 15 year.' 'the challenge is to deliver the next generation of ubiquitous and converged network and service infrastructure for communication, computing and media.' All those demand the new collaborative working environments and associated technologies.

As part of the Asia-Link and Asia IT&C projects supported by the European Commission, the research in Web-based collaborative design and manufacture was carried out by the international project consortia led by the authors' team. The work conducted includes the development of a Web-enabled environment (Su et al. 2006), online collaborative computer aided design and manufacture (Lin et al. 2006, Chen et al. 2005), effective remote-execution of large size programs (Su et al. 2003) and Web-enabled distributed product design (Mendikoa et al. 2005, Mendikoa et al. 2006).

Based on the experience gained in CWE research and to address the above issues, a new CWE framework with an upperware-middleware structure is proposed by this research, which works as a mediator and facilitator to provide collaborative services for the development of flexible, scalable and adaptable tools and applications in design and manufacturing engineering. This enables geographically dispersed team members to effectively collaborate with each other through the aid of seamless integration of ICT technologies.

The research includes the development of an overall structure of the framework, the methods how the upperware interacts with the underlying layers and the applications, and the middlewares to be developed in various Internet/Web and ICT techniques such as Grid, Web services, mobile agent and wireless computing. The techniques for resource wrapping as well as resource publishing, resource searching and resource locating are also proposed for the CWE.

2. Overview of the CWE framework

The structure of the framework is shown in Figure 1. The upperware interacts with the applications and middlewares in the underlying layer, to provide specific services for collaboration combining the basic services provided by the middlewares and the tools.

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

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

- Virtuosity: It enables interoperability across borders, encourages scientific/business collaborations between organizations, and allows remote job submission and management.
- Autonomy: It allows services to be executed without requiring the monitoring or continuous interaction of clients. The aim is to create services that display intelligent autonomous behaviour by being able to complete an assigned task by appropriately choosing from a set of possible strategies.
- 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 there.
- Security: Information services provide information about available resources and status of resources, while security services provide single sign-on, authentication, and authorization and secure data transfer.

To achieve all the above is a novel and challenging task. No current existing technology is capable of providing all these features. In order to implement the integration and collaboration amongst the partners, the system has to be provided with the following four layer architecture, as shown in Figure 1.

[Insert Figure 1 about here]

2.1 Resources layer

The bottom layer is the Resource layer. The physical resources would be wrapped into specific technology enabled ones so that they could be recognized by the collaborative working environment, and four kinds of resources are supported by default: Grid, Web Services, P2P and mobile agent resources. These resources independently exist in the system.

2.2 Middleware layer

The middleware is a class of software technologies to manage the complexity and heterogeneity inherent in distributed systems. It is defined as a layer of software above the operating system but below the application programs 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 enabling middleware techniques including Grid, Web services, mobile agents, wireless computing and pear to peer (P2P). For the large computing or sharing the distributed resources, Grid computing is the ultimate vision in distributed computing, spanning locations, organizations, hardware and software boundaries to provide tremendous power, but due to the different security mechanisms, the number of accessible nodes in a grid is generally lower than in a p2p network. For exchange of information and content files, P2P is a good choice, and a mobile agent is assigned each process engaged in the same job, monitors its execution at a different machine, and takes its periodical execution snapshot. Different techniques have different advantages in different fields. The task to determine when and which technique is to be used will be carried out in the upperware layer.

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

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

- *Resources registration, discovery and allocation*. In the system, all the resources are packed into recognizable resources and registered in a registry centre so that they could be found. Instead of storing the content on its own servers, the client makes the information available to server directly from the publisher's server via an online connection.
- *Index and repositories system*. The research focuses on creating an indexing system of clients and of available services. The initial proposition is to create an index of all clients and their resources on large indices in a main server. A query is sent to a server, and then the server looks-up the index; if the query can be resolved, then the server sends a message to the originator to inform where to get the service or file. The index and repository system will avoid the drawbacks of current centralized index approaches, although which can provide the best performance for resource discovery. The server in centralized indexes and repositories system is expensive, the bandwidth and hardware required to support large networks of clients are expensive too. If the server in the system fails to function properly, it brings down the network. To avoid the above problem, the system has a cluster of servers, and, hence, if one server fails, the rest of the servers will continue to support the network. Recent court rulings cast serious doubt about the liability involved in using centralized servers to index resources in a peer based network. The ability to monitor and enforce this requirement is quite challenging, and may be too much of a risk. Further research will be conducted to determine possible alternative index or discovery schemes to handle resource discovery and use.
- *Resource Wrapping*. Some resources may be not directly associated with any kind of the middleware technology, so resource wrapping should be taken into consideration, which migrate different CAD/CAM legacy software systems to distributed collaborative resources in CWE.

2.3 Upperware layer

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

(1) Intelligent agents for middleware

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

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

- Process control of concurrence and consistency as well as synchronous and/or asynchronous messages. Process, elements (security, services, monitoring, etc) can be shared across applications to provide horizontal services to decouple these reusable application components, facilitating more rapid changes in these processes.
- Group Task Coordinator is responsible for the task assignment of each requester, monitoring the action state of each requester and accepting the communication request and service from each provider.
- To provide an interface to the worker centric design tools and establish the connection between the design tools and the middleware techniques specified in the underlying layer.

(b) Plug and play facilities

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

It supports plug and Play. A knowledge based module is developed, and learning and adaptation capabilities using agent technologies will be further researched and developed. Investigation will be carried out to help integrate dynamic plug and play for better system performance. To make the system better distributed will enable developers to deliver rich Internet applications. The applications or resources could be easily plugged into the supporting facilities to enable them work well within the collaborative working environment.

2.4 The applications

The upperware 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 utilised 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.

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

The objective of this part of the research is to develop and to implement a generic service wrapper for automatically migrating different legacy software systems to distributed collaborative resources in the CWE.

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

3.1 The wrapping method

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

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

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

3.2 Issues to automate wrapper generation

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

- Variety of the interfaces to legacy systems. There are many interfacing styles in legacy systems. They have different implementations to each other and dedicated proprietary. Thus it is difficult for server-side application developers to implement wrapper objects for legacy systems, even when understanding some of the interfaces to the legacy systems. To cope with this problem, the Wrapper Template Classes and an automatic wrapper generator set are developed to free the providers from understanding and implementing various interfacing techniques.
- Representation of interfaces to legacy systems. To generate a wrapper automatically, a server-side developer or provider should submit interfacing information for legacy systems to an automatic wrapper generator. Thus, some representations are required to describe easily the interfaces to legacy systems. For this purpose, the Extended WSDL is proposed. In any case, a server-side provider should describe the services of legacy systems with WSDL for clients, in other words, the server-side provider has to describe the interfaces to wrapper tools. The Extended WSDL will provide a way of describing the services of legacy systems for clients and another way of describing actual interfaces to legacy systems for a wrapper generator.

3.3 Benefits of the wrapping approach

The benefits of the wrapping approach are as follows***:***

- Enhancement of reusability. By using previously developed programs through wrapper tools, they are reused at the level of executable codes.
- Reduction of software development cost. The previously developed programs have high reliability since they have been used and tested for a long time. Rather than redesign and redevelop programs with the same functionality, wrapping legacy systems should reduce both development and testing cost.
- Location transparency of server applications. Service wrapper tool for legacy applications have a location transparency feature. Thus, although a wrapper service can be located anywhere on the network, the provider who uses a wrapper service is able to construct client applications without any consideration about the location of legacy applications.
- Code security and easily joining. The wrapper automation generator helps to avoid the need of knowing about Web technology by resource provider and without revealing resource code to wrapping developer.
- Flexible wrapper for different resources to different interface. Resources are heterogeneous, i.e., written in different language, running in different platforms. In addition, due to using multiple middles and multiple protocols in this environment, Wrapper must be facilitated with adaptive procedure for different resources to proper wrapping interface. This is dependent on the need of collaborative resource function. For example, if it is computing resource, it will be wrapped in Grid protocol and interface; if it is interactive commercial CAD software system, it will be wrapped in Web services, and so on.

4. Utilisation of the Framework for Collaborative Design and Manufacture

As part of an on-going research project, the framework described above has been applied in the development of a collaborative work environment for design and manufacture. The total process of collaborative design and manufacture has been considered, including formulation of product design specification, supplier selection, conceptual design, detail design manufacture, inventory management after sale services and recycling.

4.1 Application of multiple middleware techniques within the CWE for collaborative design and manufacture

Several middleware techniques, such as Grid computing, Web services, mobile agents and CORBA, have been applied in the development of the CWE for collaborative design and manufacture. Due to the limitation of the length of this paper, the approaches and applications developed so far are outlined in this section, and one of the applications, mobile agents for collaborative design, is further detailed in sections 4.2 to illustrate how the application fits into the CWE framework.

Web-enabled product life cycle analysis, supplier selection and product specification. It consists of three aspects: (1) an information analysis model to help manufacture clarify and analysis the orientation of the customer, competitors and manufacture for strategy planning purposes; (2) the supplier selection model to help manufactures to identify appropriate suppliers for components, materials and services within the procedure of product design and manufacture; (3) a method has been developed for people who involved in product design process to efficiently share and to reuse PDS (product design specifications). Within the research, Web-services in connection with EJB (enterprise Java Beans) ISP (Java Sever Pages), MYSQL database and Java Serverlet have been utilised (Hou 2007).

A service-oriented approach for online software sharing. It enables the users to share software resources over the Internet. It consists of shared applications, client applications and three types of services: application proxy service, proxy implementation service and application manager service. With the aids of the services, the client applications interact with the shared applications to implement a software sharing task. The approach satisfies the requirements of copyright protection and reuse of legacy codes. A case study of online sharing a software package of gear design optimisation was conducted to demonstrate the approach developed (Xiong and Su 2006a).

Web-service supported collaborative CAD. With this approach, the geographically dispersed team members can collaborate over the Internet to conduct computer aided design with tetures of online modifying CAD drawings. The advanced technique of Web Services is utilized for the users to provide and to request the services, and the most popular CAD software AutoCAD is used as vehicle to illustrate the methods developed' The method developed is in principle applicable for other CAD software packages such as Pro/Engineer and Solidworks. It can be operated in computers with different operation systems located in different locations (Su et al. 2006).

A CORBA based collaborative architecture. It combines CORBA with other technologies such as Java and Genetic Algorithms to integrate engineering design programs and enable communication between them, regardless of languages they are written in and platforms they are ported on, and to leverage multiple design interests. Three main application paradigms have been addressed. Firstly, by using CORBA-Servlet, a singular large-size program can be executed remotely with interactive features such as parameter input, program execution and monitoring, and varieties of dynamic and rich forms of resultant return. It also facilitates multi-user management. Secondly, Genetic Algorithms has been employed to combine the CORBA to mediate the impacts from different domain expert considerations over the distributed environment. The CORBA based architecture integrates heterogeneous and distributed multiple design applications and enables them to communicate with each other. A GA-based optimiser is to help a main designer to conduct gear design optimisations through invoking remote and heterogeneous applications to meet user's complex needs. Thirdly, Applet-CORBA combination is utilised to provide a thin client model to enable users to do the remote invocation from a Web browser, without any setting up for CORBA in advance. As a user of this application, a designer does not need to know anything about CORBA . The rich features of applet make client developers to design varieties forms of Web browser application, such as dynamic data, dynamic drawing, and so on. (Ji 2006)

A mobile agent system for collaborative design. It is implemented based on Java technology, supporting the construction of mobile agents or mobile agent-oriented systems with the dynamic loading class for various applications. With the parametric design method, which is popular in most commercial CAD packages such as Pro/E, I-DEAS, UG and SolidWorks, this mobile agent system allows the users to input the CAD parametric design data in mobile devices and get the results from the server. The mobile agents can also utilise the design resources distributed within the network (Zheng 2007).

A Grid based system for a virtual manufacture/retailer organization. It enables a large number of manufacturers and retailers to cooperate with each other within a Virtual Organization by sharing inventory information. This system adopts a mode based on inventory management to enhance the Quality of Service, which would greatly facilitate the business processes between the manufacturers and retailers. With the support of Grid computing, the system is more powerful than other existing systems and has considerable advantages of more accurate calculations and more quick response (Xiong and Su 2006b).

4.2 Development of a Prototype Mobile Agent Collaborative Environment for Product Design

Nowadays mobile technology has been widely applied in various areas; however, little efforts have been made in utilising the technology for product design, although product design is characterized by a high degree of collaboration and mobility. Mobile devices (such as pocket computers, wireless handheld devices, mobile phones, etc) are currently being used more often as personal tools. It is a challenge task to apply mobile networking in product design. New systems aiming at such mobile devices need to be developed, in order to provide a collaborative work environment for sharing design information, data and knowledge among distributed design and manufacturing team members. With the aid of such systems, the designer can collaborate with other team members online, the manufacturing engineer in the workshop can remotely access the product database stored in the design office, the process planning engineer can utilise and modify the product data anywhere; and the manager can access the product database when he is away from his office, remaining in contact with his staff members by online communication, actually seeing drawings and relevant data, rather than by just hearing about them on the telephone.

To meet the demands of the mobile systems for collaborative product design, a prototype mobile collaborative work environment (MCWE) has been developed by utilising the CWE framework presented in section 2.

Agent technology has been recognized as a promising paradigm for next generation manufacturing systems (Shen et al. 2006) and attracted researchers great attention. For example, Hao et al. (2006) developed an agent-based collaborative e-Engineering environment for product design engineering. Liu and Tang (2005) presented an evolutionary design approach in a multi-agent design environment to support design management and communication in a distributed environments. As a proof-of-concept prototype using Java RMI technology, a mobile agent platform was proposed by Mahmoud and Yu (2006), helping the end user to reduce information overload and provided novel ways for cellular service providers to compete remote services and acquire new customers. To enhance the system's capacity, the mobile agent technique is utilised in the MCWE

4.2.1 Overview of the Prototype MCWE system

Within the CWE framework, the mobile agent technology is specified in the middleware layer. The CAD applications in server side are plugged into the upperware layer. When a user sends a request from the mobile device, the request first reaches at the upperware layer, then the system decides what type of resource needs to be revoked in the resource layer.

Within the MCWE system, a mobile agent 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. At present mobile computation relies on expensive and frail network connection, and, hence, those features become more and more valuable. Mobile agent codes are stable, robust, and fault-tolerant, and hence there will be sufficient time for the agents to move to another main host and continue working when a warning message of problems is received.

For different mobile operation systems, such as Linux and Windows Mobile, J2ME is a good choice to run the thin client application (J2ME needs to be mentioned in the following text). Although mobile devices become more and more powerful, some of them still cannot work with the agent platform. For the latest mobile devices which are able to do so, the agent codes run in the mobile device; for those devices which cannot run the agent, the MCWE divides the mobile agent codes into two parts, of which the part responsible for receiving input data from the user stays in the mobile device, while the rest stays in the server. Figure 2 shows the communication amongst the clients.

[Insert Figure 2 about here]

As shown in Figure 3, there are four main parts within the MCWE system:

Agent Manager: It is used for managing, storing, and dispatching the agent. Agent Manager must be capable of instantiating agents from an agent storage (termed as agent packages) on the local file system, as well as transmitting them through the network. Java's dynamic class instantiation capabilities make it possible to treat all agents generically, and instantiate them based on the knowledge of a common behaviour shared by all agents.

Mobile Agent: It is composed of two kinds of classes, one for the long-distance interface connection, and the other for the realization of interface connections. These agents may visit the resources or execute in the server after they are transmitted from the client.

Web server: This is used for storage of the dynamic class and integrated in the upperware plug and play facilities. When the host and the agent server need the Java class, they visit the Web Server to download it, which benefits for the long-distance transmitting of the mobile agent, or the fault-tolerance of the system, since the remote server can download the Java class form the reliable site.

RMI registration: The proxy is registered in the RMI registration. The agent gains the proxy stub, which acts as the local host to take the long-distance object use.

[Insert Figure 3 about here]

4.2.2 The operation process

Although CAD packages play an important role, little research has been conducted in mobile collaborative CAD for product design, due to the limited CAD capacity of mobile devices. This problem is resolved by this research using mobile agent and parametric design method.

Within the MCWE system, the client's requests for the CAD drawing are packaged as XML data and then transmitted to the Server via the client agent. The CAD package is located in the Server where most of the processes to produce the drawing is implemented. Therefore, it is not required for the mobile device to install the CAD software, which avoids the mobile capacity problem mentioned above.

In the process of produce the drawings, the MCWE system employs a parametric design method, which is popular in most nowadays commercial CAD packages such as Pro/E, I-DEAS, UG and Solid Works. With the parametric design method, via the mobile device, the designer inputs the design parameters to produce the CAD drawings; and by revising the parameters, the designer can also modify the CAD models online.

Those data include the parameters required to produce the CAD drawings. After having received the data, the server runs the CAD package to produce the drawing specified by the parameters. After the server finishes the process, the results are then sent to the client

The operation process is as follows: after the connection has been established between the mobile device and the server, the client sends the parametric CAD data to the server which actives the mobile agent, and the agent codes are downloaded from the Server to the mobile device; at the Server side, the CAD package is operated to produce the CAD drawings which are stored into the database, and the CAD results are then sent o the client. Since the CAD results are of the png file, the mobile user can view the file in the mobile device based on the J2ME technology; if the user wants to edit the CAD file or change the CAD information in the mobile device, the mobile device should support the mobile CAD application, such as CeCAD, which is supported by Windows Mobile 5 operation system.

A prototype MCWE system has been developed for a manufacturing company in the UK for online collaborative design and manufacture of their product. The company uses SolidWorks for producing CAD drawings. With the SolidWorks 3D parametric CAD features, the mobile user can open template parts, assemblies or drawings, and make changes.

The company maintains the control of 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 through the network, a client GUI that sends and receives inputs/outputs. The underlying software is therefore protected from attempts to reverse-engineer it and from unauthorised changes. The company can withdraw a user's access rights to the system at any time should it be necessary, for example on termination of an employment. The codes, which are written in C++, run the SolidWorks application after the agent receives the parametric design data form the mobile client. When SolidWorks completes CAD drawings, the system stores the CAD drawings into png file, which can be sent to the mobile device, from which the mobile user can view it. The operation procedure is summarised below

- Start the Web server, which is integrated with the plug and play facilities in the upperware layer, and install the dynamic class which needs to be downloaded.
- Start the server and client.
- The user, via the mobile device, inputs the CAD parametric design parameters; send the data to the server.
- Server downloads the code from the Web server, creates the agent through the RMI register, communicates with the server, executes the SolidWorks application in the server, and gets the results.
- Send the png file to the mobile device.

4.2.3 Integration of Grid and Web service into the MCWE system

It is proposed to combine Grid and Web services with the mobile agent in the further development of the MCWE, which will offer the following features: inherent consistency through any equipment, in any time, any place among various kinds of end-users; incorporating new services, allowing the system to be extensible; enhancing the security when transmitting data in the network because of the grid technology is employed for the system; utilizing Web/Grid technology to allows the application program to be upgraded and expanded into other end-devices.

There is a trend in distributed computing towards Service-Oriented-Architectures, and in particular the Grid as a vision of secure Web Services for sharing resources across virtual communities and organizations. Utilisation of the Grid offers not only advantages for e-design, but specific advantages for mobile e-design.

Due to the relative ease with which loosely-coupled secure Grid services are applicable, it would be particularly useful in a mobile context as mobile devices change their networks and neighbours far more frequently than ordinary computer installations. Mobile devices using Grid services would therefore benefit from being able to find and use services that were local to the device (for example, to utilize a nearby screen to show information that couldn't be crammed onto a handheld display).

However the current set of Grid technologies does not fit well with the loosely coupled requirements of mobile e-design and are often too heavy-weight to fit on a mobile device. Unless this is addressed it will make the emerging e-design Grid infrastructures inaccessible to mobile devices, and stunt the development of novel mobile e- design applications. We will specify our requirements to create a mobile e-design client using Grid technologies and will research the state of the art in the next generation of Grid technology to see if it fulfils the requirements for mobile e- design that we have identified.

5. Concluding remarks

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

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

As part of an on-going research project, a prototype of mobile collaborative work environment (MCWE) has been developed, which illustrates the application of the framework into the development of a CWE for collaborative design and manufacture. It has to be pointed out that the work presented in this paper is at its early development stage and more work is needed to further demonstrate the proposed CWE framework. In addition to integration of grid and Web services into the system as mentioned in section 4.2.3, further research in the MCWE will also include the development a method for online collaboration between CeCAD packages. CeCAD is a stand-alone CAD package for mobile devices, and currently it cannot be used for online collaboration. Because it can run '.dwf' files, the most popular type of CAD drawings, the online collaborative CeCAD to be developed will make the MCWE more powerful.

Acknowledgement

The authors are grateful for the support received from the EU Asia-Link programme (grant No.ASI/B7-301/98/679-023) and Asia IT&C programme (Grant No. ASI/B7-301/3152-099/71553) for carrying out the research.

References

Chen, X., Luo, T., He, Y., Zhou, W. and Sun, D., Research on 3D Feature Modeling technology for Internet-driven Collaborative Design, *Proceedings of 9th International Conference on Computer Supported Cooperative Work in Design*, 2005, pp 649-654.

Chenga, H.-C. and Fenb, C.-S., A Web-based distributed problem-solving environment for engineering applications, *Advances in Engineering Software*, 2006, **37**(2), 112-128.

European Commission, *Collaboration@Work: the 2006 report on new working environments and practices,* 2006 (European Communities: Luxembourg).

European Commission, *ICT-Information and Communication Technologies Work Programme 2007-2008,* 2007 (European Communities: Luxembourg).

Hao, O., Shen, W., Zhang, Z., Park, S-W and Lee, J.-K., Agent-based collaborative product design engineering: An industrial case study, *Computers in Industry*, 2006, **57**(1), 26-38.

Hou, J., Web oriented manufacturer contralised environment supporting SMEs, PhD thesis, Nottingham Trent University, 2007.

Ji, S., Web-based collaborative environment for integrated design, PhD thesis, Nottingham Trent University, 2006.

Li, Z. and Jin X., Architecture of collaborative design grid and its application based on LAN, *Advances in Engineering Software*, 2007, **38**(2), 121-132.

Liu, H. and Tang, M., Evolutionary design in a multi-agent design environment, *Applied Soft Computing*, 2006, **6**(2), 207-220.

Mendikoa, I., Sorli, M., Barbero, J., and Carrillo, A., Knowledge based distributed product design and manufacturing, *proceedings of 9th International Conference on Computer Supported Cooperative Work in Design,* 2005, pp 679-684.

Mendikoa, I., Sorli, M., Barbero, J. and Carrillo, A and Gorostiza, A., Inventive approach for problems detection in collaborative design, *proceedings of the International conference on Advanced Design and Manufacture*, 2006, pp 424-428.

Mahmoud, Q. H. and Yu, L., Havana agents for comparison shopping and location-aware advertising in wireless mobile environments, *Electronic Commerce Research and Applications*, 2006, **5**(3), 220-228.

Shen,W., Hao, Q., Yoon, H. J. and Norrie, D. H., Applications of agent-based systems in intelligent manufacturing: An updated review, *Advanced Engineering Informatics*, 2006, **20**(4), 415-431.

Su, D., Ji, S., Amin, N. and Hull, J. B., Multi-user Internet environment for gear design optimisation, *Integrated Manufacturing Systems*, 2003, **14**(6), 498-507.

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

Tsai, C-Y, Sun, T-H and Huang, J-X, A Web-based XML information sharing system for collaborative product development, *International Journal of Production Research*, 2006, **44**(15), 2955-2976.

Xiong, Y. and Su, D., A Web-Service Based Approach for Software Sharing, in: *Lecture Notes in Computer Science*, **3865**, *'Computer Supported Collaborative Work in Design II'*, edited by W Shen et al, pp 215-224, 2006a (Springer: Berlin Heidelberg). .

Xiong, Y. and Su, D., A Grid Based System for a Manufacture/Retailer Virtual Organization, proceedings, *proceedings of International Conference on Advanced Design and Manufacturing*, 2006b, pp 442-446**.**

Younas, M., Awan, I. and Duce, D., An efficient composition of Web services with active network support, *Journal of Expert Systems with Applications*, 2006, **31**(31), 859-869.

Yujun, Y. and Jinsong, Z., Internet-based collaborative product development chain for networked product development, *The International Journal of Advanced Manufacturing Technology*, Springer London, 2005, **28**(7), 845-853.

Zheng, Y., Development of a mobile collaborative environment for collaborative product design, MPhil/PhD Transfer Report, Nottingham Trent University, 2007.

List of Figures:

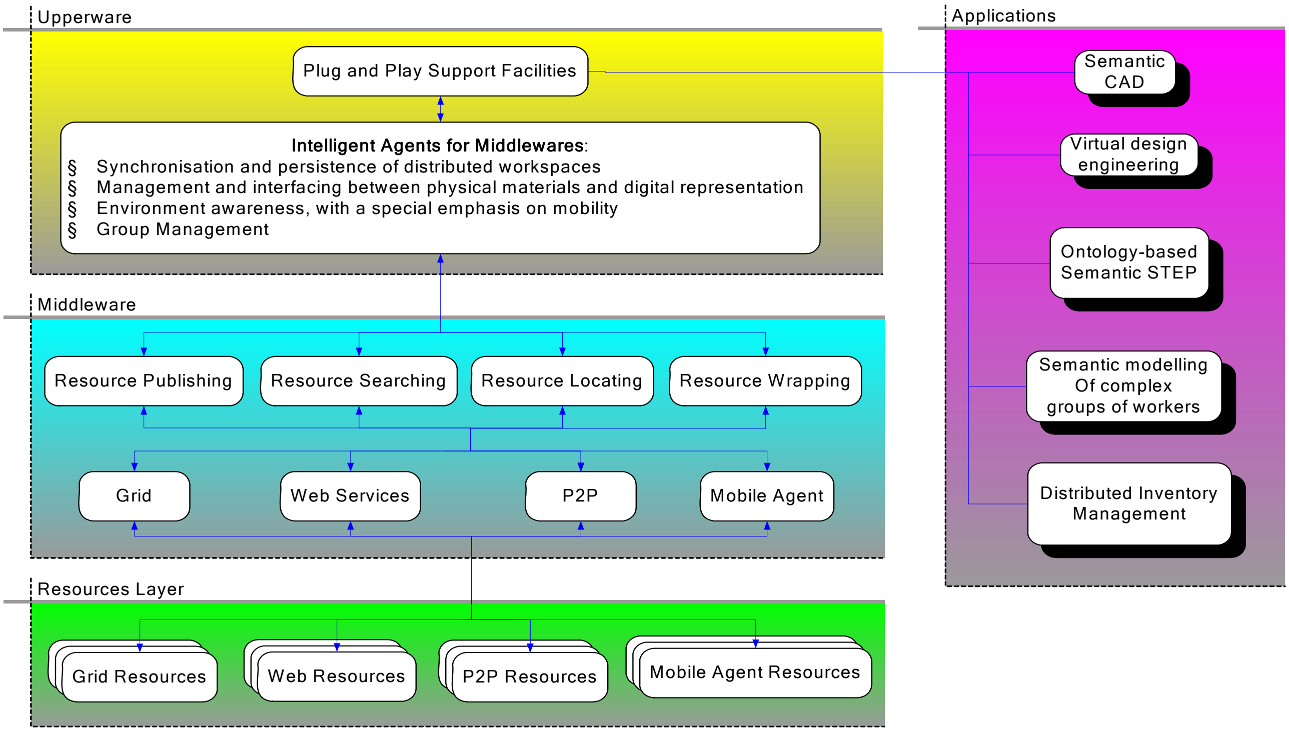
Upperware
Plug and Play Support Facilities
Intelligent Agents for Middlewares:
§ Synchronisation and persistence of distributed workspaces
§ Management and interfacing between physical materials and digital representation
§ Environment awareness, with a special emphasis on mobility
§ Group Management
Middleware
Resource Publishing
Resource Searching
Resource Locating
Resource Wrapping
Grid
Web Services
P2P
Mobile Agent
Resources Layer
Grid Resources
Web Resources
P2P Resources
Mobile Agent Resources
Applications
Semantic CAD
Virtual design engineering
Ontology-based Semantic STEP
Semantic modelling Of complex groups of workers
Distributed Inventory Management

Figure 1: Structure of the proposed framework

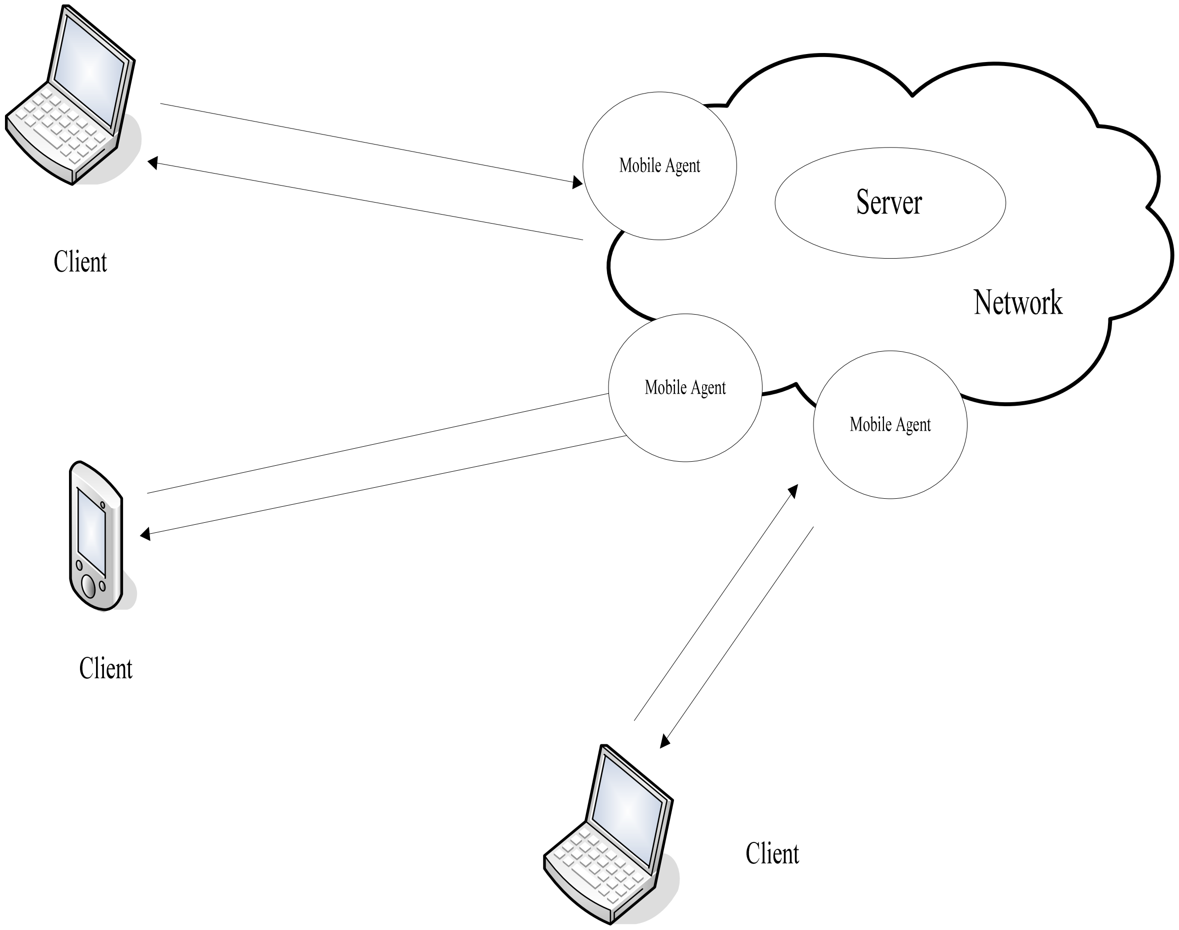

Figure 2: Communication within the MCWE

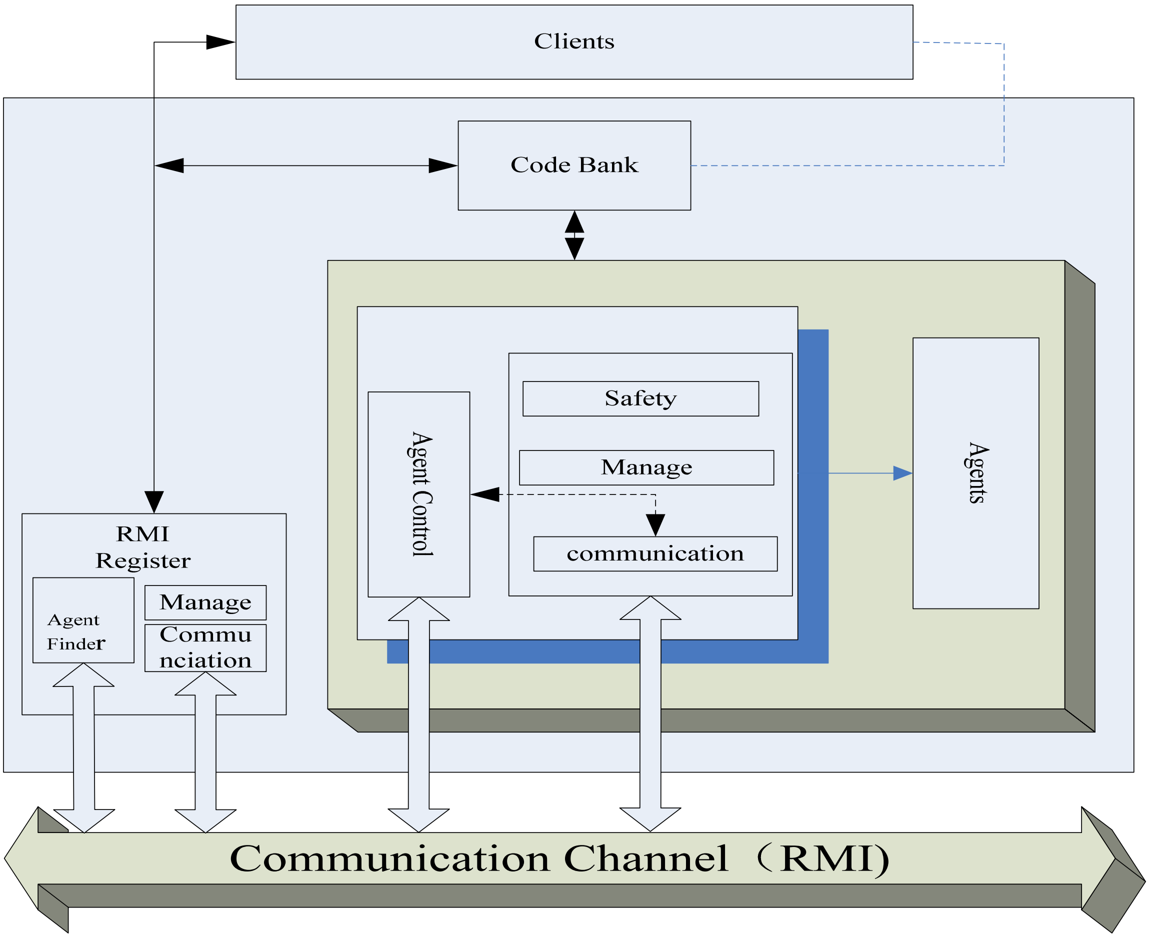

Figure 3. Agent system architecture