Chapter 10

AGENT-SUPPORTED WEB-BASED COOPERATIVE DESIGN

Weiming Shen
Integrated Manufacturing Technologies Institute
National Research Council Canada
weiming.shen@nrc.ca

Hamada Ghenniwa
Dept. of Electrical and Computer Engineering
University of Western Ontario
hghenniwa@eng.uwo.ca

Lihui Wang
Integrated Manufacturing Technologies Institute
National Research Council Canada
lihui.wang@nrc.ca

Abstract: The Internet and Web-based technologies are playing an increasingly important role in cooperative product development systems. However, most Web-based applications were developed using the client/server architecture for information sharing, rather than for real time cooperation. Indeed, the integration of agent and Web-based technologies to develop cooperative design systems will provide greater flexibility for legacy systems integration, a better environment for real time cooperation, and more secure environment for information sharing and communication. This chapter reviews the state-of-the-art of applications of the Web and Agent technologies to cooperative product design, presents some results of an on-going research project on developing an agent-supported Web-based framework for multidisciplinary design optimization (MDO). The primary objective of the proposed approach is to develop an architectural framework for integrating intelligent agents and the Internet/Web-based technologies. The principle is to leverage the right technology for the appropriate purpose. A prototype environment is presented for blow molded automotive parts design (simulation and optimization).

Y. Ye et al. (eds.), *Agent Supported Cooperative Work*

Key words: CSCW, Agents, Web, Internet, Cooperative Design, Multidisciplinary Design Optimization, Technology Integration.

1. INTRODUCTION

Real-world engineering design projects require the cooperation of multidisciplinary design teams using multiple sophisticated commercial and non-commercial engineering tools, such as CAD tools, optimization software, engineering databases, and knowledge-based systems. Individuals or groups of multidisciplinary design teams usually work, often for quite a long time, in parallel and separately with various engineering tools, which are located on different sites. In addition, at any moment, individual members may be working on different versions of a design or viewing the design from various perspectives, at different levels of details.

The Internet provides instant access to a wealth of design information, ranging from part library to 3D product models. The ability to access this information from anywhere at anytime makes the Internet an extension of the designer's reference library. The Internet becomes a unique infrastructure for resources integration, information sharing, and design collaboration. The popularity of the Internet is largely due to the influence of the World Wide Web, which has made the Internet accessible and available to mass population. Powered by the ever-improving information technologies, such as Java, search engines, email, HTML (Hyper Text Markup Language), XML (eXtensible Markup Language), and RDF (Resource Description Framework), the Web provides another familiar interface and gives us a common 'look and feel' to information exchange. As the use of the Internet and Web spreads, and because of globalization, the paradigm of the design activity is changing drastically. Specifically, there is an ever-increasing need for the continuous collaboration among geographically distributed design teams. The cooperative product design process is physically enabled by the Internet and Web technologies, and functionally supported by the technologies in the domain of artificial intelligence, such as agent technology and knowledge management. These enabling technologies serve as the wheels of the cooperative design vehicle to move forward.

This chapter introduces the cooperative product design problem (Section 2); reviews the state-of-the-art of Web-based cooperative product design (Section 3) and agent-based cooperative product design (Section 4); discusses the integration of agent and Internet/Web technologies (Section 5); describes a case study on the development of an agent-supported Web-based multidisciplinary design optimization environment (Section 6); and ends with some conclusions and perspectives.

2. COOPERATIVE PRODUCT DESIGN

Cooperative Design (also called Collaborative Design, Concurrent Design and Interdisciplinary Design) is the process of designing a product through collaboration among designers associated with the life cycle of the product. This would include those from functions as disparate as design, manufacturing, assembly, test, quality and purchasing as well as those from suppliers and customers (Sprow, 1992). The objectives of such a cooperative design might include optimizing the mechanical function of the product, minimizing the production or assembly costs, or ensuring that the product can be easily and economically serviced and maintained (Hartley, 1992). The process of cooperative design usually involves team members that are often working in parallel with different engineering tools distributed in separate locations, possibly across various time zones around the world.

Traditional design systems have used a sequential model of design generation, which breaks the design task into sub-tasks that are serially executed in a predefined pattern. Recently, researchers found that sequential design is brittle and inflexible and often requires numerous iterations. This makes the design expensive and time-consuming, and also limits the number of design alternatives that can be examined. Another undesirable aspect of the sequential design approach is its downstream information flow, in which information feedback from low-level (e.g., process planning or manufacturing at shop floor) to the high-level design is usually performed by human interactions. This may cause an insufficient design and hence inefficient product development due to the absence of manufacturability checks at the design stage. Cooperative design attempts to address these problems concurrently by detecting and considering conflicts and constraints at an earlier design stage.

To support cooperative design, computer technology must not only augment the capabilities of the individual specialists, but also enhance the ability of the collaborators to interact with each other and with computational resources. Additionally it should be able to support wide range of engineering design characteristics, such as diverse and complex forms of information, interdisciplinary collaboration, heterogeneous software tools, etc. Complicating the situation is that with the explosion of globalization, the design environment has become more complex and dynamic in which teams are highly geographically distributed and mobile. This shift of paradigm made traditional approaches and tools of sharing design information among collaborators incapable of supporting effective cooperative design. A number of emerging technologies including CSCW, distributed objects, intelligent agents, the Internet and Web have been proposed to implement distributed cooperative product design systems. In the following sections, we review the state-of-the-art of applications of the Web and Agent technologies to cooperative product design, and discuss the opportunities as well as challenges in this exciting research area.

3. WEB-BASED COOPERATIVE PRODUCT DESIGN

The World Wide Web was originally developed to allow information sharing within globally dispersed teams and the dissemination of information by support groups. Originally aimed at the High-Energy Physics community, it has spread to other areas and attracted much interest in user support, resource discovery and cooperative work areas. It is currently the most advanced information system deployed on the Internet [http://www.w3.org/].

The ability of the Web for designers to combine multimedia to publish information relevant to the spectrum of the design process, from concept generation and prototyping to virtual manufacturing and product realization, motivated the adoption of the Web as a cooperative product development (including design and manufacturing) tool. A Web-based cooperative product development system would primarily provide: (1) access to catalogue and design information on components and sub-assemblies; (2) communication among multidisciplinary design team members (including customers, designers and production engineers) on multimedia formats; (3) authenticated access to design tools, services and documents. The Web is now playing an increasingly important role in cooperative product development systems.

During the past few years, the Web-based infrastructure has been used in a number of cooperative product design systems. In most cases, the Web is primarily used by multidisciplinary team members as a medium to share design data/information/knowledge (Cutkosky et al., 1996; Roy and Kodkani, 1999; Toye et al., 1993; Wood and Agogino, 1996), and in some cases for product data management and project management through integration of the Web with the related technologies (Numata, 1996). In some other cases, the Web may only be used to monitor the status of the working system and the design process (Shen and Barthès, 1997).

A number of frameworks have been proposed for Web-based cooperative design systems (Huang and Mak, 1999b; Numata, 1996; Pahng et al., 1998; Roy and Kodkani, 1999), but most of them are still under proof-of-the-concept prototype development.

A commercial software suite called ipTeam [http://www.NexPrise.com/] is available from NexPrise Inc. for cooperative product development. However, it is primarily focused at a high level for virtual enterprise and supply chain management. It was derived from the DARPA-sponsored AIMS (Agile Infrastructure for Manufacturing Systems) program. Similar tools available through the Web for civil engineering projects have been reviewed by Novitski (2000). An extensive list of related software and web sites (extranets) including a few with cooperative design functionality is available at: http://www.extranets.cc/.

A number of CAD software vendors have recently released Internet/Web-based versions of their products, e.g., AutoDesk's AutoCAD2000i and AutoCAD2002 with eTransmit and i-drop technologies to support Internet/Web-based collaboration (http://www.autodesk.com/), PTC's Pro/Engineer based on Groove Networks Technology to support cross-firewall and real-time collaboration (http://www.ptc.com/), and EDS' Product Lifecycle Management (PLM) solutions to enable all participants in product lifecycle to work in concert (http://www.eds.com/products/plm/), etc. CoCreate's OneSpace Solution Suite supports collaborative design for most leading CAD solutions (http://www.cocreate.com/).

Some researchers have developed Web-based tools or systems based on standalone applications, e.g., Web-based DFX tools by Huang and co-workers (Huang et al., 1999; Huang and Mak, 1999c); WebCADET by Rodgers and co-workers (Caldwell and Rodgers, 1998; Rodgers et al., 1999).

Most Web-based cooperative design systems are developed using Java. Some others are developed by Common Lisp (e.g., WWDL by Zdrahal and Domingue (1997)), and Prolog (e.g., WebCADET by Rodgers et al. (1999)).

In addition to HTML and Java Applets for developing client side user interfaces, ActiveX and VRML are also widely used. Huang and co-workers have used ActiveX to develop Web-based DFX tools (Huang et al., 1999; Huang and Mak, 1999c), and Morphological Chart based cooperative conceptual design system (Huang and Mak, 1999a). A number of Web-based design systems (Allen et al., 1999; Klein, 1997; Roy et al., 1997) use VRML as a neutral representation for geometric models with the following reasons: (1) designers may work with heterogeneous software and hardware environments; (2) VRML files are easy to be viewed using standard Web browsers; (3) a number of CAD tools (latest versions) provide the functionality to export geometric models to VRML files, and no added development is needed. However, VRML can only be used to display the geometric models without the ability to edit or modify the models, though it allows designers to put some annotations or comments on the design (Klein, 1997). A more extensive review of Web-based cooperative design can be found in (Shen, 2000).

Most existing Web-based cooperative design systems adopt the client/server architecture. In order to support collaboration, Web-based design servers need to communicate the structure of the design representation so that users can pose queries about formal design concepts such as rationale and purpose or the causality between physical and functional elements. However, for a truly cooperative development environment, Web servers must engage users in a dialog-like interaction that encompasses a range of activities, such as geometric and semantic product modeling, design representation, user-interaction, and design browsing and retrieval. Unfortunately, the current Web technology cannot meet these requirements. In other words, information access is not the only major outstanding problem. In order to collaborate on a distributed project, remote engineers and designers need active help to coordinate their efforts. This coordination might involve translation of terminology among disciplines, locating/providing generic analysis services (e.g., finite element analysis), prototyping services, and project management. To the degree that Web servers are not mere repositories of information but engage users in active dialogue while providing such remote services in order to solve design problems, such servers may be called agents. We discuss agent based cooperative design in more details in the next section.

4. AGENT-BASED COOPERATIVE PRODUCT DESIGN

Agent technology was used to develop cooperative design systems before the Web. In fact, a number of earlier projects on agent-based cooperative design started when the Web was not yet available, e.g., PACT (Cutkosky et al., 1993) and DIDE (Shen and Barthès, 1994).

The case for using agents in industry is well founded by Parunak (1998). Considering that the main trends are increased product complexity and diversity, changes in supply networks, and increased product variety over time, Parunak has analyzed where agent technology can best be used in design and operation activities. His answer has been "agents are best suited for applications that are modular, decentralized, changeable, ill-structured, and complex". The reasons often given for adopting an agent approach are linked to their being pro-active object systems and to the simplification of the architecture of the software systems, knowing the cost of software is a major component of the total cost of constructing or operating a factory. The real gain obtained from an agent-based approach, however, often comes from a better description of the real world by focusing on objects rather than functions. When used appropriately, this leads to the desired modularity, allowing flexible simulations, and to better response and improved software reusability. In addition, the fact that agents can cope with a dynamically changing world by performing dynamic linking allows them to handle ill-structured or rapidly changing situations in a more economical way (Shen et al., 2000).

In agent-based cooperative design systems, agents have mostly been used for supporting cooperation among designers to provide semantic glue between traditional tools, or to provide better simulation platforms. An earlier review of multi-agent cooperative design systems can be found in (Lander, 1997). The book on "Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing" (Shen et al., 2000) provides a detailed discussion on issues related to the development of agent-based cooperative design systems as well as a comprehensive review of some related projects or systems. Here, we provide an overview of up-to-date advances in this area, with a specific attention to the integration of agent technology with other emerging technologies, such as the Web, CSCW (Computer Supported Cooperative Work) and Groupware, to develop cooperative product development systems.

The use of agents in design has been demonstrated by various research projects. PACT (Cutkosky et al., 1993) might be one of the earliest successful projects in this area. The interesting aspects of PACT include its federation architecture using facilitators and wrappers for legacy system integration. SHARE (Toye et al., 1993) was concerned with developing open, heterogeneous, network-oriented environments for concurrent engineering, particularly for design information and data capturing and sharing through asynchronous communication. SiFAs (Brown et al., 1995) was intended to address the issues of patterns of interaction, communication, and conflict resolution using single function agents. DIDE (Shen and Barthès, 1994, 1997) was developed to study system openness, legacy systems integration, and geographically distributed collaboration. ICM (Fruchter et al., 1996) provides a shared graphical modeling environment for cooperative design. Co-Designer (Hague and Taleb-Bendiab, 1998) was intended to support localized design agents in the generation and management of conceptual design variants. Concept Database (Varma et al., 1996) provides strategic design support for version control, workflow management and information gathering. A-Design (Campbell et al., 1999) presents a new design generation methodology, which combines aspects of multi-objective optimization, multi-agent systems, and automated design synthesis. It provides designers with a new search strategy for the conceptual stages of product design that incorporates agent collaboration with an adaptive selection of designs. A recent comprehensive review of related projects can be found in (Wang et al., 2002).

Similar to the Web-based design systems, agent-based systems also provide a cooperative environment for the sharing of design information, data, and knowledge among distributed design team members. In fact, much of the research conducted in building agent-based cooperative design systems has also focused on sharing information and data among agents. However, this could easily be achieved using the Web technology.

Unlike the Web-based design systems using the client/server architecture, an agent-based design system is "a loosely coupled network of problem solvers that work together to solve problems that are beyond their individual capabilities". Agents in such systems are communicative, cooperative, autonomous (or semi-autonomous), reactive, proactive, and intelligent. Different approaches have been proposed for system organization. Most systems use a federation approach utilizing facilitators, mediators, brokers, and other types of middle agents (Shen et al., 2000). Some systems use approaches similar to the blackboard architecture or client/server architecture, e.g., the Design Board approach in SiFAs (Brown et al., 1995); the shared graphical modeling approach in ICM (Fruchter et al., 1996); and the shared database approach by Varma et al (1996). The Autonomous Agent approach used in DIDE (Shen and Barthès, 1997) is different. Although various definitions have been proposed for autonomous agents, an autonomous agent should normally have the following characteristics: (1) it is not controlled or managed by any other software agents or human beings; (2) it can communicate/interact directly with any other agents in the system and also with other external systems; (3) it has knowledge about other agents and its environment; and (4) it has its own goals and an associated set of motivations.

Although agent technology has been recognized as a promising approach for cooperative design systems, those agents that have so far been implemented in various prototype and industrial applications actually either support a weak sense of, or lack the aspect of "intelligence".

Both agent technology and Web technology are very useful in developing cooperative design systems. The attractiveness of the Web for propagating information makes it appropriate to integrate with agents for accessing and manipulating information automatically. The challenge is how to build the Web-based environment that will enable and support seamless interaction between designers, agents and servers using the available emerging technologies.

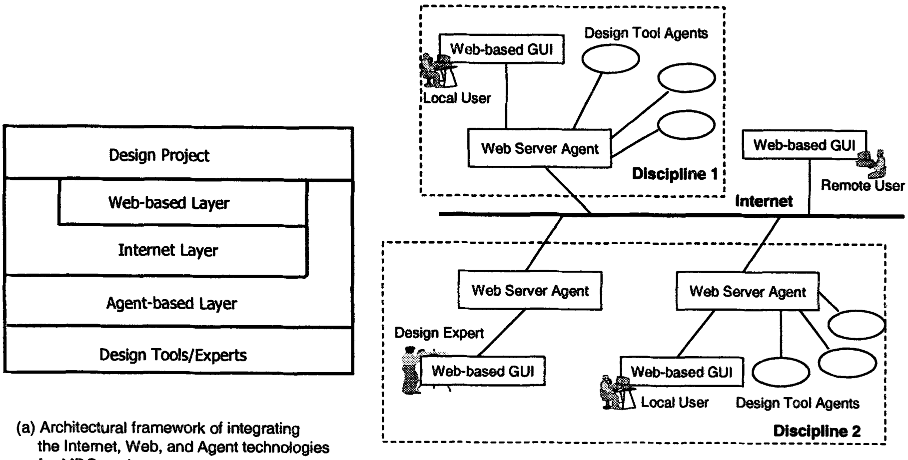
5. INTEGRATION OF AGENT AND INTERNET/WEB TECHNOLOGIES

To provide value without undue burden, it will be essential that the integration architecture assume responsibilities for the components interaction, promote cooperative behavior among them, and permit users to express their needs as high-level directives as opposed to component-oriented tasks. What is needed is a framework that enables agile, flexible, dynamic composition of resources and permits their interaction in a variety styles to match present and changing needs of design projects. This framework should go beyond the traditional views of integration; it should provide a level of abstraction at which a multidisciplinary design environment can be viewed collectively as a coherent universe.

Our research focuses on the idea of integration from the perspective of rapidly deployable cooperative distributed systems. The main integration aspects of a multidisciplinary design environment are composition, coordination, cooperation, and adaptation. Composition is the construction of a design solution from a set of entities that might heterogeneous and belong to different disciplines. Coordination is the ability of these entities to manage their interactions in order to solve the design problem. Cooperation is the ability of the entities to share their capabilities and knowledge as related to the problem. Adaptation is the ability of entities to recompose themselves, to improve cooperation with each other, and to re-task their activities, based upon an evaluation of their performance.

To overcome the shortcomings of the conventional approaches, we set an important objective to develop an architectural framework for integrating enabling technologies, namely, the Internet/Web, and Agent-orientation design paradigm for cooperative design or MDO (multidisciplinary design optimization) environments, as shown in Figure 1. This framework will merge these technologies in a way that each will play the appropriate role in designing an integration environment. In this work we strongly believe that agent-orientation is a very promising design paradigm for such an environment. The first principle of agenthood is that an agent should be able to operate as a part of a community of cooperative distributed systems environment, including human users. In fact, such a paradigm is essential to model open multidisciplinary design environments, especially considering the multiple dynamic and simultaneous roles a single expert or design tool may need to participate in given design project sessions. In addition, agent technology is rich in the sense of supporting and enabling the automation of complex tasks and developing systems that are reliable and able to take responsibilities of the design project in which they cooperate.

Therefore, in the proposed framework, Figure 1 (a), we view a design project as a collection of multidisciplinary design tools and experts that can be integrated to serve the objective of a design project. The proposed framework allows design tools and experts to collaborate and coordinate their activities collectively and coherently through their correspondence or representative agents, shown in Figure 1 (b). It is equally important that the framework supports platform- and language-independent environment as well as "anytime and anywhere" metaphor. To this end, as depicted in Figure 1, the agents will be enabled through the Internet technology to communicate and coordinate their activities for remote environments. Thus, a design project will be able to exploit the potentially vast body of expert forces that are distributed over different geographical and computational environments, whereas the Web-based technology, such as HTML and XML, will complement the Internet by providing the communication harmony between human experts and users, and the design tools.



(a) Architectural framework of integrating the Internet, Web, and Agent technologies for MDO environments

(b) Technology integration to develop MDO environments

Figure 1. Architectural framework for technology integration in MDO environment.

6. A CASE STUDY: DEVELOPMENT OF A DISTRIBUTED MDO ENVIRONMENT

Multidisciplinary design optimization (MDO) is an appropriate design methodology, especially for complex engineering systems that are geographically distributed and are governed by mutually interacting physical phenomena and made up of distinct interacting subsystems (Li and Hopper, 1998). These subsystems are usually implemented in different, possibly heterogeneous, computers connected through a network to support multidisciplinary design projects carried out by different teams.

This section presents some results of an international collaborative research project on the development of an MDO methodology as well as a distributed MDO software environment for blow molded automotive parts design optimization.

6.1 The Project

Blow molding is the most popular and efficient process for manufacturing commodity hollow plastic parts such as bottles, containers, toys, etc. More recently, this forming process has been applied to the manufacture of complex automotive parts such as fuel tanks, seat backs, air ducts, windshield washer and cooling reservoirs, etc. The quality of these complex hollow parts is governed by several parameters such as:

- Material properties;
- Operating conditions;
- Tooling parameters associated with the mold and the die design;
- Mechanical performance of the final part.

Blow molded part designers in today's global environment are under increasing pressure to beat the triangle dilemma, minimize both development time and cost while ensuring better quality. In almost all industrial sectors, simulation technologies have proven to be powerful tools for achieving a small-scale attainment of this goal. The use of optimization-based design allows the designers to treat complex design criteria via simulation and is expected to increase dramatically.

In order to address industrial concerns about design tools for blow molding, an international collaborative research project has been established between two institutes of the National Research Council of Canada (NRC) and two universities in Taiwan (Thibault et al., 2001). The primary objective of this project is to build a multidisciplinary design optimization (MDO) software environment for the design of blow molded automotive parts. The proposed methodology includes distributed system integration using intelligent agent and Internet/Web technologies; multiple optimization methods including gradient-based optimization techniques and soft computing techniques (neural networks, fuzzy logic, and genetic algorithms, etc.).

The main requirements of blow molded automotive components usually combine part weight, thickness distribution, part dimension, shape and mechanical performance. These design variables can be controlled by manipulating the mold shape and operating conditions in order to minimize part weight subject to mechanical performance. The proposed MDO software environment will integrate process simulation and performance simulation tools into an optimization procedure in order to minimize the part weight subject to mechanical constraints in service.

The process modeling technology is capable of simulating the consecutive phases of the forming process as a function of the mold geometry, material properties, and operating conditions. It can predict part quality such as weight, thickness distribution, shrinkage and warpage. The performance modeling technology can simulate the mechanical behavior in service and predict the part resistance for a given load.

BlowSim is a finite element software simulation package for extrusion blow molding, injection stretch blow molding, and thermoforming processes. This software has been developed at the Industrial Materials Institute (IMI) of National Research Council of Canada (Thibault et al., 2001).

The optimization process phase consists of minimizing the part weight subject to the minimum thickness distribution of the part produced by the mechanical performance optimization process. In addition to traditional gradient-based design optimization techniques, the project team has developed alternative new optimization methods including:

- A fuzzy optimization algorithm to determine the optimal die openings of the programming points in the extrusion blow molding process (Hsu et al., 2001).
- A soft computing strategy to determine the optimal die openings in the extrusion blow molding process. This method improves search efficiency by applying engineering knowledge to genetic algorithms using fuzzy rules (Yu et al., 2001). The adopted Fuzzy Neural-Taguchi strategy establishes a back propagation network using Taguchi's experimental array to predict the relationship between design variables and response.

6.2 Benefits of Using the Web and Agents

As mentioned above, the Web technology is becoming more and more popular in implementing cooperative product design environments. Web-based approaches for the implementation of a blow molded automotive parts design system have several advantages:

- Software does not need to be installed on the client site, which in turn reduces design costs for user companies (particularly SMEs) by eliminating both software/hardware purchase and installation costs.
- Software upgrades need to be done only on the server site, which will save time and money for both the software suppliers and user companies.
- Software suppliers will be able to protect their software from illegal duplications and distributions.

In fact, several new businesses have been established based on providing this kind of services, usually called ASP (Application Service Providers).

ASPs provide user companies with a leasing vehicle for large-scale complex applications, using Web browsers to run the applications on the ASPs' hosted infrastructure. A major problem facing ASPs is the decline of the user companies interest due to their concern about the confidentiality and security of their data. Nonetheless the cooperative design environment approach is still of great value if applied within an enterprise with multiple design and development teams, or multiple departments, that might be geographically distributed.

Although Web technology plays an important role for promoting and supporting sharing of design information, it is not sufficient to support real-time cooperative design. It is not flexible enough for legacy systems integration as well as for computing resource management. To deal with these issues, we propose an agent-oriented approach for Web-based cooperative design systems that will:

- provide greater flexibility for legacy systems integration through socket-based communication among local resource agents behind a Web server;
- enable real-time collaboration through communication among active Web servers which are implemented as autonomous intelligent agents communicating with each other actively. Such an implementation also provides a way to integrate various legacy systems separated by firewalls;
- improve significantly the performance of development and design process, particularly in the cases of multiple projects and multiple users working at the same time.

6.3 The Prototype System

Figure 2 shows the logical system architecture of the current prototype implementation. This system extends traditional Web-based cooperative systems by introducing four types of agents:

1) Web servers are active in the sense that they are able to participate in peer-to-peer interaction by sending dynamic information to passive client side browsers as well as to other Web servers and agents.
2) Design Tool Agents are used to represent design and simulation tools for finite element method based simulation, optimization, and geometry manipulation, etc. Most of these design tools are legacy systems, and are encapsulated into agents using wrappers (Shen and Barthès, 1997). Representative agents, such as BlowSim agents provide Blow Molding simulation services as needed. Agents of different design tools might "collaborate" with each other to accomplish a simulation task. Also, they might "compete" to attain a better overall result or performance.

Negotiation-based scheduling techniques (Shen and Norrie, 2001) are used to achieve load balance among these agents.

3) Yellow Page Agent(s) register participant design tool agents (e.g., BlowSim Agents) and provide a look-up service for active Web servers (e.g., BlowSim servers). The advantage of registering design tool agents with a YP agent rather than with active Web servers is to ensure the system performance and consistency (by having multiple Web servers to contact the same YP agent) and to increase the system reliability.
4) (User) Interface Agents provide human users, such as project managers and designers, with appropriate Web-based interfaces to interact with the system. A subproject is underway to develop different types of interface agents as intelligent advisors to support different end users.

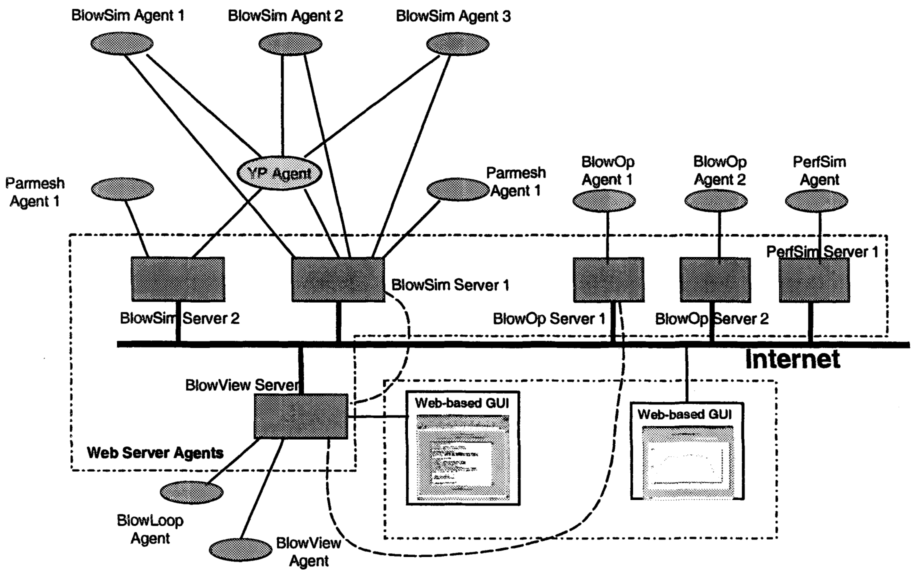


Figure 2. Prototype Implementation.

6.4 A Scenario

Here is a scenario showing the current working prototype environment:

- A user should work on a computer installed with the BlowView Server, the BlowView and BlowLoop agents.
- The user needs to use the BlowView locally to prepare all definition files.
- The prepared definition files will be sent to BlowSim Server (1 or 2) for process simulation.

- BlowView Server will collect the results back from BlowSim Servers. The BlowLoop Agent will be asked to extract useful data (initial parameter values and evaluation function values) from the results. The related data will then be sent to BlowOp Server (1 or 2) for optimization (to generate an improved solution with new parameter values for next simulation, or a set of new solutions in the case of GA based optimization).
- The BlowView Server will collect the results back from the BlowOp Server.
- BlowView Server asks the BlowLoop Agent to prepare new definition files and send them to BlowSim Server (1 or 2) for next round of simulation(s).
- Above processes will be repeated for 5 to 7 times.
- The final results will be sent back to the BlowView Server.
- The BlowView Agent will then be activated automatically to display the optimization results in a 2D graphical interface.

The above optimization process will be done automatically after the user launches the project through a Web browser. The user will be able to monitor the simulation and optimization process through a Web browser (see Figure 3 as an example of the Web-based monitoring interfaces), and to interrupt the process if necessary. The monitoring and interruption actions can be done on a Web browser anywhere through the Internet. However, the project launch and final results display in a 2D graphical interface can only be done on the computer installed with the BlowView Server.

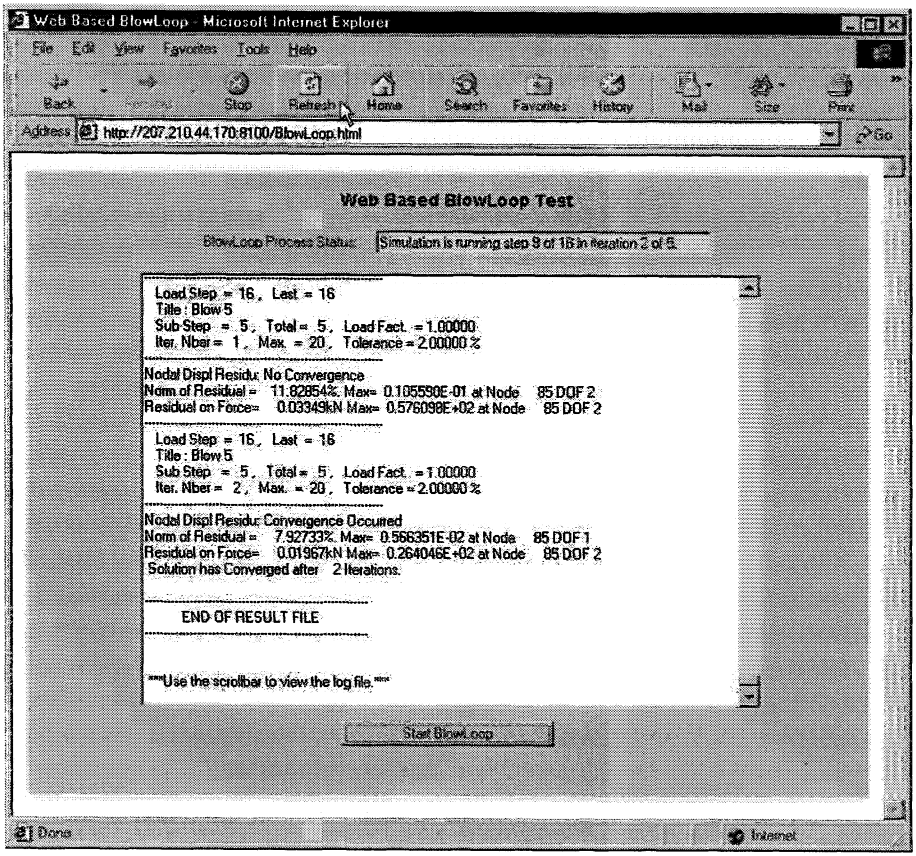


Figure 3. A Web-based Monitoring Interface.

Under the current prototype implementation, browser access through the Internet (rather than the machine installed with the BlowView Server and the BlowView Agent) can only be used to monitor the simulation/optimization process in a text window (Figure 3) or in a graph window (Figure 4). However, it is expected that these limitations will be eliminated in a late prototype implementation.

The integration of multiple optimization software packages as well as the performance simulation will be investigated in the next stage.

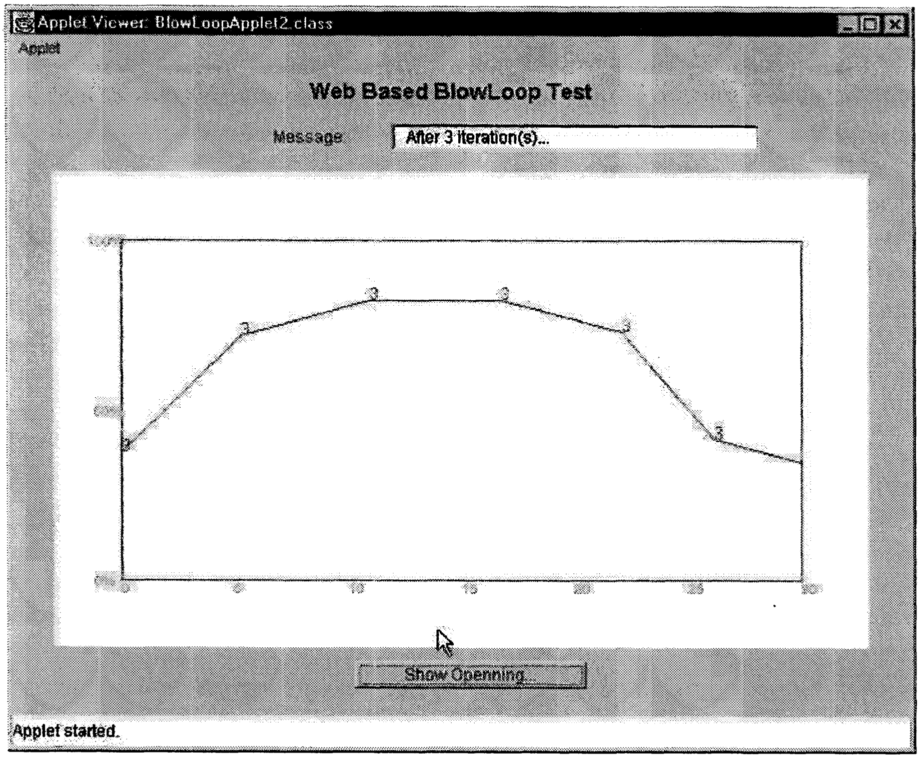


Figure 4. A Sample Graph Window.

6.5 Some Implementation Issues

6.5.1 Hardware and Software

The hardware and software environment is set up as requested by the current users of the standalone software:

- The current prototype is being implemented on a network of PCs and SUN workstations.
- The JRun application server is used due to its free license (at the time of prototype implementation) and sufficient performance.
- The primary programming language is Java, but Fortran and C++ are also used for legacy software integration.

6.5.2 Web-based User Interfaces

Web-based user interfaces are developed in Java Applet for project managers and designers to start the design, simulation and optimization processes, and monitor the real time summary report of this process in both text and graph formats. Java Applet is selected for this purpose because an Applet-Servlet communication is needed to display real time information. Intelligent Interface Agent techniques are implemented to provide powerful Web-based user interfaces. More details are to be reported separately.

6.5.3 Web Access and Security Issues

A major concern of implementing Internet-based systems is the assurance that proprietary information about the intellectual property owned by the organization is available only to authorized individuals. For general acceptance of the Internet-based approach, the secrecy of the proprietary information must be maintained. We consider the following issues during the prototype implementation:

- All accesses to the prototype environment are password protected.
- Most web servers will be accessible only to the registered computers (IP addresses). This is easy to be done using the JRun application Server.
- All control inputs could only be done on the computer with the BlowView Server. Remote access using Web Browsers is only used to monitor the simulation and optimization process.
- TCP/IP socket communication is used in the prototype for communication among BlowSim agents. All these BlowSim agents should be behind a same firewall. Same techniques cannot be used for communication among Web servers, which are usually behind different firewalls.
- Other security issues need to be studied during the following phases of the project.

6.5.4 Multi-User and Multi-Project Issues

In the first prototype implementation, multiple users will be allowed to monitor the simulation and optimization process.

Although it is not a problem to have several users to run several projects at the same time by having several similar client side servers (BlowView servers), the actual prototype implementation will have only one client side Web server (BlowView Server). Because of heavy calculation load of BlowSim servers, other techniques and mechanisms are needed. The Contract Net based negotiation among BlowSim agents is implemented as the first author implemented previously for agent-based dynamic manufacturing scheduling (Shen and Norrie, 2001).

Each design project may also have multiple design versions. A simple versioning mechanism is to be implemented to allow users to retrieve results of previous designs and navigate among different design versions to make a final decision.

6.5.5 Intermediate Data Storage

In order to facilitate the monitoring process and ensure the system reliability, a database is used by each Web server to store all intermediate simulation and optimization results. A project manager or designer is therefore able to access the intermediate results through the related Web server. This implementation strategy raises the system centralization concern, but it facilitates significantly the system implementation, and the current end users also requested this kind of implementation.

6.5.6 Post-Processing of Results

NRC-IMI is working on the automatic generation of graphic files (e.g., VRML, JPEG, or BMP, etc.) based on the optimization results so that the results will be visually available on the Web when the process is finished.

6.5.7 XML Based Message Services

Since XML documents can be exchanged and used across dissimilar platforms and applications, it is very suitable to use it as a standard for data exchange when applications are related to business transitions, e.g., for electronic commerce, supply chain management or virtual enterprises (Shen et al., 2001). Although XML itself is still evolving and its supporting tools are still under development, it is being promoted by leading industrial companies like Microsoft (http://www.microsoft.com/xml/) and IBM (http://www.ibm.com/xml/). It will become a widely accepted international standard for all kinds of Internet-based applications. XML will be used for formatting messages among all cooperative agents. According to our recent R&D work in other related projects (Shen et al., 2001), XML holds several advantages over traditional Java Object Serialization, including:

- XML is easily understood and modified by both humans and agents, while Java Serialization is unreadable by humans;
- XML is platform and implementation neutral. This allows for agents to be implemented in virtually any language and any platform and still maintain 100% interoperability.

- XML is extensible by design, and offers a natural versioning mechanism. As new agents and data fields are introduced to the system, legacy agents that do not use these new options do not have to be recompiled and redeployed. This type of version control comes very easily with XML (in the form of XML Namespaces), while it requires much more effort to implement with Java Object Serialization.

6.5.8 XML Based Data Management

Another very important issue is the data management, including the management of input definition files, optimization results, as well as the information related to project management, design versions management, and users management. The traditional way is to have a database management system with a relational database. However, since XML is being used to provide message services among agents and Web servers. It is nature to consider the implementation of such a data management system using XML, i.e., all above mentioned data will be managed through an XML file (instead of a database) with links to detailed data files (definition files, and result files, etc.).

7. CONCLUSIONS AND PERSPECTIVES

This chapter presents an overview of state-of-the-art approaches integrating and applying Agent and Web-based technologies to cooperative product design. Also, we presented some results of our on-going research on the development of a Web/Agent-based distributed multidisciplinary design optimization (MDO) framework. The key contributions of this research include:

- the integration of agent technology with Web-based systems to provide greater flexibility for legacy systems integration;
- supporting real time collaboration of product development by enabling communication among active Web servers;
- load balancing among computation resources through agent-based negotiation;
- the ability to support multiple optimization software packages from different vendors or service providers with the ability to choose the appropriate one for the product development.

The proposed approach is distinct from other Web-based cooperative design systems in that software agents are integrated with Web-based systems to provide greater flexibility through agents' promising features such as autonomy, modularity and scalability. When the proactiveness of agents is implemented, i.e., the problem-solving agents (design tool agents in the case study) are implemented to be able to take initiatives to participate in a working project, the system performance will be further enhanced.

The proposed approach is also different from classical agent-based cooperative design systems since it takes full advantages of XML and Web-based technologies to ensure the system stability, reliability and usability. However, in continuing the research, the following are key challenges and the current concerns of this project:

- experimental investigation and pragmatic issues related to environments with multi-user in multi-project situations;
- trade-off analysis related to the effective communication among active Web servers (limitations of the high level "request forwarding" and firewall barriers of raw socket communications);
- application of the concepts of cooperation and competition to achieve effective load balancing among BlowSim agents;
- experimental investigation and pragmatic issues related to applying autonomy and proactiveness of design tool agents;
- analysis of pragmatic issues related to systems security and privacy.

8. REFERENCES

Allen, R.H., Nidamarthi, S., Regalla, S.P. and Sriram, R.D. (1999) 'Enhancing collaboration using an Internet integrated workbench', *Proceedings of DETC99 – 1999 ASME Design Engineering Technical Conference*, Las Vegas, NV.

Brown, D.C., Dunskus, B., Grecu, D.L. and Berker, I. (1995) 'SINE: Support for single function agents', *Proceedings of Applications of AI in Engineering*, Udine, Italy.

Caldwell, N.H.M. and Rodgers, P.A. (1998) 'WebCADET: facilitating distributed design support', *IEE Colloquium on Web-Based Knowledge Servers*, London, U.K., pp. 9/1-9/4.

Campbell, M.I., Cagan, J. and Kotovsky, K. (1999) 'A-Design: An agent-based approach to conceptual design in a dynamic environment', *Research in Eng. Design*, Vol. 11, pp. 172-192.

Cutkosky, M.R., Engelmore, R.S., Fikes, R.E., Genesereth, M.R., Gruber, T.R., Mark, W.S., Tenenbaum, J.M. and Weber, J.C. (1993) 'PACT: An experiment in integrating concurrent engineering systems', *IEEE Computer*, Vol. 26, No. 1, pp. 28-37.

Cutkosky, M.R., Tenenbaum, J.M. and Glicksman, L. (1996) 'Madefast: Collaborative Engineering over the Internet', *Communications of the ACM*, Vol. 39, No. 9, pp. 78-87.

Fruchter, R., Reiner, K.A., Toye, G. and Leifer, L.J. (1996) 'Collaborative mechatronic system design', *Concurrent Engineering: Research and Applications*, Vol. 4, No. 4, pp. 401-412.

Hague, M.J. and Taleb-Bendiab, A. (1998) 'Tool for management of concurrent conceptual engineering design', *Concurrent Engineering: Research and Applications*, Vol. 6, No. 2, pp. 111-129.

Hartley, J. (1992) *Concurrent Engineering*, Cambridge, Mass., Productivity Press.

Huang, G.Q., Lee, S.W. and Mak, K.L. (1999) 'Web-based product and process data modelling in concurrent 'design for X'', *Robotics and Computer-Integrated manufacturing*, Vol. 15, No. 1, pp. 53-63.

Huang, G.Q. and Mak, K.L. (1999a) 'Web-based morphological charts for concept design in collaborative product development', *Journal of Intelligent Manufacturing*, Vol. 10, pp. 267-278.

Huang, G.Q. and Mak, K.L. (1999b) 'Web-based collaborative conceptual design', *Journal of Engineering Design*, Vol. 10, No. 2, pp. 183-194.

Huang, G.Q. and Mak, K.L. (1999c) 'Design for manufacture and assembly on the Internet', *Computer in Industry*, Vol. 38, No. 1, pp. 17-30.

Hsu, Y.L., Liu, T.C., Hsu, M.S., and Thibault, F. (2001) 'A fuzzy optimization algorithm for blow moulding process', *Proceedings of the First International Workshop on Multidisciplinary Design Optimisation*, London, ON, Canada,, pp. 9-14.

Klein, M. (1997) 'Capturing geometry rationale for collaborative design', *Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE'97)*, pp. 24-28.

Lander, S.E. (1997) 'Issues in multiagent design systems', *IEEE Expert*, Vol. 12, No. 2, pp. 18-26.

Li, S.F. and Hopper, A. (1998) 'A framework to integrate synchronous and asynchronous collaboration', *Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE'98)*, pp. 96-101.

Novitski, B.J. (2000) 'Team Building', *Computer Graphics World*, Vol. 23, No. 4, pp. 34-40.

Numata, J. (1996) 'Knowledge Amplification: an information system for engineering management', *Sony's Innovation in Management Series*, Vol.17, Sony Corporation, Japan.

Pahng, G.D.F., Bae, S. and Wallace, D. (1998) 'A web-based collaborative design modeling environment', *Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises (WET ICE'98)*, pp. 161-167.

Parunak, H.V.D. (1998) 'What can agents do in industry, and why? An overview of industrially-oriented R&D at CEC', *Cooperative information agents II: learning, mobility and electronic commerce for information discovery on the Internet: Second International Workshop, CIA'98*, Klusch, M. and Weiss, G. (ed.), Paris, France, Springer, pp. 1-18.

Rodgers, P.A., Huxor, A.P. and Caldwell, N.H.M. (1999) 'Design support using distributed Web-based AI tools', *Research in Engineering Design*, Vol. 11, pp. 31-44.

Roy, U., Bharadwaj, B., Kodkani, S.S. and Cargian, M. (1997) 'Product development in a collaborative design environment', *Concurrent Engineering: Research and Applications*, Vol. 15, No. 4, pp. 347-365.

Roy, U. and Kodkani, S.S. (1999) 'Product modeling within the framework of the World Wide Web', *IIE Transactions*, Vol. 31, No. 7, pp. 667-677.

Shen W. and Barthès J.P. (1994) 'A Distributed Architecture for Design Environment Using Asynchronous Cognitive Agents', *Proceedings of the Second Singapore International Conference on Intelligent Systems*, Singapore, pp. B327-334.

Shen, W. and Barthès, J.P. (1997) 'An experimental environment for exchanging engineering design knowledge by cognitive agents', *Knowledge Intensive CAD-2*, Mantyla M., Finger S., and Tomiyama T., (eds.), Chapman & Hall, pp. 19-38.

Shen, W. (2000) 'Web-Based Infrastructure for Collaborative Product Design: an Overview', *Proceedings of the Sixth International Conference on Computer Supported Cooperative Work in Design*, Hong Kong Polytechnic University, Hong Kong, Nov. 29 – Dec. 1, pp. 239-244.

Shen, W., Norrie, D.H. and Barthès, J.P. (2000) *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*, Taylor and Francis, London, UK.

Shen W. and Norrie, D.H. (2001) Dynamic Manufacturing Scheduling Using both Functional and Resource Related Agents', *Integrated Computer-Aided Engineering*, Vol. 8, No. 1, 17-30.

Shen, W., Brooks, C., Li, Y., Lang, S., and Wang, L. (2001) 'XML-based message services for internet-based intelligent shop floors', *Proceedings of the SPIE International Conference on Internet-Based Enterprise Integration and Management*, Boston, MA, SPIE Vol. 4566, pp. 135-144.

Sprow, E. (1992) 'Chrysler's Concurrent Engineering Challenge', *Manufacturing Engineering*, Vol. 108, No. 4, pp. 35-42.

Thibault, F., Gauvin, C., Laroche, D., and DiRaddo, R. (2001) 'Development of an MDO Software Environment for the Blow Moulding Process', *Proceedings of the First International Workshop on Multidisciplinary Design Optimisation*, London, ON, Canada, pp. 1-8.

Toye, G., Cutkosky, M., Leifer, L., Tenenbaum, J. and Glicksman, J. (1993) 'SHARE: A methodology and environment for collaborative product development', *Proceeding of 2nd Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, IEEE Computer Society Press, pp. 33-47.

Varma, A., Dong, A., Chidambaram, B., Agogino, A. and Wood, W. (1996) 'Web-based tool for engineering design', *Proceedings of AID'96 Workshop on Agents and Web-Based Design Environments.*

Wang, L., Shen, W., Xie, H., Neelamkavil, J. and Pardasani, A. (2002) 'Collaborative conceptual design: a state-of-the-art survey', *CAD*, Vol. 34, No. 13, pp. 981-996.

Wood, W.H. and Agogino, A.M. (1996) 'A case-based conceptual information server for concurrent engineering', *CAD*, Vol. 28, No. 5, pp. 361-369.

Yu, J.C., Chen, X.X., and Hung, T.R. (2001) 'Optimization of extrusion blow molding using soft computing and Taguchi method', *Proceedings of the First International Workshop on Multidisciplinary Design Optimisation*, London, ON, Canada, pp. 15-22.

Zdrahal, Z. and Domingue, J. (1997) 'The World Wide Design Lab: an environment for distributed collaborative design', *Proceedings of 1997 International Conference on Engineering Design*, Tampere, Aug. 19-21