A Distributed MDO Environment Based on Web and Agents

Weiming Shen, Ying Wang, Yan Ma and Yinsheng Li
Integrated manufacturing Technologies Institute, National Research Council Canada
800 Collip Circle, London, Ontario, Canada, N6G 4X8
[weiming.shen, ying.wang, yan.ma, yinsheng.li]@nrc.ca

Hamada H. Ghenniwa
Dept. of Electrical and Computer Engineering, University of Western Ontario
1151 Richmond Street, London, Ontario, N6A 5B9,Canada
hghenniwa@eng.uwo.ca

Francis Thibault
Industrial Materials Institute, National Research Council Canada
75 boul. de Mortagne, Boucheville, Quebec J4B 6Y4, Canada
francis.thibault@nrc.ca

Yeh-Liang Hsu
Dept. Of Mechanical Engineering, Yuan Ze University
135 Yuan-Tung Road, Chung-Li, Taiwan
mehsu@saturn.yzu.edu.tw

Jyh-Cheng Yu
Dept. of Mechanical Engineering, National Taiwan University of Science and Technology
43 Keelung Road, Sec. 4, Taipei, Taiwan
jcyu@mail.ntust.edu.tw

Abstract

This paper presents some results of an ongoing NRC-NSC joint project on Multidisciplinary Design Optimization. It focuses on the development of a Web/agent based distributed Multidisciplinary Design Optimization (MDO) environment called WebBlow which is being implemented using a number of emerging technologies such as the Internet, Web, XML, Java, and intelligent agents. The Web provides the infrastructure for implementing the distributed MDO environment over the Internet. It provides an anytime/anywhere solution for project managers and designers working on multiple design and optimization projects. It can also ensure the system security and mitigate the difficulties with firewalls encountered by conventional software agents. Intelligent agents are used to encapsulate process simulation and performance simulation as well as optimization software packages in order to facilitate the integration of various legacy systems, and more importantly realize the dynamic computing load balancing among a cluster of computers behind the Web server. XML is used for data management at the server side as well as for message exchange among simulation and optimization software agents. A prototype environment is presented for blow molded automotive parts design.

Keywords: *Multidisciplinary Design Optimization, Web, Agents, Concurrent Engineering, Collaborative Design, Computing Load Balancing.*

1 Introduction

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

In order to coordinate activities of multidisciplinary design teams and to guarantee the interoperability among the different engineering tools, it is necessary to have efficient collaborative design environments. These environments should not only automate individual tasks, in the manner of traditional computer-aided engineering tools, but also enable individual members to “share information”, “collaborate” and “coordinate” their activities within the context of a design project. This paper reports some results of an ongoing NRC-NSC joint project on Multidisciplinary Design Optimization. It focuses on the development of a Web/agent based distributed Multidisciplinary Design Optimization (MDO) environment called WebBlow which is being implemented using a number of emerging technologies such as the Internet, Web, XML, Java, and intelligent agents. The rest of this paper is organized as follows: Section 2 reviews the research literature; Section 3 provides a brief project description; Section 4 describes the system requirements; Section 4 presents the system design; Section 5 reports the system implementation results; and Section 6 gives some conclusions and perspectives.

2 Research Literature

An extensive literature survey on Web/agent based collaborative design has been done by the project members during the past three years. The detailed survey results have been presented at CSCWD 2000 (Shen, 2000; Wang et al., 2000). This section only provides an overview of the research literature.

The Web is now playing an increasingly important role in developing collaborative product development systems. A number of frameworks have been proposed for Web-based collaborative design systems, but most of them are either to support fundamental aspects of collaborative design or still under proof-of-the-concept prototype development (Shen, 2000).

A number of CAD software vendors have recently released Internet/Web based versions of their software 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 using most leading CAD solutions (http://www.cocreate.com).

Web-based collaborative design systems, in general, are usually based on client/server architecture, in which the interaction among components is predefined. This kind of approach is not sufficient enough to support integration in a multidisciplinary design environment. Tasks in such an environment are usually involving complex and non-deterministic interactions, producing results that might be ambiguous and incomplete. In addition, the dynamic nature of the environment requires that the components of the system be able to change configuration to participate in different, often simultaneous roles in design projects.

Agent-based integration architecture is an appropriate foundation framework for building flexible, dynamic communities of distributed software systems to meet these challenges. Detailed discussions on the advantages of using intelligent agents in collaborative design and an extensive survey of related projects can be found in (Shen et al., 2000; Wang et al., 2000).

3 Project Description

Blow molding is a popular and efficient manufacturing process for hollow plastic parts such as bottles, containers, toys, etc. More recently, this forming process has been applied to produce complex automotive parts such as fuel tanks, seat backs, air ducts, windshield washer and cooling reservoirs and even planar parts with carpet adhered to one of the sides. The quality of these complex hollow parts is governed by several parameters such as material properties, operating conditions, tooling parameters associated with the mould and the die design, and mechanical performance of the final part.

Blow molded parts designers in today’s global environment are under increasing pressure to beat the triangle dilemma, i.e., minimizing both development time and cost while ensuring best 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 advanced design tools for blow molding, an international joint research project was initiated in 2000 under the NRC-NSC Joint Research Program. The principal participants are from National Research Council Canada's Industrial Materials Institute (NRC-IMI) and Integrated Manufacturing Technologies Institute (NRC-IMTI), National Taiwan University of Science and Technology and Yuan Ze University in Taiwan (Thibault et al., 2001).

The primary objective of this project is to build a multi-disciplinary design optimization (MDO) software environment for the design of automotive interior blow molded 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 contribution of each research team is illustrated in Figure 1 (Thibault et al., 2001).

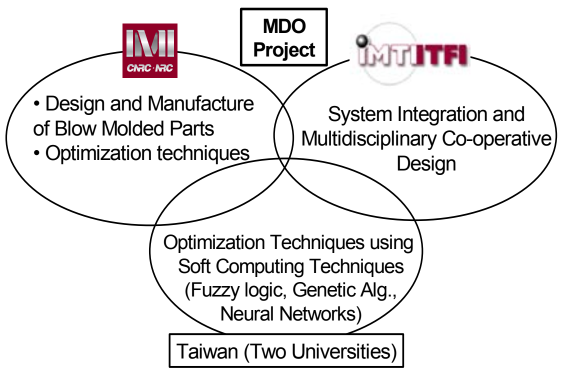

Figure 1. The Project Team

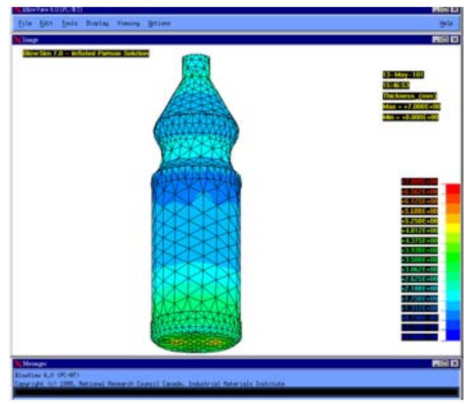

Figure 2. The Sample Blow Molding Problem

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 mould shape and operating conditions in order to minimize part weight subjected to mechanical performance. The proposed MDO software environment will integrate process simulation tools, performance simulation tools into an optimization procedure in order to minimize the part weight subjected 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 mould 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 NRC-IMI. Figure 2 shows the thickness distribution of a sample problem simulated by BlowSim when the gap opening is set at 75% at all time.

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 include:

- 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 algorithm using fuzzy rules to (Yu et al., 2001). The adopted Fuzzy Neural-Taguchi strategy, first establishes a back propagation network using Taguchi's experimental array to predict the relationship between design variables and response.

One of the major tasks of this joint research project is to develop a software environment that integrates the blow molding process simulation, performance simulation and the optimization algorithms in a way that reduces the development time of the blow molding and improve the part quality of the product.

4 System Requirements

The objective of the current project is to build a distributed multidisciplinary design optimization (MDO) software environment (called WebBlow) for the design of blow molded parts. The proposed methodology includes distributed system integration using intelligent agents and Internet/Web technologies. A detailed description of the system requirements can be found in (Wang, 2002a). These requirements can be summarized as follows:

- Web based user interfaces are the only interfaces between users and the WebBlow software.
- User should be able to input user name, group name and password for user validation.
- User should be able create a new project; find an existing project from the system and view project related data; view an existing project's optimization data.
- User should be able to choose an optimization method/tool from existing tools including Gradient-based, Fuzzy Logic, and FUNTGA (Yu et al., 2001), and choose an optimization objective from process optimization, performance optimization, and warpage optimization.
- When user chooses process optimization as the optimization objective, user should be able to choose an optimization target from parison length optimization, part thickness variance minimization, and part weight minimization.
- For any defined optimization method, optimization objective and target, user should be able to input necessary data so that the definition files can be generated.
- User should be able to submit optimization data to the WebBlow system; start an optimization request; interrupt an optimization process.
- When multiple computing resources are connected to the WebBlow system, user should be able define a resource selection criteria from computation cost and computation time; define a earliest start time and a latest stop time.
- User interfaces should be able display the status (iteration and step) of the design and optimization process; display "die opening" in a 2D graphic interfaces; display "part weight" in a 2D graphic interfaces; display the log file; allow users to download result files and view the results locally.
- A console interface is needed for the system administrator. This interface could be Web based or standalone application. System administrator should be able to: create new user(s); create new user group(s); set/update privileges for users.
- Project Management: Application should be able generate new design project number; store multiple existing design project data; allow existing design project data being updated mutual exclusively; retrieve same existing design project data.
- Project Version Management: Application should be able generate version number for design project; store multiple existing versions' design project data; allow existing design project version data being updated mutual exclusively; retrieve same version existing design project data.
- User Management: Application should be able create new user's privilege profile; store multiple existing users' privilege profile; allow existing user's privilege profile being updated mutual exclusively; retrieve user profile.
- Multiple optimization computing resources should be able to connect to the WebBlow system dynamically. WebBlow should be able to find the optimal computing resource for a given task.
- The output of the WebBlow software is an optimized design of a blow molded part in general, and numeral result files in particular. These result files include: report.txt, prog_points.txt, *.log, *.res, *.pat, *.nod.
- There are three levels of privileges: A, B, and C. Level A users should be able add/modify/delete user/user group; set/modify privilege for entities of levels B and C. Level B users should be able create a new project; find an existing project; view an existing project; choose an optimization method; choose an optimization objective; choose an optimization target; define design parameters; submit optimization data; start an optimization request; interrupt an optimization process. Level C users should be able view optimization results on Web browser; download result files from WebBlow system.
- A System Administrator has only the level A privileges. A Design Project Manager has the level C privileges. A Blow Molded Parts Designer has the level B + level C privileges.

5 System Design

This section presents a brief description of the system design. Detailed system design is presented in (Wang, 2002b).

As mentioned above, the Web technology is becoming more and more popular to implement collaborative 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 (user company) 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 supplier and user companies.
- Software suppliers will be able to protect their software from illegal duplications and distributions.

Although Web technology plays an important role in promoting and supporting sharing of information and design, it is not sufficient enough to support real-time collaborative design. It is not flexible enough for legacy systems integration as well as computing resource management or load balancing. To deal with these issues, we propose an agent-oriented approach for Web-based collaborative design systems. The proposed approach has a number of advantages:

- It provides greater flexibility for legacy systems integration through socket-based communication among local resource agents behind a Web server.
- It can enable real-time collaboration through communication among active Web servers. These active Web servers 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, since socket based communication is not possible for agents behind different firewalls.
- It can significantly improve the performance of development and design process, particularly in the cases of multiple projects and multiple users working at the same time.

5.1 System Architecture and Components

WebBlow system is composed of agents, Applets, Servlets and XML databases as shown in Figure 3. Each of them has own responsibilities and they work together collaboratively. There are eleven major entities in this system:

- *Interface Agent (Applet):* It is responsible to collect information from user, display results to user, and validate the data at entry level based on business logic, etc. Intelligent Interface Agent technology is going to be applied in this portion. An interface agent models a user's behavior by establishing user profile that is relevant to the user's interests. It usually includes the ability to: acquire and retain an interest profile of its user and act upon one or more goals based upon that profile, act autonomously pursuing the goals posed to it by its user, irrespective of whether the user is connected to the system where the agent is based, apprise its user of progress towards outstanding goals and present preliminary results, etc. Note that user interface agents are investigated in a separate project in our research group and the results are to be reported separately.
- *BlowDesign Server Agent*: It is responsible to receive request from user interface; communicate with EDM Agent; feedback to user interface; create a Job Agent when the incoming request is an optimization request and all data are valid.
- *Job Agent*: It is created by BlowDesign Server Agent. It communicates with EDM Agent for storing and retrieving data; with DF/YP Agent for finding competent service providers (BL Agents); and with BL Agents for negotiation based task allocation and job progress monitoring; and finally, it dissolves when the optimization job is accomplished and the results are saved with EDM Agent.
- *EDM Agent (also called Collaboration Oriented Data Agent)*: It is a reactive database agent, functioning like an interface between a database and its user. Other agents may query user or project related information from XML database through EDM Agent, or store information into XML database through it. It provides a set of XML database query and manipulation standard methods.
- *Ontology Agent*: An ontology agent is needed to decipher the content of exchanged messages. The possible enhancement direction is: (1) the transfer of knowledge between agents; (2) learning; and (3) interfacing agents and humans. (Not to be implemented in the current version.)

- *DF/YP Agent*: It is responsible to register BL Agent when a BL Agent is active; provide Job Agents with all active BL Agents' information; keep up-to-date BL Agent information on hand, etc.
- *Image Generator*: It is responsible to generate 2D or 3D images from simulation results. It will be implemented when the automatic image generation software module becomes available.
- *Console Agent*: It is used to configure and maintain system parameters.
- *BL Agent (BlowLoop Agent)*: It is a major problem solving agent in WebBlow. Most other pre-existing software modules are to be invoked by this agent. It executes optimization simulation based on the parameters provided by Job Agent; sends optimization results to Job Agent. It is developed by wrapping a legacy system developed in C and FORTRAN.
- *Databases*: Three XML databases are to be implemented: DF/YP database, Ontology database and EDM database. Query and manipulation operation is done by standard methods defined in three agents (DF/YP Agent, Ontology Agent and EDM Agent).
- *Servlet*: It is the gateway for receiving messages from Interface Agent (Applet) over HTTP protocol and forwarding messages to BlowDesign Server Agent by socket communication, and vice versa.

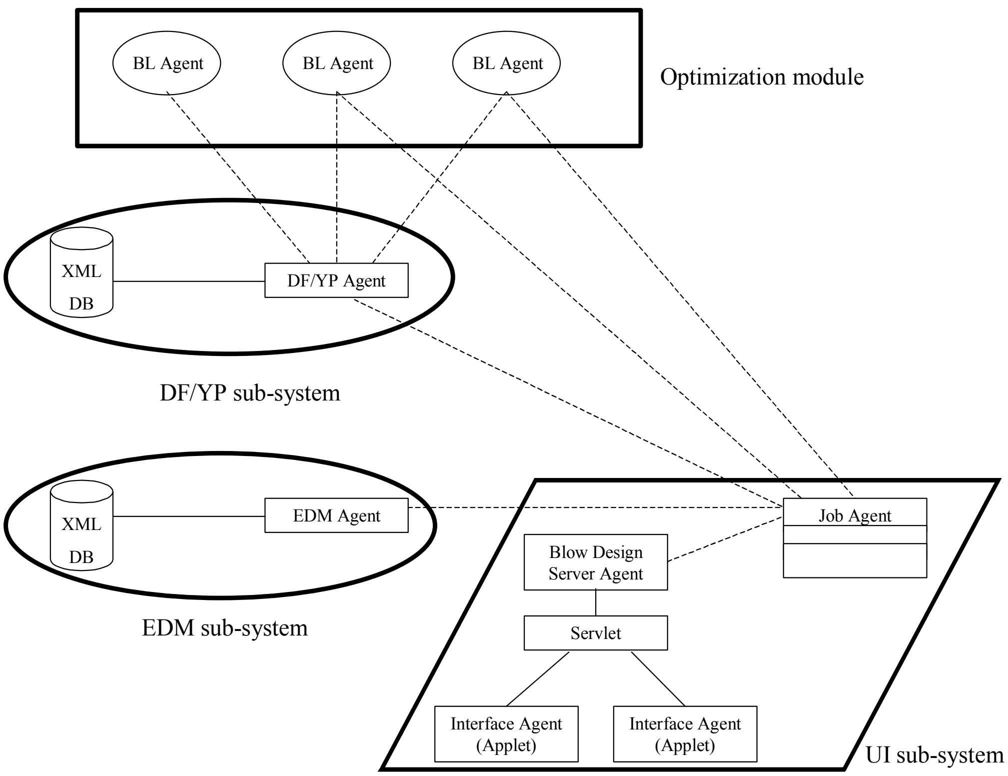

Figure 3. System Architecture

The agents are separated into two groups from physical location perspective. Some agents are located only within the optimization service provider organization site, or say within the firewall, and others may be anywhere in the world as long as the Internet access is available. The physical communication is designed depending on these two kinds of agents locations. The communication between agents over a firewall is through Applet/Servlet communication using HTTP protocol. On the other hand, the communication between agents within the firewall is through socket communication.

The relationships between agents are also different depending on their state during the software running life cycle. Some agents are initialized when system is started and destroyed when system is stopped. These agents are static to the system. Some other agents' existence depends upon the optimization job, and they are dynamic to the system. The relationship between two static agents is a static relationship, and if one party of the relationship is a dynamic agent, the relationship would be called a dynamic relationship. For example, DF/YP Agent is initialized when system is started, no matter there is a BL Agent or not, it exists independently. And DF/YP database exists too in the form of a XML format text file. The relationship between these two entities is static. In other words, this relationship is there all the time during the system's life cycle. However, BL Agent is a dynamic entity, its existence is not guaranteed and its life cycle is unpredictable. When a BL Agent is started, there is a relationship between BL Agent and DF/YP Agent. But this relationship usually will last only until the BL Agent is terminated. In Figure 3, the solid line shows the static relationship between entities, and dashed line means the relationship is dynamic while the system is working.

5.2 Web Based User Interfaces

The Graphic User Interface of WebBlow system is made up of 6 interfaces, and 15 classes (Figure 4). Figure 5 shows two examples of the Web based user interfaces.

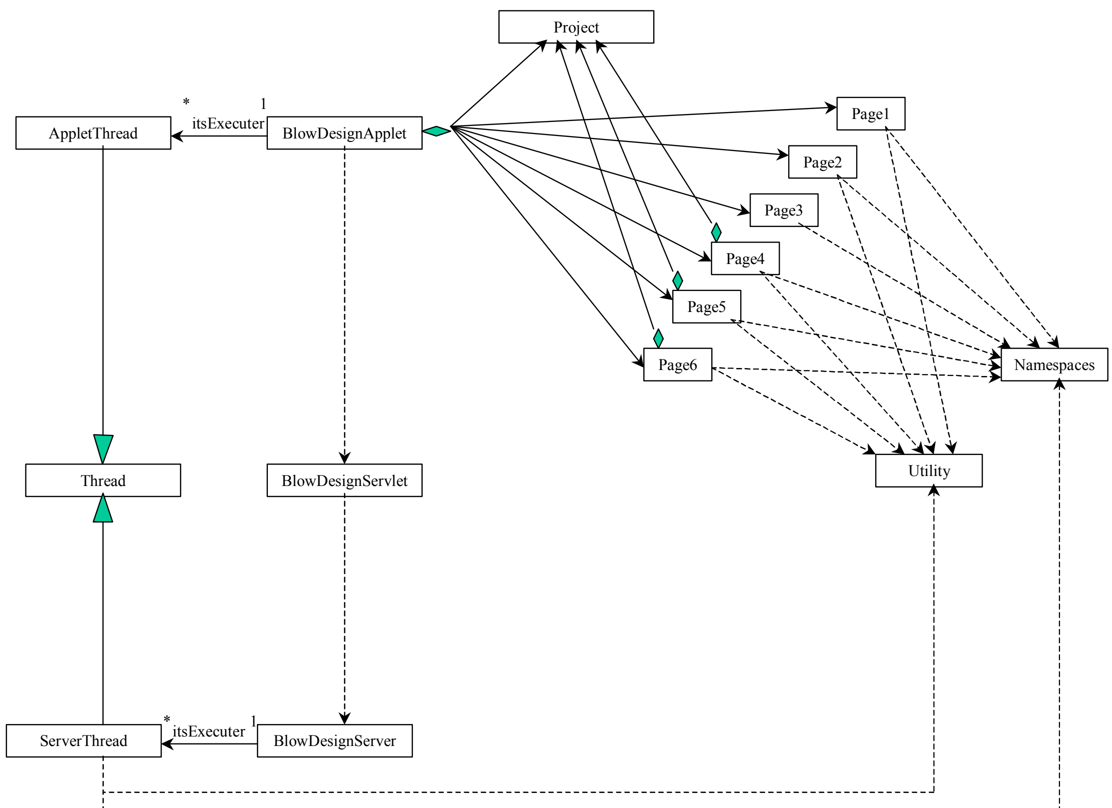

Figure 4. GUI Class Diagram

5.3 Computing Load Balancing

Agent based negotiation is used to implement the computing resource management or load balancing behind the Web server. Same techniques used in agent-based distributed manufacturing scheduling (Shen and Norrie, 2001) are to be adopted in system implementation. Different negotiation methods are to be investigated in addition to the basic Contract Net Protocol (Smith, 1980). More details are to be reported separately.

5.4 Distributed Data Management

For addressing the distributed data management problem concerning the management of multiple projects, multiple users as well as multiple versions of an individual design project, we propose a collaboration oriented data agent (CODA) – also called EDM (Engineering Data Management) agent. CODA is designated with three main features – XML-based structured data infrastructure, central service and distributed data management, and event-condition-action rules-based reactive transaction mode, all of them are bringing out the best in each other. This part of the work has been presented in a separate paper (Li and Shen, 2002).

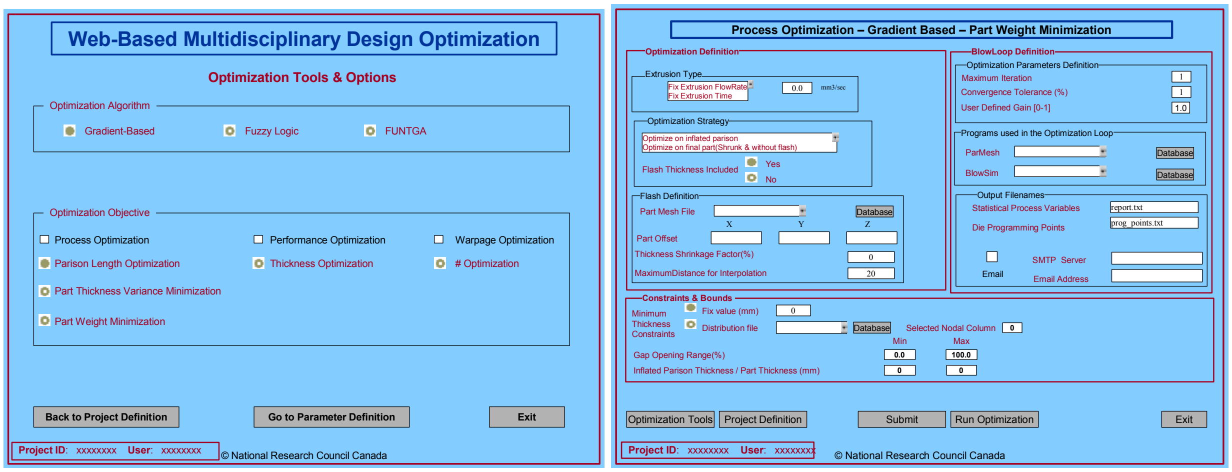

Figure 5. Examples of Web Based Graphic User Interfaces

6 System Implementation

The current software prototype is implemented in a local network of PCs with the Windows NT/2000 platform. The primary programming language is Java, but FORTRAN and C++ are also used for legacy software integration. Apache^{TM} and Tomcat^{TM} are adopted for server site implementation.

All Web based user interfaces are implemented using Applets. Applet-Servlet communication techniques are used for communications between user interfaces (Interface agents) and the server, and socket communication is used for message exchanges among WebBlow agents including Job agents, BlowLoop agents, the DF/YP agent, the EDM agent, and the BlowDesign Server agent. Some agents (including the Ontology agent, Image Generator) have not been implemented yet. All databases are implemented using XML. XML is also used for messaging among all WebBlow agents. Computing load balancing (or computing resource management in general) is realized through negotiation among job agents and BlowLoop agents using a modified Contract Net Protocol.

7 Conclusions and Perspectives

This paper reports the second prototype implementation of our on-going project on the development of a distributed MDO environment. This prototype has been redone from system requirements definition to system design and implementation. The major work includes Web based user interfaces design and implementation, agent-based computing resource management or load balancing, and XML based data management (including the management of multiple projects, multiple users as well as multiple versions of an individual project). A working prototype is still under final implementation and validation. The future work includes finalizing the prototype implementation and validation through an industrial user. The final version is to be used by industrial companies next year.

The system integration methodology and general system architecture are proposed not only for this particular application, but also for many other similar applications, and many software modules (e.g., XML based data management, computing resource management, etc.) can be reused in even more other applications.

References

[1] Y.L. Hsu, T.C. Liu, M.S. Hsu, and F. Thibault, A fuzzy optimization algorithm for blow moulding process, in Proceedings of the First International Workshop on Multidisciplinary Design Optimisation, London, ON, Canada, pp. 9-14, (2001).

[2] Y. Li and W. Shen, "CODA: A Collaboration Oriented Data Agent", *Proceedings of the Seventh International Conference on Computer Supported Cooperative Work in Design*, Rio de Janeiro, Brazil, pp. 252-257, (2002).

[3] W. Shen, "Web-based infrastructure for collaborative product design: an overview", *Proceedings of the Sixth International Conference on Computer Supported Cooperative Work in Design*, Hong Kong, pp. 239-244, (2000).

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

[5] W. Shen and D.H. Norrie, "Dynamic Manufacturing Scheduling Using both Functional and Resource Related Agents", *Integrated Computer-Aided Engineering*, 8(1), 17-30, (2001).

[6] R.G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", *IEEE Transactions on Computers*, C-29(12), 1104-1113, (1980).

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

[8] L. Wang, W. Shen, H. Xie, J. Neelamkavil and A. Pardasani, "Collaborative Conceptual Design: A State of the Art Survey", *Proceedings of the Sixth International Conference on Computer Supported Cooperative Work in Design*, Hong Kong, pp. 204-209, (2000).

[9] Y. Wang, WebBlow: A Web/Agent Based MDO Environment for Blow Moulded Parts Design, Requirement Document V1.0, NRC-IMTI, (2002a).

[10] Y. Wang, WebBlow: A Web/Agent Based MDO Environment for Blow Moulded Parts Design, Design Document V1.1, NRC-IMTI, (2002b).

[11] J.C. Yu, X.X. Chen and T.R. Hung, "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, (2001).

Author Biography

Dr. Weiming Shen is a Senior Research Officer at NRC-IMTI. He obtained his Bachelor (1983) and Master (1986) degrees from Northern Jiaotong University, China and his PhD degree (1996) from the University of Technology of Compiègne, France. He has been working on intelligent agents and their applications to concurrent engineering design, intelligent manufacturing for about 10 years. He has published one book and about 100 papers in scientific journals and international conferences/workshops, and co-edited 7 conference/workshop proceedings in the related areas. He is an editorial board member of the International Journal of Networking and Virtual Organizations and served as guest editor for other three international journals. He is a Senior Member of IEEE and a member of ACM and AAAI. He is Adjunct Professor at both University of Western Ontario and Lakehead University.

Ying Wang is a guest worker at NRC-IMTI. She is finishing her Master degree in Computer Science at Lakehead University and becoming a PhD candidate in Dept. of Electrical and Computing Engineering of the University of Western Ontario. She obtained her Bachelor degree in Computer Science from Beijing University of Chemical Engineering.

Yan Ma is a guest worker at NRC-IMTI. She is finishing her Master degree in Computer Science at Lakehead University. She obtained her Bachelor degree in Computer Science from Shanghai Jiaotong University.

Dr. Yinsheng Li is an NSERC Visiting Fellow at NRC-IMTI. He obtained his Bachelor degree in Machinery Design and Manufacturing from Chongqing University (1988); Master degree in Industrial Robotics and Bionics from Southeast University (1994); and PhD degree in Computer-Aided Design from Tsinghua University (1999).

Dr. Hamada Ghenniwa is an Assistant Professor at the University of Western Ontario. He has expertise in agent-based cooperative distributed systems. He is also working on advancing the state of the practice of software engineering to improve the quality of cooperative distributed systems that depend on software. He has authored and co-authored several papers in refereed journals and conference proceedings as well as technical and industrial project reports. Dr. Ghenniwa is also focusing on solving problems related to task allocation and scheduling in multi-machine systems with partially overlapping capabilities. The result of his research has been applied to solve various problems in cellular and flexible manufacturing. Dr. Ghenniwa is currently leading research and industrial projects concerned with integration in distributed information systems, business-to-business e-commerce, pervasive computing, real-time systems, and multi-agent systems for manufacturing control.

Dr. Francis Thibault received his Ph.D. degree in 1999 from Chemical Engineering department at Ecole Polytechnique of Montreal, Canada. His work focused on the numerical and experimental mixing characterization of a new coaxial mixing system developed for coating color makedown in pulp and paper industry. Finally, in 2000, he has completed a Post. Doc. with the Elf Atochem Society in France working on the flow modeling in twin-screw extruders. Mr. Thibault is now working at the Industrial Materials Institute of the National Research Council of Canada located in Boucherville (Montreal). He is responsible of the development of optimization capabilities in extrusion blow moulding, injection-stretch blow moulding and thermoforming.

Dr. Yeh-Liang Hsu received his PhD from Stanford University in 1992. He is professor and chairman of ME Department, Yuan Ze University, Taiwan. He specializes in design optimisation.

Dr. Jyh-Cheng Yu is an Associate Professor at the Dept. of Mechanical Engineering, National Taiwan University of Science and Technology.