

Integration, Management and Communication of Heterogeneous Design Resources with WWW Technologies

Shuyan Ji, Daizhong Su and Jiansheng Li

Advanced Design and Manufacturing Engineering Centre, School of Built Environment,
Nottingham Trent University, Burton Street, Nottingham, NG1 4BU, UK
E-mail: {shuyan.ji2, daizhong.su, jiansheng.li}@ntu.ac.uk
URL: <http://www.admec.ntu.ac.uk>

Abstract. Recently, advanced information technologies have opened new possibilities for collaborative designs. In this paper, a Web-based collaborative design environment is proposed, where heterogeneous design applications can be integrated with a common interface, managed dynamically for publishing and searching, and communicated with each other for integrated multi-objective design. The CORBA (Common Object Request Broker Architecture) is employed as an implementation tool to enable integration and communication of design application programs; and the XML (eXtensible Markup Language) is used as a common data descriptive language for data exchange between heterogeneous applications and for resource description and recording. This paper also introduces the implementation of the system and the encapsulating issues of existing legacy applications. At last, an example of gear design based on the system is illustrated to identify the methods and procedure developed by this research.

1 Introduction

As networks proliferate all over the world, it is inevitable to implement some enterprise activities such as product design within the virtual spaces supported by computer networks. Relying on such a virtual environment, it will be possible to effectively utilize dispersed resources to quickly respond to clients' requests and to greatly reduce time-to-market. In engineering area, there have been numbers of existing applications for engineering calculation, computer aided design, computer aided manufacture and computer aided evaluation, and numbers of new ones are still continue to emerge. A challenging issue in this field is to enable these diverse heterogeneous legacy applications to be integrated at lower cost and to work together smoothly. Its solution lies in the combination of distributed object technology with common information exchange languages.

The Common Object Request Broker Architecture (CORBA), defined by the Object Management Group (OMG), is a standard for the distributed computing and systems integration [1]. It seeks to provide a platform-independent and language-independent framework to enable object components to be operated from anywhere in

a network without concern for the operating system or the programming language. This allows heterogeneous legacy applications to be integrated together without re-writing the essential codes and hence the development cost can be reduced.

CORBA has gained a great number of distributed applications in engineering. For example, Li implemented a collaborative design system based on network using CORBA and Java [2]. Pahng et al developed a Web-based collaborative Design Modeling Environment [3]. In the research of Kim et al, the system integrates multiple clients, application servers, and databases together over three-tier structure. The communication between clients and application servers is done via CORBA [4]. Li et al discusses the idea that the collaborative product development mode on Internet is a process of collaborative decision making in stages. CORBA is used to establish an environment for collaborative problem solving [5]. Yoo developed Web-based knowledge management for sharing data in virtual enterprises, where CORBA interface helps Java agents communicated with the knowledge base system [6]. Hauch et al explored communication between integrated software tools using CORBA. The proposed system allows the encapsulated components in different processes on different machines to directly communicate in a high-level manner [7]. Lee's collaborative optimization approach for multidisciplinary design optimizations allows diverse optimizing system belonging to different disciplinary co-optimize a single problem [8]. Sang focused on the CORBA wrapping of legacy scientific applications, especially the procedures for wrapping the Fortran codes using CORBA and C++ [9].

The most of the above systems are still under proof-of-the-concept prototype development stages. Recently, some commercial distributed collaboration applications emerge at market. Windchill enables the ProEngineer users to collaborate over the Internet [10]. Autodesk has incorporated Netmeeting inside Autodesk Inventor R2 in order to provide a degree of collaboration [11]. These systems are limited among the same CAD/CAM systems.

In the fast-moving IT world, new object and exchange paradigms emerge rapidly. A wide acceptance of approaches based on the extensible Markup Language (XML) appears into the world. The W3C defines a set of XML-based protocols and standards that are the foundation for the current notion of Web services, in which recent development contends with CORBA [12]. Ouyang et al presented a design web service based distributed collaborative CAD system, employing geographical features as collaborative elements [13].

There have been many arguments about the advantages and disadvantages between the CORBA and the new emerging technology XML-based Web services [14-15]. The mostly accepted idea is that the Web services and CORBA are not exclusive and neither of them will replace the other, but rather should be seen as complementary technologies that need to be used together. CORBA is an enabling technology for creating sophisticated, distributed object systems on heterogeneous platforms. XML is a technology for conveying structured data in a portable way. CORBA allows users to connect disparate systems and form object architectures. XML will allow users to transmit structured information within, between and out of those systems, and to represent information in a universal way in and across architectures. Both technologies are platform-, vendor- and language-independent. CORBA tie together cooperating computer applications invoking methods and exchanging transient data transient data that will

probably never be directly read by anyone, while XML is intended for the storage and manipulation of text making up humane-readable documents like Web pages. In addition, portable data storage and exchange in XML will relieve CORBA-based systems from low efficiency much data causes in.

The author group have been contributing Web-enabled design environment for design and manufacture and implemented the Web-based design environment that facilitated invocation of large size computing program, multi-users management, and product data exchange mechanism [16-18]. Our current work focuses on a new form of hybrid tool architecture to allow effective interoperation of the integrated heterogeneous applications and flexible data exchange services for virtual enterprises and their collaborative design environment, through combining XML technology into CORBA-based infrastructure. In this paper, following the overview of the system architecture, encapsulating the existing applications into the components that can be operated in the system, developing the dynamical GUI (graphical user interface) for them, and exchanging the data format between them are presented. An example of gear design is finally presented to illustrate the architecture and methods developed.

2 Hybrid Architecture of the System

In the framework of the system, existing resources, no matter where they reside in a network, are integrated together and allowed to communicate with each other, relying on CORBA ORB (the Object Request Broker). Figure 1 illustrates the basic components that make up the architecture. In the CORBA-based distributed system, existing resources are encapsulated into objects, e.g. components that can be found and invoked between each other. IIOP (the Internet Inter-ORB Protocol), an underlying protocol, is the standard protocol that specifies how objects communicate across TCP/IP, the standard connection-oriented transport protocol for the Internet.

In the client/server distributed architecture, a client is a process that wishes to perform an operation on a distributed object, and a server, an object provider, is a process that provides this object to the client. The system allows a client to find, further connect and invoke the selected object that a server hosts. With the standard interface new coming objects and clients could work along with the already existing components. As the system augments there may be the need of one or more managers for the collaboration. A collaborating server provides a set of server-centralized services such as user authentication, object selection, and data-related operations.

Since lots of data files used in applications may be incompatible to each other, we have employed XML (the eXtensible Markup Language) on top of CORBA as a neutral data descriptive language. XML is independent from application and its platform and thus it is a proper option for such a heterogeneous environment. Moreover, the CORBA facilities such as Name service and Event service are utilized so components can be located and found easily.

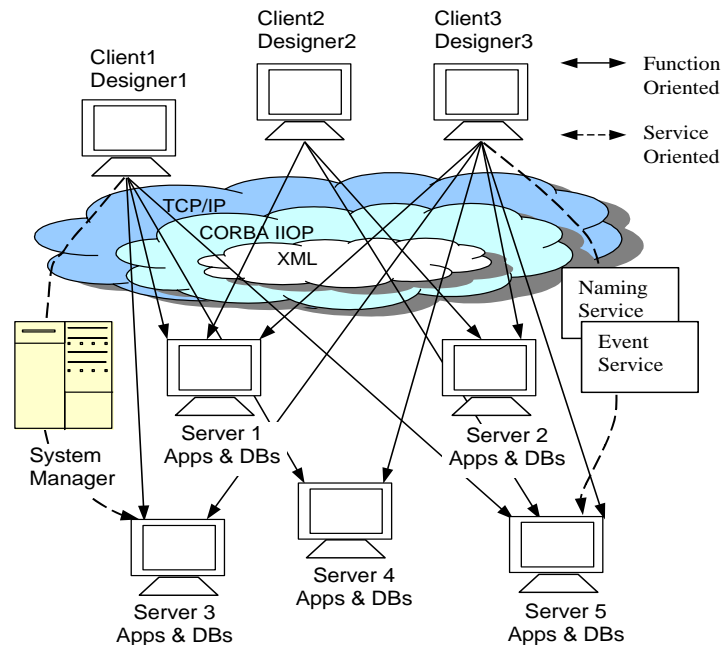


Fig. 1. Overview of the proposed system

2.1 Components and Their Communication

The relationship between all the components is illustrated in Figure 1. This hybrid system incorporates the best features of peer-to-peer with the reliability and security of server-based system. The peer-to-peer structure between clients and object servers provides functional facilities and the server-centralized structure through the system server helps to offer administrative services.

The system allows the direct interaction for function between clients (object user) and servers (object Providers) in de-centralized model. This peer-to-peer architecture not only provides a performance mechanism but also avoids a central bottleneck.

Using service-oriented mechanism the system server helps to publish the object, and to facilitate the global data warehouse and its updating and retrieving services. A set of CORBA services is standard CORBA objects in the system to facilitate the services. The Name service provides a means for objects to be referenced by name within a given naming structure. The Event service provides a mechanism through which CORBA objects can send and receive event messages. The connections linking to the system manager server and to the CORBA service server are initiated by the client and are kept alive during the client request session.

2.2 Data Exchanging Within the Heterogeneous Environment

The communication between applications is implemented using the CORBA, described in the above section. The data exchange between them will be implemented with the XML technology, shown in Figure 2. XML is of cross-languages and cross-platforms. It can be read directly or parsed by the Xerces-J library and the Xerces-C++ library. The XML file is transferred using the File Transfer Protocol (FTP).

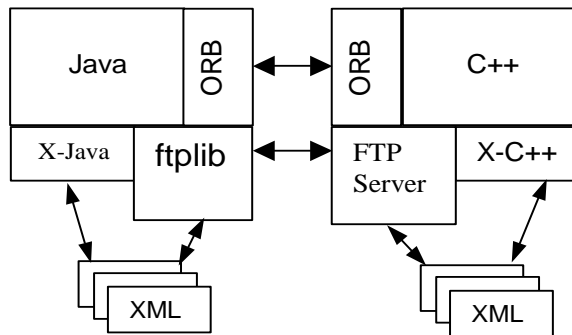


Fig. 2. Data exchange on the heterogeneous environment

3. Encapsulation of Heterogeneous Legacy Application

In the development of a distributed application, it is an essential step to bring existing legacy software to distributed object system. According to Sneed [19], there may be three strategies for doing so, e.g. redevelopment, reengineering and encapsulating. The redevelopment strategy is to start from scratch and redevelop all of the applications with the distributed object concepts. This approach is the most expensive and time consuming. The reengineering strategy is to convert the existing programs to object-oriented programs and distribute objects appropriately. This approach is to port the code from the old environment to new one. Code conversion is not easy and few tools and methods are available. The encapsulating strategy is to encapsulate the existing applications and to invoke them from the object-oriented distributed environment. The advantage of encapsulating is that legacy systems become part of the new generation of applications without discarding the value of the legacy applications.

In this development the encapsulating strategy is applied to deal with the existing programs. Encapsulation base on CORBA separates interface from implementation, as shown in Figure 3. It hides differences in programming language, location, and operating system with the interface defined in IDL (Interface Definition Language). Server-side application developers only need to understand the services of legacy systems and to describe them in the standard interface. Each component can be encapsulated separately and then integrated using object-based communications.

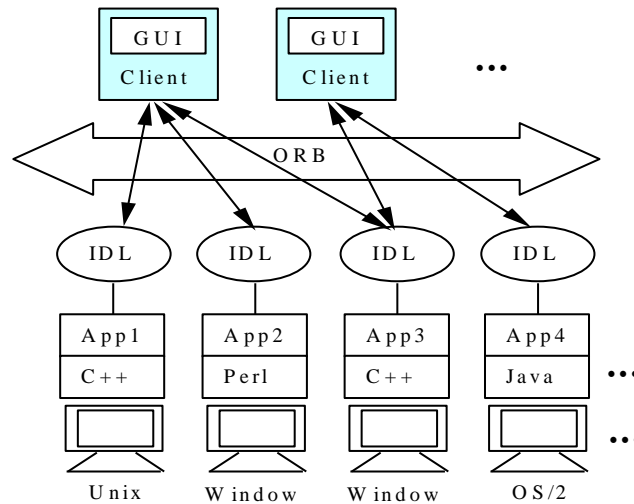


Fig. 3. Encapsulating structure

4. System Management

Conducting collaborative design with the distributed system usually utilizes more than one program to implement a complicated design task. Therefore the system needs to provide an administration-oriented architecture for the collaboration. In this research, a set of XML-based services is described.

When an interested client contacts an object server for the first time with the intention to include its functionality into its graphical user interface, or to develop a more complex application utilizing the available design resources, the relative information of objects should be captured easily. It includes general information about the component and its functionality. In order to invoke an object, the client also needs to know about everything that is necessary to run the object, such as the reference name of the object, the number and type of input parameters and return parameters for every method on the object, and the input and output files if there are ones. Figure 5 gives an example of component data information encoded in XML format. XML file can be read from XML-enabled browser on any platform and any operating system.

A client-specific GUI needs to be developed for a user to invoke one or more objects, as described in section 3. The system provides a mechanism to create dynamically a client GUI, accordingly to the selected object, for common user to use in the client side. Figure 5 gives the resource list and the GUI for the chosen object. The content of the graphical user interface can be set up at run-time. The GUI according to the description in the XML shown in Figure 4 could be automatically created, as shown in Figure 5.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<component_management>
  <component>
    <name>Gear Design Optimisation</name>
    <command>
      <name>optim</name>
    </command>
    <in_para>Maximum Centre Distance</in_para>
    <in_para>Minimum Centre Distance</in_para>
    <in_para>Maximum Module</in_para>
    <in_para>Minimum Module</in_para>
    <in_para>Maximum Thread</in_para>
    <in_para>Minimum Thread</in_para>
    <in_para>Maximum Helix Angle</in_para>
    <in_para>Minimum Helix Angle</in_para >
    <out_para>Result Centre Distance</out_para>
    <out_para>Result Module</out_para>
    <out_para>Result Thread</out_para>
    <out_para>Result Helix Angle</out_para>
    <out_para>Result Stress</out_para>
    <out_para>Result Mean Stress</out_para>
  </component>
  <component>
    ? ? ? ? ?
  </component>
</component_management>

```

Fig. 4. An example of object data structure

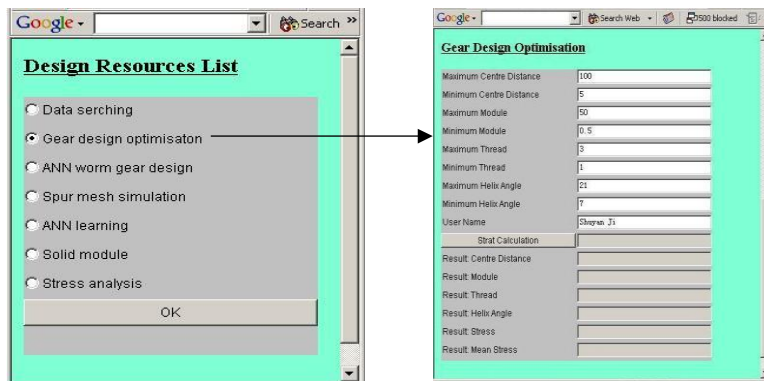


Fig. 5. Design resource list and the GUI for the chosen object

5. An instance over the Integrated Design Environment

With the system, we have integrated gear design related applications adhering to the CORBA standard and conducted an example of gear design on the distributed design resources. Figure 6 shows the legacy applications integrated into the system and their working procedure.

The main designer, the user of the client 1 in the system, wishes to conduct gear design including design optimization calculation and geographical model design over the distributed system. The client application includes the GUI (on the upper-left in Figure 6) for the design optimization, the design optimization procedure, data files, and the client favorite CAD commercial software (on the upper-right) and its service pro-

gram. After registration, the client user runs the GUI program. Firstly the designer could input parameters and set up the design objectives from available distributed design calculation resources list, including the remote calculation resources for gear contact stress, gear contact stress, the slide/roll ratio and gearing interference check. The remote programs are written in C++, Java, or Perl and are ported on Windows, Linux or OS2 while the GUI is written in Java application. The CORBA architecture enables the communication between the GUI and the heterogeneous design objective calculation programs. Data exchange between the remote resources and the client is based on the XML mechanism, as described in Section 2.2. The global data warehouse on the server is for XML data storage.

Unlike the design optimization application, App1, App2, and App3 are all of interactive-type programs for geometrical design and need human-involved actions during the execution. They are not easy to be CORBA-enabled. The individual service programs, i.e. App S1, App S2 and App S3, provide the communication between commercial CAD software.

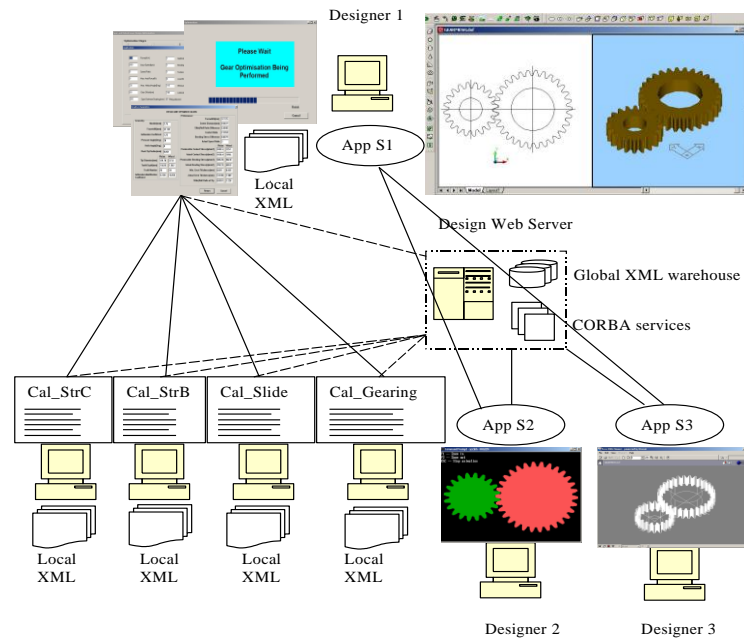
The GUI is for implementing all interactive design activities including inputting parameter, monitoring the calculation progress and viewing resultant data, and the Genetic Algorithm (GA) procedure is to fulfill the design optimization and to invoke multiple design objective programs. The resultant data is retrieved, recorded in XML format and then transferred into the system server for storage. Application 1 and application 3 are two kinds of CAD design tools for geometrical design, respectively located in the different operating systems. Application 2 is a gearing simulation tool for the cutting interfere analysis. The service programs App S1, App S2 and App S3 are for retrieving XML file from the system server and transforming the XML format into the respective CAD software specific format to the application. The main designer will implement the final modification in its own CAD application 1 and exports the final graphical file in a DXF file, a kind of mediate CAD data file.

6. Conclusion

There have been numbers of existing product design applications in engineering, which lead in the challenges of development of effective integrative environments. This research utilizes the CORBA and XML technologies to address the challenges. The CORBA technology allows the integration of existing heterogeneous diverse applications, regardless the operating system they are running on or the programming language they are written in, and enables the communication between the heterogeneous applications. XML is a good choice to store and transport the common data between the applications in a flexible and platform neutral way. XML is also suggested to record object information for the system to provide a dynamical standardized interface between the client application and the server.

The concept considering CORBA and XML proposed by this research has the features of efficient communication between applications and flexible data exchange services. It allows multiple design applications to contribute to the overall system; thus, it can react to customer's complicated design requirements much faster. Ever-

increasing applications can be integrated with existing ones without having to change most of the application's code. XML-style data format provides flexible data exchange between applications in heterogeneous environment and in humane-readable format. The hybrid architecture combining peer-to-peer model and server-centralized structure enables the efficient functional operation, the resource administration and



services.

Fig. 6. Gear design applications integrated into the distributed system

7. Acknowledgement

This research is supported by the EU Asia IT&C programme (Grant No. ASI/B7-301/3152-099/71553), which has been carried out at The Nottingham Trent University, UK in cooperation with Harbin Institute of Technology and Chongqing University in China, and Foundation LBEIN in Spain.

References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. *Int. J. Digit. Libr.* 1 (1997) 108–121
2. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): *Theoretical Aspects of Computer Software. Lecture Notes in Computer Science*, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438

3. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
1. Orfali, R., Harkey, D.: Client/Server Programming with Java and CORBA, 2nd edn. John Wiley & Sons Inc., New York Chichester Weinheim Brisbane Singapore Toronto (1998)
2. Li, J.: Collaborative Design Method on Network", Journal of Tsinghua University Xuebao. 9 (2000) 93-96
3. Pahng, F., Bae, S., Wallace, D.: A Web-based Collaborative Design Modeling Environment. Proceeding of the 7th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Stanford, CA., June 17-19, 1998, pp. 161-167
4. Kim, H., Yoo, S.-B., Lee, H.-C.: Web-enabled Collaborative Design Environment. ETRL Journal. 3 (2000) 27-40
5. Li, J., Zhang, H., Wang, J., Xiong, G.: Collaborative Design Method on Network, Journal Of Tsinghua University. 9 (2000) 93-96
6. Yoo, S. B., Kim, Y.: Web based Knowledge Management for Sharing Product Data in Virtual Enterprises. International Journal of Production Economics. 1 (2002) 173-183
7. Hauch, R. M., Jacobs, S. W., Prey, S. W., Samsel, H. L.: A Distributed Software Environment for Aerospace Product Development", AIAA/ASME/ASCE/AHS/ASC Structures Struct Ural Dynamics & Materials Conference. St. Louis Mo USA, April 12-15, 1999, pp. 1385-1394
8. Lee, K.-T., Roh, M.-L., Cho, S.: Multidisciplinary Design Optimisation of Mechanical Systems Using Collaborative Optimisation Approach. International Journal of Vehicle Design. 4 (2001) 353-368
9. Sang, J., Follen, G., Kim, C., Lopez I., Townsend, S.: CORBA Wrapping of Legacy Scientific Applications Using Remote Variable Scheme. International Conference of Parallel & Distributed System. Kyongju, June 26-29, 2001
10. <http://www.ptc.com/> accessed on 01/Feb/2005
11. <http://www.autodesk.com/> accessed on 01/Feb/2005
12. <http://www.w3.org/> accessed on 01/Feb/2005
13. Ouyang, Y., Tang, M., Lin, J., Dong, J.: Distributed Collaborative CAD System Based on Web Service. Journal of Zhejiang University SCIENCE SCI. 5 (2004) 579-586
14. Gisolfi, D.: Web services architect, Part 3: Is Web services the reincarnation of CORBA?. <http://www-106.ibm.com/developerworks/webservices/library/ws-arc3/>
15. Elenko, M. Reinertsen, M.: XML&CORBA. <http://www.omg.org/library/adt.htm> accessed on 31/07/2005
16. Su, D., Ji, S., Amin N., Hull, J. B.: Multi-user Internet Environment for Gear Design Optimisation. Integrated Manufacturing Systems. MCB UP Limited 6 (2003) 498-507
17. Su, D., Ji, S., Amin, N., Chen, X.: A Framework of Web Support for Collaborative Design. Proceedings of the 5th International Conference on Frontiers of Design and Manufacturing, Dalian, China, 2002, pp. 492-498
18. Li, J., Su, D., Henshall, J. L., Xiong, Y.: Development of a Web-enabled Environment for Collaborative Design and Manufacture. Proceedings of 8th International Conference on Computer Supported Cooperative Work in Design, Xiamen, China, May 26-28, 2004, pp. 540-545
19. Sneed, H.: Encapsulation of Legacy Software: A Technique for reusing legacy software components. Annals of Software Engineering, Vol. 9, 2000, pp. 293-313