Integrated Worm Design Based on Distributed Heterogeneous System

Professor Daizhong Su and Dr Shuyan Ji
Advanced Design and Manufacturing Engineering Centre
School of Architecture, Design and the Built Environment
Nottingham Trent University
Nottingham, UK

Abstract - *Worm gear design to professional standards normally involves the cooperation amongst the experts from multiple disciplines such as geometrical design, stress analysis, material selection, manufacturing, vibration simulation, etc. These experts might be in different organizations, located in different geographical regions and utilize different software tools and applications, which might be written in different languages, ported on different platforms and linked to different networks. In order to construct a collaborative environment for integrated worm gearing design, heterogeneity of these resources has to be taken into consideration. This paper presents a collaborative system for integrated worm gearing design based on CORBA distributed computing technology. All the software-based design applications and tools can be integrated together and communicated with each other, regardless of languages and platforms. This makes it easy to develop the complex distributed system using multiple programs and to construct an integrated collaborative design environment for implementing complex design tasks.*

Keywords: collaborative design, CORBA, worm gear, heterogeneous systems.

1 Introduction

The rapid development of computer aided techniques, such as Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM), enables engineers to effectively fulfill various design tasks such as concepts and detail design, and manufacturing simulation at individual stages of product development. However, the increasing complexity of modern products, the cruel competition pressure and the globalization of product development necessitate a collaborative design environment, where different computer programs and distributed experts need to be collaboratively involved to conduct a design task.

Worm gear design, like other complex engineering design, involves the cooperation amongst the experts from multiple disciplines such as geometrical design, stress analysis, material selection, manufacturing, vibration simulation, etc. The experts might be in different organisations and located in different geographical regions. In addition, different experts might use different resources including computing applications, CAD systems such as Pro-E, UG, AutoCAD, ANYSIS, Abacus, etc. These applications might be written in different languages, ported on different platforms and linked to different networks. In order to construct a collaborative environment to implement the worm gear design, heterogeneity of these resources has to be taken into consideration.

The Common Object Request Broker Architecture (CORBA), a distributed object computing architecture, provides a communication mechanism between applications, regardless of languages, platforms and protocols [1]. It allows heterogeneous legacy applications to be integrated together and invocated by each other, without rewriting the essential codes and hence the development cost can be reduced.

This paper presents a heterogeneous system for the integrated worm gear design based on CORBA distributed computing technology. It starts with literature review, followed by the presentation for the integration architecture and the communication structures of heterogeneous applications, the illustration of the integrated worm gear environment, and ends with the conclusion part.

2 Literature Review

CORBA has been successfully applied to build complex services in a number of areas, ranging from telecommunications, finance, e-commerce, healthcare, to the graphical user interface of Linux desktop (GNOME). However, most of the CORBA-based systems in engineering are still in prototype development stages. 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]. Yoo developed Web-based knowledge management system for sharing data in virtual enterprises, where CORBA interface helps Java agents to communicate with the knowledge base [4]. Hauch et al explored communication between integrated software tools using CORBA. The system allows encapsulated components in different processes on different machines to directly communicate in a high-level manner [5]. Lee's collaborative optimisation approach for multidisciplinary design optimisations allows diverse optimising system belonging to different disciplinary co-optimise a single problem [6]. Sang focused on the CORBA wrapping of legacy scientific applications, especially the procedures for wrapping the Fortran codes using CORBA and C++ [7].

Recently, in the fast-developing IT world, new 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 [8]. Ouyang et al presented a design Web service based distributed collaborative CAD system, employing geographical features as collaborative elements [9].

Web Services provides interpretabilities between various software applications running on disparate platforms, with the unique features such as open standards and protocols through many common firewall security measures, however, it is still in their infancy comparing with distributed computing such as CORBA and suffers from poor performance, and thus a trade-off needs to be taken into consideration between flexibility and performance. Since the integrated worm design needs high efficiency communication between different applications, CORBA is a better candidate to construct a collaborative infrastructure for the high performance requirements.

3 The Architecture of Application Integration

There are basically two types of architectures possible to realize the integration of design resources based on the CORBA, i.e. point-to-point structure and server-centralised structure. Centralised Web server takes on the task of a mediator and brings together application providers and interested clients. Communication between client applications and design applications must go through the centre server. It provides convenient administration and secure controls for all the interactions. However this structure suffers from poor performance due to the central bottleneck. The point-to-point structure provides a totally decentralized system, which is used in this system design to obtain the direct linkage between clients and applications, as shown in Figure 1. Clients could contact application servers without going through a central station. This point-to-point architecture would avoid a central bottleneck, however, each application component server would have to implement its own administrative system.

Collaborative design engineering not only augments the capabilities of the individual specialists in the structural combination, but as an extension of concurrent engineering, it also enhances the ability of collaborators to interact with each other and with computational resources, whether in a sequential or concurrent working flow. This point-to-point structure opens the possibility to communicate with each other between any two partners.

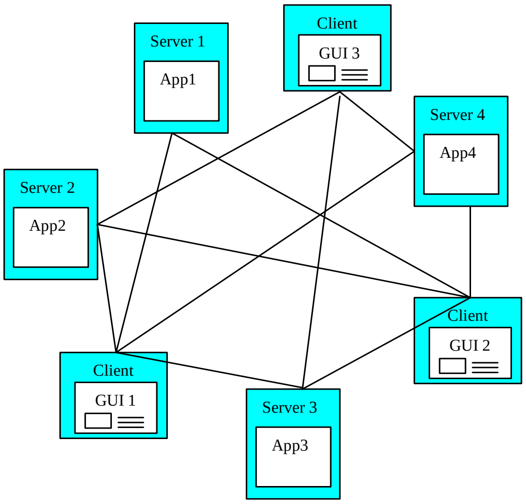

Figure 1 Point-to-Point Architecture

Based on CORBA, applications can be wrapped into distributed objects, and then can be found and invocated. CORBA IDL (Interface Definition Language) is utilised to define the interface for the objects. A designer (user) can use a client application to directly invocate the remote applications through CORBA ORB (Object Request Broker) and IIOP (Internet Inter-ORB Protocol). The linkage between client application and object implementation is transparent, regardless of languages, platforms and protocols.

4 The Communication between Heterogeneous Applications

Worm design involves many experts such as product design specification (PDS), concept design, detail design, manufacture simulation, cost analysis, etc. These experts may utilise different design tools, locate geographically in different places and belong to different organisations. They need to collaborate with each other to find rational solutions, as shown in Figure 2. In order to realize the collaboration between different experts, all the different computer applications have to be integrated within a suitable framework and thus communicate with each other.

4.1 Communication between Java Client Programs and C++ Service Applications

The integrated design starts with design demands identification and then proceeds to transform many aspects of consideration such as the user demands, technical requirement, cost consideration, quality control, etc into Product Design Specification (PDS). The PDS is stored in a database file such as SQL. In this research, when conducting design optimisation, input parameters are required from the PDS data resource. There is a service program, which is written in C++ and in charge of retrieving and updating data in the PDS database file.

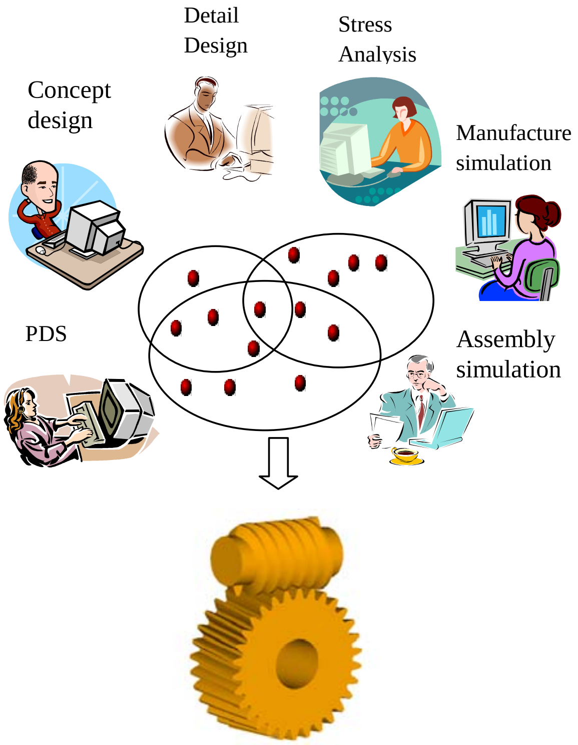

Figure 2 Many experts working collaboratively to conduct a common design task

CORBA enables the communication between heterogeneous applications. Taking the linkage between C++ program and Java application as an example, the communication architecture and its implementation, as shown in Figure 3, are presented in this section.

The communication between the C++ program and PDS database is implemented by ODBC (Open Database Connectivity) mechanism. During the parameter input procedure, product design parameter can be retrieved from the PDS database. On the optimisation site there is an optimisation service program, written in Java application, to access remote PDS service program. The communication between the Java application and the C++ program are implemented by CORBA, as shown in Figure 3. Based on CORBA, the C++ program is wrapped into an object. IDL is utilised to define the object interface. The IDL file can be compiled and mapped by respective compilers into corresponding ORB, responsible for the communication. On the optimisation side, the IDL file is mapped into Java ORB; while on the PDS side, the IDL file is mapped into C++ ORB. The developer on the C++ program provider side has to write a server program and the object implementation, i.e., the PDS database service program, while the developer on the client side writes a Java client program to invocate remote object program.

The server program accepts the requests from the client and further invokes the object program. IIOP, the underlying communication mechanism of CORBA, is used to specify how different ORBs communicate with each other.

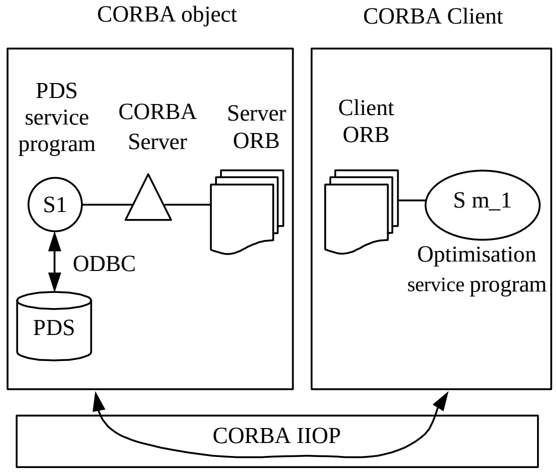

Figure 3 The CORBA based communication between PDS service program and optimisation service program

The C++ ORB, C++ object, and C++ server program are compiled and built by C++ developing tool while Java client are compiled and built by Java compiler. The setting and debugging of CORBA-oriented procedures are implemented by CORBA developing tool Visibroker [1].

4.2 Communication Mechanism for the Optimization Implementation

In order to implement the design optimisation, the main designer invokes a remote design optimisation program, which is implemented in C++ and executed on Linux. It is wrapped into a CORBA object to be invoked by a CORBA client. On the client side, the CORBA client is a Java applet. Based on CORBA, the communication between applications is transparent, regardless of languages and platforms. Figure 3 gives the interaction structure based on CORBA between a Java application and a C++ program. Using CORBA IIOP, normally a client program can access remote objects directly without any interference from servers. However, Java applet clients cannot directly access the remote object because of Web browser security restrictions placed on Java applets, i.e., the so-called Java sandbox security. When an applet runs in Web browser or applet viewer, it cannot do the following things.

- To open connection to other remote server (CORBA server), except for the Web server from which it was downloaded);
- To accept any incoming connections such as CORBA IIOP connection.

This is not a problem for Java application client other than applet client. Also it is not a problem when Java applet client and server run on the same computer. However this is a problem when Java applet and CORBA server run on different hosts over Internet.

On the other hand, in the presence of firewalls, setting up CORBA IIOP connection probably fails, because most firewalls block every communication, except for some well-known ports like those for HTTP or FTP, which the Web server is based on. Therefore there is a need of gateway between CORBA IIOP and WWW HTTP to enable this applet client to access remote CORBA objects. Visibroker offers such a gateway named Gatekeeper.

Visibroker GateKeeper [1] can be configured to enable Visibroker CORBA applets to communicate with object servers across networks while still conforming to the security restrictions imposed by Web browsers and firewalls. The Gatekeeper serves as a gateway from applet to server objects even if a firewall is restricting access. In case of firewalls Gatekeeper automatically switches from IIOP protocol to HTTP implementing so-called HTTP tunnelling mechanism, as shown in Figure 4.

As HTTP Tunneling, Gatekeeper accepts and decodes the HTTP requests. IIOP requests from clients are wrapped in HTTP so that they can get through firewalls. What is more, the Gatekeeper can be used as a Web server.

5 Integrated System for Worm Gear Design

All the essential heterogeneous tools and databases can be linked to the Internet within the CORBA point-to-point framework and thus communicates with each other, as shown in Figure 5. The system design aims to provide rich forms of collaboration adaptive to the need of practical design.

In this research, the point-to-point direct connection structure is utilized for multiple programs integration to ensure its higher efficiency . Within the structure, the worm design resources integrated include PDS databases, optimization computing program, modeling design software, stress analysis software, and common reviewing system.

Product data specifications are defined and saved as a database file, as shown in the upper-left part of Figure 5. The PDS database file contains the performance requirements, material information, and quality assessment information. On the PDS side, there is also a service program “S1” for the PDS database operation. The service program is written in C++ and could conduct the writing and reading operations of databases, through the ODBC (Open Database Connectivity). The PDS database files and C++ service program S1 are ported on Windows platform.

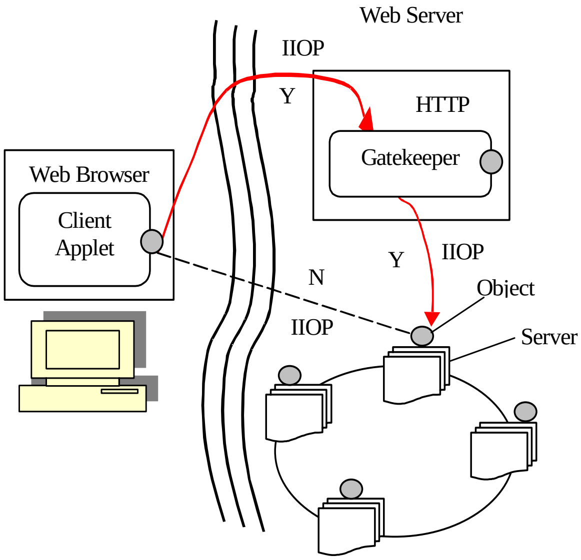

Figure 4 CORBA Applications Extends Access the Firewall

The optimisation designer conducts the design optimisation through the optimisation program, as shown in the upper-middle part in Figure 5. The design resources, including data and programs, are not resides with the optimisation designer and he has to invocate remote design resources to implement the design optimisation task through three operations: retrieving the required data, invocating the optimisation program, and generating 2D graphics. Design input parameters are from the PDS formulation engineer, through the communication bridge between the service program S m-1 and S1 on the PDS side. S m-1 written in Java is linked and communicated with S1 written in C++ with CORBA’s language-independent feature. A remote designer possesses the design optimisation program, written in C++ and ported on Linux, as shown in the upper-middle part of Figure 5. The 2D graphics generation program is written in Java Applet and ported on Windows, as shown in the upper-right part of Figure 5.

After completion of the design optimisation, the resultant data are passed to Pro-E modelling designer and ANSYS stress analysis specialist. The 3D model is shown in the lower-left part and the stress analysis in the lower-right part in Figure 3. Pro-E and ANSYS are commercial software, could not be directly invocated by remote programs but through the service programs S2, Sm-2, S3 and Sm-3, which are implemented in C++ and the software interface-developing tool.

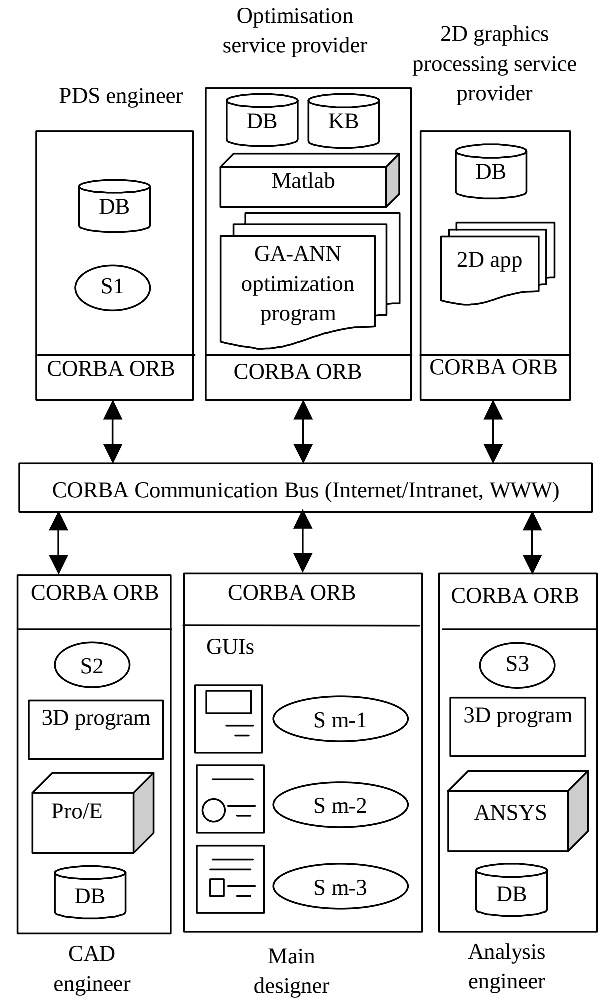

Figure 5 The integrated work design system framework

All communications between any two points, implemented with CORBA mechanism, is of language-independent and platform-independent. This effective and efficient communication between two points enables collaborative design between design team members.

5.1 Collaboration between Matlab and a Program for ANN-GA Based Design Optimization

An approach combining artificial neuro-network (ANN) and genetic algorithm (GA) has been developed to optimise the worm gear design parameters based on finite element analysis results.

The neuro-network is written in C++, while the GA is conducted using Matlab version 7.0. With the ANN-GA approach, the GA interacts with the ANN to get the objective function values with a set of design parameters during each iteration, and finally obtain the optimised design parameters. The communication between GA program and Matlab is shown in Figure 6.

The ANN-GA approach has the advantage of optimising large number of design parameters with multiple objective functions. The design parameters to be optimised include module, number of teeth, tooth modification coefficients, pressure angle, centre distance, etc.; the objective functions include minimising the transmission errors, reducing the contact and bending stresses, and less amount of deflection.

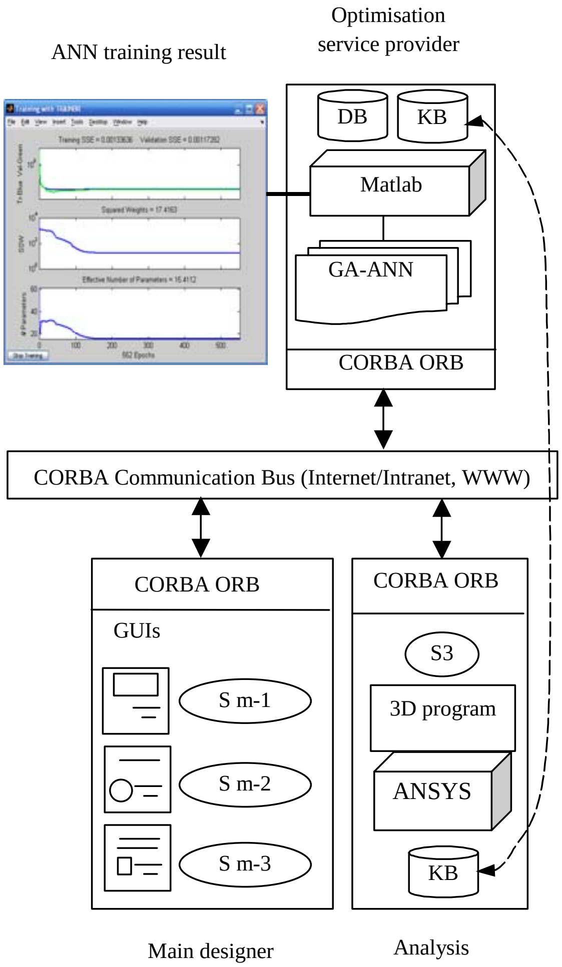

Figure 6 The integration framework of Matlab-GA-FEA

The optimisation is conducted using the GA programme. During the optimisation process, considerable iterations for calculation of objective functions, i.e., the contact stress, bending stress and deformations are necessary; for example, to get the final optimisation results, 10 generations of optimisations and hundreds sets of calculation results of the objective functions within one generation may be necessary. However, to get the calculation results using finite element analysis method is very time consuming, which may take one day to get one set of such results. To speed up the process, an ANN method is developed to get the results of the objective functions.

The ANN method is based on a large number of results obtained already from finite element calculations. In a case study conducted in this research, 117 such sets of data obtained before the construction of the ANN. Each set of data includes design parameters (module, number of teeth, etc. mentioned previously) and corresponding values of objective functions, i.e. bending and contact stresses and deformation obtained from the finite element analysis. A back propagation ANN has been developed which consists of three layers: the input layer consists of 6 neurons, each of which represents a design parameter; hidden layers consists of 10 neurons; and the output layer consists of 3 neurons, each of which represents contact stress, bending stress and tooth deflection. Amongst the 117 sets of data, 80 sets were used to train the neuro-network and the 37 sets were used to validate the network trained. After successful completion of the training, the neuro-network can be used to predict the objective function values, in other words, with a random set of design parameters provided to the input layer, a corresponding set of values of objective functions can be obtained at the output layer. Although the ANN is based on the FEA results, but there is no FEA calculation involved. Therefore the speed to get the objective function results via ANN is dramatically faster that FEA calculations.

5.2 Collaboration between the Main Designer Optimization Experts

The main designer can conduct design optimisation through invoking remote optimisation service and can view the results from the Browser by invoking remote 2D graphics processing application. The communication between the client program and the service programs is based on the mechanism demonstrated in Section 4.2.

The GUI client program is written in Java applet, which is designed for the main designer to interact with while the optimization program and 2D processing program is written in C++, as shown in Figure 7. The system supports multiple designers to conduct design in the mean time. The harsh table mechanism is designed to provide the dynamic management of the user identification. The session can be kept until to stop deliberately. This style provides an environment to quickly respond to the requirements.

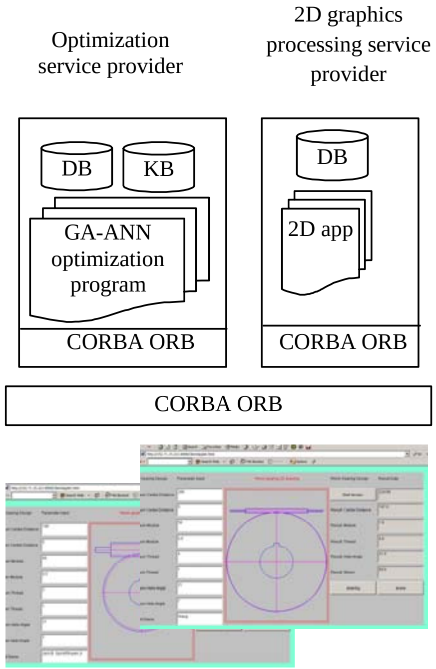

Figure 7 The collaboration between Java applet GUI, optimisation program and 2D graphics processing program

5.3 Collaboration between Software ProE, ANSYS through Service Program

The CAD software Pro/Engineer is used for the 3D solid model, as shown in Figure 8, using the resultant data from optimization. The FEA software ANSYS is used for the stress analysis of worm gearing pair. The collaboration

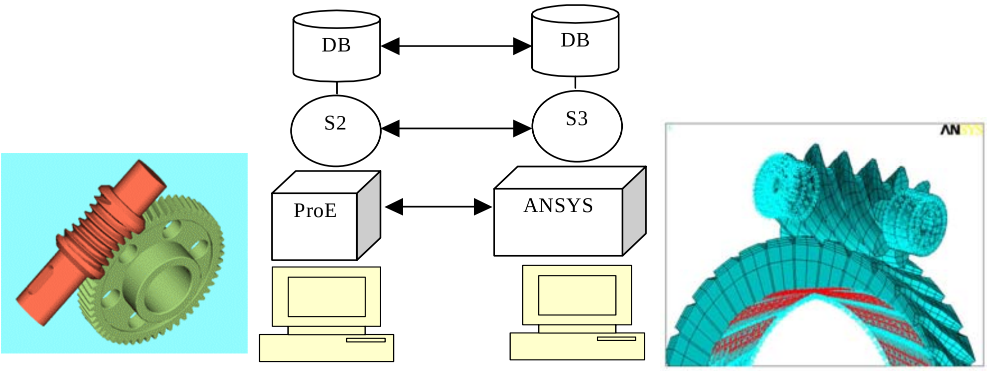

Figure 8 The collaboration between CAD modelling and FEA analysis

between the two commercial software systems is still the challenge in current research. There possibly are three levels of communication.

STEP is used for the data exchange and product data sharing asynchronically between the two systems. S2 and S3 are respectively service programs for the two systems and written in the software programming API. Two programs are responsible for processing the parameterized model. Based on the feature-based data there are two versions of 3D model constructed and processed by the service programs. The direct communication between the two systems is implemented based on agent-based technology and it is currently under ongoing in our research. The CAD software and the CAE software can be wrapped as software agent, along with the 3D model, rule-based knowledge mechanism, and agent-enabled routines. The software agents can interact with each other. The analysis 3D model is constructed and shared in the CAD and CAE systems. The underlying communication behind these collaborations is implemented based CORBA point-to-point mechanism.

6 Conclusions

An integrated worm gearing design system based on distributed system is implemented and presented in this paper. In the system, geographically dispersed design experts and their specific design resources including design tools, applications, and commercial software systems, and database files are integrated together over the Internet to conduct collaborative design for a common design task. Any applications between two sites can be directly communicated with each other across the barriers such as languages, platforms, or protocols. The system enables experts from different domains, such as PDS, design optimisation, geometrical modelling, and stress analysis, to effectively collaborate with each other. Design resources considered in the system are mostly heterogeneous, i.e. written in different languages and ported on different platforms. The applications can be communicated with each other within the CORBA communication architecture.

The optimisation programs are collocated in three sites. These programs can be directly communicated with each other. CORBA Gatekeepter enables a Java Applet client to be communicated with the remote object by getting through the client enterprises firewall.

This model makes it easily to develop the complex distributed system using multiple programs and enables to construct integrated collaborative design environment for implement complex design tasks.

7 Acknowledgement

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

8 References

[1] R. Orfali and D. Harkey, *Client/Server programming with Java and CORBA*, 2nd ed. John Wiley & Sons Inc., New York Chichester Weinheim Brisbane Singapore Toronto, 1998.

[2] F. Pahng, S. Bae and D. Wallace, “A Web-based Collaborative Design Modeling Environment”, Proceeding of the 7th Workshop on Enabling Technologies: Infrastructure for Collaborative Engerprises, Stanford, CA., pp. 161-167, June 17-19, 1998.

[3] Ji. Li, “Collaborative Design Method on Network”, *Journal of Tsinghua University Xuebao*, Vol 9, pp.93-96, 2000.

[4] S. B. Yoo and Y. Kim, “Web based Knowledge Management for Sharing Product Data in Virtual Enterprises”, International *Journal of Production Economics*, Vol 1, pp. 173-183, 2002.

[5] R. M. Hauch, S. W. Jacobs, S. W. Prey and H. L. Samsel, “A Distributed Software Environment for Aerospace Product Development”, AIAA/ASME/ASCE/AHS/ASC Structures Struct Ural Dynamics & Materials Conference, St. Louis Mo USA, pp. 1385-1394, 12-15 April, 1999.

[6] K.-T. Lee, M.-L. Roh and S. Cho, “Multidisciplinary Design Optimisation of Mechanical Systems Using Collaborative Optimisation Approach”, *International Journal of Vehicle Design*, Vol 4, pp. 353-368, 2001.

[7] J. Sang, G. Follen, C. Kim, I. Lopez and S. Townsend, “CORBA Wrapping of Legacy Scientific Applications Using Remote Variable Scheme”, International Conference of Parallel & Distributed System, Kyongju, 26-29 Jun, 2001

[8] http://www.w3.org/ accessed on 01/Feb/2005.

[9] Y. Ouyang, M. Tang, J. Lin and J. Dong, “Distributed Collaborative CAD System Based on Web Service”, *Journal of Zhejiang* University *SCIENCE SCI*. No.5, pp.579-586, 2004.