International Journal of CAD/CAM Vol. 3, No. 2, pp. 51~60 (2003)

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

Application of Multi-Frontal Method in Collaborative Engineering Environment

Ki-Eak Kwon*, Seong Wook Cho, Young Choi and Gyu-Bong Lee[1]
School of Mechanical Engineering, Chung-Ang University, Seoul, Korea
[1]Korea Institute of Industrial Technology, Chon-An, Korea

Abstract – The growth of the World Wide Web and the advances in high-speed network access have greatly changed existing CAD/CAE environment. The WWW has enabled us to share various distributed product data and to collaborate in the design process. An international standard for the product model data, STEP, and a standard for the distributed object technology, CORBA, are very important technological components for the interoperability in the advanced design and manufacturing environment. These two technologies provide background for the sharing of product data and the integration of applications on the network. This paper describes a distributed CAD/CAE environment that is integrated on the network by CORBA and product model data standard STEP. Several prototype application modules were implemented to verify the proposed concept and the test result is discussed. Finite element analysis server are further distributed into several frontal servers for the implementation of distributed parallel solution of finite element system equations. Distributed computation of analysis server is also implemented by using CORBA for the generalization of the proposed method.

***Keywords*: Finite elements, CAE, CORBA, STEP, JAVA, Distributed engineering, Multi-Frontal**

1. Introduction

A rapid advancement in computer technology is providing motivations for researchers and engineers to explore a new paradigm in manufacturing environment. Especially, technologies including object oriented software development, distributed application development, and new international standard for representing product model data offers main thrust for implementing distributed and concurrent engineering environment. Terminology such as CIM and concurrent engineering has been around for many years and at least the concept has been sufficiently established already. However, the concept has not been fully implemented to everybody's satisfaction due to the lack of software tools and inadequate infrastructure of manufacturing organizations. The emerging technologies prompted various research projects for distributed and collaborative engineering in diverse areas in design and manufacturing.

Hardwick *et al.* (1996) proposed a prototype information infrastructure for virtual manufacturing enterprises [1]. The main focus is on establishing the standard data protocols for diverse application systems. Combination of the Internet with the Standard for the Exchange of Product Model Data (STEP) and the Common Object Request Broker Architecture (CORBA) is the key idea. At the lower level, combined usage of two languages, i.e. data modeling language EXPRESS for STEP and CORBA interface definition language IDL, enables such integration in the distributed heterogeneous engineering environment. They proposed the use of information infrastructure with STEP and CORBA. However, the proof of the concept was not shown with application development.

A paper by Pahng *et al.* (1998) describes more focused research for product design problems in a computer network-oriented design environment [2]. The proposed framework is based on the extension of object-based modeling and evaluation concept to the distributed environment. This paper shows a good example of collaborative product design that encompasses potentially competing objectives in diverse disciplines.

The usage of 'feature' concept in the distributed environment is another application area that several researchers are investigating. A feature can be defined as a characteristic shape or attribute relevant to design or manufacturing process. Researches are tempted to utilize the feature concept for the distributed engineering environment because the feature concept provides a multiple view of product data for multiple applications that might also be distributed. The paper by Martino *et al.* (1998) addresses the integration of design and analysis by feature-based intermediate representation in the distributed environment [3]. Design-by-features and feature recognition approach is supported in the distributed, object oriented feature-based system architecture. Han and Requicha (1998) described a feature recognition system that runs in a distributed environment, through uniform API that wraps around geometric modeler [4]. Gadh and Sonthi (1998) proposed to use different levels of geometric abstraction by feature recognition

*Corresponding author:
Tel: +82-(0)2-816-4973
Fax: +82-(0)2-817-5101
E-mail: kwonke@ms.cau.ac.kr

for Internet-based design [5]. None of the feature applications in the distributed environment mentioned above used the CORBA for communications between applications.

Managing and sharing of design and product information in enterprise level and inter-enterprise level is also a great concern in the Internet age. Dong and Agogino (1998) proposed the framework that merges the processes of the design documentation and design data management in a networked design environment [6]. Then the information is accessed in a client/server environment. The RISESTEP [7] project aims at developing a framework for sharing product data and collaborating in a distributed heterogeneous environment. The collaboration is achieved by sharing standardized product data among various partners in the extended enterprises. The basis for the integration in the project is STEP and CORBA. A similar approach for VR application is discussed in the paper by Marache *et al.* (1997) [8].

The aim of the SHARE [9] project at Stanford was to build a real-time collaboration environment upon the WWW. The collaboration was accomplished by sharing of information by Web browsers and by multimedia communication between participants and teams. The MADEFAST [10] project shows an example of successful usage of WWW for collaborative design by teams of multiple disciplines who are geographically distributed. The Next-Link [11] project at Stanford was to develop coordination agents in an infrastructure for distributed design and concurrent engineering.

While many engineering applications including above surveys are finding their way into distributed engineering environment, the numerical analysis such as finite element analysis in mechanical design process has been performed still in the traditional stand-alone fashion. The focus of this paper is to propose a framework for a distributed finite element analysis system over computer network for mechanical design and to discuss the result from the prototype implementation. Of course, the proposed framework for distributed analysis will be well fitted in the overall distributed engineering environment by means of using standard data and communication protocols.

2. Enabling Technologies

2.1. Software development with CORBA and Java

Fig. 1 shows application software development process with CORBA [12, 13]. Orbix and OrbixWeb from IONA [14] Technologies were used as CORBA tool, and the Java was used as a development language in our research.

Java and CORBA are mutually complementary technologies in distributed software development [15]. These technologies allow development of powerful distributed engineering applications on the WWW. Communication between clients and remote server is supported by CORBA protocols. Clients can be developed as Java applets and can be downloaded from remote server through Web browsers.

Fig. 1 shows how CORBA interface for distributed applications is defined and necessary client and server

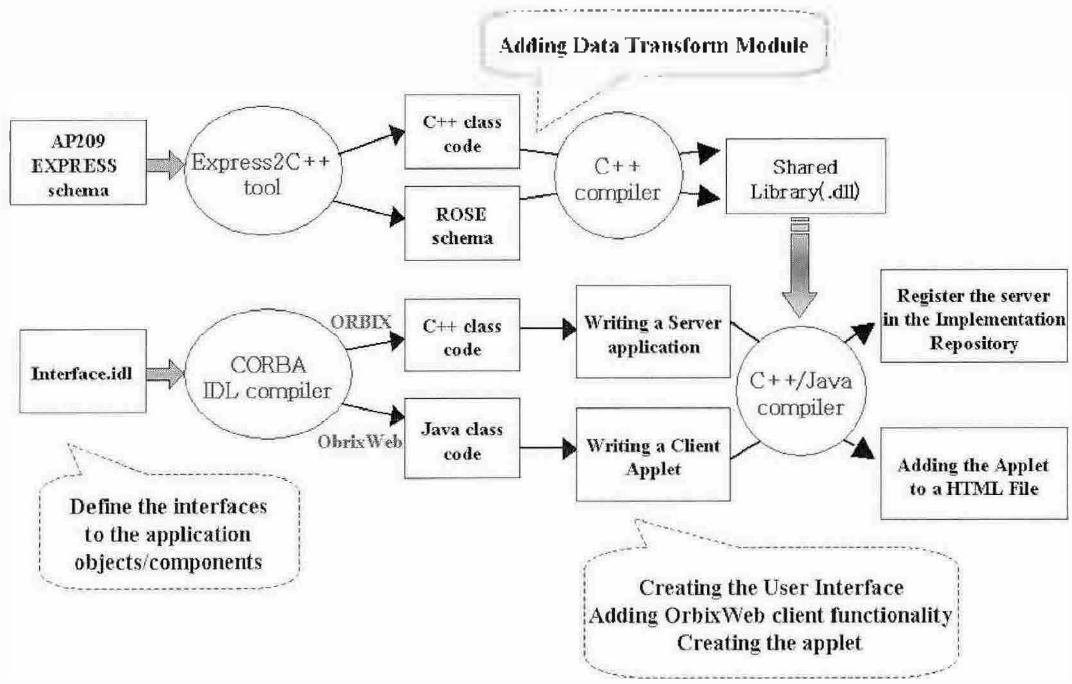


Fig. 1. Software development with CORBA.

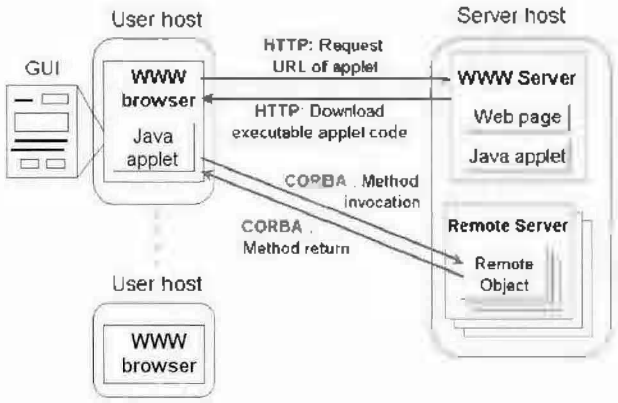


Fig. 2. Distributed applications with CORBA and Java.

code are developed with implementation language, in this case, Java.

Java is an object oriented programming language rapidly accepted by software developers. One of the characteristics of Java in developing distributed applications is platform independent byte-code. The byte-code is interpreted by Java interpreter and executed. Another characteristic is the mechanism of Java applet. Java applets can be downloaded in real time through Web browsers and executed on host computer automatically. Java applets make development and installation of client components easier in distributed application systems. Fig. 2 depicts mechanisms of how remote CORBA applications can be invoked in distributed application environment implemented with Java and CORBA on the WWW.

2.2. Finite element analysis with STEP

One of the important aspects in engineering collaboration in distributed environment is how to store and efficiently share product data between distributed applications. One approach is to use standard product model data. ISO 10303, STEP, covers various product model data in many application areas. Product model data for engineering analysis is also covered by Application Protocol 209 in STEP [16]. STEP AP209 [17] provides a standard data model that satisfies the need for the exchange of computer-interpretable structural product definitions, including product shape, associated FEA models, material properties and analysis results. The nature of standard representation of FEA models and analysis results allows easy integration and data sharing with other existing FEA applications. To use STEP for developing finite element analysis software, it is necessary to parse and generate STEP physical files according to the AP209 schema defined with EXPRESS.

ST-Developer from STEP Tools Inc. was used in our implementation to generate C++ classes for entities in the AP209 schema. An interface between STEP data and finite element analysis module is necessary to translate STEP data into generic analysis data used in our finite analysis modules.

3. Distributed Finite Element Analysis

Finite element analysis of large structures usually requires huge amount of computer resources. For the cost-efficient analysis of large structures, various parallel computational scheme have been proposed. Since the specific hardware was needed for the parallel computation, the application of the algorithm was highly limited in the past. However recent development of distributed environment over the network provides promising motivation in the system equation solution fields.

In this paper, distributed finite element analysis system is proposed for the solution of the system equations. In the proposed system, heterogeneous hardware systems are integrated over the network for the simultaneous calculation of the large structural system.

3.1. Parallelized finite element analysis

In the first stage of the parallelization, studies were focused on the system matrix modification for the utilization of multi-CPU facilities. Recent studies are based on the subdivision of the analysis domain. Domain Decomposition Method is one of those proposals, which is based on the subdivision of the structures into the number of substructures. Each domain is independently analyzed and assembled to form a complete sub-matrices. Degrees of freedom along the boundaries are collected through internet for the solution of the system matrix. Pre-conditioned Conjugate Gradient (PCG) method is also proposed for the analysis of large structures [18].

In this paper we proposed a parallelized system equation solution scheme in distributed environment. In the solution process multi-frontal algorithm has been used for the main solution method [19-21]. For the multi-frontal solver, efficient domain decomposition is required. Solution efficiency is highly depends on the nature of domain decomposition. Hence proper construction of wave front decomposition and multi-frontal solution scheme is necessary. In this paper, domain decomposition is performed by graph decomposition theory [22].

3.1.1. Parallelization by Multi-frontal method

Parallelization of the multi-frontal solver is realized by use of its inherent characteristics, that is, independent calculation of each wave front and independent assembly of adjacent frontal degrees of freedom. Entire domain is decomposed into the number of CPU's used. Assembly between different hardware platform requires relatively small amount of data transfer. In this way, efficiency of parallel computation can be highly improved.

In the frontal solution procedure only those equations that are actually required for the elimination of a specific degrees of freedom are assembled, the degrees of freedom considered statically condensed out, and so on. An advantage of the wave front solution is that

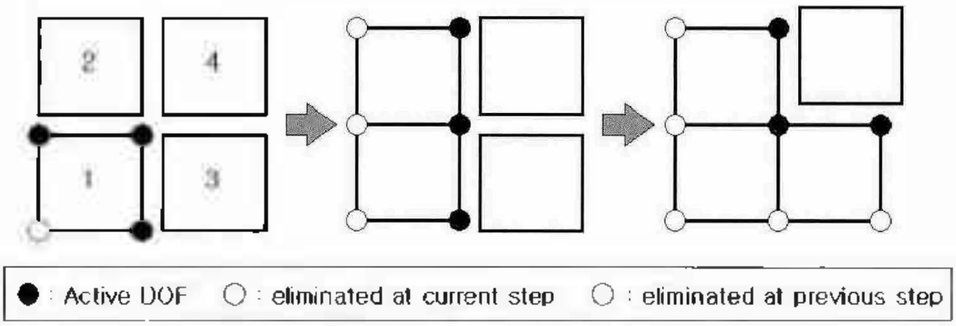


Fig. 3. Successive elimination in the frontal method.

elements can be added with relative ease because no nodal point renumbering is necessary to preserve a small bandwidth. Further, it does not require the assembled global stiffness matrices, the method can be implemented with relatively small amount of hardware resource like high speed main storage and auxiliary devices.

Fig. 3 shows the procedure of frontal solution algorithm where internal degrees of freedom are statically condensed out as soon as it is assembled. The wave front or frontal method is an arrangement of Gauss elimination in which assembly of structural equations alternates with their solution. The first equations to be eliminated are those associated with element 1 only. Then the adjacent element, element 2, makes its contribution of stiffness coefficients to the system of equations. If any additional equations are fully summed – that is, if any additional degrees of freedom are shared by elements 1 and 2 only – these equations are eliminated. The next elimination awaits contributions from one or more additional elements. This repetitive alternation between assembly and solution can be viewed as a "wave" that sweeps over the structure in a pattern dictated by the element numbering.

Multi-frontal solution procedure is an extension of fontal solution scheme. In this procedure, several independent wave front are employed in the first stage and each wave front is assembled altogether later on. Fig. 4 shows the procedure of multi-frontal solution algorithm where four independent wave fronts are employed for example.

Multi-frontal solution algorithm is the process of reducing the number of d.o.f by substitution, for example, by starting a Gauss elimination solution of equations for unknowns but stopping before the stiffness matrix has been fully reduced. Multi-frontal solution process is described as follows.

1. Evaluate $[\mathbf{k}]$ and $\{\mathbf{f}\}$ for each domain, where $[\mathbf{k}]$ and $\{\mathbf{f}\}$ pertain to all d.o.f. of the domain. Let the equation $[\mathbf{k}^A]\{\mathbf{u}^A\}=\{\mathbf{f}^A\}$ represent a portion of the domain A. And Let d.o.f. $\{\mathbf{u}^A\}$ be partitioned so that $\{\mathbf{u}^A\}=[\mathbf{u}_i, \mathbf{u}_e]^T$, where $\{\mathbf{u}_i\}$ are interface d.o.f. to be retained and $\{\mathbf{u}_e\}$ are internal d.o.f. to be eliminated by condensation. Thus $[\mathbf{k}]\{\mathbf{u}\}=\{\mathbf{f}\}$ become.

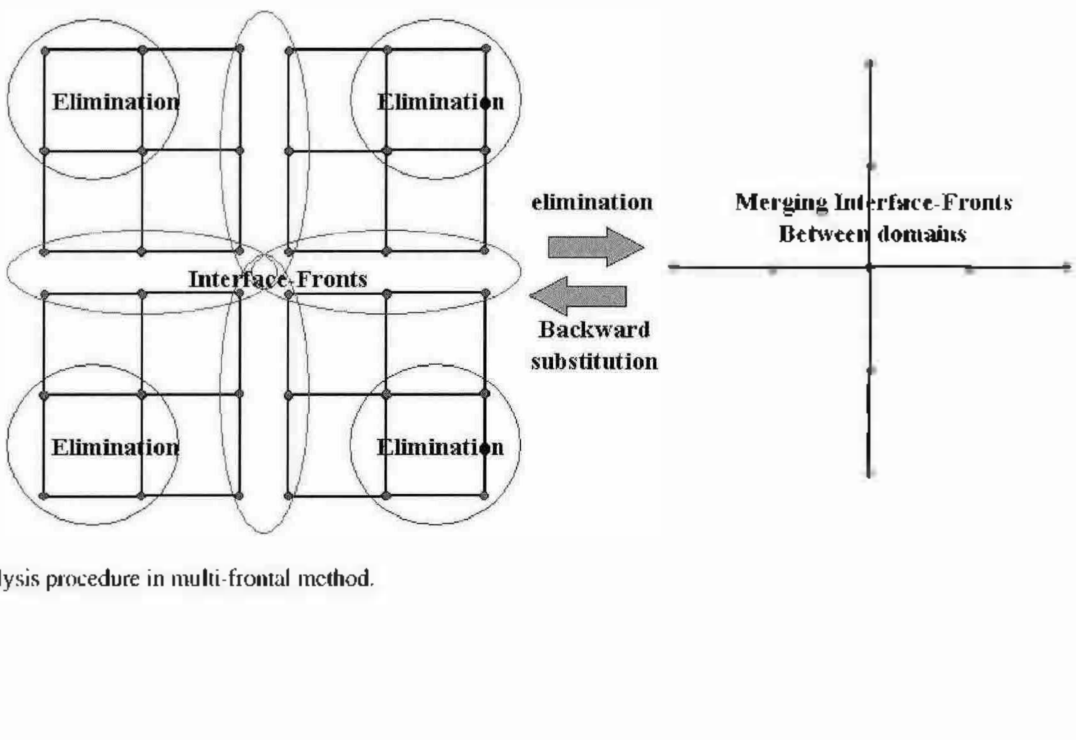


Fig. 4. Analysis procedure in multi-frontal method.

$$\begin{bmatrix} K_{ii}^A & K_{ic}^A \\ K_{ci}^A & K_{cc}^A \end{bmatrix} \begin{Bmatrix} u_i^A \\ u_c^A \end{Bmatrix} = \begin{Bmatrix} f_i^A \\ f_c^A \end{Bmatrix} \tag{1}$$

Eliminate internal d.o.f. $\{\mathbf{u}_c\}$ by static condensation;

$$u_c^A = [K_{cc}^A]^{-1}[f_c^A - K_{ci}^A u_i^A] \tag{2}$$

The condensed $[\mathbf{k}]$ and $\{\mathbf{f}\}$ pertain to only the interface d.o.f. $\{\mathbf{u}_i\}$ of the domain.

$$\bar{f}_i^A = K_{ic}^A [K_{cc}^A]^{-1} f_c^A \tag{3}$$

$$\bar{K}_{ii}^A = K_{ii}^A - K_{ic}^A [K_{cc}^A]^{-1} K_{ci}^A$$

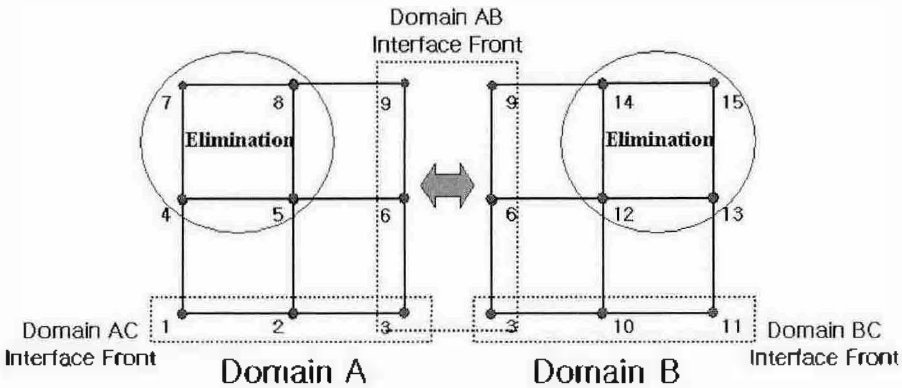
$$f_i^A - \bar{f}_i^A = \bar{K}_{ii}^A u_i^A \tag{4}$$

2. Assemble domains by connecting interface nodes (i.e., nodes shared by domains). Thus generate structural equations $[\mathbf{K}_i]\{\mathbf{U}_i\}=\{\mathbf{F}_i\}$, in which $\{\mathbf{U}_i\}$ contains the interface d.o.f. of all domains. Solve for $\{\mathbf{U}_i\}$.

$$F_i - \bar{F}_i = \bar{K}_{ii} U_i \tag{5}$$

3. For each domain, extract from $\{\mathbf{U}_i\}$ the interface d.o.f. $\{\mathbf{u}_i\}$ of that domain. Use eq. (2) to compute interior d.o.f. $\{\mathbf{u}_c\}$ (Backward-Substitution). Now all d.o.f. of the domain are known. Hence, stress calculation proceeds in the usual way.

Fig. 5 shows the procedure of merging fronts between neighbored domains. Each domain is composed of four-element assuming d.o.f. of each node is one.



$$ID^A = [\ 1\ \ 2\ \ 3\ \ 6\ \ 9\ ] \qquad ID^B = [\ 3\ \ 6\ \ 9\ \ 10\ \ 11\ ]$$

$$K^A = \begin{bmatrix} k_{11}^A & k_{12}^A & k_{13}^A & k_{14}^A & k_{15}^A \\ & k_{22}^A & k_{23}^A & k_{24}^A & k_{25}^A \\ & & k_{33}^A & k_{34}^A & k_{35}^A \\ & Symm. & & k_{44}^A & k_{45}^A \\ & & & & k_{55}^A \end{bmatrix} \qquad K^B = \begin{bmatrix} k_{11}^B & k_{12}^B & k_{13}^B & k_{14}^B & k_{15}^B \\ & k_{22}^B & k_{23}^B & k_{24}^B & k_{25}^B \\ & & k_{33}^B & k_{34}^B & k_{35}^B \\ & Symm. & & k_{44}^B & k_{45}^B \\ & & & & k_{55}^B \end{bmatrix}$$

$$ID^{A+B} = [\ 1\ \ 2\ \ 3\ \ 6\ \ 9\ \ 10\ \ 11\ ]$$

$$K^{(A+B)} = \begin{bmatrix} K_{11}^A & K_{12}^A & K_{13}^A & K_{14}^A & K_{15}^A & 0 & 0 \\ & K_{23}^A & K_{23}^A & K_{24}^A & K_{25}^A & 0 & 0 \\ & & K_{33}^A + K_{14}^B & K_{34}^A + K_{12}^B & K_{35}^A + K_{13}^B & K_{14}^B & K_{15}^B \\ & & & K_{44}^A + K_{22}^B & K_{45}^A + K_{23}^B & K_{24}^B & K_{25}^B \\ & & & & K_{55}^A + K_{33}^B & K_{34}^B & K_{35}^B \\ & & Symm. & & & K_{44}^B & K_{45}^B \\ & & & & & & K_{55}^B \end{bmatrix}$$

Fig. 5. Front merging procedure.

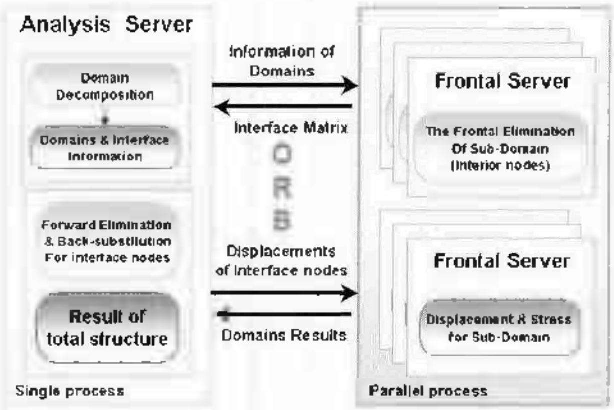


Fig. 6. Distributed parallel computation over the network.

Parallel libraries like MPI (Message Passing Interface) or PVM (Parallel Virtual Machine) are implementation of parallel finite element solution scheme based on the multi-frontal procedure. But those previous implementations were mainly used for the multi-CPU super computer facilities.

In this paper, multi-frontal procedure has been implemented over the distributed environment by using CORBA which has already been used for the distribution between various servers. Fig. 6 shows schematic diagram of the distributed parallel computation over the network by CORBA.

Upon analysis data are provided to the analysis server in the form of STEP Physical file, the analysis server divides the calculation domain according to the status of active frontal server. Total number and the capabilities of available active frontal servers are considered in this domain decomposition. Information on each analysis domain are transferred from analysis server to selected frontal severs. Individual frontal servers are invoked through multi-thread in order to perform the independent computation, i.e. distributed parallel system by CORBA.

Here, each frontal server is provided with the data which is an object defined through CORBA-IDL interface. In this way, efficiency of frontal solution itself can be improved. After complete solutions are obtained at Analysis Server, the results are returned to the client in the form of STEP physical file.

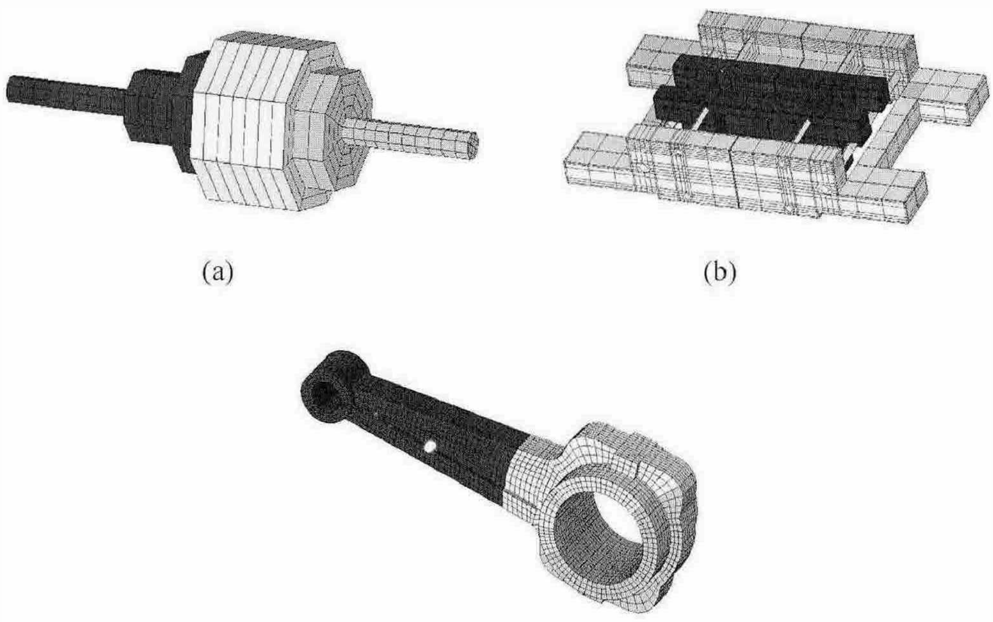
For the effective use of distributed frontal servers, proper distribution of the calculation load are crucial since each servers are built on heterogeneous platforms and have different capabilities. Multi-graph theory is employed for the efficient domain decomposition of the analysis domain.

3.1.2. Domain decomposition by graph theory

In the parallel computer, proper division of the domain is necessary in order to maintain the balance between distributed hardware platforms. It is also required to minimize the size of the wave front between boundaries of each wave front.

In this paper, multi-stage graph theory was adopted for the efficient domain decomposition of multi-frontal solution [23]. Actual implementation was done by using the freeware Metis [24].

Fig. 7 shows an example of domain decomposition by using multi-stage graph theory. In the example, finite elements are divided into 4 regions. Information on the boundary nodal points are transferred to the



(a) (b)

(c)

Fig. 7. Domain decomposition by graph theory. (a) 3564 DOF, (b) 19968 DOF, and (c) 38331 DOF.

Table 1. Hardware configuration for test run

Server	CPU	Memory
Analysis	Pentium III 700	256 Mb
Frontal [1]	Pentium III 700	256 Mb
Frontal [2]	Pentium III 500	256 Mb
Frontal [3]	Pentium III 500	196 Mb

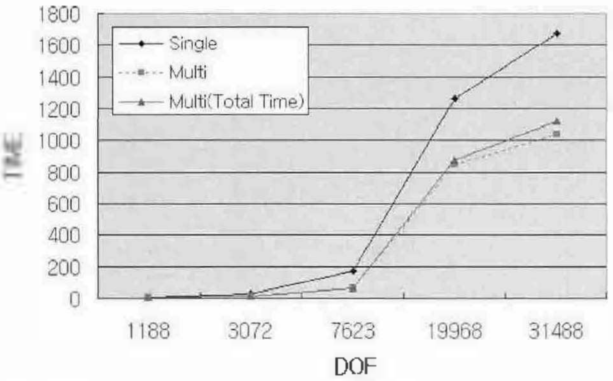


Fig. 8. Performance comparison for DOF Increment.

frontal servers.

3.2. Distributed finite analysis element using CORBA

A test bed is developed for the implementation of multi-frontal solver based on multi-step graph theory and CORBA. Three examples in Fig. 7 are analyzed in order to demonstrate the efficiency of the multi-distributed multi-frontal procedure. Table 1 lists the development environment for the frontal servers. Fig. 8 shows the analysis results comparison that demonstrates the performance of the distributed multi-frontal solver.

4. Distributed Analysis System Architecture

4.1. System architecture

A test bed has been developed to demonstrate the proposed scheme. Current implementation of the proposed system architecture includes database server and application servers that are completely separated from the client software. Current server applications include CAD server, database server and Finite Element Analysis server [25]. Each server is composed of several independent modules that can interact with other server modules or client. Design data is stored in the database server as STEP AP203 physical file. Analysis data including loading, boundary conditions, analysis parameters and mesh data is generated and managed with STEP AP209 schema. Analysis results, such as displacements, stress distribution data are also managed as STEP AP209 physical file in the database server.

CAD server is composed of visualization module and meshing module. When a client want to see the shape of the part, visualization module generates Java 3D rendering data from the corresponding STEP data in

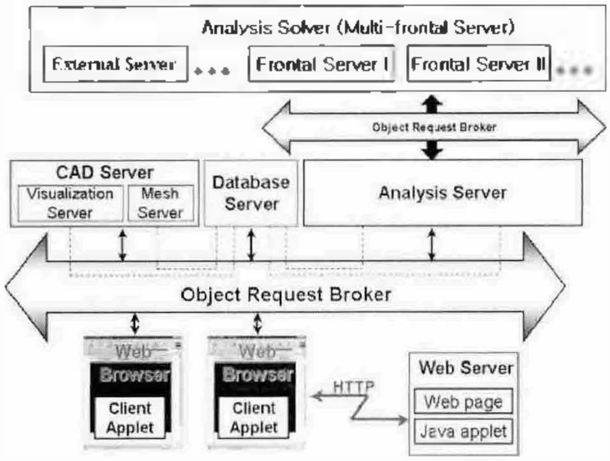


Fig. 9. Distributed finite element analysis system architecture.

the database and send it to the client applet. Meshing module generates finite element meshes for the geometry retrieved from the database after receiving data including element type and mesh density from the remote client.

Our current implementation of analysis server includes external port for the external commercial analysis code, main solution module, and multi-frontal solver module. Multi-frontal solver module also distributes the solution domain over the network through CORBA.

Current client also implements Java3D and Swing package for the user interface and data output.

4.2. Scenario for the distributed analysis

The following is work scenario for the prototype implementation of the proposed distributed analysis system architecture.

(1) Client connects to the Web server which provide client Java applets with common Web browser.
(2) Java applets for the user interface is downloaded to the client Web browser.
(3) Client connects to the DB (STEP data) server to retrieve project information. The project data contains meta data that points to relevant CAD data, mesh data, analysis result data and etc. Model data can be viewed through Visualization server where model data are transformed into Java 3D rendering data and downloaded into client applet.
(4) Upon client's request for meshing or solving, application server requests necessary data (STEP data) list from DB server. Operations requested by client are performed at each computer where corresponding application modules reside. Analysis parameters are also provided.
(5) Finite element analysis is performed by invoking the Analysis server.
(6) Analysis server invokes Multi-frontal server for the distributed parallel process. Analysis server also divides the analysis domain into substructures for the parallel computation.

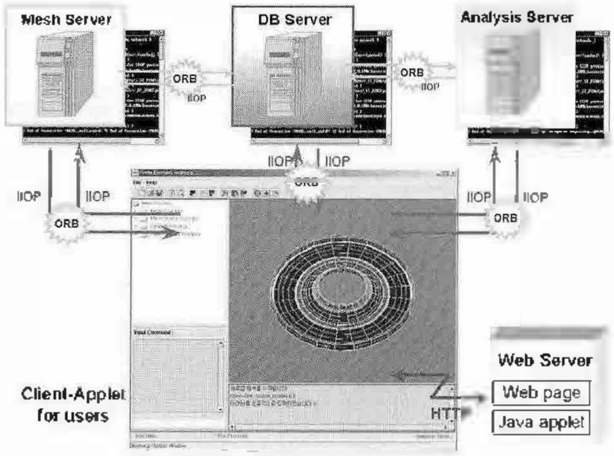


Fig. 10. Structure of distributed analysis.

(7) Analysis Server translates the domain information into the objects defined according to the IDL Interface. Client may receive results from application modules after computation is finished.

(8) Objects translated from each domain information according to the IDL Interface are provided from the analysis server to the selected active frontal.

(9) Client may receive results (STEP Physical File) from application modules after computation is finished.

(10) Client may save all the operation results at DB server. Analysis results are saved in STEP AP209 physical file format. Project data may also be renewed.

Interface between the server applications and client software is facilitated by ORB middleware. Clients only need to request service to ORB regardless of the type or location of the servers. Fig. 10 shows the process between various servers and the client.

4.3. Scenario for collaborative analysis

The following is work scenario for collaborative analysis.

(1) Clients connects to the Web server which provide

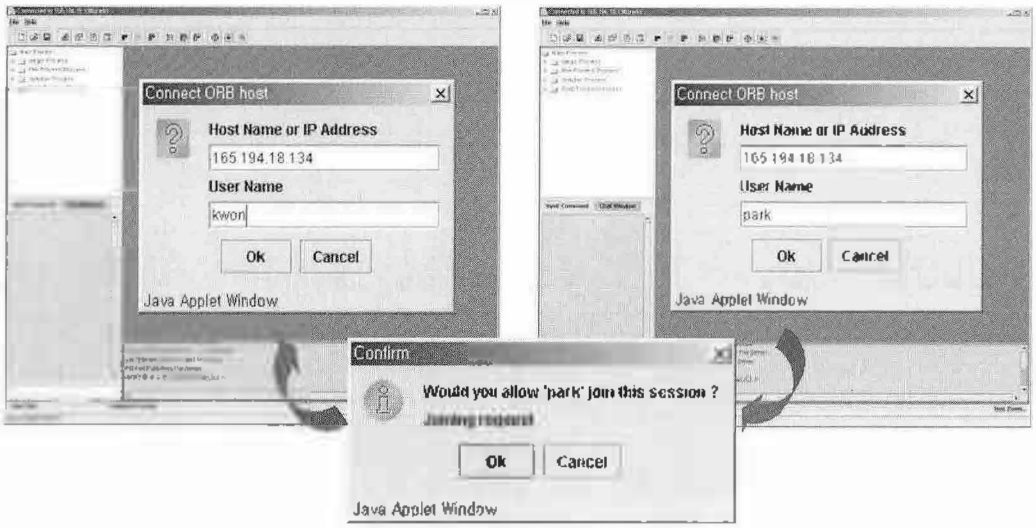
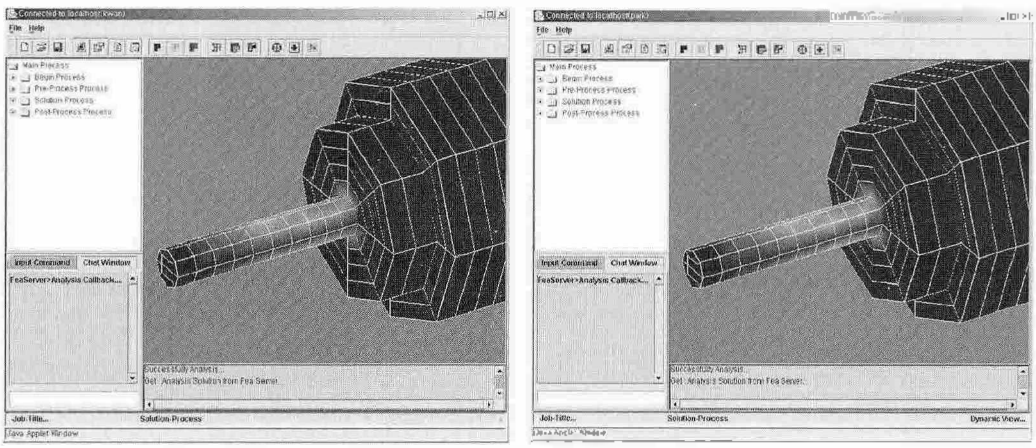


Fig. 11. Creating/joining analysis session.



(a) (b)

Fig. 12. Collaborative analysis sharing analysis results (a) client 1, (b) client 2.

client Java applets with common Web browser. And Java applets for the user interface is downloaded to the client Web browser.

(2) Clients enter or join session for collaborative analysis.

(3) Clients connect to the DB server to retrieve analysis model and result data.

(4) Clients discuss analysis results with clients in the session.

Fig. 11 shows that a client joins a session.

Fig. 12 shows analysis conference session sharing and discussing analysis results between remote analysis clients.

5. Conclusion

In this research, we tried to demonstrate the feasibility of STEP-based integrated engineering environment through the development of distributed finite element analysis system with STEP and CORBA. STEP product model data, including finite element analysis results data, in the form of physical files or database generated from the proposed system may be shared by other distributed application modules without special translation efforts. The enabling technologies such as CORBA and Java play key roles in the development of integrated and geographically distributed application software.

In addition to the distribution of analysis modules, numerical solving process itself was again divided into parallel processes using multi-frontal method for computational efficiency. In contrast to the specially designed parallel process for specific hardware, CORBA-based parallel process is well suited for heterogeneous platforms on a network.

The idea of Web-based distributed analysis system may be applied to the engineering ASP for design and analysis in the product development processes. Rapid progress in the network technology, including high bandwidth, will accelerate this trend. The trend of using pay-per-use engineering ASP through the Web is another driving factor for the further development of this approach.

We believe that the proposed approach for the analysis can be extended to the entire product development process for sharing and utilizing common product data in the distributed engineering environment, thus eventually provide bases for virtual enterprise.

Acknowledgements

This work was supported by Advanced Ship Engineering Research Center of the Korea Science & Engineering Foundation.

References

[1] Hardwick, M., Spooner, D. L., Rando, T. and Morris, K. C. (1996), "Sharing manufacturing information in virtual enterprises," *Communications of the ACM*, **39**(2), 46-54.

[2] Pahng, H., Senin, N. and Wallace, D. (1998), "Distribution modeling and evaluation of product design problems," *Computer-Aided Design*, **30**(6), 411-424.

[3] Martino, T. D., Falcidieno, B. and Hainger, S. (1998), "Design and engineering process integration through a multiple view intermediate modeller in a distributed object-oriented system environment," *Computer-Aided Design*, **30**(6), 437-452.

[4] Han, J. H. and Requicha, A. A. G. (1998), "Modeler-independent feature recognition in a distributed environment," *Computer-Aided Design*, **30**(6), 453-464.

[5] Gadh, R. and Sonthi, R. (1998), "Geometric shape abstractions for internet-based prototyping," *Computer-Aided Design*, **30**(6), 473-486.

[6] Dong, A. and Agogino, A. M. (1998), "Managing design information in enterprise-wide CAD using 'smart drawings'," *Computer-Aided Design*, **30**(6), 425-435.

[7] Debras, P., Huguet, P., Amar, V. and Poyet, P. (1997), RISESTEP (EP 20459) : a user driven project to develop a platform for product data sharing in a concurrent engineering context. Proceeding of the Product Data Technology Days : 23-30. French Riviera, France.

[8] Marache, M., Amar, V. and Diard, F., A CORBA-based infrastructure managing STEP distributed models for virtual reality applications. *Proceeding of the Product Data Technology Days* : 119-128. French Riviera: France.

[9] SHARE: A Methodology and Environment for Collaborative Product Development. http://cdr.stanford.edu/SHARE/share.html.

[10] Cutkosky, M. R., Tenenbaum, J. M. and Glicksman, J. (1996), "Madefast: Collaborative engineering over the Internet," *Communications of the ACM*, **39**(9), 78-87.

[11] NEXT-LINK, http://cdr.stanford.edu/NextLink/NextLink.html.

[12] CORBA/IIOP 2.2 Specification, (1998), OMG, http://www.omg.org/ library/c2index.html

[13] Jon Siegel, (1996), *CORBA fundamentals and programming*, John Wiley & Sons, Inc.

[14] IONA Technologies, (1996), *OrbixWeb programming & reference guide*.

[15] Evans, Eric and Rogers, Daniel, (1997), *Using Java Applets and CORBA for Multi-user Distributed Applications*, http://computer.org/internet/.

[16] STEP Tools Inc. (1996), *STEP Programmer's toolkit - ROSE library reference manual*.

[17] ISO, 1996, ISO 10303-209 - Part 209: Application Protocol: Composite and metallic structural analysis and related design.

[18] Papadrakakis, M. and Bitzarakis, S. (1994), "Domain decomposition PCG methods for serial and parallel processing," *Advances in Parallel and Vector Processing for Structural Mechanics*.

[19] Ingle, N. K. and Mountziaris, T. J. (1995), "A multifrontal algorithm for the solution of large systems of equations using network-based parallel computing," *Computers Chem. Engng.*, **19**, 671-681.

[20] Amestoy, P R., Duff, I. S. and L'Excellent, J. Y. (1998), Multifrontal parallel distributed symmetric and unsymmetric solvers. *Technical Report RAL-TR-1998-051*, http//www.numerical.rl.ac.uk/reports/reports.html.

[21] Scott, Jennifer A. (2002), Parallel frontal solvers for large sparse linear systems. *Technical Report TR-2002-012*, http//www.numerical.rl.ac.uk/reports/reports.html.

[22] Karypis, G. and Kumar, V. (1995), "A fast and high

quality multilevel scheme for partitioning irregular graphs," *Technical Report TR 95-035*. University of Minnesota : Department of Computer Science.

[23] Karypis, G. and Kumar, V. (1995), *Metis ver. 4.0 : A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*, Department of Computer Science, University of Minnesota.

[24] Kim, S. J. and Kim, J. H. (April 28-May 2 1997), "Large-scale Structural Analysis using Domain Decomposition Method on Distributed Parallel Computing Environment," *Proc. of HPC Asia 1997*. Seoul, Korea.

[25] Kwon, K. E., Park, M. J., Yang, S. W., Choi, Y. and Cho, S. W. (June 1998), "Distributed CAD/CAE Environment Based on STEP and CORBA," *Fourth World Congress on Computational Mechanics*.