International Journal of the Korean Society of Precision Engineering Vol. 4, No. 6. November 2003.

Web-based Simulation System for Multibody Systems

Hyung-Suk Han[1,#]

[1] New Transportation System Group, Korea Institute of Machinery & Materials (KIMM), Daejeon, South Korea

ABSTRACT

A web-based dynamic simulation system, called O-DYN, for multibody systems is developed. All the interfaces of the system are accessible via web browsers, such as Netscape or Explorer. The system uses a block-diagram type O-DYN/Modeler developed in JAVA Applet as a preprocessor. The O-DYN postprocessor composed of O-DYN/Plotter and O-DYN/Animator is developed in JAVA Applet. The O-DYN/Solver for predicting the dynamic behavior is run on the web server. Anyone who wants to simulate the dynamics of multibody systems or share results data can access the analysis system over the internet regardless of their OS, platform, or location.

Key Words : Multibody system , Dynamics, Web-based, JAVA, Internet, Web browser

1. Introduction

Due to advances in multibody dynamics simulation technologies and the development of commercial software, the dynamics simulation of multibody systems has become prevalent. Multibody dynamics is applied in a variety of fields such as the automotive industry, heavy industry, aerospace, electronics, bioengineering and others. Multibody dynamics simulation plays an important role in virtual engineering, and its importance is increasing gradually. Therefore, multibody dynamics simulation is no longer a field limited only to experts. It would be desirable for all engineers involved in developing a given machine to easily take part in the process of multibody dynamics simulation and share the results of the simulation in order to collaborate with one another.

At present, the development of internet and web application technologies offers a new opportunity for sharing and distribution of information. Web applications have the advantage of overcoming the limits of the user's heterogeneous computing environments and location. Furthermore, web-based collaboration can be achieved. Thus, web application technologies offer a new opportunity in the field of simulation. Although web-based simulation is already established in the field of industrial engineering, it is still in the initial stages in mechanical engineering.[1-7] It is considered that web-based simulations for mechanical engineering will become widespread within a few years.

A web-based system makes engineering calculations possible through a web server and a web browser regardless of the limitations of the computer system and the location of the engineer. In addition, expansion to distributed simulation, which is based on communication among the web servers, is possible. Large number of users can share the results simultaneously and engage in discussion with each other, so that collaboration and concurrent engineering are possible. Furthermore, the simulation results and relevant data can be stored in databases to increase the reusability of the information.

Currently, there is no web-based multibody dynamics simulation system that is available. Although commercial multibody dynamics simulation systems presently exist, such as DADS[8], ADAMS[9], RecurDyn[10] and SIMPACK[11], they are all stand-alone programs that can only work in

Manuscript received: September 8, 2003 ;
Accepted: October 14, 2003
Corresponding Author:
Email: hshan@kimm.re.kr
Tel: +82-42-868-7814, Fax: +82-42-868-7844

computers with installed simulation systems or by remotely using an auxiliary program. Since stand-alone programs can only be used by a computer with an installed system, it is difficult for many users to use it. In addition, they are difficult to share information. Stand-alone programs also save data in an independent computer; hence the technical information cannot be managed systematically. The commercial systems described above allow only one user, which means that as the number of users increases, the expenses increase proportionally. Recently, some leading analysis software developers started charging ASP (Application Software Provider) services using auxiliary programs. The introduction of these services shows that even engineering simulation systems are shifting from stand-alone programs to web-based systems.

In this paper, a web-based, general-purpose multibody dynamics simulation system is introduced. The objective of this paper is to introduce system architecture and implementation techniques in order to present ways of applying web application techniques to multibody dynamics simulation. Fig. 1 shows the system that this work is focused on. As shown in Fig. 1, the multibody dynamics analysis solver is installed in a web server and runs through the internet. All engineering calculations and data management are done in this web server. All user interfaces are serviced via a web browser. The user, through a web browser, is able to perform modeling, solving and evaluation of multibody systems. As a result, several users can perform multibody dynamics simulation through the web regardless of the computer system and the locations of the users. In addition, several users can share engineering information at the same time or at different times regardless of their locations, so that the implementation of collaboration and concurrent engineering is possible.

2. System Development

2.1 System architecture

Fig. 2 shows the architecture of O-DYN that is introduced in this paper. The system is mainly composed of PRE Module, POST Module, JOB Control Module and SOLVER Module. The PRE Module performs the function of defining a model, which the user tries to simulate using O-DYN, and is composed of the O-DYN/Modeler that creates the input data of O-DYN/Solver, a dynamics solver.

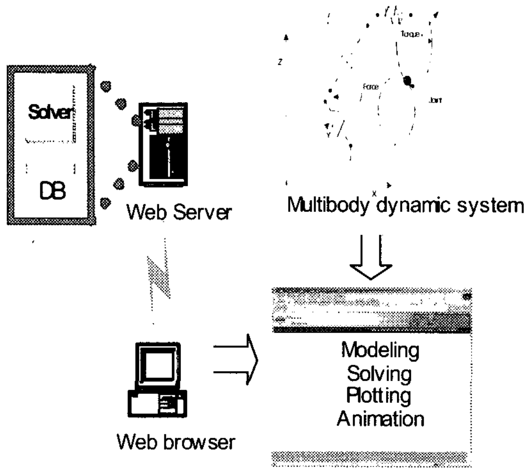

Fig. 1 Target web-based multibody dynamic simulation system

The POST Module provides analysis and visualization functions of the results of O-DYN simulation and is composed of the O-DYN/Plotter and O-DYN/Animator. The O-DYN/Plotter provides 2D/3D graphs for the analysis of simulation results. The O-DYN/Animator provides motion in a multibody system with 3D animation. The JOB Control Module manages the jobs requested from the PRE Module and POST Module. It is composed of the Job Manager, which manages the simulation demands of PRE Module and the invocation of the SOLVER Module according to simulation demands. The Job Manager also manages the transmittance of simulation results to the POST Module, the User Manager, which deals with the user's management and security, and the Data Manager, which takes care of the user's job data (simulation model, result data, input data) and the system data. The SOLVER Module performs a dynamic simulation using the user's simulation model submitted through the PRE Module. It is composed of the O-DYN/Solver and Solver Manager that manages the solver.

2.2 System Operations and Development Environment

As shown in Fig. 2, the PRE Module and POST

Module, which are the user's interfaces, run in the user's web browser, while the JOB Control Module and SOLVER Module run in the server. In order to connect to the system, the user should use a web browser that can use JAVA without being obstructed by the operating system or the type of web browser. When connected to the system, the user needs to first go through user authentication, and then select the PRE Module to perform a dynamics simulation or the POST Module to being performed analysis and visualization of the simulation results.

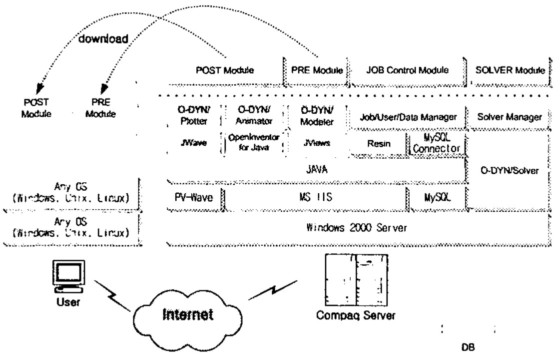

Fig. 2 System architecture

According to the selected job, the O-DYN/Modeler, O-DYN/Plotter and O-DYN/Animator will be transmitted to the user's web browser. The supported JAVA version is JDK 1.3.1.[12] OpenInventor for JAVA[13], JWave[14], and JViews[15] run time library, which are used in the PRE Module and POST Module, are transmitted to the user's web browser without any additional installation. As for the server where the user's job is being processed, a Compaq ML-570 Server was used with Windows 2000 Server as the OS. The software used in O-DYN are as follows: MySQL as the database server; PV-Wave[14] as the graph server; Resin[16] as the Servlet/JSP engine; the in-house O-DYN/Solver for dynamics simulation; and MS-ISS as the web server. The MS-ISS, a web server, sends the PRE Module and POST Module to the user and provides the JOB Control Module. The O-DYN used Microsoft products for the web server and the OS, but used JAVA and ANSI C++ for the system development. Therefore, it can be easily ported to non-Microsoft systems.

O-DYN is a web-based system that allows users to use the system simultaneously. Therefore, problems such as limitations on the number of users and load balancing can occur. The load balancing and number of users are limited by the server's hardware performance, network speed and number of run-time licenses. The number of simultaneous users refers to the number of users who use each module simultaneously while connected to the system. The O-DYN/Modeler and O-DYN/Animator are executed after transmitted to the user's web browser so that they do not affect the system. However, the O-DYN/Plotter uses a graph server while the users collaborate. Thus, it is related to run-time licenses of the graph server. In addition, since it uses the system connected to the internet, the network speed and transmission quantity are very important. The following describes the size of each module. The module of the ODYN/Modeler is about 2MB, including the GUI library, JViews, while the module of the O-DYN/Plotter is 300KB, including the graph library, JWave. The module of the O-DYN/Animator, including the graphic library, OpenInventor for Java, is about 3MB. Each module is saved in the cache of the user's web browser and is not transmitted whenever used, so that the actual data size of each module determines the transfer size. The O-DYN/Modeler is determined by the user's simulation model but the size of the model data is extremely small. In the O-DYN/Plotter and O-DYN/Animator, the size of the result data determines the transfer size. In particular, the O-DYN/Animator receives the geometric information for animation so that the transfer size drops significantly. From the results of the system's test operation, it is seen that inducing the split transmission of geometric information and the compression method can increase the efficiency of the system.

In web-based engineering systems, unspecified users can gain access to the system because of the characteristics of the internet. Hence, a method that only allows permitted users needs to be presented. In O-DYN, assuming that network security (firewall or security of other network levels) is assured, an ID and password should be given to the users and the user's network addresses are added to the server. Therefore, unregistered network addresses are denied, and even registered network addresses are denied when the user's ID and password do not match. Even when the system is accessed, it is not effective for all users to share the job data. Thus, in O-DYN, a method that allows a user to select another person with whom the job data is shared should be used. From the test operation of the system, it is considered that collaboration should be extended with the induction of the sharing method of the job data, versioning and working group.

Under this circumstance, the advantages of O-DYN as a web-based engineering system are as follows. First, the user does not need to install extra S/W, so expenses on S/W and installation costs can be reduced. Second, the upgrade of S/W occurs only at the server, so time and expenses on upgrades can be saved. Third, since it uses an in-house solver, it is free from copyright concerns unlike expensive commercial solvers. Fourth, the computer of each user does not have to be high-performance, and with a high-performance server, reductions on expenses and more effective usage are possible. Fifth, the work results are saved and managed in the server, so the management and authentication of the work data are improved and sharing of the work data is possible. Sixth, information can be synchronously and asynchronously shared by many users, so collaboration and concurrent engineering can be supported.

Table 1 shows the hardware and software used to develop and operate the system. The programming languages used to develop the system were C++ and JAVA. C++ was used to develop the O-DYN/Solver and all other system modules were developed with JAVA. As for web programming methods, HTML, JAVA Applet, JSP, and Servlet were used. The PRE Module and POST Module, which are the user's interfaces, were developed with JAVA Applet. The JOB Control Module and SOLVER Module except for the O-DYN/Solver were developed with JAVA Servlet and JSP.

The characteristics of the JAVA Applet used in the user's interface are as follows. JAVA Applet is attached to a web browser and used as ActiveX; it can run anywhere where the JAVA Virtual Machine (JVM) is available. It has the advantage of downloading necessary modules without the installation of extra programs. However, JAVA Applet, due to security, cannot control the local computer files and cannot communicate with systems other than the web server. Therefore, all the data of the system are managed in the web server. Through this, systematic management and authentication of data can be guaranteed. On the other hand, when data from the user's computer needs to be uploaded (user's curve data, geometric information, etc.), uploading is impossible due to JAVA Applet's security policy, and a file uploader of JSP form is used. The characteristics of the JAVA Servlet are as follows. The JAVA Servlet can perform the server's application programs that JAVA Applet cannot process, and through communication with JAVA Applet, data can be transmitted. Through session and cookie control, it can provide user's authentication. Currently, the O-DYN works in a unitary system but can later be divided into the web server system, the database management system, and the dynamics analysis system. For this, the user interface of the JAVA Applet has to overcome the security restriction. In other words, if a dynamics solver is located in a system other than the web server, the PRE Module and POST Module cannot use the database server and dynamics solver. This problem can be solved with JAVA Servlet.

Visual C++ 6.0 and JDK 1.3 were used as the programming development tools, and the system was developed based on ANSI C++ so that a generalized C++ compiler could be used instead of Visual C++. MySQL, as a relational data management system, which was used to manage user's information, user's work data (model data, result data, job history data), multibody modeling element, and log data, can be used without additional expenses. Users can share data simultaneously and it is easy to backup or manage data. The libraries used in the system are as follows: JViews, a GUI library, was used to develop the user interface of block-diagram type; and JWave, a graph library, could use IMSL, which is a numerical analysis library in JAVA version. It only transfers result data that is expressible in the user's web browser. So it can improve network load and effectiveness. It also has the advantage of providing basic functions for collaboration. OpenInventor for Java, a 3D graphics library, is a library based on OpenGL that assures more functions and more effective operating speed than OpenGL and JAVA3D. It also has the advantage of reading VRML (Virtual Reality Modeling Language) format provided by CAD and iv (OpenInventor File Format) format in geometric information.

2.3 PRE Module

The PRE Module provides the O-DYN/Modeler which is a user interface of block-diagram type.

Table 1 System development environment

OS	Windows 2000 Server
H/W	Compaq Server, 1 CPU
Web Server	MS IIS 5.0
Servlet/JSP Engine	Resin 2.1
Programming Tools	Visual C++ 6.0 with SP4, JDK 1.3
DBMS	MySQL 3.23
Library	JWAVE 3.5 (PV-Wave 7.5 included) OpenInventor for Java 3.6,JViews 5.5

The O-DYN/Modeler, with JViews, was developed into a JAVA Applet and it allows users to share the model data. In order to prevent modification of the model data at the same time, only the creator of the model data is able to modify the model data and all other sharing users are only permitted to read the data. Fig. 3 shows an example of the O-DYN/Modeler screen. The operation scenario of O-DYN/Modeler is as follows. The user, after logging in, receives the O-DYN/Modeler. The O-DYN/Modeler reads the necessary multibody element information for modeling from the DB and forms a multibody element tree on the left side of the screen. The user selects the necessary element from the element tree to model a multibody system and save the model data in the DB. To input each element parameter, a block is clicked and the input is entered, as shown in the left of Fig. 4. After validating the model, if the user selects 'Analysis' for dynamic analysis, the Job Manager is executed to prepare the dynamics simulation. The developed O-DYN/Modeler reads the necessary multibody element information from the DB for modeling so that it can be easily expanded.

2.4 JOB Control Module

The JOB Control Module is implemented with the Job Manager which controls the interactions among the PRE Module, POST Module and SOLVER Module, the User Manager, which deals with user authentication, and the Data Manager, which manages all the data with the DB, in JAVA Servlet. The Job Manager is the communication hub that redirects user requests from PRE Module, POST Module to User Manager, Data Manager, and Solver Manager and mediates each interaction.

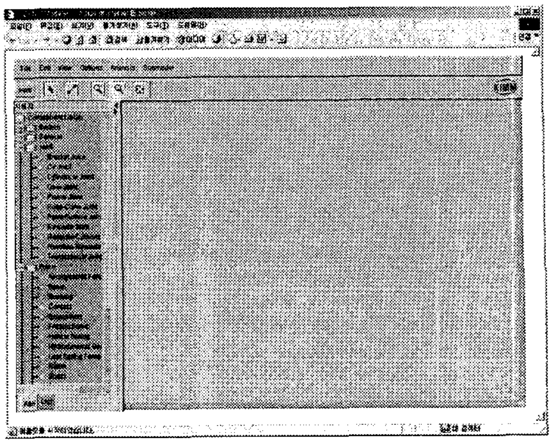

Fig. 3 O-DYN/Modeler

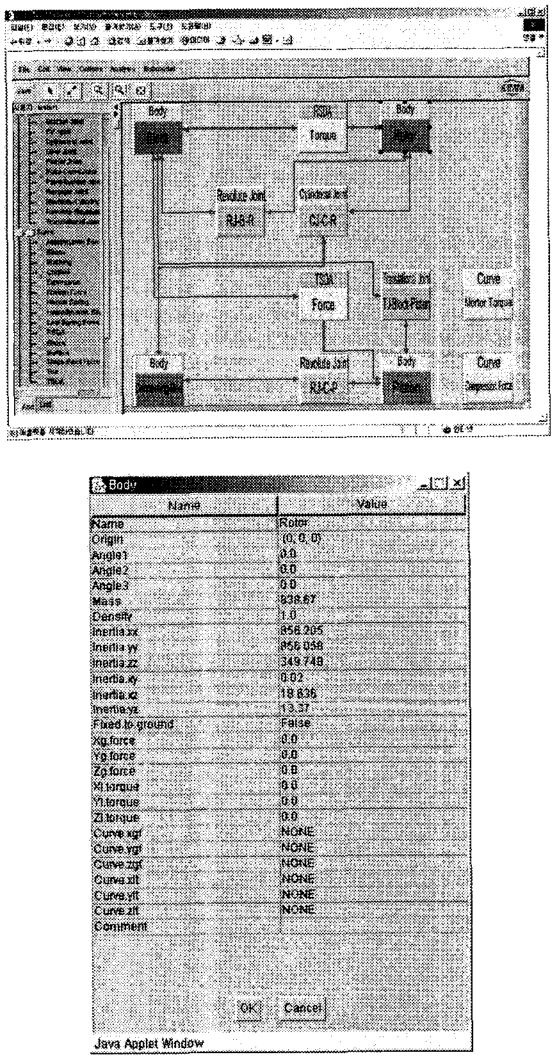

Fig. 4 Block-diagram type modeling in O-DYN/Modeler

A typical operation flow is as follows: The Job Manager receives simulation request from PRE Module via 'JAVA Applet-Servlet communication' and sends the model data saved in the PRE Module to the SOLVER Module and invokes the SOLVER Module to perform the simulation. When the SOLVER Module completes its simulation, the SOLVER Module signals the Job Manager to notify the end of simulation. The Job Manager saves the simulation results in the DB and sends a simulation-end message to the PRE Module. Now, the user can request the analysis of simulation using the POST Module. These are the control flows of the system, which are shown in Fig. 5 along with the necessary data files. To provide user authentication from the PRE Module and POST Module, the User Manager performs two levels of authentication using the user's network address and ID/password. The Data Manger performs the functions of loading and saving the data, which are used in the PRE Module, POST Module and JOB Manager, in the DB. It also offers the functions of writing the model data saved in the DB into an input file of the O-DYN/Solver and saving the analysis result files of the O-DYN/Solver in the DB.

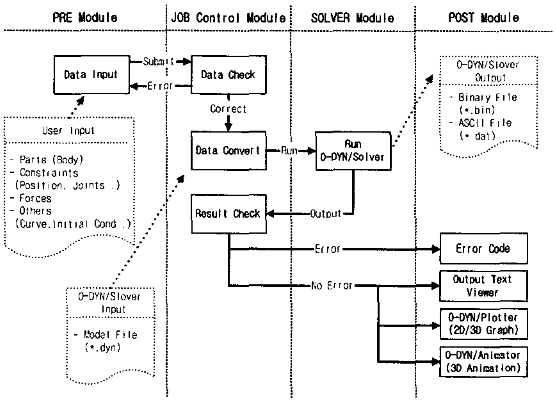

Fig. 5 Control flow of O-DYN system

2.5 SOLVER Module

The O-DYN/Solver, a dynamics solver, and Solver Manager, which runs the solver, have been developed. The O-DYN/Solver utilized object-oriented modeling and object-oriented programming for multibody dynamics simulation and was developed with C++. As shown in Table 2, it has the commonly used Body, Constraints, Joints that use the constraints, and Force elements in the libraries.

The Solver Manager runs the O-DYN/Solver in batch job format and it is implemented in JAVA Servlet. The following provides a simple description of the O-DYN/Solver.

The equations of motion of multibody dynamics systems are defined as Eq. (1).

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^{T} \\ \mathbf{\Phi}_{\mathbf{q}} & 0 \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q} \\ \boldsymbol{\gamma} \end{Bmatrix} \tag{1}$$

Here,

$\mathbf{q}$: *position vector*

$\dot{\mathbf{q}}$: *velocity vector*

$\ddot{\mathbf{q}}$: *acceleration vector*

$\mathbf{M}$: *mass matrix*

$\mathbf{\Phi}_{\mathbf{q}} \equiv \left[\partial \Phi_j / \partial q_i \right]_{m \times n}$: *constraint Jacobian matrix*

$\boldsymbol{\lambda}$: *Lagrange multiplier*

$\mathbf{Q}$: *generalized force*

$\boldsymbol{\gamma}$: *right side of constraint acceleration*

For the solution to these equations of motion of multibody dynamics systems, refer to Haug's work.[17] All commercial programs based on these equations of motion are based on procedural programming. In order to obtain the dynamic solution of a multibody system, a numerical analysis of Eq. (1) is needed. In this paper, equations of motion of multibody systems and numerical analysis were done with classes. In this paper, a separate class was defined for the derivation and numerical analysis of Eq. (1) and dynamic analysis progressed according to the order shown in Fig. 6. The class that performs numerical analysis consists of the vector, matrix and numerical methods for the derivation of equations of motion. Fig. 7 shows the flow of the O-DYN/Solver.

Table 2 O-DYN/Solver classes

Class type	Classes
Body	Rigid Body
Constraint	Ground, Position, Point, Revolute joint, Cylindrical joint, Translational joint, Bracket joint, Spherical joint, Rack-and-pinion, Gear joint, Driver
Force	Translational-spring-damper, Rotational-spring-damper,Beam, Bushing

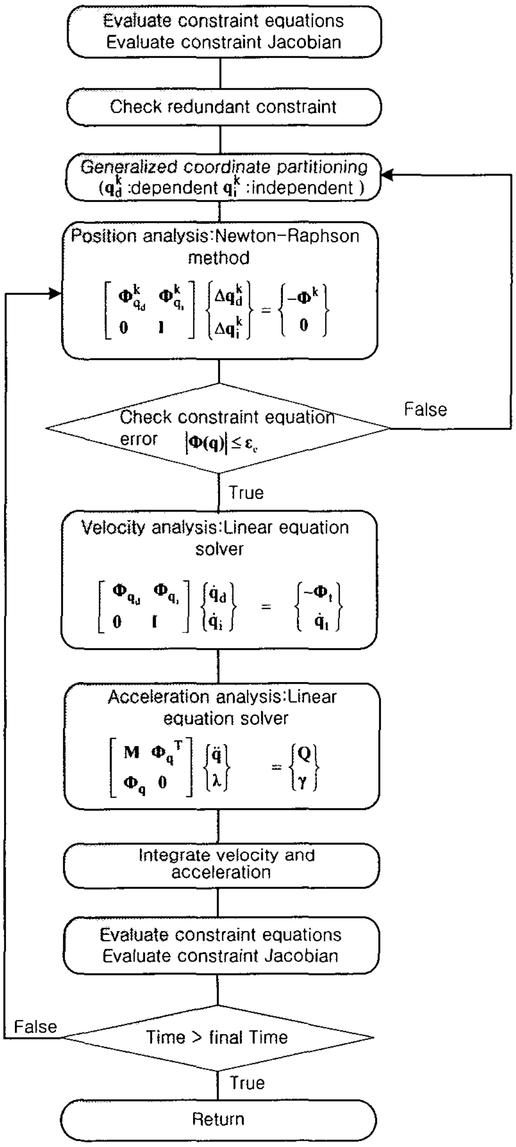

Fig. 6 O-DYN/Solver numerical analysis flow

2.6 POST Module

The Post Module provides the analysis and visualization of simulation results in 2D/3D graphs and 3D animation and is developed into the JAVA Applet. The O-DYN/Plotter, which provides the result analysis in 2D/3D graph format, uses the JWave library and makes the sharing of simulation results, collaboration and information exchange possible among users. Fig. 8 shows an example of O-DYN/Plotter screen. For collaboration and information sharing, if user A zooms on a specific area of the simulation result graph, the same result will be provided to user B who can simultaneously view the same simulation result through the internet. Furthermore, using the annotation function, discussion on the simulation result is possible. The O-DYN/Animator, which provides a motion visualization function using multibody dynamics analysis results, has been developed. The O-DYN/Animator provides analysis results, analysis and visualization with 3D animation on a web browser. The OpenInventor for Java based on OpenGL was used. Geometric information of each body uses iv or VRML format. With dynamics motion information (x, y, z, θ_x, θ_y, θ_z), which is the result of geometric information and O-DYN/Solver's simulation, 3D dynamics motion can be visualized by animation. Fig. 9 shows the system architecture of the O-DYN/Animator. The O-DYN/Animator is composed of Importer Module, Creator Module and Viewer Module. The Importer Module generates the geometric information by uploading the iv or VRML files generated from CAD. The Creator Module lets the user define and generate the geometric information. The Viewer Module, based on the geometric information, has the function of visualizing in 3D animation.

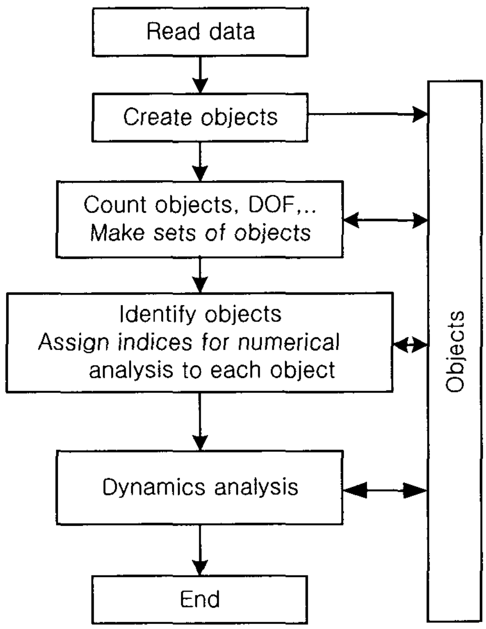

Fig. 7 O-DYN/Solver flow

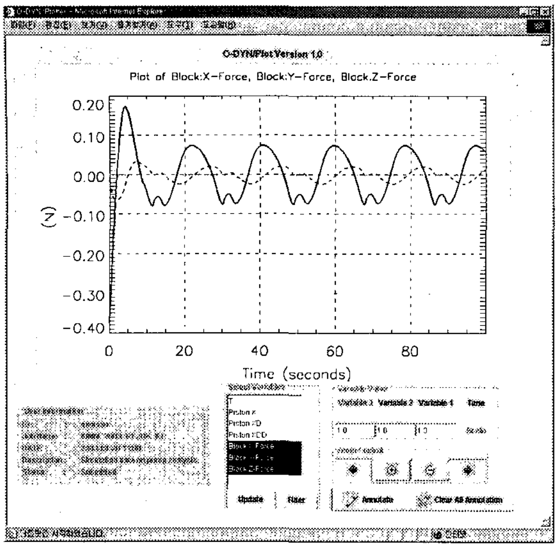

Fig. 8 O-DYN/Plotter

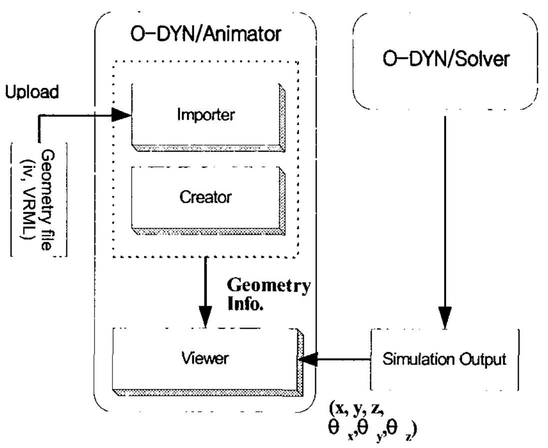

Fig. 9 O-DYN/Animator system architecture

The geometric information is made in the Importer Module and Creator Module and the simulation result files of the O-DYN/Solver. Fig. 10 shows an example of a 3D animation that used the results of the analysis.

3. Application

Using O-DYN, the web-based multibody dynamics simulation system introduced in this paper, dynamics simulation of a reciprocating compressor was performed. The reciprocating compressor, as shown in Fig. 11, is used in a lot of refrigerators. One of the most important tasks in advancing the compressor technology is reducing the vibration of the compressor.

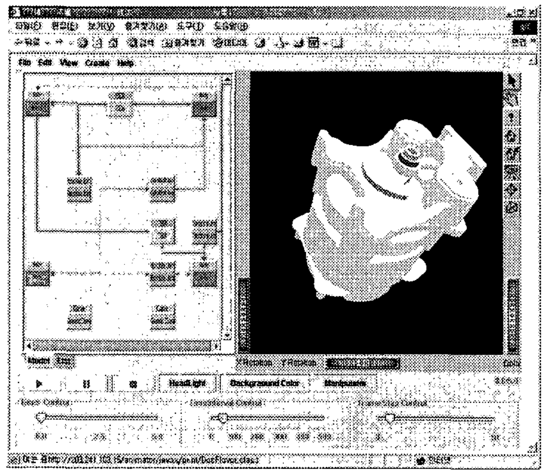

Fig. 10 O-DYN/Animator

The vibration of a compressor is mainly caused by the pressure fluctuation of a cylinder followed by motor torque vibration.[18-20] In order to achieve vibration reduction, the dynamic motion and exciting force need to be predicted under a condition where the refrigerant gas pressure and motor performance are given. In this paper, in order to predict the exciting force, the main body was fixed with the ground constraint and the reaction force was evaluated as the exciting force. Fig. 12 shows a coordinate that can be a reference of the summarized figure and the input/output of a compressor. Table 3 and Table 4 show the mass properties and multibody dynamics modeling of the compressor. The performance of compressor motor torque was obtained through experiments and is shown in Fig. 13. The motor torque is defined in terms of the angular velocity of the rotor as shown in Fig. 13. The pressure within the cylinder is a function of the rotation angle of the rotor and uses a specific curve obtained from the experiment, as seen in Fig. 14. The compressor experiences rotational resistance due to the contact of the components. This paper focuses on the gross motion, so based on the results of the experiment, it was defined as the damping due to the relative angular velocity of the rotor and the fixed block. The motor angular velocity and variation of the angular velocity were compared with the analysis to obtain the damping coefficient of 2.5 Nms/mm as the rotational resistance. Fig. 4 shows the modeling of this compressor using the O-DYN/Modeler. As in Fig. 4, the multibody modeling elements of Table 3 and Table 4 are modeled as block diagrams. By clicking each block, the input parameter of each element is set. After doing this, if an analysis is requested, the O-DYN/Solver performs the dynamic analysis. After the O-DYN/Solver performs the analysis, the O-DYN/Plotter analyzes the result. Fig. 15 shows the position of a piston and Fig. 16 shows the reaction force in x direction obtained with the O-DYN/Plotter. As described in the previous chapter, using the O-DYN/Plotter makes collaboration support possible. The position of the compressor moves periodically and the reaction force has the maximum value when the rotor is at 0° and 180°. It can be seen that the maximum value is about 70N. This reaction force will work as the exciting force of the compressor. This paper focuses on introducing a system that simply explains the result of the dynamic analysis of the compressor. Fig. 10 shows the visualization of motion performed by the O-DYN/Animator. Here, the O-DYN/Plotter and O-DYN/Animator make collaboration possible since several users can have access to the same result at the same time.

Table 3 Inertia properties of the compressor

Body	Mass(kg)	I_{xx} (kgmm^2)	I_{yy}	I_{zz}
Block	5.0	20.0	0.05	15.02
Rotor	0.04	500.0	500.0	30.0
Connecting-Rod	0.03	1.1	10.0	3.5
Piston	0.03	3.0	3.5	4.0

Table 4 Dynamic model of the compressor

Bodies	Block, Rotor, Connecting-rod, Piston
Translational joint	Block-Piston
Revolute joint	Block-Rotor
Revolute joint	Connecting-rod-Piston
Cylindrical joint	Rotor-Connecting-rod
Motor torque	Fig.13
Compression Pressure	Fig. 14
Rolling resistance	Block-Rotor (2.5 Nms/mm)

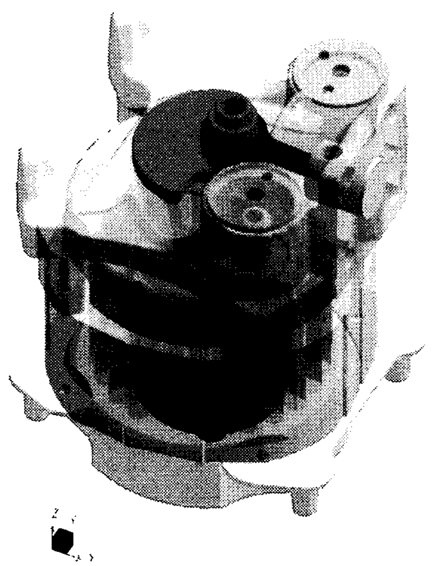

Fig. 11 Reciprocating compressor

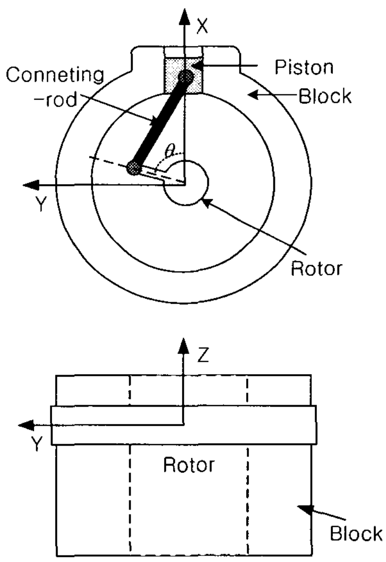

Fig. 12 Schematic diagram for the compressor

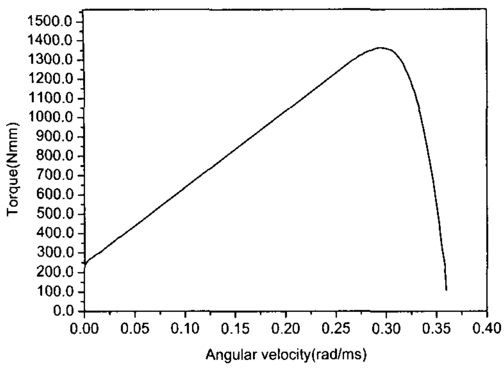

Fig. 13 Characteristics of motor

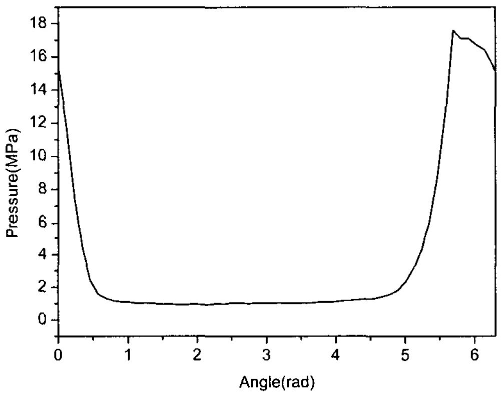

Fig. 14 Pressure versus rotor angle

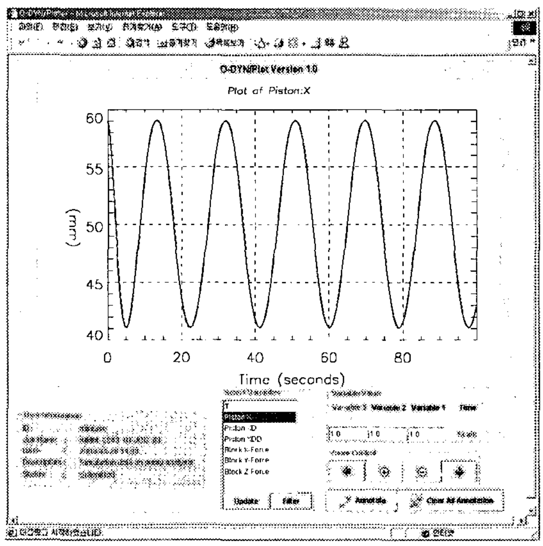

Fig. 15 x position of piston versus time

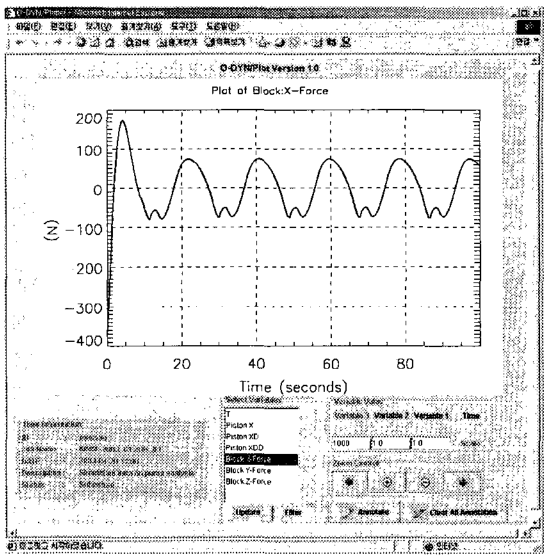

Fig. 16 x reaction force on block at origin

4. Conclusion

In this paper, a web-based, general-purpose multibody dynamics simulation system was introduced. All user interfaces of the simulation system were implemented through a web browser and the internet. All numeric calculations needed for the dynamics simulation were done in the web server. All user interfaces in the web were developed based on JAVA Applet. In conclusion, the user can perform the dynamic simulation of a multibody system through the internet and a web browser regardless of the user's computer environment and location. Furthermore, simultaneous sharing of information, implementation of collaboration and concurrent engineering are made possible, as well as the systematic management of information using the database.

References

1. Miller, J.A., Fishwick, P. A., Taylor, S. J. E., Benjamin, P. and Szymanski, B., Research and Commercial Opportunities in Web-Based Simulation, Simulation Practice and Theory, pp. 55-72, 2001.
2. Stormont, D., "Web-Based Mobile Robot Simulator," Proceedings of the National Conference on Artificial Intelligence, pp. 937-938, 1999.
3. Zhou, H., Zhang, Y. and Li, D., "A Remote Injection

Moulding Simulation System Based on the Internet and the Web," International Journal of Advanced Manufacturing Technology, Vol. 19, Issue. 10, pp. 722-726, 2002.

4. Zeng, H., Alarcon, V. J., Kingery, W., Zhu, J. and Magdi Selim, H., "A Web-Based Simulation System for Transport and Retention of Dissolved Contaminants in Soil," Computers and Electronics in Agriculture, Vol. 33, Issue. 2, pp. 105-120, 2002.
5. Kumara, S. R. T., Lee, Y. H., Tang, K., Dodd, C., Tew, J. and Yee, S. T., "Simulation Anywhere Any Time : Web-Based Simulation Implementation for Evaluating Order-to-Delivery Systems and Processes," Winter Simulation Conference Proceedings, Vol. 2, pp. 1251-1259, 2002.
6. Henriksen, J. O., Lorenz, P., Hanisch, A., Osterburg, S. and Schriber, T. J., "Web-Based Simulation Center : Professional Support for Simulation Projects," Winter Simulation Conference Proceedings, Vol. 1, pp. 807-815, 2002.
7. Callaghan, M. J., Harkin, J., McGinnity, T. M. and Maguire. L. P., "An Internet-Based Methodology for Remotely Accessed Embedded Systems," Proceedings of the IEEE International Conference on Sysyems, Man and Cybernetics, Vol. 6, pp. 158-163, 2002.
8. http://www.lmsintl.com/
9. http://www.adams.com/
10. http://www.functionbay.com/
11. http://www.simpack.de/
12. http://java.sun.com/
13. http://www.tgs.com/
14. http://www.vni.com/
15. http://www.ilog.com/
16. http://www.caucho.com/resin/
17. Haug, E. J., Computer Aided Kinematics and Dynamics of Mechanical System, Allyn and Bacon, USA, 1989.
18. Padhy, S. K., "On the Dynamics of a Rotary Compressor: Part 1-Mathmatical Modeling," Advances in Design Automation, ASME, Vol. 1, pp. 207-217. 1993.
19. Padhy, S. K., "On the Dynamics of a Rotary Compressor: Part 2-Experimental Validation and Sensitivity Analysis," Advances in Design Automation, ASME, Vol. 1, pp. 219-227, 1993.
20. Yangisawa, T., Mori, M., Shimizu, T. and Ogi, T., "Vibration of a Rolling Piston Type Rotary Compressor," International Journal of Refrigeration, Vol. 7, No. 4, pp. 237-244, 1985.