A Web-based agile system for rolling bearing design

P.Y. Pan
Department of Engineering, The University of Liverpool, Liverpool, UK
K. Cheng
School of Engineering, Leeds Metropolitan University, Leeds, UK
D.K. Harrison
Department of Engineering, Glasgow Caledonian University, Glasgow, UK

Keywords
Artificial intelligence, Agile production, Internet, Operating systems

Abstract
Rolling bearings are basic mechanical elements for supporting rotational and reciprocating motion in engineering products. Many approaches have been proposed by researchers in order to facilitate the procedure of bearing design, but mainly concentrating on the development of PC-based systems or relative software packages.The authors present a novel approach to implementing a design support system for rolling bearings based on the World Wide Web so as to achieve agility in rolling bearing design. The system was developed based on the philosophy of agile manufacturing by integrating various information resources, artificial intelligence, and Web technology with expertise. This Web-based distributed system will be globally accessible by the user on the Internet and can be automatically sized according to the demand of application requirements. The system implementation issues are discussed in detail in this paper.

Received July 2000
Revised November 2001
Accepted March 2002

Integrated Manufacturing Systems
14/6 [2003] 518-529

[ISSN 0957-6061]
[DOI 10.1108/09576060310491388]

1 Introduction

Rolling bearings are basic mechanical elements for supporting rotational and reciprocating motion in engineering products. The procedure for the selection and design of journal bearings is very complicated and relies heavily on specialist knowledge and practical experience. Current methods of selection and design of rolling bearings can be loosely classified as manual or computer aided. The manual method normally uses manufacturers' catalogues (book type) or electronic catalogues (CD-ROM media) and many complicated factors have to be considered by the designer in the decision-making process toward the final selection and design outcome. Obviously it is very tedious, ineffective and even difficult to make the trade-off among these factors. Computer aided methods are usually implemented in PC-based algorithm systems or expert systems. These computer programs provide automated design procedures or rule-based expertise. However, a PC-based design support system can only be used by individuals and will not be able to meet the requirements of severe competition in this area.

With the increasing competition in the global market, bearing companies have to produce high-quality, competitively priced products quickly and efficiently to achieve the agility which is accomplished by integrating all of the resources available to a company – technological and human; internal and external – into a coordinated interdependent system capable of achieving short product development cycle times and responding quickly to sudden market opportunities.

With the great progress of Internet and World Wide Web technologies, various attempts have been made to implement Web-based supporting systems for different applications in the areas of design and manufacturing. For example, Chui and Wright (1999) have developed an Internet based multimedia educational tool for the design of simple mechanical parts. Reed and Afjeh (1998) have used a Web-based interactive engineering tool for engineering simulation and teaching and learning purposes. Many researchers and practitioners (Iuliano and Jones, 1996; Hardwick *et al.*, 1996; Erkes *et al.*, 1996; Yu and Cox, 1997; Booth, 1996) have also made other efforts in Web-based engineering applications. The results of all these attempts have illustrated the great potential of an Internet-based virtual manufacturing enterprise, with Web-based design and manufacturing support systems in particular.

There exists a notable fact, however, that although the Web has made a variety of information accessible to multiple users simultaneously, it still does not offer any simple way to use stand-alone applications (Tomarchio and Vita, 1998), particularly those computationally and graphically intensive applications most required at various levels in a rolling bearing design and manufacturing environment. There are many factors that affect the implementation of Web-based standalone support systems for rolling bearing design. The obstacles to the application are also from the heterogeneous design environments existing in a company or among enterprises. To date, a number of methods, from the programming point of view, for dealing with heterogeneous environments have been proposed by researchers, including message-passing libraries, remote procedure call and heterogeneous distributed shared memory systems. These methods, however, failed to

The Emerald Research Register for this journal is available at
http://www.emeraldinsight.com/researchregister

The current issue and full text archive of this journal is available at
http://www.emeraldinsight.com/0957-6061.htm

hide the hardware differences and the distributed nature of a system from the users or code generators.

This paper describes a new approach to implementing a Web-based agile system for supporting rolling bearing design in heterogeneous design environments. Its implementation issues are discussed and explored in detail. The proposed approach aims at enhancing the design agility and manufacturing responsiveness of an enterprise with the support of Internet and Java-based systems. By using this approach, rolling bearing design and manufacturing environments in heterogeneous platforms could be effectively managed and maintained. The work presented is a part of a joint research project being undertaken at Leeds Metropolitan University and Glasgow Caledonian University, UK.

2 Proposed system architecture based on Java programming

2.1 Why is this system based on Java?

Although the Internet provides a platform for information transfers between different clients, the requirements of rolling bearing design are far beyond that. For example, graphical interfaces, product data management and interactive intelligent computing support are more necessary than just plain textual query or data searching. The implementation of a Web-based agile system for rolling bearing design needs agile information technologies. Java is a tool and programming environment that supports the distributed processing for designers and manufacturers via the Internet/intranet. Java has the following attractive features that are most suitable for integrating heterogeneous design environments:

- Visual input could be used for engineering design/manufacturing process because Java contains libraries for developing GUI.
- Java supports embedded applets and thus allows designers/users to remotely access embedded applications such as bearing selection codes and product data accumulated in other companies via the Internet.
- Java is platform independent. Therefore only a single version of the source code needs to be maintained for a program execution on different operating systems.
- Java's built-in TCP/IP, HTTP and FTP make it suitable for Internet deployment.
- Multiple designing/machining systems can be served simultaneously via the Internet because Java has the capability of multithreads.
- Design/manufacturing operations for complex bearing products can be decomposed into small pieces based on Java's object-oriented capabilities. Successful design codes can be reused.
- As a revolutionary strategy, Java's server-based application deployment technology can lay the foundation of a network computing architecture that, in the long run, can dramatically cut an organisation's total cost of ownership of client platforms.

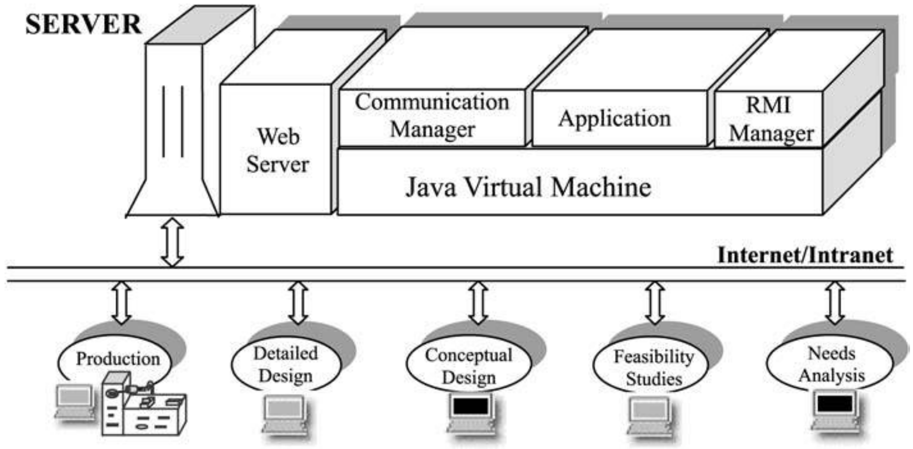
Java is both a platform-independent and application development platform which uses the Web as a deployment medium. Its platform-independent and server-based architecture enables designers to deploy applications across virtual organisations. That includes business partners, whose computing environments may be completely different, and internal users, whose standalone computers or dedicated data entry applications can be upgraded with Java-based and graphical applications. As a very promising programming language, almost every major computer and operating system provider, such as Windows, Unix and Macintosh, supports Java (Pan *et al.*, 1999).

2.2 Engineering design support systems

The design and manufacturing of new bearing products frequently requires the talents of many specialists, various techniques, specialised software tools and appropriate solutions. A bearing design process includes several phases, starting from marketing analysis, feasibility study, conceptual design, detailed design, production and final product distribution and consumption (Cheng and Rowe, 1996). During the entire process, the operation at each design phase might be carried out in a corresponding department in co-operation with others. Because of the heterogeneous environments among them, the design operations such as product modelling, simulation and transferring design data from one design phase to another could be very time-consuming. The transferred data could be misunderstood. Information resources and design tools could not be easily shared.

Figure 1 shows a proposed design scenario with the support of design systems based on Java programming. Java is a promising language for developing Web-based systems. The most attractive feature of Java is its platform independence. Using Java, only a

Figure 1
A design scenario: client-server architecture



single version of the source code needs to be maintained for a program execution on different operating systems (Pan *et al.*, 1999). This scenario is a client-server architecture, e.g. a central Web server co-ordinates access to local or distant design data or codes resources. The individual design phase located in different design groups or departments is a client to the server. The only requirement for a client is to install a Java-enabled Web browser. The core server site architecture, however, needs a few functional modules:

- The Web browser is an HTTP server that replies to the requests issued by the design clients.
- The Java virtual machine is the Java execution environment and the runtime environment.
- The application is the server-side codes to be made available on the design platform.
- The communication manager runs the communication with the design client.
- The remote method invocation (RMI) manager controls the access to the remote design facilities.

With the Java-enabled Web browser, designers, as clients, will be able to interface to server-side user accounts and run the design platform in a convenient manner. Once they supply the required user name and password, they can log into their accounts and use the design facilities such as database searching, spreadsheets, CAD tools, open computing, from the same interface, the Web browser. Clearly, this architecture is ideal for supporting rolling bearing design practices in current manufacturing industry, e.g. collaborative design activities of designers distributed at different locations.

3 Implementation issues for the Web-based system

3.1 Information resources integration through three-tier architecture

The purpose of developing this system is to provide a comprehensive design support environment for the users and in order to achieve the bearing design and manufacturing agility via the speed and accessibility of Internet technology. For the system, linking with the large bearing manufacturers' existing electronic catalogues and product data database is one of the important issues in its development. Actually, these catalogues and databases could be distributed on different servers anywhere over the Web. Such a system is therefore in nature a distributed system and can be resized in the light of an application's requirements.

Currently, all the links on the Web are based on their corresponding Web address or universal resource locator (URL). HTML forms and the CGI (common gateway interface) protocols have been used to retrieve and view information on Web servers based on their URLs. All the information submitted to and obtained from a server is through an HTML form. Therefore, for a simple information searching requirement, e.g. input parameters are textual, this mechanism can be used to:

- provide an interface to a searchable database, and return the result of a search to the client in HTML pages;
- process the information submitted in a query form and return the state-of-the-art result from a manufacturer's electronic catalogue or product database; and
- process inputs from a guest book, including the guest's username, password

and ID number, etc. for logging into the system, and return a confirmation message to the client.

However, the information resources required by the Web-based design support system are more than just textual. Design information and data include every aspect of a bearing product and exist in different media types such as spreadsheets, CAD drawings, and photographs, but not just in string. The authors therefore proposed a three-tier architecture as shown in Figure 2 for searching a design database or manufacturing database via a RMI manager. Through contacting a Java Database connectivity (JDBC) package, the RMI manager will be able to provide designers with a convenient information-searching mechanism on objects they require. JDBC defines a set of application programming interface (API) objects and methods to interact with an underlying database. Through RMI, a Java program first opens a connection via JDBC to a database, makes a statement object, then passes structured query language (SQL) statements to the underlying DBMS, and finally retrieves the searching results as well as design information about the result sets. The major advantage of using RMI technology for database searching in an engineering support system over CGI is that RMI can pass and return any design or manufacturing objects developed in Java instead of just textual in CGI (Yu and Chen, 1998).

3.2 Remote access of design packages via RMI mechanism

As mentioned earlier, this system provides several design functional modules for users to use. Some modules are developed by the authors, but some functional modules are implemented through integration with existing application packages and databases. The authors hope that this Web-based design support system can not only share resources residing in other systems, such as that in a manufacturer's catalogue, but also finally allow individual functional modules, such as an intelligent bearing selection module, to be remotely invoked or integrated with the heterogeneous design environment under certain conditions. So the RMI mechanism is used in the development of this system. RMI is a technique for using objects which may reside on different platforms anywhere on the Internet.

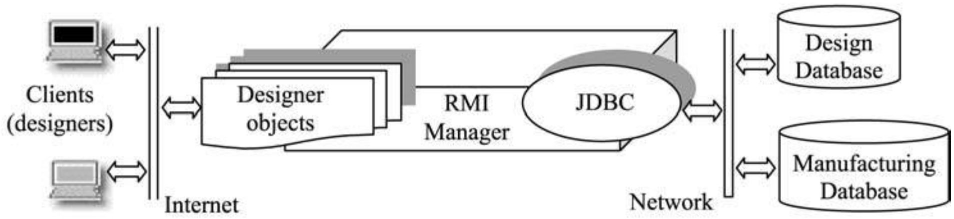
There are a few steps to implementing a remote object:

- Define an interface that extends the remote interface. Each method of this new interface must declare that it will throw a RemoteException.
- Define a class that implements the design interface. This class should fulfil the requirements for making the new class a remote object. Currently the class UnicastRemoteServer is available for implementing this function.
- Generate the stubs and skeletons that are needed for providing the RMI-application design interface to the Web-based system. Stubs are responsible for initiating the call to a remote design object and handling other interactions with the remote objects. Stubs also marshall and unmarshall arguments used to communicate with the server. Skeleton is a server side entity responsible for dispatching calls to the actual design object, marshalling and unmarshalling design arguments.
- Create a client program that will make RMI calls to the server.
- Start the registry and run the remote server and client.

3.3 Open computing and programming

Web pages programmed using HTML can only transfer messages such as text and images, i.e. static Web pages, but cannot implement highly interactive and executable ability on the Internet. One of the major objectives of developing this Web-based intelligent design support system is to make the system globally accessible and executable by users on different platforms. The system aims at not only providing bearing design data and information, but more important, at being a design computing environment for

Figure 2
The three-tier architecture of integrating information resources



the selection of bearing type and configuration, bearing parameters optimisation, and the simulation of bearing lubrication and mounting details, etc. Platform independent and open computing are very useful features for an engineering system to be broadly applied. JavaScript and Java programming are therefore extensively used in developing this Web-based design support system. Client-server technology is another indispensable technique for implementing the features and functions described above.

3.3.1 Using JavaScript

JavaScript is a new scripting language based on the Java programming language initially developed by Sun Microsystems. This powerful up-and-coming scripting language is being further developed by Netscape Communications Corporation. Its products, Netscape Navigator 2.0 or later versions, support JavaScript.

JavaScript is a cross-platform, object-based scripting language for client and server application. JavaScript enables the creation of applications that run over the Internet. For example, client applications run in a browser such as Netscape Navigator, and server applications run on a server such as Netscape Enterprise Server (Stanek, 1999), which all use JavaScript programming. JavaScript can recognise and respond to mouse clicks, form inputs, and page navigation. With JavaScript, not only interframe communications, but also client-side error checking, the ability to launch multiple frame-targeted executables simultaneously, can be implemented. Using JavaScript makes creating intelligent dynamic HTML pages possible. For instance, with JavaScript programming, a user's inputs can be processed and persistent data be maintained using special objects, files, and relational databases. JavaScript is normally solely used on the client (in Navigator or another Web browser). In the Web-based design support system, however, there are more complex needs such as communicating with a relational database and providing continuity of information from one module to another in the design exercise, etc. For these more demanding situations, server-side capabilities should be supported and server-side JavaScript (SSJS) should therefore be used.

Server-side JavaScript runs as a part of a Netscape Web server with access to database servers or other server-based applications. Figure 3 shows in more detail how the server-side JavaScript environment and associated applications fit into the Netscape Web server architecture.

3.3.1.1. Advantages of using JavaScript against CGI in this system. CGI technology has been widely used in developing Web-based applications and server-side programming. It has some attractive features:
- it is very portable – can be run on any CGI-enabled Web server; and
- it can be programmed in a variety of languages such as Perl, C/C++, Java, TCL.

However, the following disadvantages affect its feasibility in this Web-based support system:
- its speed is slow because a CGI program launches a new process every time it is executed. Without doubt this will result in a heavier burden on servers and affect the communication speed over the Internet;
- it makes state management difficult to implement and therefore it will definitely cause more time on the system's development; and
- it makes database access or electronic catalogue searching available only via separate libraries. Consequently, the development and running environment of the system will become complex.

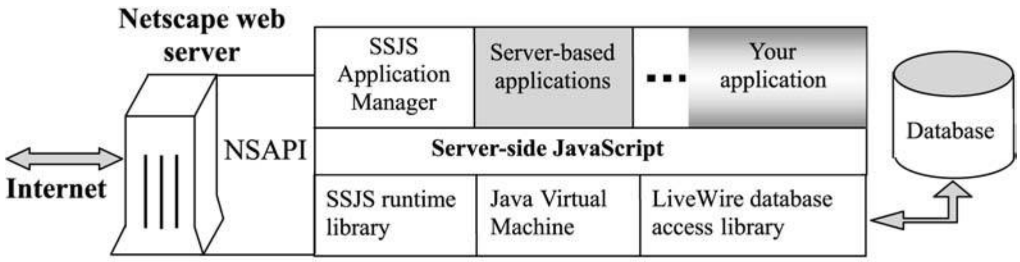
To develop an interactive support system running on the Internet, interactive or communication speed between the client and the system is an essential factor. The Internet is globally shared by millions of users at the same time and "traffic congestion" often happens, especially during the "rush hour". It is not unusual sometimes to wait half an hour or even hours to retrieve information from a server on the Internet. Therefore more consideration should be given on how to enable Web-based application systems to run as fast as possible.

SSJS runs faster than CGIs. In internal Netscape tests, SSJS was three times faster than Perl CGIs for database access. Compared with CGI, SSJS has the following advantages in integrating databases or other server-side applications into this Web-based support system (Stanek, 1999; Netscape, 1999):
- provides state management capabilities and therefore makes it much easier to implement the management;
- supports built-in database access;
- allows code reuse between client and server applications, and will be able to extend functionality using Java via LiveConnect; and
- runs three times faster than CGI and scales much better than CGI.

3.3.2 Using Java open computing

Java is an object-oriented programming language based on C++, which enables users

Figure 3
Server-side JavaScript in the Netscape server environment



to create standalone applications or "applets" that run on virtually any computer on the Internet (Weber, 1996). Java offers many essential object-oriented features such as data abstraction, encapsulation, polymorphism and inheritance. Object-oriented programming simplifies the development of application systems. Using object-oriented programming, an engineering system is decomposed into separated constituent objects, which are then mapped to the computational domain. This approach follows very well with the decomposition requirement of the Web-based bearing intelligent design support system.

In this design support system, all the functional modules are decomposed into individual objects. For example, in the bearing selection module which is implemented with neural networks and fuzzy logic technologies, each of the bearing selection engine components is represented as an object. Each component's characteristics, such as its data and computing methods, are encapsulated within each object respectively. By organically combining specified types of component objects, such as the fuzzyfication objects for fuzzifying the inputs of bearing operating factors, neural networks objects for the selection computation, the module's kernel, i.e. the bearing fuzzy-neural selection engine, can be easily modelled and the engine can easily adapt to the selection inputs from different users.

3.3.3 The application of client-server technology

The major benefit of client-server technology is that the processing load can be shared between the client and the server. A client can be any programs, such as GUI applications, Telnet, that request services from a server application. Examples of server applications include database servers, application servers, and communication (ftp, telnet) servers (Cheng *et al.*, 1997).

Like JavaScript, Java programming provides not only client-side ability, but also classes for server-side processing. For developing client-server applications, using the java.net package and Java Server API are inevitable. The former is the package that makes Java a network-based language. It provides the capability to communicate with remote resources by creating or connecting to sockets or using URLs. The latter offers uniform and consistent access to the server and administrative system resources.

Low-level network communications of a client-server application are normally performed using sockets. For example, the open fuzzy-neural computing for bearing parameters residing on a server can be accessed by and communicated with the client in this way. Another way to deal with communications at a higher level is using RMI. The mechanism of RMI is used in the Web-based system for database searching and design packages' invocation. Through both socket and RMI, the communications between the client and the server could be effectively and efficiently implemented. The following socket related classes are supported by Java.net:

- *ServerSocket.* This class represents the socket at the server side. Methods in this class provide facilities to create and close socket connections and accept connections from clients.
- *SocketImpl.* This class is an abstract interface used to define different socket implementations.
- *SocketImplFactory.* This class is an interface provided for different socket implementations. The SocketImplFactory available in Java is PlainSocketImpl which makes use of the TCP protocol and provides for reliable data transfer.
- *Socket.* This class represents the socket at the client side. A socket can be created and closed by the methods in this class. The input and the output streams associated with this socket are secured by these methods as well.

Because of the daily heavy burden and working load on the Web, more consideration should be given to relieve the burden as much as possible in developing a Web-based support system. For developing this Web-based bearing design support system, the following measures were carefully planned and balanced among using client/server-side scripts, and Java applets and Java applications in particular:

- using client-side scripts and Java applets to reduce the load on the server and the network backbone by allowing the client to process inputs locally;
- using client-side scripts and Java applets to dramatically improve performance for the client because the client does not have to wait for responses from the server and can process inputs immediately; and
- scripts and Java applets to simplify the design and implementation process by reducing the interaction with a single source document.

In the bearing selection module of the system, for example, the object of neural networks in the bearing selection engine is trained with the data from hundreds of successful bearing selection cases. The bearing selection engine and its inherent neural computing object could be simultaneously accessed by multiple potential users on the Internet. So there is the need to code this object as a Java application rather than an applet, and to train it and then install it as a server-side application. As for the fuzzyfication object, which is also a part of the bearing selection engine, different users may have different inputs concerning bearing operating requirements. The fuzzyfication object should be coded in applets or in client-side scripts and only be used at the client-side to achieve the benefits listed above.

3.4 User interface design and HCI aspects

The system user interface development includes interaction development and interface software development. Interaction development is concerned with how the user interface works, its "look and feel" and behaviour in response to what a user sees and hears and does while interacting with the system. The interface software development is the means for implementing the code that instantiates the interaction components (Hix and Hartson, 1993).

Design of the Web-based design support system's user interface was considered from functional, aesthetic and structural perspectives. Each of the perspectives affects the quality of the overall user interface design. The following discussions focus on some principles applied in developing the system user interface.

Direct manipulation

From the psychological point of view, system users expect their physical manipulations to have a response from the system. For example, when an iconic button is pressed, users like to see the corresponding process, e.g. the button being graphically pressed down on the screen. Combination of a physical activity and feedback reinforces the sense of the manipulation, which may improve a user's efficiency and effectiveness in using the system. System users want topics of interest to be highlighted in a system. The users want clues which tell them that a particular command is being carried out, or if not, they want to be told why not and what they can do instead. In this Web-based system developed, for example, a button manipulated by click is highlighted by colour change, hand pointer and corresponding information displayed in a status window simultaneously. By means of direct manipulation, therefore, the system users could be provided with a feeling of confidence that they are in charge rather than being driven by the system.

Consistency

There should always be a coherent way for the user to implement actions because users want to rely on familiar and straightforward ways to interact with or manipulate the system. For example, icon functions should be consistent in appearance. The style of the system screen layout should be consistent, otherwise the displays may be confusing and distracting. In the Web-based design support system, functional buttons in different modules and even different pages within a same module have a consistent appearance and the same highlighted colour when clicked. Buttons for changing windows, going to the home page and going back to each module front page keep the same animation style.

Simplicity

Designing a successful Web-based highly interactive system is actually very complex in terms of the cognition and human-computer interaction (HCI) issues involved. It is important to design the system for the user's easy usage and to further simplify the user's operation tasks. For example, some design tasks can possibly be simplified by using icon-driven, word and sound interpretation, animation, action effects, and other HCI means that are natural to the user on specific design tasks. A complex design task can be broken down into several simple subtasks by modelling the system and staging its built-in design procedures.

4 A developed Web-based agile system for rolling bearing design

Based on the proposed architecture in section 2 and implementation issues investigated in section 3, a Web-based bearing design support system has been developed by the authors. Figure 4 shows the front page of the system. The system includes seven functional modules which are electronic catalogue, intelligent selection, mounting details, sealing devices, lubrication, manufacturing database and design module. The system aims to provide comprehensive, responsive and rapid bearing selection and design support for the large engineering community by taking advantage of the speed, ease of access and low cost of Web-based techniques. The system is developed using HTML, JavaScript, and Java programming combined with AI techniques. All modules in the system are hyperlinked in an organic and associative way. A system user can enter any module by clicking the module icon on the system front page. The modular structure allows the system to be easily modified, updated, and is user-friendly.

The mounting details, sealing devices and lubrication modules include comprehensive application examples and case studies, and their design history is presented in multimedia style. The users can go through them and may get some ideas as they are needed. Manufacturing database and electronic catalogue modules can be searched and even modified by users. Inside the intelligent selection module, there is a selection inference engine based on neural networks and fuzzy logic. It is the kernel of the intelligent selection module. The module has the ability to automatically process the user selection requirements (inputs) which may be well defined and with uncertainty, and generate the corresponding selection results based on the inputs. Figure 5 shows the architecture framework of the system. Some screen copies of the system are shown in Figure 6.

Inside the mounting details module, for instance, the module incorporates mounting techniques with multimedia style Web pages including 2D and 3D animations and simulations. The module provides detailed design information on the four following bearing mounting methods for supporting rotational machinery:

1 *Mounting by fixed and floating bearings.* Mounting by fixed and floating bearings are the most common methods used to fix

Figure 4
A Web-based design support system for rolling bearings

Figure 5
The architecture framework of the Web-based system

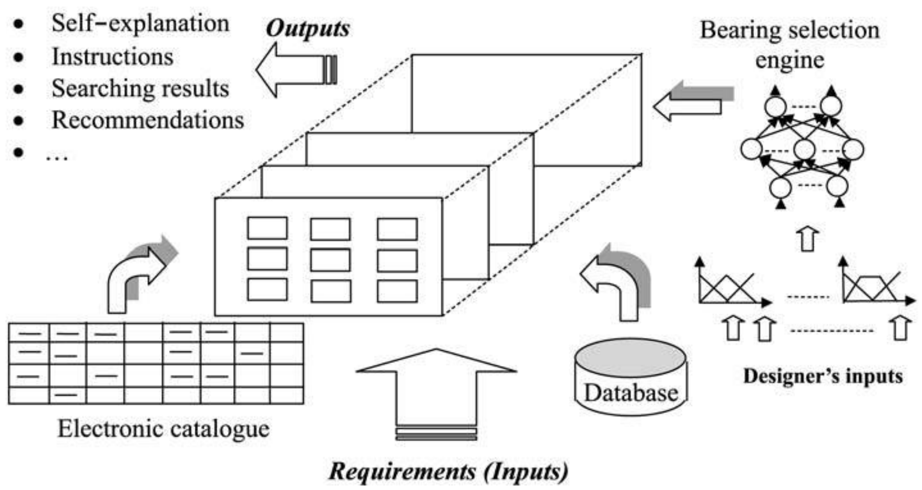
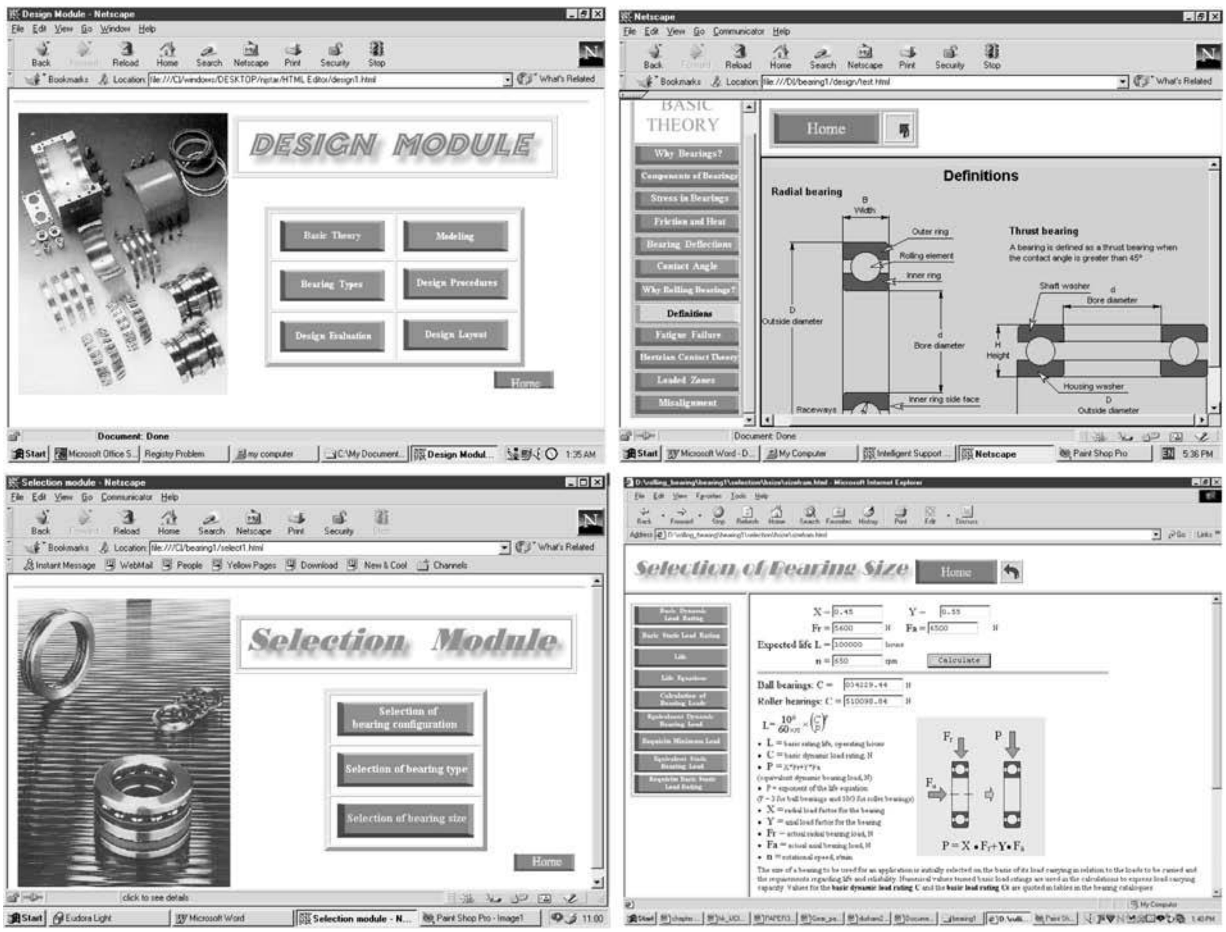


Figure 6
Some screen copies from the system



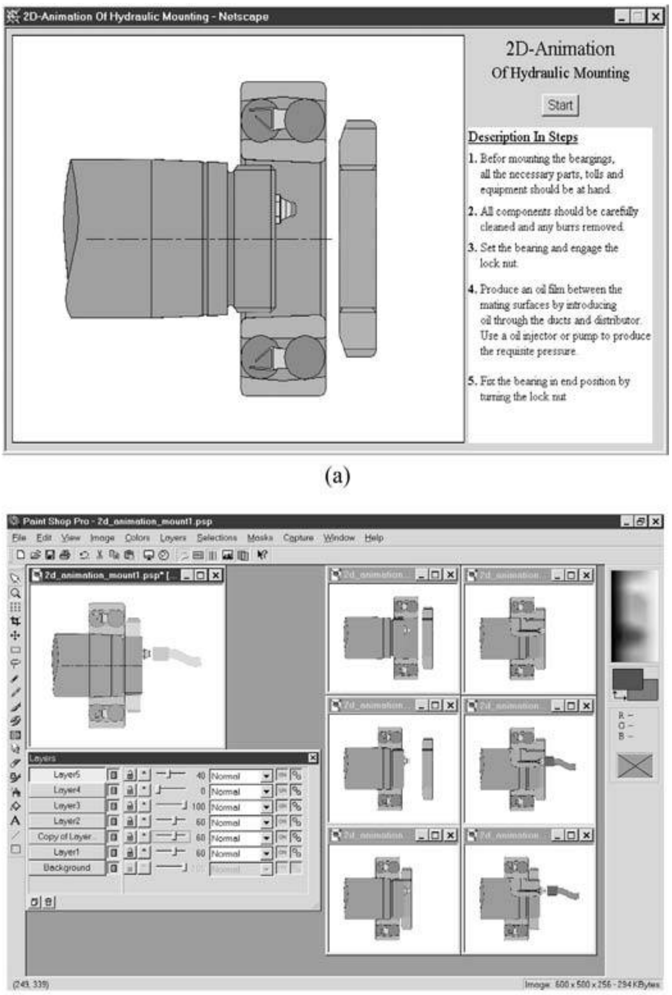
a rotating mechanical part into the bearing housing. The fixed bearing takes over its radial load proportion and at the same time all occurring axial forces in both directions. Locating bearings are fixed on the shaft as well as in the housing against side movements. Radial bearings that can take up radial and axial loads are suitable for fixed bearings. Also combinations with radial bearings and axial bearings can be used. If the shaft is fixed in the housing by more than two bearings, then only one of the rolling bearings may be designated as the fixed bearing. All other rolling bearings have to be designated as floating bearings. The floating bearing takes over only the radial load proportion. Floating bearings have to allow axial movements in order to avoid a mutual tension between the rolling bearings. Floating bearings either allow axial movement in the bearing itself (e.g. cylindrical roller bearings) or one of the races is a sliding fit either on the housing or the shaft.

2 *Support mounting (cross-locating)*. Supporting mounting is presented, if each of the two bearings is guiding the shaft in one of the axial directions. This kind of mounting is mainly suitable for short shaft, e.g. for the mounting of a pinion shaft. All rolling bearings are usable for the supporting mounting, which will be able to receive forces in one axial direction. In the use of tapered roller bearings or single-row angular contact ball bearings, pre-loading is necessary in certain cases.
3 *Floating mounting*. Both ball and roller bearings are mounted as floating bearings with small side clearance through which a limited movement of the shaft in the longitudinal direction is possible. The axial clearance is limited on 0.5mm up to 1mm. This kind of mounting bearings is used to reduce the manufacturing costs by machining the housing bore in one operation.
4 *Preload mounting*. Preload mounting (normally designated with angular contact bearings) offers possibilities for an exact adjustment of a clearance or a preloading. Normally the operational clearance should be positive, i.e. the bearing should have a minor clearance during the operating condition. But in some cases it is necessary to have a preload, i.e. a negative operational clearance (e.g. work spindle bearings on machine tools and bearing arrangements for oscillatory movement, etc.). The main effects of bearing preload are as follows:
   - to increase the stiffness;
   - to decrease the noise developed;
   - to increase the accuracy of shaft guidance; and
   - to enhance the service life.

These four methods mentioned above are clearly described and comprehensively illustrated in this module through the media of text, graphics, 2D and 3D Web-based animations and simulations. With 2D and 3D animations and simulations, the details of mounting and dismounting can be illustrated physically and realistically. The user will therefore be able to easily obtain the knowledge of mounting details and to understand in-depth various mounting and dismounting operations.

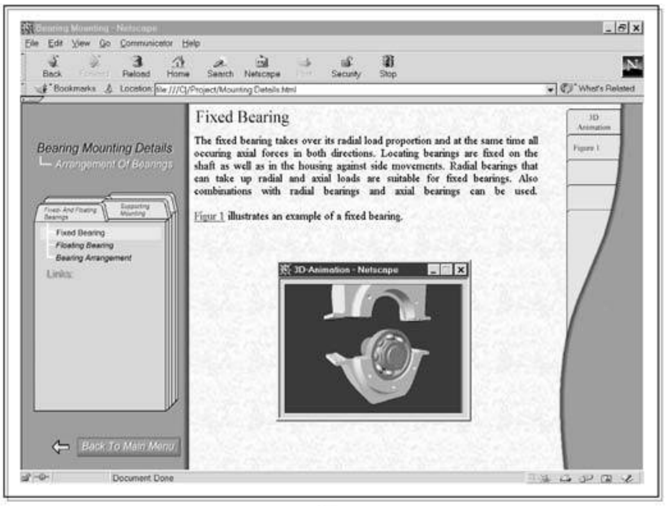
Figure 7 shows 2D Web-based animations and simulations screen copied from the module. Figure 8 shows 3D Web-based animations screen copied from the module.

Figure 7
Mounting details presented with 2D animations and simulations

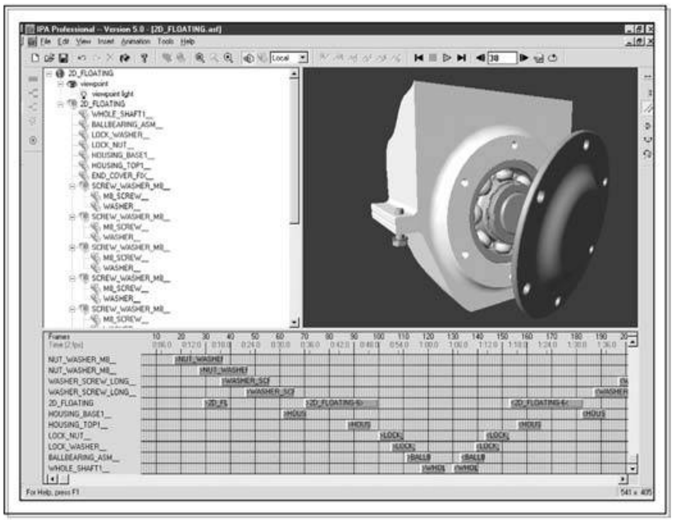


(a)
(b)

Figure 8
Mounting details presented with 3D animations



(a)


(b)

5 Concluding remarks

A Web-based engineering approach is proposed for implementing a Web-based design support system. The distributed architecture for the system is presented with respect to its implementation with Internet and Java programming in particular. The approach is further described and investigated in detail with implementing issues for a Web-based bearing design support system developed by the authors, which are of significance for developing Web-based engineering systems, especially running in heterogeneous environments. The development and trial results have shown that these implementation issues are critical in the attainment of a Web-based design support system applied in industry. The authors would like to present these results in the near future.

References

Booth, R. (1996), "Agile manufacturing", *Engineering Management Journal*, Vol. 6 No. 2, pp. 105-12.

Cheng, K. and Rowe, W.B. (1996), "A designer based approach to product design in a concurrent engineering environment", *Proceedings of the Second International Conference on Managing Integrated Manufacturing: Strategic, Organisation and Social Change*, Leicester University, Leicester, 26-28 June, pp. 125-30.

Chui, W.H. and Wright, P.K. (1999), "A WWW computer integrated manufacturing environment for rapid prototyping and

Mr K. Markus implemented some animations and simulations for the bearing mounting details module and Mr J. Toussaint implemented the lubrication module of the system presented. Thanks are also extended to SKF (UK) Ltd for providing useful bearing design and manufacturing data.

education", *International Journal of Computer Integrated Manufacturing*, Vol. 12 No. 1, pp. 54-60.

Cheng, K., Harrison, D.K. and Pan, P.Y. (1997), "An Internet-based architecture of implementing design and manufacturing agility for rolling bearings", *Journal of Materials Processing Technology*, Vol. 73 No. 1-3, pp. 96-101.

Erkes, J.W., Kenny, K.B., Lewis, J.W., Sarachan, B.D., Sobolewski, M.W. and Sum, R.N. (1996), "Implementing shared manufacturing services on the World-Wide Web", *Communications of the ACM*, Vol. 39 No. 2, pp. 34-45.

Hardwick, M., Spooner, D.L., Rando, T. and Morris, K.C. (1996), "Sharing manufacturing information in virtual enterprises", *Communications of the ACM*, Vol. 39 No. 2, pp. 46-54.

Hix, D. and Hartson, H.R. (1993), *Developing User Interface*, John Wiley & Sons, London.

Iuliano, M. and Jones, A. (1996), "Controlling activities in a virtual manufacturing cell", *Proceedings of the 1996 Winter Simulation Conference (WSC'96)*, NJ, USA, pp. 1062-7.

Netscape (1999), "LiveWire 1.01 for Windows32 – release notes", available at: http://home.netscape.com/eng/LiveWire/relnotes/win-1.01.html (accessed February).

Pan, P.Y., Cheng, K. and Harrison, D.K. (1999), "Java-based systems: an engineering approach to the implementation of design agility and manufacturing responsiveness", *Proceedings of the 15th International Conference on Computer-aided Production Engineering*, Durham University, Durham, pp. 19-21.

Reed, J.A. and Afjeh, A.A. (1998), "Developing interactive educational engineering software for the World Wide Web with Java", *Computer in Education*, Vol. 30, pp. 183-94.

Stanek, W.R. (1999), "Server-side JavaScript", available at: www.zdnet.com ... es/articals/0,4413,2251991,00.html (accessed January).

Tomarchio, P.O. and Vita, L. (1998), "Increasing application accessibility through JAVA", *IEEE Internet Computing*, Vol. 2 No. 4, pp. 70-7.

Yu, S.S. and Chen, W.C. (1998), "Java based multi-tier architecture for enterprise computing: a case study from a university academic information system", *Proceedings of the 17th Conference on Consumer Electronics*, NJ, pp. 252-3.

Yu, W.M. and Cox, A. (1997), "Java/DSM: a platform for heterogeneous computing", *Concurrency: Practice and Experience*, Vol. 9 No. 11, pp. 1213-24.

Weber, J. (1996), *Using JAVA*, 2nd ed., Que Corporation, New York, NY.

This article has been cited by:

1. V. M. M. Thilak, S. R. Devadasan, N. M. Sivaram. 2015. A Literature Review on the Progression of Agile Manufacturing Paradigm and Its Scope of Application in Pump Industry. *The Scientific World Journal* **2015**, 1. [CrossRef]
2. S. Vinodh, K.R. Arvind, D. Rajanayagam. 2011. Development of digital product catalogue for enabling agility in a manufacturing organisation. *Journal of Engineering, Design and Technology* **9**:2, 143-156. [Abstract] [Full Text] [PDF]
3. S. Vinodh, D. Kuttalingam. 2011. Computer-aided design and engineering as enablers of agile manufacturing. *Journal of Manufacturing Technology Management* **22**:3, 405-418. [Abstract] [Full Text] [PDF]
4. S. Vinodh, S.R. Devadasan, C. Shankar. 2010. Design agility through computer aided design. *Journal of Engineering, Design and Technology* **8**:1, 94-106. [Abstract] [Full Text] [PDF]
5. S. Vinodh, G. Sundararaj, S.R. Devadasan. 2009. Total agile design system model via literature exploration. *Industrial Management & Data Systems* **109**:4, 570-588. [Abstract] [Full Text] [PDF]
6. S. Vinodh, G. Sundararaj, S.R. Devadasan. 2008. Total agile design system in contemporary organisational scenario. *Industrial Management & Data Systems* **108**:8, 1111-1130. [Abstract] [Full Text] [PDF]