Int J Adv Manuf Technol (2004) 24: 148–160
DOI 10.1007/s00170-003-1737-4

ORIGINAL ARTICLE

S. Q. Zhou · A. P. Zhao · K. S. Chin
P. K. D. V. Yarlagadda · Z. Peng

A solution for knowledge resources provider over the internet

Received: 8 January 2002 / Accepted: 11 March 2003 / Published online: 4 May 2004

Abstract Product development is complicated systematic engineering. The development of a successful product is achieved through cooperation between various design teams and utilisation of a number of design resources that exist in various disciplines, organisations and locations. As product complexity increases, product designers are required to collaborate with others to develop high-quality products. In this process, an effective solution is required from a knowledge resources provider for them to be able to acquire and share knowledge resources in various teams, locations, and domains. Based on the Internet, a solution for a knowledge resources provider, called a web-based product development support system (WPDSS), is proposed in this paper. The structure and implementation of the WPDSS system are both discussed in detail. Issues concerning development of the WPDSS system such as system structure, knowledge representation, and knowledge searches are investigated. Based on design of a rotor-bearing system of turbine machinery, a prototype of the WPDSS is developed to validate the feasibility of the proposed solution.

Keywords Product development · Knowledge resources · Integrated system · Internet

S. Q. Zhou (✉) · Z. Peng
School of Engineering, James Cook University,
Townsville, QLD 4811, Australia
E-mail: zhoushouqin@263.net

A. P. Zhao
Department of Electric Power Engineering,
School of Electronics and Information and Electric Power Engineering, Shanghai Jiaotong University,
Shanghai 200030, P.R. China

K. S. Chin
Department of Manufacturing Engineering Engineering Management, City University of Hong Kong,
83 Tat Chee Avenue, Hong Kong, P.R. China

P. K. D. V. Yarlagadda
School of Mechanical, Manufacturing and Medical Engineering,
Queensland University of Technology, Brisbane,
QLD 4001, Australia

1 Introduction

Product development is a complicated activity. Its failure or success is heavily determined by design competence. Successful product development is achieved through synthetic utilisation of a wide variety of knowledge in various fields and integration of the efforts of design teams in various design stages. To support product development, in addition to an effective collaborative design team, a knowledge support environment which integrates all kinds of design knowledge in various sources, locations, domains, and disciplines is needed. As product complexity increases, this demand is obvious and urgent. In most organisations, however, knowledge is distributed among many individuals such as product designers, tool engineers, manufacturing engineers, performance analysts, etc. Due to the difficulties in collecting and building a huge central knowledge base, efforts at developing and building centralised knowledge repositories that integrate all of these individual knowledge resources have failed in recent years [1]. The developed centralised systems are only suitable for a narrow scope and they were found to be impractical. In this paper, a distributive knowledge resources provider solution, called a web-based product development support system (WPDSS), is proposed. It focuses on a series of issues on how to integrate distributive individuals to form a league of knowledge resources over the Internet. The ultimate objective of WPDSS is to construct a distributive knowledge resource shared system to meet the needs of product designers on various knowledge resources based on the Internet network, advanced information technology and artificial intelligence technology.

Both shared knowledge and knowledge on its own are power [2]. Strategic alliances that share design knowledge resources have a great advantage in global competition, especially to small medium enterprises (SMEs) since they generally lack information technology support and in-house developing expertise [3]. The development of a WPDSS system will be of great benefit to companies, especially for SMEs, since it will provide robust support of knowledge resources to product designers. Moreover, as global competition increases, a product designer is required to develop high quality output with a short lead-time. Traditional methods of knowledge acquisition, such as design handbooks, single analysis programs or software, cannot meet current designer demands regarding range and quantity of knowledge resources. The WPDSS system has been developed to provide sufficient information and knowledge resources that refer to all aspects of product development and that will be of great benefit to product designers in developing outputs quickly and efficiently. However, in the past decade there has been very little research focusing on this area of study. The rapid development of the Internet as a global network and of Intranet as a corporate network has created a chance for industries to meet this challenge. The Internet technique provides a cheap way of linking and interconnecting with industrial organisations and resources. Based on the Internet and related tools, a web-based product development support system, integrating individual knowledge resources distributed in various disciplines, domains, and locations is developed in this paper.

In recent years, with the great progress of the Internet and Web technologies, a number of attempts have been made to implement web-based support systems for product design and manufacturing. Among them, a number of studies and projects investigated methods of constructing a rational collaborative framework for product development. For example, Qiang et al. [4] developed a web product design system that mainly supports CAD-based collaborative design through the Internet. But the CAD platform discussed in that system is single. Mills [5] adopted an engineering concept for collaboration within the fields of design, engineering, manufacturing and others within an Internet environment. It discussed a series of issues for collaborative engineering such as system structure, information interaction, knowledge sharing, etc. Yet, it did not consider collaborative knowledge engineering within a distributive environment. Huang et al. [6] presented a web-based and synchronised system to provide designs for X guidelines on the Internet. However, its outcome is still in the primary stage, and it did not consider knowledge acquisition and utilisation within a collaborative design environment. All of the studies mentioned above emphatically investigated construction of a system framework to integrate various teams and to achieve seamless information sharing among them. The organisation, construction, and utilisation of knowledge resources were not the core issues considered in these projects, rather, these issues were almost ignored.

In addition, based on Internet network and technology, many studies attempted to build an Internet-based expert system for product design. For example, some web pages were developed to aid the designer in material selection [7]. Pan [8] presented an online catalogue system and a demo for the design of a timing belt. Tumkor [9] examined the development of an online design catalogue which combines an expert system and real-time application on the Web. Su et al. [10] developed a gear design optimisation over the Internet and Pan et al. [11] presented a novel approach to developing an agile design system for rolling bearings based on artificial intelligence and Web technologies. However, these studies mainly investigated the use of Internet networks and relevant technologies to build an automatic design system or tools. The knowledge base in these systems is single, standalone, and self-contain. Obviously, a real life product design needs to adopt a great deal of knowledge concerning the entire life cycle of product development from customer requirements to product maintenance. The automatic expert systems only complete a part of the overall design work. To support the entire life cycle of product development, a complete distributive knowledge resources support system should be provided to product designers so that they can search, retrieve, and utilise the knowledge they need during the product development process.

However, most current studies focused on investigation of the logical structure of a collaborative product development system within an Internet environment and Internet-based automatic expert system for a specific single design application (e.g. bearing selection). There are few investigations on the construction of a distributive knowledge resource integrated system for product development. In this paper, the web-based product development support system (WPDSS), which integrates a great deal of distributive knowledge resources existing in various disciplines, domains, and location, is proposed. The study takes a detailed look at issues regarding organising, building and integrating various distributive knowledge resources to form a web-based league of knowledge resources, searching for a piece of knowledge precisely and quickly and efficiently utilising retrieved knowledge in different design platforms.

2 The structure of the WPDSS system

Product design is a mapping process among customer domain, functional domain, physical domain, and process domain [12, 13]. To develop a successful product, it is necessary to synthetically utilise knowledge resources in various domains. Figure 1 shows product development, which is a process of mapping customer requirements to structure details of a product. All knowledge resources are stored in three different repositories: local repositories, domain repositories and common repositories. Obviously, it is impossible for any product designer to hold and master all knowledge resources existing in various local, domain and common repositories. In addition to the local knowledge that he maintains, the product designer must acquire knowledge from domain, common, and other local repositories. Hence, an effective solution such as the WPDSS system

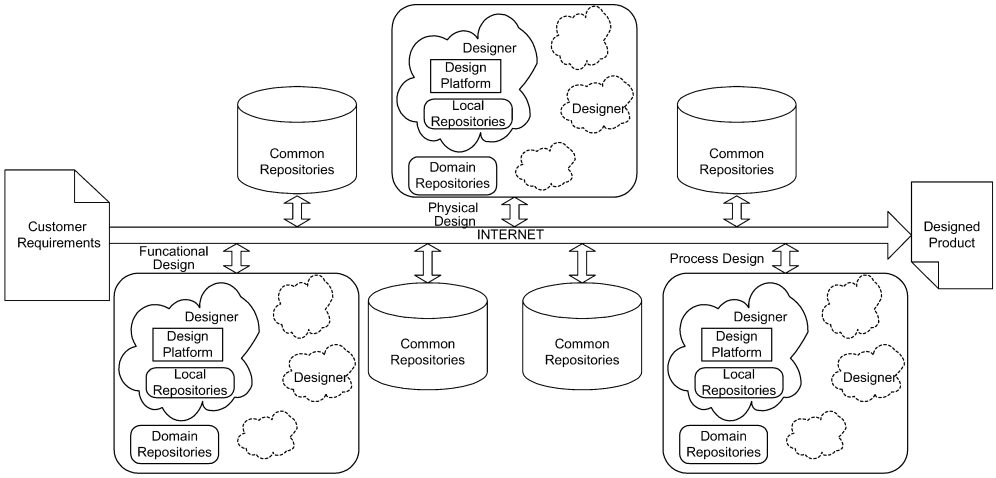

Fig. 1 Tropological organization of knowledge repositories for product development

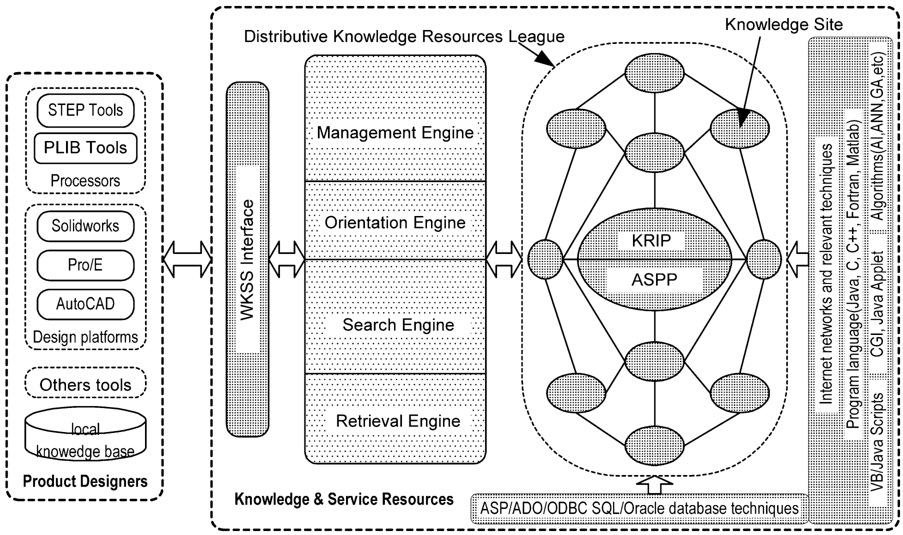

Fig. 2 Basic stucture of web-based design development support system

is necessary to aid product designers in acquiring and utilising knowledge resources in the product development process.

The basic structure of the WPDSS system is shown as Fig. 2. Individual knowledge sites store special knowledge resources in different locations, domains, and disciplines such as customer requirement analysis knowledge, product performance evaluation knowledge, sample model of product, product design specification, product parts/components, etc. All these knowledge sites are connected and integrated via a Internet network to form a distributive knowledge resource integrated league. The engines hidden in the WPDSS system are used to manage knowledge resources stored in various repositories and to provide orientation, search and retrieval services for system users. When developing a product, the product designer first relies on his local knowledge resources and design platform (e.g. Pro/E). If the local knowledge resources cannot meet his demand, then he logs into the WPDSS system to search for and retrieve the necessary knowledge according to his requirements. Since the WPDSS forms a knowledge league that integrates a great deal of knowledge resources in various locations, domains and disciplines, it is feasible to provide robust knowledge support for various product designers. Figure 2 shows that the WPDSS system consists of a large number of special knowledge sites. These various knowledge sites are the basis of the WPDSS system. The main task in developing the WPDSS system is building these special knowledge sites so as to create a complete knowledge network to support the demands of product development.

Generally, product design can be divided into five stages: concept design, scheme design, structure design, detail design, and performance evaluation. The former four stages can be attributed to design stages, and the latter stage can be attributed to the evaluation stage. The design stage completes the detailed design of product according to customer requirements. The evaluation stage achieves the performance analysis and simulation to evaluate the designed product as to whether it meets design requirements. From this viewpoint, in this paper, the knowledge resources contained in local, domain and common repositories are divided into two categories: design resources and service resources. Design resources mainly refer to knowledge for the design stage, which mainly consists of design principles, design procedures, design specifications and part/complement models of product. The service resources in the evaluation stage mainly refers to analysis and calculation programs, such as the analysis of the stability of a rotor-bearing system for a turbine that is used to evaluate the vibration performance of turbine, analysis programs to determine the ability to assemble and manufacture a designed mould, etc. Establishment and implementation for the two kinds of knowledge resources are considered separately in this paper. Knowledge resources related to overall product design in individual knowledge sites are provided through two types of platforms: the design resources integrated platform (DRIP) and the application services providers platform (ASPP), respectively. A detailed structure of certain individual knowledge sites is shown in Fig. 3. It is shown that there are two different platforms in every individual knowledge site used to provide knowledge resources.

As stated above, the WPDSS system is a web-based distributive knowledge resource integrated league used to provide robust knowledge support service for product designer. It consists of a number of individual knowledge sites. In every site, there are two platforms, the DRIP platform and ASPP platform, which are used to supply design resources and service resources, respectively. Product designers may acquire knowledge resources from the system via hidden engines. The Web-based WPDSS system breaks through the limitations of traditional knowledge acquisition methods such as handbooks or single programs. Moreover, the WPDSS is an open system. Any knowledge site can be added to it through an authentication procedure. The administrator of the WPDSS system can authorise a new knowledge site to join in it to expand knowledge resources in system. As the number knowledge sites increases, enriching knowledge resources, a robust knowledge resource network can be formed. The following sections will discuss the key issues of the development of the WPDSS system and its implementation in detail.

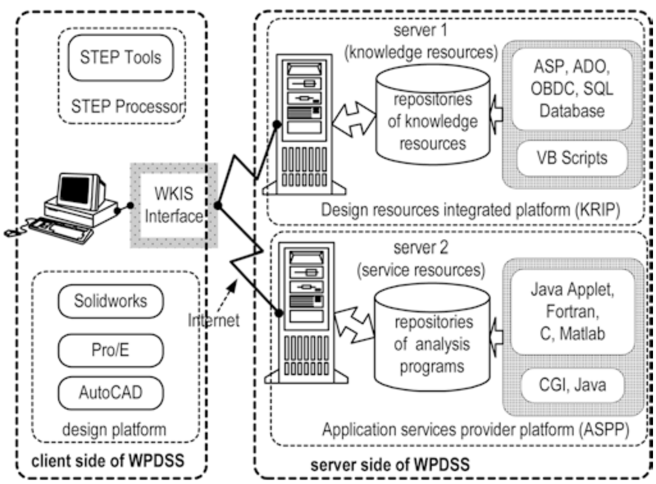

Fig. 3 Structure of individual knowledge site

3 Key issues of the WPDSS system

To implement the WPDSS system, a core task is to build various individual knowledge sites based on certain specifications. In addition, a feasible and effective web-based network needs to be constructed to integrate a great deal of individual knowledge sites. Also, since there are a number of knowledge resources available in the WPDSS system, an efficient solution for knowledge search should be provided to retrieve the knowledge in terms of requirements of product designers.

3.1 Establishment of an individual knowledge site

As stated in Sect. 2, the individual knowledge site consists of two platforms: the design resources integrated platform (DRIP) and the application services providers' platform (ASPP). The establishment of a knowledge site includes two parts: establishment of the DRIP and the ASPP. The structure of the DRIP is proposed in Fig. 4. Based on Web technology and client/server databases such as the SQL server database and Oracle, all kinds of knowledge resources concerning product design stage such as design specifications, design principles, part/component model and other relevant information are stored in various repositories. In this paper, the design resources are divided into two different types: product date model (PDM) and discrete information model (DIM). For instance, the structure and details of a standard rolling bearing can be expressed with an AutoCAD file "bearing.dxf". In certain product design platforms such as AutoCAD, the user can open this file and get detailed information about the rolling bearing. As a piece of the knowledge resource, the "bearing.dxf" file used to describe the rolling bearing is based on a specific data structure that can be directly imported into the product designers' platform (e.g. AutoCAD). Hereby, this kind of design resources is classified as part of the product data model. However, some design resources, such as a mathematical equations or design text cannot be expressed with a certain data structure. These design resources, are then classified as discrete information models. In this paper, different storage and representation solutions are provided corresponding to different types of information. The discrete information model is described with plain text, pictures, or in table format. They are expressed with XML, HTML or GIF files. The design resources of product data models such as parts/components are represented with the EXPRESS language based on the standard of exchange for product

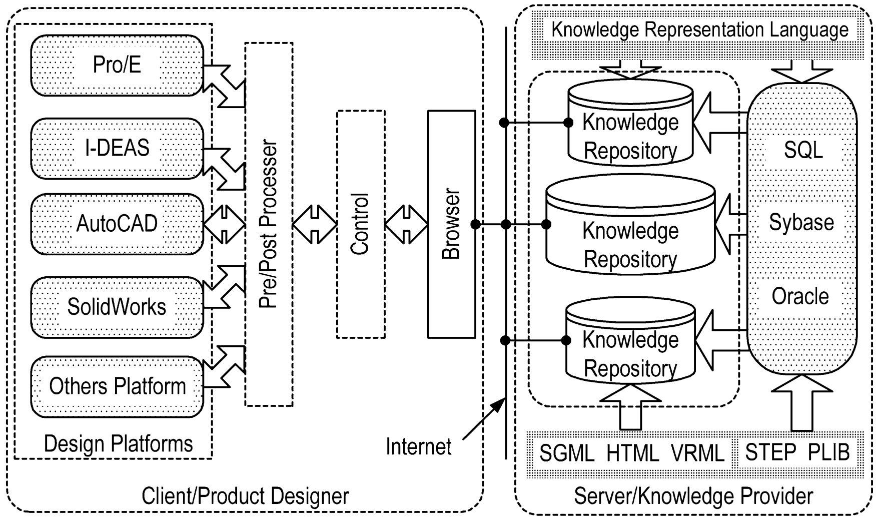

Fig. 4 Structure of design resources integrated platform

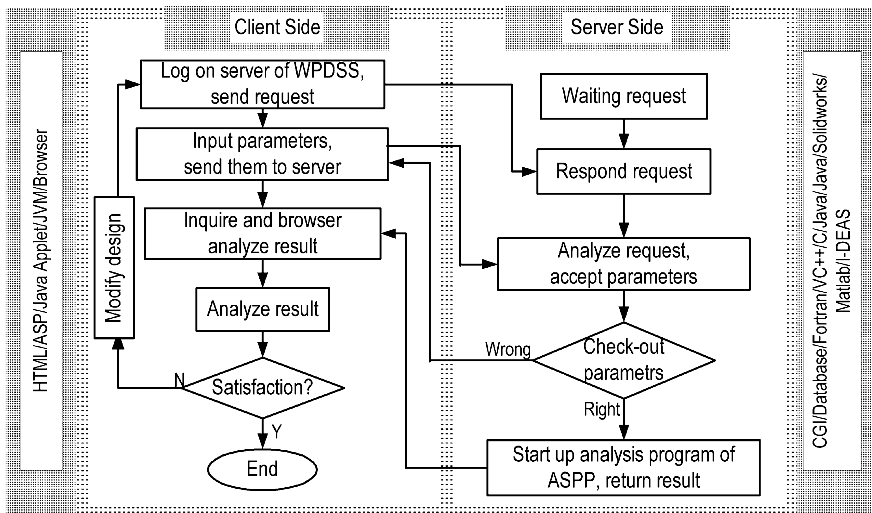

Fig. 5 Structure of application services provider platform

model data (STEP) [14]. Product designers can browse discrete information resources via their Internet browser (e.g. Internet Explorer) and download and import product data model resources into their product design platform.

The structure of the ASPP is shown in Fig. 5. It is used to remotely provide service resources to product designers so that they can analyse and evaluate the designed product as to whether it meets design requirements. A series of analysis programs are built into the server side of the ASPP. When a product designer wants to validate or evaluate the performance of a designed product, he can enter the ASPP platform, fill out the parameter forms, and submit the parameters. The ASPP server will start the program and return the evaluation results. The users can then decide whether the deigned product meets design requirement or need to be re-designed according to returned analysis results. If the designed product satisfies the performance requirements, product design is finished. Otherwise, the design process must be iterated again until the designed product meets the requirements. Traditionally, analysis programs of product performance are designed for one user. The cost of this kind of program is high and its maintenance is difficult. The service resources in the ASPP platform of a WPDSS system are built over the Internet networks. Any authorised users can run them remotely. Moreover, the core programs are maintained by the vendor. This manner of service resource provider is very convenient for users so that they may obtain the up-to-date version of service resources. This is one of a series of benefits such as low cost, easy maintenance, and no limitation on quantity of users.

3.2 The knowledge search approach over the Internet

As the amount of knowledge sites and knowledge resources in the WPDSS system increase, being able to quickly and precisely searching for a piece of knowledge according to user requirements is a key issue that affect the performance of the WPDSS system. An efficient and effective approach toward knowledge search should be provided to ensure that the WPDSS system provides a perfectly service for product designers.

Currently, there are a number of knowledge search mechanisms already in place on the Internet, including keyword search engines, such as AltaVista, and catalogue search engines, such as Yahoo. Although keywords and catalogue searches have many advantages, they both face problems. For instance, keyword search engines often return a large number of inappropriate pointers; whereas the catalogue, or taxonomic approach, requires a great deal of time in hand-assembling, although both automatic and manual refinement of keyword searches exist, such as WebCompass, where users can define sub-categories of their search, they still return a large number of inappropriate 'hits'. Due to the lack of content-awareness in traditional hypertext documents, search engines have to inspect all of the content of each page and consequently, they return vast numbers of inappropriate hits. In contrast to the above search approaches, in this paper, a two-step strategy for knowledge search is proposed to aid product designers in searching and retrieving knowledge from the WPDSS system quickly and exactly. The goal is to orientate an exact knowledge site according to product designer requirements for the knowledge they need. The next step is searching for a piece of knowledge that exactly meets the designer's demand within this orientated site. Corresponding to the two steps, the weighted keywords orienting approach (WKOA) for orientating knowledge site within the WPDSS system and the constraint driven search approach (CDSA) for retrieving knowledge from repositories of orientated site are proposed. Detailed descriptions of WKOA and CDSA are presented in the following sections.

3.3 Weighted keywords orienting approach

The weighted keywords orientating approach (WKOA) is used to orientate a knowledge site that contains certain knowledge required by a user within the distributive knowledge site league. A flowchart of WKOA is shown in Fig. 6. Its basic scheme is introduced as follows:

First, set up a keyword repository and fix every knowledge site with 5–10 keywords used to express its property. When a user wishes to search for a specific piece of knowledge, he first enters a paragraph that represents his requirements into the interface of search engine. Second, the keyword retrieving model picks out several keywords from the paragraph and returns them to user. The user assigns weights to individual keywords according to their importance in the scale of 0.0 (least important) to 1.0 (most important). After the user inputs the weighted keywords through the interface of the search engine, the search engine calculates the match degree of knowledge sites with the keywords, and returns the address of knowledge sites (results) with the highest match degree to the user. Consequently a knowledge site that contains the required knowledge is orientated within the WPDSS system according to the requirements of designer on knowledge. Once the site is orientated, the designer continues to search for the information he needs within this site.

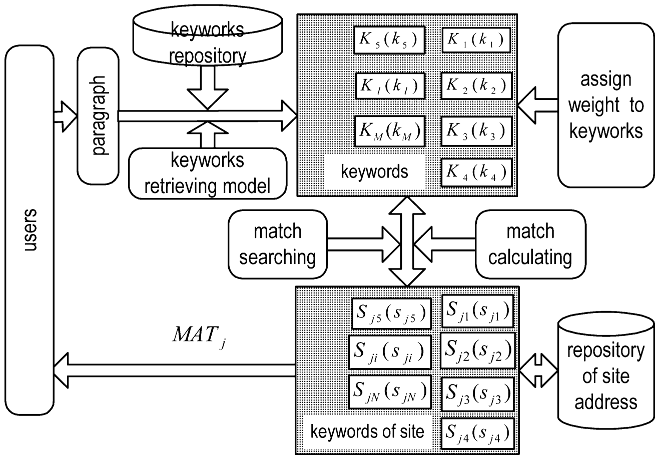

Fig. 6 Flowchart of weighted keywords orienting approach

In the WKOA approach, the keywords used to orientate knowledge sites are picked up from user's requirement statements. For example, if a user enters, "a knowledge site that can be used to tutor the structural design of turbine machinery and performance analysis of a rotor-bearing system", keywords such as "structural design", "turbine-machinery", "performance analysis", "rotor-bearing system", are then picked up from the statement. It must be emphasised that the degree of importance of these keywords that reflects user's requirements is different between among the various keywords, so that the primary and secondary relationship of the selected keywords needs to be determined. Otherwise, the search engine will simply return the knowledge sites related to these keywords no matter what their degree of match within user requirements. To overcome this limitation, a weighted keywords approach is applied. When the search engine picks up the keywords from the users' statement, the list of keywords is returned to the user. The user then assigns weight values to each keyword according to his demand to determine the degree of importance of each keyword. Sequentially, the search engine returns the knowledge sites to the user in terms of degree of match within knowledge sites using the weighted keywords.

The address record of knowledge sites, which is stored in the address repository of the knowledge sites, is expressed as Eq. 1.

$$VD(S_i) = \{\mathrm{Name}(S_i), \mathrm{Keywords}(S_i), \mathrm{Abstract}(S_i), \mathrm{Address}(S_i)\} \quad (1)$$

The parameter S_{ji} represents the ith keyword of the site j. S_{ji} is the weight value of keyword S_{ji}, which reflect the match degree of knowledge site $j$with keyword S_{ji}. The value of S_{ji} is confined within [0, 1]. The parameter K_l represents the keywords picked up from the user's input. K_l is the weight value of keyword Kl, which is determined by user. It is also confined within [0, 1]. If K_l is 1, it indicates that the keyword K_l is most important for matching the knowledge site. If K_l is 0, it indicates that the keyword K_l can be ignored. Once the parameters K_l,K_l are determined, the search engine calculates the match degree in term of the parameters S_{ji},S_{ji},K_l,K_l. The calculation model is shown in Eq. 2.

$$MAT_j = \sum_{i=0}^{M} k_l \cdot k_l \cdot s_{jl} \qquad (2)$$

In Eq. 2,M is the total quantity of keyword K_l. MAT_l represents the match degree between knowledge site j and keywords K_l,$l \in [1,M]$. In terms of Eq. 2, the WPDSS system returns a knowledge site list to the user according to the match degree of the knowledge site for the user requirements.

3.4 Constraint driven search approach

In the two-step strategy, after the knowledge site is orientated, the knowledge needs to be further confined within this orientated site. To search and retrieve a piece of knowledge from a certain concrete knowledge site, a constraint driven search approach (CDSA) is proposed in this paper. The standpoint of the CDSA is: a user (designer) generally searches for a piece of knowledge according to a series of requirements that the knowledge must meet, such as its function, property, structure and condition. These requirements, in this paper, are called constraints. The knowledge search is thus actually a constraint-driven procedure. Confirmation of knowledge is realised by iteratively selecting and validating until the selected knowledge is coincident with all constraint conditions. Employing the selection of the rotary support-part as a case, an application of the CDSA approach is described as follows.

Supposing a product designer needs to select a rotary support part in a machine design and this part does not exist in the current database of design platform. He needs to search and retrieve it from the WPDSS system. Using the WKOA approach, he first confines the knowledge site that contains the knowledge that he needs. Then within this knowledge site, using the CDSA approach, he determines the function of the support part as "rotary support". Several pieces of knowledge are found that correspond to this function requirement such as rolling bearing, journal bearing, and magnetic bearing. Then, he determines the constraints that the knowledge should meet, such as method of lubrication, working load, running speed, etc. In term of these constraints, he could perform a step-by-step search and confine the knowledge to a piece of exact knowledge (such as rolling bearing 6215 in this example). The

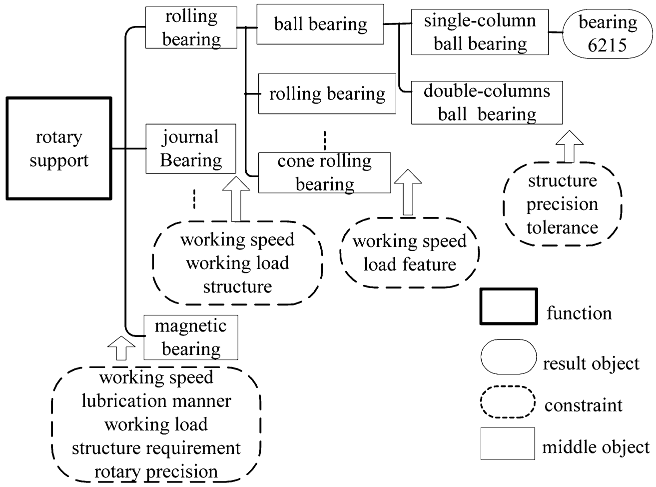

Fig. 7 The flowchart of support-part search on constraint driven

search flowchart for this rotary support part is shown as Fig. 7. With the CDSA approach, the user can search and retrieve the knowledge that matches the required functions and constraints from the knowledge repository of WPDSS once he determines the function of the knowledge and the constraints that the knowledge is required to meet.

Since the WKOA approach can capture the user needs more clearly and precisely than traditional approaches such as the keyword search approach and catalogue search approach, it can effectively orientate knowledge sites according to the demand of users. The CDSA approach obeys the real process of knowledge search. Through the mapping match between constraints and knowledge, it can return knowledge exactly according to designers' requirements within the knowledge site. As mentioned above, through this two-step search strategy, product designers can quickly and efficiently search for and retrieve a piece of knowledge within the Internet network environment.

4 System implementation

4.1 Prototype of the WPDSS system

To develop the prototype of the WPDSS system, a demo of the product development system is built as follows. SolidWorks is adopted as the client-side platform of product development and it is assumed that the product designer uses this commercial software to design and develop products. Internet Explorer is applied as the information browser on the client side. ST-Developer [15] software provided by STEP Tools Inc. is applied as the information converter among various design platforms. Because the Windows NT environment provides an effective Internet solution, supports the active server pages (ASP) technique and is used popularly, the server for the prototype of the WPDSS system is constructed using the Peer Web Server 2.0 and IIS 4.0 within the Windows NT 4.0 operating system.

A design resource integrated platform (DRIP) is constructed via an SQL server database. The design resources in the DRIP platform are stored in SQL server databases. SQL-92 standard language is applied as a search language in SQL databases. The WPDSS server is linked with SQL databases via the data component ActiveX data object (ADO) hidden in active server pages (ASP) and open database connection (ODBC). The service resources in the application service provider platform (ASPP) are built via a web programming language and sub-layer mathematical programs programmed with Fortran and Matlab. Communication between the sub-layer programs and the server of the WPDSS system is achieved via a common gateway interface (CGI).

To construct the WPDSS system, the main activities include establishing the server side of system so as to supply knowledge resources for product development. The constructed knowledge resources in this prototype of a WPDSS system contain part models (e.g. standard rolling bearing) and analysis programs concerning rolling bearings, journal bearings and electric magnetic bearings. These resources are valuable for product designers of turbine machinery, especially for designers of rotor-bearing systems. The design resources in the prototype of the WPDSS system can be browsed and downloaded to aid rotor-bearing system design. Service resources can be used to remotely analyse characteristics of bearing and the performance of the rotor-bearing system.

4.2 Representation of design resources in the WPDSS system

In the prototype, design resources concerning the discrete information model are described with text, diagrams, pictures, and tables and are expressed with an HTML file. The user can browse and download this kind of knowledge. Design resources concerning the product data model are expressed in the traditional manner. In addition, a STEP-based digital information package that contains all information on this piece of knowledge is provided. Not only can the user browse the product data model in the traditional form via an Internet browser, but he can also download the digital package and import it into his product design platform directly. In the following section, taking a rolling bearing as an example, a STEP-based knowledge representation scheme is described.

Based on the STEP standard, the product data model of rolling bearing (design resources) is expressed in Table 1.

In Table 1, the STEP information model of a rolling bearing contains complete information that can be understood with various design platforms since it is in a neutral format. The design resource of the product data model in the prototype of the WPDSS system is represented in Fig. 8. The user can browse the information for design resources via their Internet Browser to confirm whether the searched design resource meets the design demand. If so, he can download the STEP information model which contains the complete information about the design resource and can import the

Table 1 Brief STEP information model of rolling bearing

```
CHEMA rolling_bearing_schema
REFERENCE FROM geometry_schema(
axis2_placement_3d,
cartesian_transformation_operator_3d,
curve,
geometric_representation_context,
point_on_curve,
point_on_surface,
surface,
rectangular_trimmed_surface,
trimmed_curve)
USE FROM support_resource_schema(
bag_to_set,label)
...
ENTITY assistant_feature
SUBTYPE OF (rolling_bearing_entity)
code                                  STRING
price                                 REAL
vendor                                STRING
note                                  STRING
END_ENTITY
ENTITY characteristic_feature
SUBTYPE OF (rolling_bearing_entity)
limited_speed                         REAL
rotary_precision                      REAL
permited_temperature                  REAL
stiffness_coefficient                 REAL
frication_coefficients                REAL
rated_dynamic_load                    REAL
rated_static_load                     REAL
centripetal_ability                   STRING
END_ENTITY
ENTITY construction_feature
SUBTYPE OF (rolling_bearing_entity)
style                                 STRING
basic_size                            basic_size_entity
install_size                          install_size_entity
secondary_size                        secondary_size_entity
tolerance                             tolerance_entity
parts                                 parts_entity
radial_windage                        REAL
axial_windage                         REAL
heat_treatment                        STRING
machining_method                      STRING
lubrication                           lubrication_entity
seal                                  seal_entity
inner_ring_ match                     STRING
exteral_ring_match                    STRING
END_ENTITY
ENTITY basic_size_entity
D,d,B,rmin,r1 min,F,E                 REAL
C,T,r2 min,A,H                        REAL
END_ENTITY
ENTITY install_size_entity
D1, a, b, rs, da                      REAL
Da, Db, a1, ras                       REAL
END_ENTITY
.........
END_SCHEMA
```

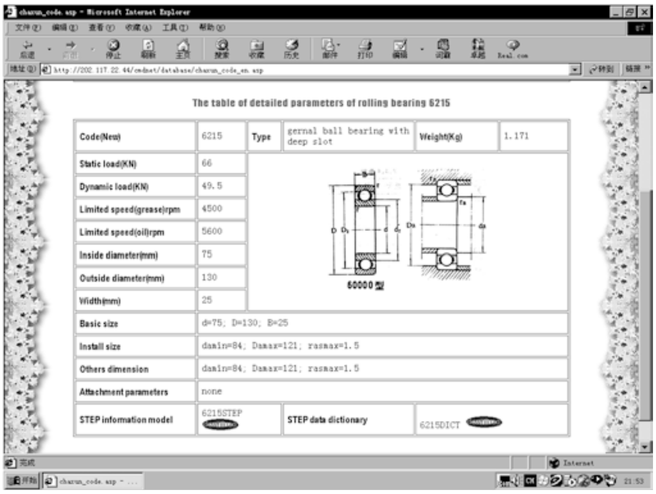

Fig. 8 Representation of design resources of product data model

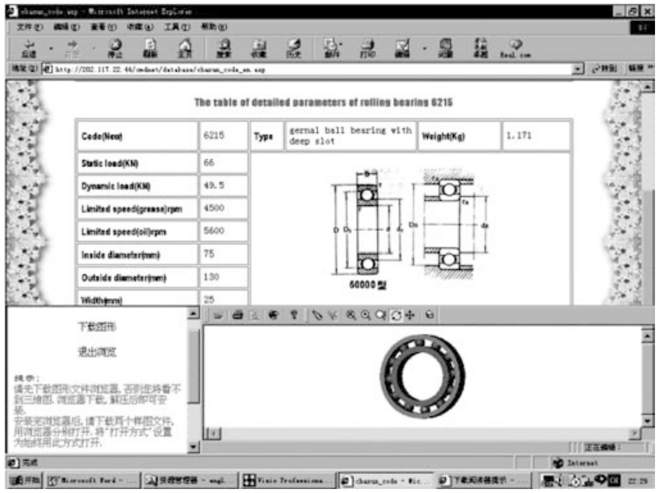

Fig. 9 A piece of searched design resource of rolling bearing

model into his product design platform, such as SolidWorks, Pro/E, AutoCAD, etc. The knowledge provided in this format can greatly improve utilisation efficiency since it can be imported into a design platform directly via the STEP pre/post processor, consequently greatly shortening lead time for product development.

4.3 Storage manner of design resources in the DRIP platform

Because the WPDSS system is based on a client/server mechanism, the database and knowledge bases of the system are constructed via the database software with a C/S mechanism such as SQL, or Oracle. In this paper, the SQL server 7.0 was applied to store a part of the design resources. The databases are connected with the WPDSS system server via open database connectivity (ODBC). The user on the client side fills out the HTML forms and sends a request to the server. The server runs the active server pages (ASP) scripts and common gateway interface (CGI) programs to retrieve the design resources from the database via the ActiveX data object (ADO) component hidden in the ASP. Because the knowledge resources are stored in the C/S mechanism database, it is very easy for the user to remotely browse and download them via their Internet browser.

In this paper, SolidWorks, a type of CAD software, is adopted as a product design platform for the prototype of the WPDSS system. The design resource for the rolling bearing, shown in Fig. 9, is stored in three formats: traditional table, SolidWorks file, and STEP information model. The user can browse the traditional table with an Internet browser and directly utilise the downloaded SolidWorks file in a product design platform. If the user adopts platforms other than SolidWorks, he can download the STEP information model and process it via a pre-post processor so as to import it into the design platform.

4.4 Implementation of the knowledge search

Taking the selection of a rolling bearing as an example, an ANN based trial scheme of knowledge search was developed to aid the user in searching through knowledge resources. The concrete implementation is described as follows.

First, the mapping relationships between styles of rolling bearing and constraint parameters such as lubrication condition, rotary speed, operation load, etc are built. Second, the artificial neural network (ANN) is adopted to code these mapping relationships and a compiler language such as Fortran, or C is applied to program ANN. Third, sufficiently accurate samples are collected to train the ANN program so that the trained ANN can select the appropriate bearing style according to the constraint parameters determined by users. Finally, the trained ANN is immigrated to the Internet environment so that the user can utilise the knowledge search approach over the Internet. In this paper, the Fortran compiler language within Fortran PowerStation 4.0 is adopted as a program language for ANN. The trained ANN program is re-written with VB Scripts over the Internet. A constraint input interface for the knowledge search based on ANN is shown in Fig. 10. The return interface of results corresponding to the inputted constraints is shown in Fig. 11. Through several iterative procedures, shown in Fig. 9, a piece of concrete knowledge that meets the user requirements can be found. Figure 12 shows the structural design of a rotor-bearing system utilising the downloaded knowledge resource for rolling bearings from the DRIP platform of the WPDSS system.

4.5 Implementation of an application service provider platform (ASPP)

Generally, the designed product needs to be evaluated to confirm whether it meets design requirements. For in-

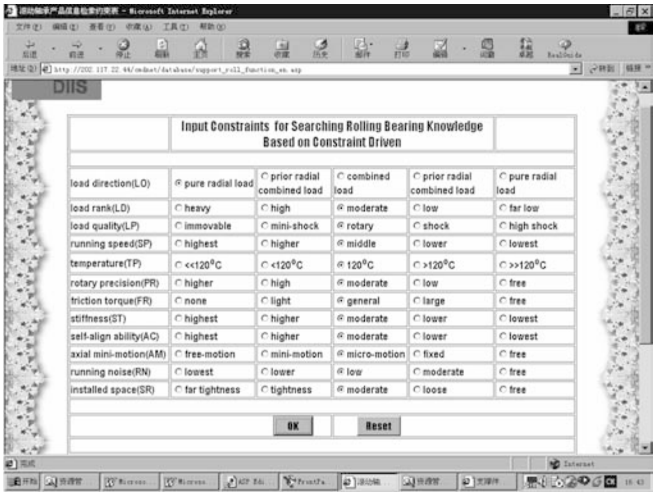

Fig. 10 Contraint input interface of knowledge search

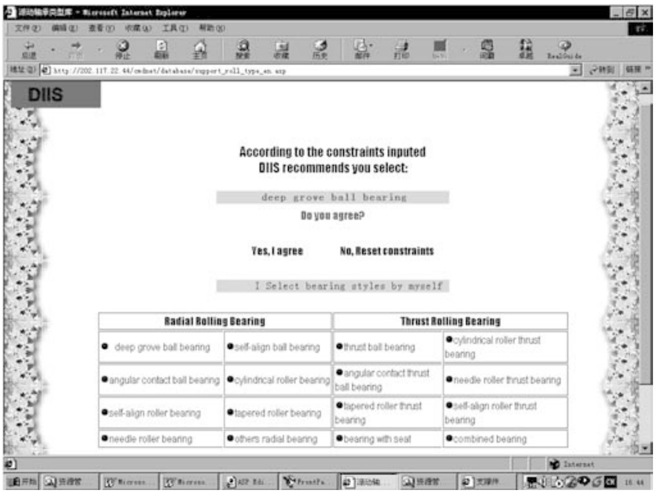

Fig. 11 Result return interface corresponding to inputted constraints

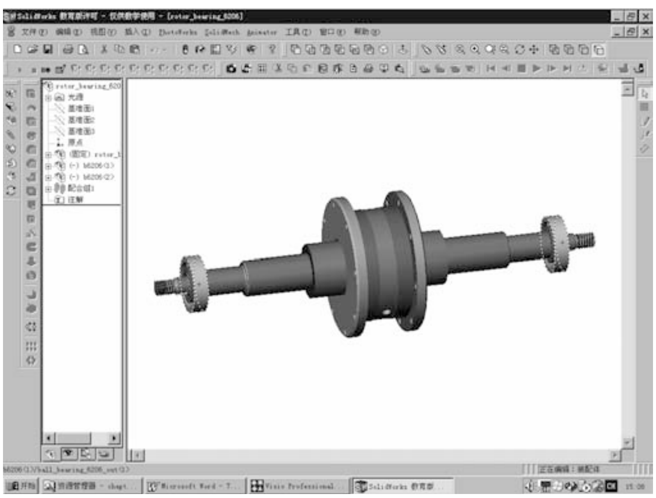

Fig. 12 Strucutre design of rotor system using searched knowledge

stance, the structure of a rotor bearing system can be determined according to user requirements and expert knowledge. The size, style and number of bearings can be determined by a turbine engineer based on the handbook. The shape, structure and material of the rotor can be determined according to an expert's experience and knowledge. It is important that the designed rotor-bearing system be assessed with a stability analysis program to ensure perfect vibration performance of the turbine machinery. For this kind of application, the application service provider platform is developed to offer a remote analysis program to assess product performance of the designed product.

In this paper, several application services that used to remotely analyse the stability of the rotor-bearing system and the characteristics of journal bearing and electric magnetic bearings are developed with Java and the Matlab language. In the developed prototype of the ASPP platform, the core programs are coded with Fortran and Matlab. The communication between core programs and the WPDSS system server is achieved with ODBC, CGI, and ASP/ADO. When a user wants to log into the WPDSS system to utilise the application services in the ASPP platform, he remotely fills out the HTML form or Java Applet form to send the parameter values to the ASPP platform server. Then the service program in the server runs and returns the analysis results to the user.

The prototype of the ASPP platform is constructed on a Windows NT operating system, the server of ASPP platform is built with a personal web server PWS32/3.0.2.926 and IIS4.0. The service resources, which include the stability analysis program for the rotor-bearing system, the characteristics analysis program for the journal bearing and performance analysis program for the magnetic bearing systems (MBS), all of which are stored in two servers. The service resources concerning stability analysis of the rotor-bearing system and bearing characteristics are stored in one server and the service resources concerning MBS performance analysis are stored in another server. The core programs for the service resources are located in server 1, shown in Fig. 13, and are coded with Fortran language. The core programs of the MBS performance analysis in the server 2 are coded with Matlab. The input interface is shown in Fig. 14. The following content will describe the establishment of both servers, respectively.

To support turbine engineers in developing a rotor-bearing system for turbine machinery efficiently, a series of service resources are provided to aid engineers in evaluating the designed rotor-bearing system and the selected bearing. Because the core programs for this kind of service resource are very complex, they are programmed with Fortran so as to increase the running efficiency of program. Communication between the core program and the ASPP server is achieved via CGI scripts. The CGI scripts receive the inputted parameter values and code them in a form understood by the core

Fig. 13 Enter interface of server 1 of the ASPP platform

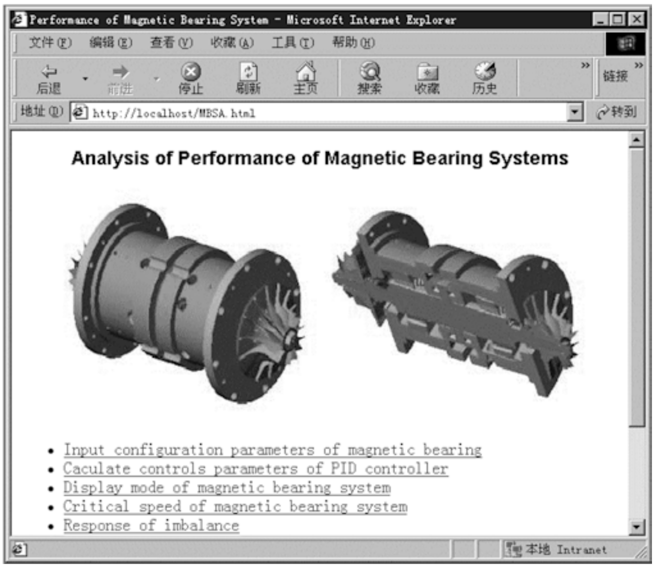

Fig. 14 Enter interface of server 2 of the ASPP platform

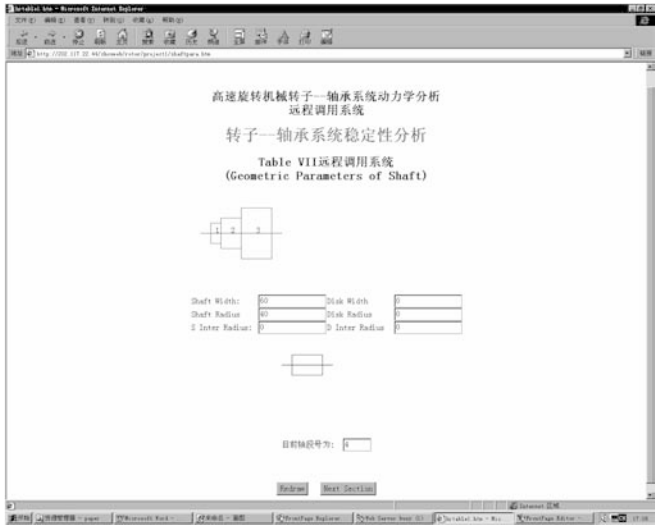

Fig. 15 Parameter input page of stability analysis of RBS

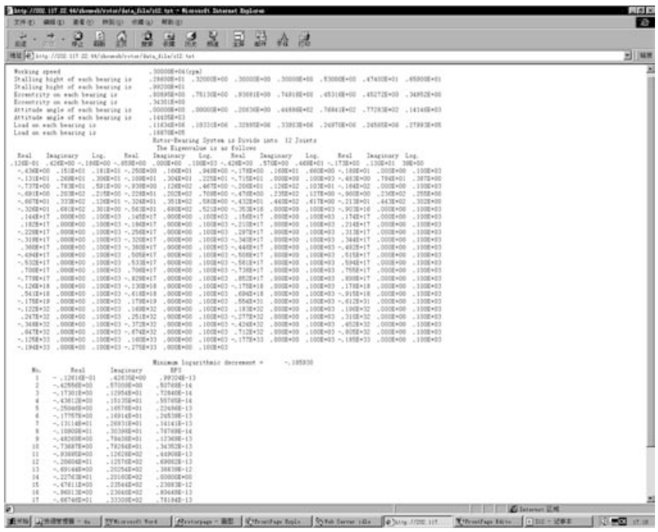

Fig. 16 Results page of stability analysis of RBS

program, control running of the core program and return the results to the user.

Figures 15 and 16 state the parameter input interface and returned result interface on stability analysis of rotor-bearing system (RBS), respectively. In this application, because the rotor is required to be divided into many elements according to the mathematical model of dynamic analysis of the rotor-bearing system to analyse vibration performance, the quantity and form of input parameters varies and is determined by the user. If a user divides rotors into different quantities of sub-units, the core program of the mathematical model requires different input parameters and values. Hence, in this service resource for stability analysis of a rotor-bearing system, the input interface of parameter is constructed via an HTML form and a Java Applet, since a Java Applet can achieve dynamic input of various parameters. The parameters that require user input are separated into several groups. Parameter name and type for each group is expressed with a Java class. All of these Java classes are expended from a Java Applet class. The Java classes are used to receive the value of parameters determined by the user and to send them to the CGI interface program. The CGI program then, packs all of the parameter values and provides them to the core program for stability analysis of the rotor-bearing system. Finally, the core program returns the analysis results to the user according to the entered parameters.

The service resources in the ASPP platform stated above are built via HTML form, Java Applet and CGI. The HTML form and Java Applet classes are used to collect parameter values from the user. The CGI program achieves communication between the core program and teh ASPP platform server. It packs the collected parameters and sends them to the core program and releases the analysis results of the core program to the user. The core programs are coded with Fortran. In the following section, a kind of service resource is introduced that is constructed in Matlab. In this kind of service resource, the core program is created with Matlab.

Due to the fact that Matlab is an easy language tool to program, although its running efficiency is lower than Fortran, a lot of analysis programs are coded with Matlab in industry. In this paper, a paradigm of web-based Matlab service resources is developed. The service resources are built on the Matlab web Server hidden in the ASPP platform. The Matlab web server application links the browser and the Matlab-based core program, collects inputted parameter values and send them to the core program, controls running of the core program and returns user results.

A Matlab-based service resource of magnetic bearing system (MBS) performance analysis is shown in Fig. 14. The parameter input interface and returned result interface are shown in Figs. 17 and 18, respectively. The parameters are acquired from the user via HTML forms. The program in the server analyses and translates the inputted data and checks it. If any of the input data does not satisfy the required format, an error is produced. Otherwise the corresponding calculation will run and the result is returned to the Web browser. An example of analysing the performance of an industrial turbo-expander supported by MBS is given as follows.

First, the user logs into the WPDSS system and enters the ASPP server interface as shown in Fig. 14. He can click different hyperlinks in the page and finish different sub tasks. For example, he can enter the parameter form interface of to input parameters to analyse MBS performance. Then he can run different calculation models, each of which will return corresponding results to the Web browser such as modal of

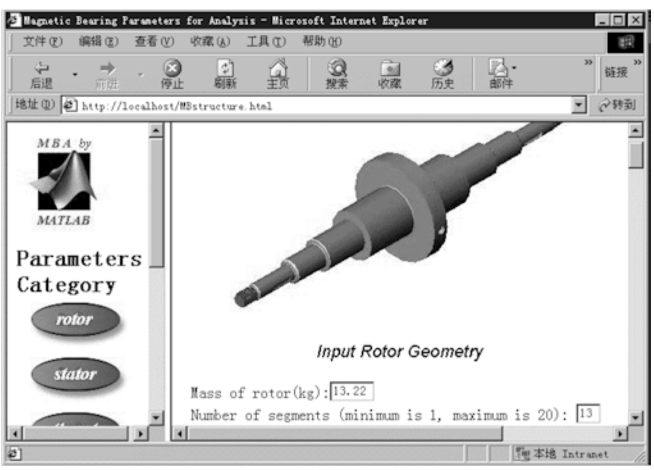

Fig. 17 Input interface of web-based MBS performance analysis

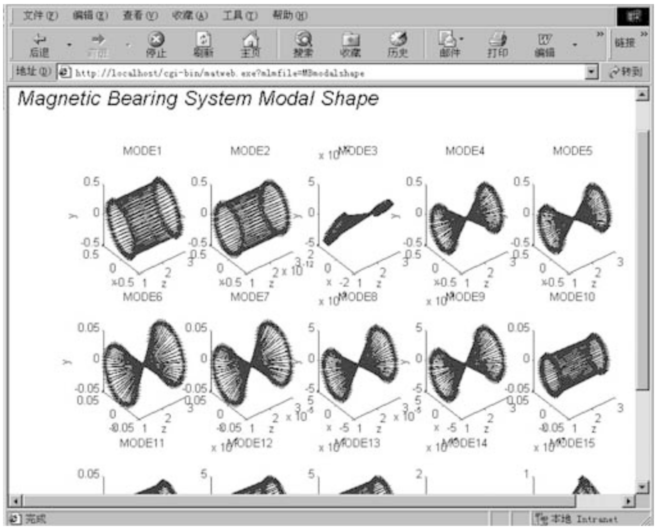

Fig. 18 Returned result interface of web-based MBS performance analysis

rotor-bearing system (Fig. 18), parameters of controller, unbalance response, etc. The Matlab service resources in the ASPP platform can remotely run Matlab applications effectively. This makes the communication and cooperation between designers easy, shortens the lead time for product design and improves the reliability of product development.

5 Conclusions

This paper proposed a solution for a knowledge resources provider over the Internet: the Web-based product development support system that is used to provide complete knowledge resources support for a product designer. The structure of the WPDSS system and its sub-platforms are discussed. Two platforms: the DRIP and the ASPP are constructed to provide design resources and service resources. Key issues concerning implementation of the WPDSS system, DRIP platform, and ASPP platform are discussed in detail. A prototype of the WPDSS system is developed. The works in this paper provide a feasible integrated knowledge scheme for product development.

The two-step knowledge search strategy provides an efficient solution for searching and retrieving knowledge within an Internet environment. The proposed weighted keywords orientating approach (WKOA) captures users requirements more clearly and precisely than more traditional approaches. It also effectively orientates knowledge sites according to the demand of users. The constraint-driven search approach (CDSA) approach obeys the real process of a knowledge search. Through the mapping match between constraints and knowledge, it can provide designers with knowledge exactly according to their requirements within the knowledge site.

The proposed STEP-based knowledge representation method provides an effective scheme to express these product data models used in various platforms in a neutral form independent of product development platforms.

The developed prototype validated the feasibility WPDSS system development. The construction of the design resources provider in the DRIP platform and the service resources provider in the ASPP platform provide solutions that can be referenced to provide knowledge resources to product designer over the Internet.

Currently, the knowledge in the developed prototype of the WPDSS system only contains a few resources which were used to aid turbine engineers in developing a rotor-bearing system for turbine machinery such as a rolling bearing, an analysis program of journal-bearing characteristics, a stability analysis program for the rotor-bearing system and a performance evaluation program of a magnetic bearing system. Since the ultimate objective of the WPDSS system is to form a robust web-based distributive knowledge site league to provide complete knowledge resources for product development, the seamless integration of various knowledge resources and sites, information interaction among various design platforms and expandability of knowledge sites must be further investigated.

Acknowledgements The financial support through the Merit Research Grant of James Cook University (grant No. 14085), Post Doctoral Research Support Grant of Queensland University of Technology of Australia (grant No.1407200018108) are gratefully acknowledged.

References

1. Breslin J, McGann J (1998) The business knowledge repository. Quorum Books, Westport, CT
2. Aguirre JL, Brena R, Cantu FH (2001) Multiagent-based knowledge networks. Expert Syst Appl 20:65–75
3. Marri HB, Gunasekaran A, Grieve RJ (1998) An investigation into the implementation of computer integrated manufacturing in small and medium enterprises. Int J Adv Manuf Technol 14:935–942
4. Qiang L, Zhang YF, Nee AYC (2001) A distributive and collaborative concurrent product design system through the WWW/Internet. Int J Adv Manuf Technol 17:315–322
5. Mills A (1998) Collaborative engineering and the Internet. Society of manufacturing engineers, Dearborn, Michigan
6. Huang GQ, Shi J, Mak KL (2000) Synchronized system for design for X guidelines over the WWW. J Mater Process Tech 107:71–78
7. http://www.matweb.com
8. Pan Y (1998) Integrating distributed online catalogs and design handbooks. Masters thesis, University of Utah
9. Tumkor S (2000) Internet-based design catalogue for the shaft and bearing. Res Eng Des 12:163–171
10. Su D, Amin N (2000) Gear design optimization over the Internet. In: Su D (ed) Internet-based engineering: applications and case studies. Nottingham Trent University
11. Pan PY, Cheng K, Harrison DK (2000) An AI and Internet-based agile system for rolling bearing design. Su D (ed) Internet-based engineering: applications and case studies. Nottingham Trent University
12. Suh NP (1998) Designing-in of quality through axiomatic design. IEEE Trans Reliab 44:256–267
13. Suh NP (1998) Axiomatic design as a basic for universal design theory. In: universal design theory, Shaker Verlag, Aachen, pp 3–24
14. Owen J (1993) STEP: an introduction, information geometers. London, Winchester SO22 5LB
15. http://www.steptools.com