Database and knowledge-base for supporting distributed intelligent product design

Congdu Nguyen and Sungdo Ha
CAD/CAM Research Center, KIST, Seoul, 130-650, Korea
Tel:. +82-02-958-5633 Fax : +82-02-958-9649 E-mail: {congdu, s.ha}@kist.re.kr

Abstract:

This research presents distributed database and knowledge-base modeling approach for intelligent product design system. The product design information in this study is described by a collection of rules and design knowledge that are utilized according to the product development procedures. In this work, a network-based architecture has been developed to enable dispersed designers to simultaneously accomplish remote design tasks. A client/server communication diagram has also been proposed to facilitate consistent primary information modeling for multi-user access and reuse of designed results. An intelligent product design system has been studied with the concepts of distributed database and network-based architecture in order to support concurrent engineering design and automatic design part assembly. The system provides the capability of composing new designs from proper design elements stored in the database and knowledge-base. The distributed intelligent product design is applied to the design of an automobile part as an example.

Keywords: Database, knowledge base, intelligent CAD, and distributed system.

1. INTRODUCTION

Designing a product is a process in which a product is created in order to suit customer's needs. The product may be totally new, or a modification from another products, or a synthesis from preexisting design elements of available entities. The synthesis method is an effective way for all. The fundamental of this method is that designers use their knowledge and experiences to produce outcome under various constraints [1]. However, the method has the limitations of reuse and sharing design knowledge and preexisting designs. Furthermore, design is an interactive process. When a problem is encountered, the designer must go back to an earlier stage and revise design decisions. Current CAD systems, even those specialized for particular applications, oblige the designers to repeat much of his or her work. What is required is the ability to modify an early decision and to confirm that change propagated throughout the design.

Information technology is now continually being developed and attempts to utilize computer in every research areas including design area. In particular, network techniques become widely popular and comfortable for distributed CAD database and design information. Hence, we can effectively work in individual design teams. Anywhere on the network, the staff of a design team can work with other members at a long distance location. The design information can be shared throughout via a network environment (Intranet/Internet) [2, 3, 4, 5, 6,11]. There are many CAD systems developed based on the Internet by applying web technology [2,3]. These systems are very helpful for concurrent engineering designs. It is synonymous with the decrease of product lead-time and the improvement of life-cycle quality of product.

To achieve the desires and address those problems, the paper suggests an intelligent product design system (***IPDS***) based on the network-based architecture with the concepts of distributed database and knowledge-base in order to support concurrent engineering designs and automatic part design assembly. Product designers can easily access to CAD servers to simultaneously accomplish remote design tasks such as making a new design by synthesizing preexisting design elements.

2. INTELLIGENT PRODUCT DESIGN SYSTEM

2.1. Distributed intelligent product design

The development of distributed system relies increasingly on middleware support through the use of software frameworks that provide abstractions such as distributed shared objects, data/information, and services including secure communication, authentication and access control [4]. In mechanical engineering, many applications have been developed in distributed environments [3,5,9].

In a distributed system components located at networked computers can be accessed by relay of messages. Resources are organized or encapsulated as objects in servers and are accessed and managed by the workstations. We can share both hardware components (such as disks, scanners, and printers) and software-defined entities (such as files and database).

Distributed intelligent product designs are sharing CAD database and design knowledge by supporting of database and knowledge-base throughout a network environment. CAD database and design knowledge are

managed separately in different repositories at different locations. A product design in this research is defined as a process of data mapping thereby designers can make multiform products by creating relational data and design knowledge along with design ideas based on a number of design elements.

2.2. Decomposition of design information

IPDS communicates by exchanging messages through a network environment. A part is a packet of geometric shapes and design information. However, we want to use a geometric shape in various application products by controlling its design information. Upon the traditional design, this idea is very difficult and not effective to be put into practice. Consequently, the paper proposes a method of *'decomposition design information (DDI)'*. It is a solution for distributed and reuse of product designs.

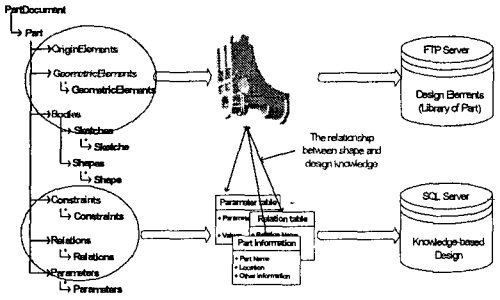

Fig 1 the decomposition structure of a part

The DDI is a significant step for distributed intelligent product designs. It is an approach of dividing design into the two individual parts: geometric shapes and design information. The geometric shapes are essential bricks that constitute design objects. Design information is the necessary structural adhesives, which are design-factors to control the modification of the parts of products [1,7].

Figure 1 shows the DDI method. The geometric shapes and design information of a part are stored at separated locations and different repositories. Because of design classification, the system must generate management information called *relational data* to manage and map correctly design information to the original parts.

2.3. Knowledge base

Design knowledge is the knowledge that represents a design as the information and functions that activate the features in parts and products. The features in mechanical engineering design were concerned with modeling of single parts [7]. Feature-based modeling of single parts was a significant improvement when compared with traditional CAD systems. With features, the designers can speed up the design process as well as provide a means for standardization, thus reducing cost and time-to-market. Also the improvement of the quality of the design and the linking between design and applications such as analysis is expected to be the advantages of feature-based design [8].

The principle of design knowledge in IPDS that is the designer must think about design object for getting the requirements as the design information. In fact, a design object may have several states that are related to its styles. Therefore, we need to compile design information into the knowledge base as functions and entities of design objects. These functions are described as a script language along with parametric designs.

The figure 2 shows a *framework of capturing design knowledge*. It illustrates the conception of ideas for new or revised products. The ideas are from the customers, employers, and new technology. The designers collect all information for analyzing and designing products. The first step is to define the functions called *functional requirements* [1]. The second step is to define the geometric entities of parts, arrange the connections among entities, and group the parts of a family into a group such as subassembly. A complete product is also similar to a subassembly, which is a composite of a series of subassemblies and parts. When all the definitions about design-ideas are finished we then capture these definitions into the knowledge-base system such as the raw design knowledge. Design step is a process wherein we can make parts using the 3D modeler. This step does not only involve making of geometric entities but also other activities of parts. In the whole period of design process, updating the parts in the design library is needed. Design library is a storehouse used to keep all designed products and parts.

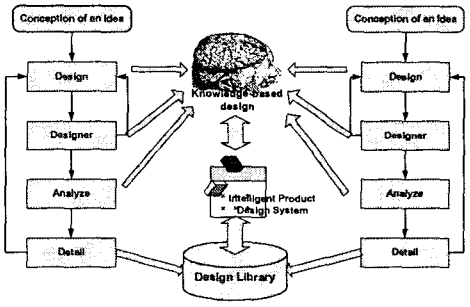

Fig 2. Sequence of events prevalent for organizing and extracting design knowledge

Controlling the parts of a product is a manipulation on their relations and parameters [13]. The relations can be *Rules, Checks and Formulas*. Check is used to test a variable or an equation that satisfies a condition. If it does not satisfy a condition the system should inform a message to product designers. For instance, when mass is a variable.

mass < 10kg : "Ensures that mass is less than 10 kg"

Formula's content is an assigning expression that assigns a value to a variable. For instance, when we have variables such as *mass, width, height, depth* and *density*

*mass = (width*height*depth)*density*

Rule can include many commands, and also it may be a rule, i.e., *IF (condition) {command} ELSE {}*, or may be *an assigning expression*. For instance,

if (mass>2kg) { depth = 2mm } else { depth = 1mm }

or it can assign values between two variables:

`OP1_FLFloorMtg_X` = `Brkt\IP2_FLFloorMtg_X`

3. SYSTEM CONFIGURATION

3.1. A network architecture

Recently, Internet technologies have been applied for many fields of studies. Mechanical engineering is one of the fields, which has explored so much in the areas of CAD, CAM and CAE [9, 10, 11]. The advances of network and information technology make it possibly construct large-scale high performance distributed computing environments [9]. At the same time emerging applications require the ability to exploit diverse and geographically distributed resources.

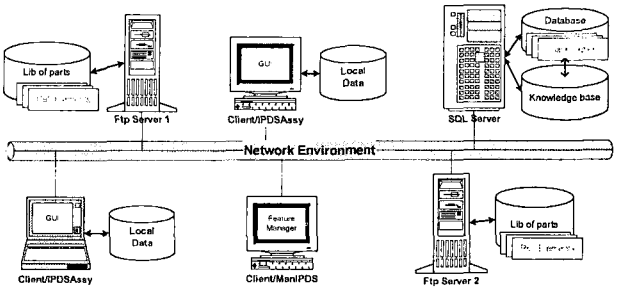

Fig 3. A network-based architecture of the IPDS

The extraordinary growth of the Internet technologies and the consequent creation of new tools and applications enable the development of network-based systems including commercial and corporative information systems. They are constructed as integrated systems. Correspondingly, we build new modeling techniques to assist the IPDS by mining the defined design knowledge and data files. Currently, there are no methodologies that deal with the entire process of modeling these applications. Recognizing that, the paper introduces the network-based architecture as a distributed computing environment for setting up the IPDS (see figure 3).

3.2. Client/server communications

The IPDS is client/server applications connecting remote parts, database, and knowledge base to a local CAD system. The communication of client and server is shown in figure 4. The typical communication structure of the IPDS is implemented client/server communication architecture that offers multiple users access to the same server as well as easy modification its functionalities. Figure 4 shows a schematic architecture of client machine. Intelligent CAD system allows designers to handle CAD database and knowledge base.

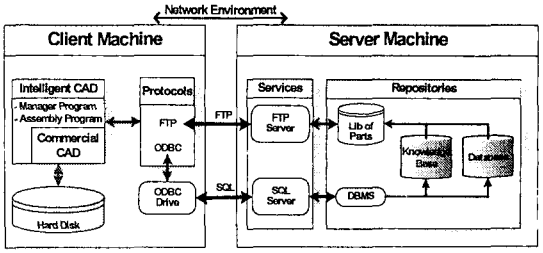

Fig 4. A client/server communication

The components of client machines are user-specific applications and network protocols used to communicate with server machines. The ODBC Drive in client machine is a getaway supporting client machine to freely communicate with SQL servers. The components of server machine are *FTP service*, *SQL service* and *repositories*. The CAD geometries and the complete products are stored in *Lib of Parts*. Design knowledge and relational data should be managed by DBMS (Database Management System).

4. IMPLEMENTATION

4.1. Integrated database and knowledge base system

CAD database and knowledge-base are product design information used to control design process. Basically, they are data in the *database tables*, and the relations among product design information are the relationships among these tables. In this work, to create the relational database tables for storing all product design information is need. The information we have to consider is the products' information and constraints, the parts' profiles, the parameters and the relations of products and parts.

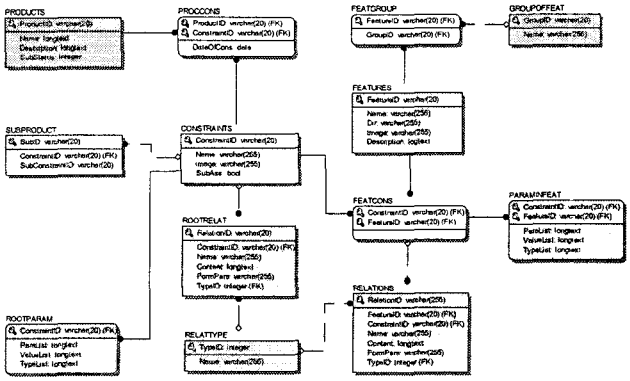

Fig 5. Database tables of the IPDS

We arrange that the reference files of the parts are stored in distributed file system and the whole information related to the parts and products as well as the constraints should be collected in the integrated knowledge-base and database system. The figure 5 shows the relational database tables of IPDS.

4.2. Client application

Client application is a part of the IPDS which consists of two programs: ManIPDS and IPDSAssy. The two programs connect to CAD servers for getting product

design information and communicate with a solid CAD system for modeling product designs. ManIPDS program guides the product designers for error-free design and manages the design elements and products' properties. The IPDSAssy program is useful for assembling designed products. The product assembly algorithm is presented in IPDSAssy program. The whole process of product assembly is automatic which means that the intervention of product designers is unnecessary. The programs are coded by Visual Basic 6.0. CATIA system is used as a 3D modeler. The programs handle product designs on the CATIA system by the supporting of functional automation clients [12].

The figure 6 depicts *Product Assembly Algorithm (PAA)*. This algorithm holds for both complete product (assembly) and incomplete product (subassembly). It is a method of searching in depth, presented in IPDSAssy program. The execution steps of the algorithm are shown in table 1.

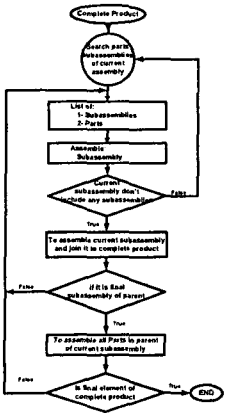

Fig 6. Product assembly algorithm

Table 1. The execution steps of the PAA

First step:
To search parts, subassemblies in complete products
Second step:
To assemble structure of subassemblies then assemble parts
1. To assemble structure of each subassemblies
2. To check if subassembly includes any subassemblies
   - o if Yes: go to first step
   - o if No: assemble all parts in this subassembly and do to third step

Third step: to check
1. If the current subassembly is final assembly of parent
2. If not go to second step

Fourth step: to check
1. If it is final elements of complete go to END
2. If not then go to second step

5. A CASE STUDY

5.1. Control of parts and features of a product

To fix a product design is to control the relations of its parts. The relation is composed of three types: rule, check, and formula. In this work, we describe the technique of controlling the parts of a product as well as features of a part by managing parameters/values. The Figure 7 describes the relationship between *Bracket* and *Metaliner*. Here, the position and the offset of *Metaliner* depend on their locations on *Bracket*. Hence, how can we decide the relationship among them? And how can we locate the Metaliner to the correct position whenever its position on Bracket is to be changed? These problems can be solved as follows.

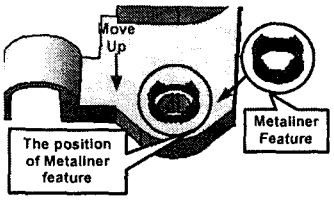

Fig 7. The relationship between parts & features

Assume that Metaliner and Bracket are of an assembly level. To match Metaliner on Bracket we need to get the position values of Metaliner on Bracket by compiling the rules in the relations. The two parts have the same parent that is 'Brkt_Assy'. In Bracket, we define two parameters: *IP1_FLFloorMtg_Y* and *IP1_FLFloorMtg_X*. In the parent 'Brkt_Assy', we also define two output parameters that are *OP1_FLFloorMtg_X`* and *OP1_FLFloorMtg_Y*. The following script gets the position values of Metaliner on Bracket to the parent.

`OP1_FLFloorMtg_X` = `Brkt\IP2_FLFloorMtg_X`
`OP1_FLFloorMtg_Y` = `Brkt\IP2_FLFloorMtg_Y`

In Metaliner, we define two parameters: *IP2_FLFloorMtg_X* and *IP2_FLFloorMtg_Y*. We now assign the output values in the parent to the Metaliner by the following relation script.

`MetalLiner\IP2_FLFloorMtg_X`=`OP1_FLFloorMtg_X`
`MetalLiner\IP2_FLFloorMtg_Y`=`OP1_FLFloorMtg_Y`

5.2. Design of automobile part

The design of an automobile part applied in the IPDS is an example. The automobile part applied in this study is an Automatic Transmission Lever (ATL). There are several types; here we have shown one style of it that is *GK product type*. GK product type is constructed from four subassemblies and one part: *GK_Lvr_Assy*, *ThreePieces*, *Brkt_Assy*, *BTSI_Assy*, and *GK_Knob*, respectively. All the information of GK product type can be controlled and managed by ManIPDS program by creating a new one, selecting its parts and subassemblies, and inserting design knowledge. The form in figure 8, it is one of management forms, supports managing product design information.

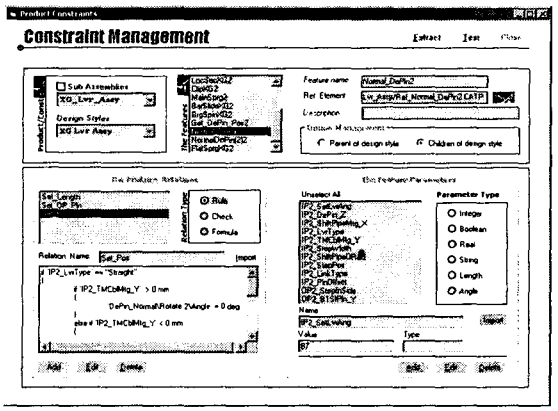

Fig 8. Constraint management form for handling product design information

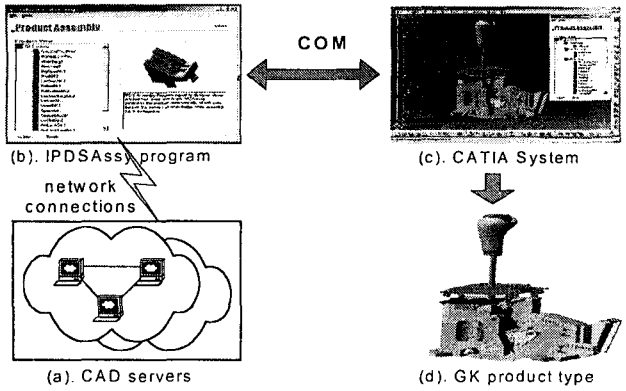

Fig 9. The activities and the result of assembling GK product type

With product design information, we create GK product type using IPDSAssy program. IPDSAssy connects to CAD servers in order to mine product design information while creating product. The figure 9 shows the activities and the result of assembling GK product type.

6. CONCLUSIONS

The paper has developed an intelligent product design system based on a network-based structure and the concepts of distributed database and knowledge-base. The system provides designers with the capability of composing new designs from proper design elements stored in databases and knowledge bases. A product design in our system is defined as a process of data mapping thereby designers can design multiform products by creating relational data and design knowledge along with design ideas based on a number of design elements.

The client application, Intelligent CAD system, has developed. The Intelligent CAD system is a combination of ManIPDS and IPDSAssy programs. The methodology to deal with diversified knowledge has been presented in the ManIPDS program that helps designers make design decisions. The product assembly algorithm has been applied to IPDSAssy program. Finally, the paper has shown the result of designing an automobile part.

References

[1] Won-Ki Kim, Taesoo Kim, Sung Woon Cha, Sungdo Ha, Development of Intelligent Product Design System Using Functional Features, CIRP Proceeding, Vol 35, pp 383-386, 2002.

[2] G.Q. Huang, K.L. Mak, Design for manufacturing and assembly on the Internet, Computer In Industry, 38, pp 17-30, 1999.

[3] D. Xue, Y. Xu, Web-based distributed system and database modeling for concurrent design, Computer-Aided Design, 35, pp 433-452, 2003.

[4] James Ang, Distributed Environment: A concept framework using objects, Data & Knowledge Engineering, 14, pp 191-202, 1994.

[5] Z. Adamczyk, D. Jon´czyk, K. Kociotek, A new approach to a CAD/CAM system as a part of distributed environment: Intranet database, Journal of Materials Processing Technology 133, pp 7–12, 2003.

[6] Wang Xiaotong, The IDS model of Intelligent Design System, Computers & Structures, 66, pp 579-586, 1996.

[7] Chongsu Kim, Peter J.O'Grady, representation formalism for feature-based design, Computer-Aided Design, 28, pp 451-460, 1995.

[8] G. Brunetti, B. Golob, A feature-based approach towards an integrated product model including conceptual design information, Computer-Aided Design, 32, pp 877-887, 2000.

[9] Shouqin Zhou, Kwai-Sang Chin, Youbai Xie and Prasad K. D. V. Yarlagadda, Internet-based distributive knowledge integrated system for product design, Computer in Industry, vol 50, Pages 195-205, 2003.

[10] F. Mervyn, A. Senthil kumar, S.H. Bok, A.Y.C. Nee, Development of an Internet-enabled interactive fixture design system, Computer-Aided Design, vol 35, Pages 945–957, 2003.

[11] N. Shyamsundar and Raid Gadh, Internet-based collaborative product design with assembly features and virtual design spaces, Computer-Aided Design, vol 33, Pages 637-651, 2001.

[12] Automate Application with Visual Basic, Visual C++: http://msdn.microsoft.com/officedev/, 2000 & http://support.microsoft.com/support/officedev/offdevout.asp.

[13] Dassault System, CATIA V5 Online Help, 2000