

Functionality-Based Modular Design for Mechanical Product Customization Over the Internet

Yan Wang

Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA
15261

Bartholomew O. Nnaji*

Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA
15261

Received: February 20, 2001; Accepted: May 2, 2001.

Abstract: There is now greater demand for product customization. For large-scale mechanical product customization, there is great need to develop new methods that will ensure shorter design cycle, shorter time to market, reduced life cycle cost, and higher product quality. The emergence of the Internet makes mass customization possible. It is essential that tools that will support customer participation in product design and realization be developed. The critical tool is an easy-to-use Computer-Aided Design (CAD) system over the Internet where customers' preferences can be captured by functionality-based formalism. A new conceptual design methodology, functionality-based modular design, is presented in this article with the capability of high flexibility and speed, which supports mechanical product customization. A model is developed for modularization to accommodate different system behavior requirements from users of CAD. An XML-compatible language, PML, which has good properties of interoperability, scalability, and extensibility, is employed for product representation. This modular conceptual design approach is implemented in a new design platform.

Key Words: Conceptual Design • Customization • Modularization • Functionality • Computer-Aided Design • Computer Integrated Manufacturing • Rule-Based Reasoning • Case-Based Reasoning • Knowledge-Based Design • PML • XML

I. INTRODUCTION

The emergence of Internet technologies is having great impact on manufacturing industries. The world-wide computer network has led to a considerable shortening of time and space and has also led to significant changes in the traditional structure of supply chain management. This revolution is also facilitating the evolution of "paperless" manufacturing, which has been the enduring dream of Computer-Integrated Manufacturing (CIM). Customers will be able to participate in product

design activities at different stages over Internet without the restriction of time and space.

Normally, in the development of new product, the designer is presented with specifications that essentially are preferences and constraints of the customer. However, there is no existing mechanism to transform these conceptual specifications into form. This is largely due to the complexity of the design problem that requires capturing design intents and multidisciplinary constraints. Currently, the typical way to help customers to reorganize the functional requirements along

* Corresponding author.

with other requirements (cost, manufacturability, etc.) comprehensively is the direct consultation of domain experts. The internet can serve as a new medium for the communication among customers, designers, manufacturers, and vendors. One of the challenges for product customization over the Internet is the process of capturing customers' intentions, requirements, and specifications on a new product, especially for those customers who may not have engineering design experience. To understand customers' requirements and accommodate them in the design process, good Computer-Aided Design (CAD) systems, which should be human-friendly, easy-to-use, intelligent, and experienced assistants, are crucial. In this article, functionality is perceived as the hub of design. A method of functionality-based modular design is presented that advocates modularization and knowledge reuse in mechanical design. This methodology is implemented in a new design platform.

II. THEORIES OF DESIGN

Considerable research effort has been spent on general theories about product design. At different levels of abstraction, there are different categories. According to Alexander,¹ design is based on the idea that every design problem begins with an effort to achieve fitness between two entities: the form in question and its context. The form is the solution to the problem; the context defines the problem. In other words, when we speak of design, the real object of discussion is not the form alone, but the ensemble comprising the form and its context. Good fit is a desired property of this ensemble that relates to some particular division of the ensemble into form and context. Freeman and Newell⁸ defined the design process as the process of transformation of functional requirements into structure. They were among of the early researchers to introduce the notion of functional reasoning as the basis for design. The Topological Design Theory²⁴ is based on a topological model of human intelligence. It starts with some definitions (entity set, item of attribute, etc.). Based on some axioms (recognition, correspondence, and operation), it deduces theorems that model designer's designing process. The designing is the process of mapping between functional space and attribute space. Suh¹⁸ defines design as the culmination of synthesized solutions in the form of product, software, processes, or system by the appropriate selection of design parameters that satisfy perceived needs through the mapping from functional requirements in the functional domain to design parameters in the structural domain. The Domain Theory² uses Function/Means-

Law and three domains for describing and synthesizing a machine system. In the transformation domain the purpose of any product or machine, which is to support a transformation or process, is captured. In the organ domain, a materialized composition of products, which creates functions or effects, is the focus. In the part domain, machine parts and their assembly relations are considered. Design is a purposeful activity;⁴ it involves a conscious effort to arrive at a state of affairs in which certain characteristics are evident. Analysis, synthesis, and evaluation are three phases of design.

Although researchers are using different models (e.g., Function-Attribute,²⁴ Function-Behavior-State,²⁰ Function-Behavior-Structure^{3,15}), it is generally accepted that design is the mapping process from functions to forms according to designers' specifications or requirements. Typically, these requirements will be examined for functional, technical, operational, and financial feasibility by the assistantship of domain knowledge sources. For product customization, functionality is the starting point of the design and the ultimate goal of a new product. Functionality is the center of the design activity. As shown in Figure 1, the original functions should be considered in different design stages, whether for form generation, material selection, reliability analysis, or cost management.

Another reason that functionality plays an important role in product customization is that it is intuitive for human beings, as new design is launched by the sense of not satisfying the existing ones. Customers can utilize functionality as a channel to participate in design so as to specify requirements, issue verifications, and make decisions effectively with the computer.

III. CURRENT CAD SYSTEMS

Computer-Aided Design (CAD) software is playing an important role for product design process. The early CAD systems just aided designers to finish drafting. Current CAD tools now include features that express the mechanical meanings of geometric configurations on a part.

The hub of the conventional mechanical CAD systems is the geometric modeler. The task of the computer is focused on capturing geometric information, whereas nongeometric technical information (e.g., material properties, functional requirements, manufacturing methods, etc.) and administrative information (e.g., bill of materials, process planning and scheduling, cost estimation, etc.) are mostly neglected. This brings up the following problems of current CAD systems:

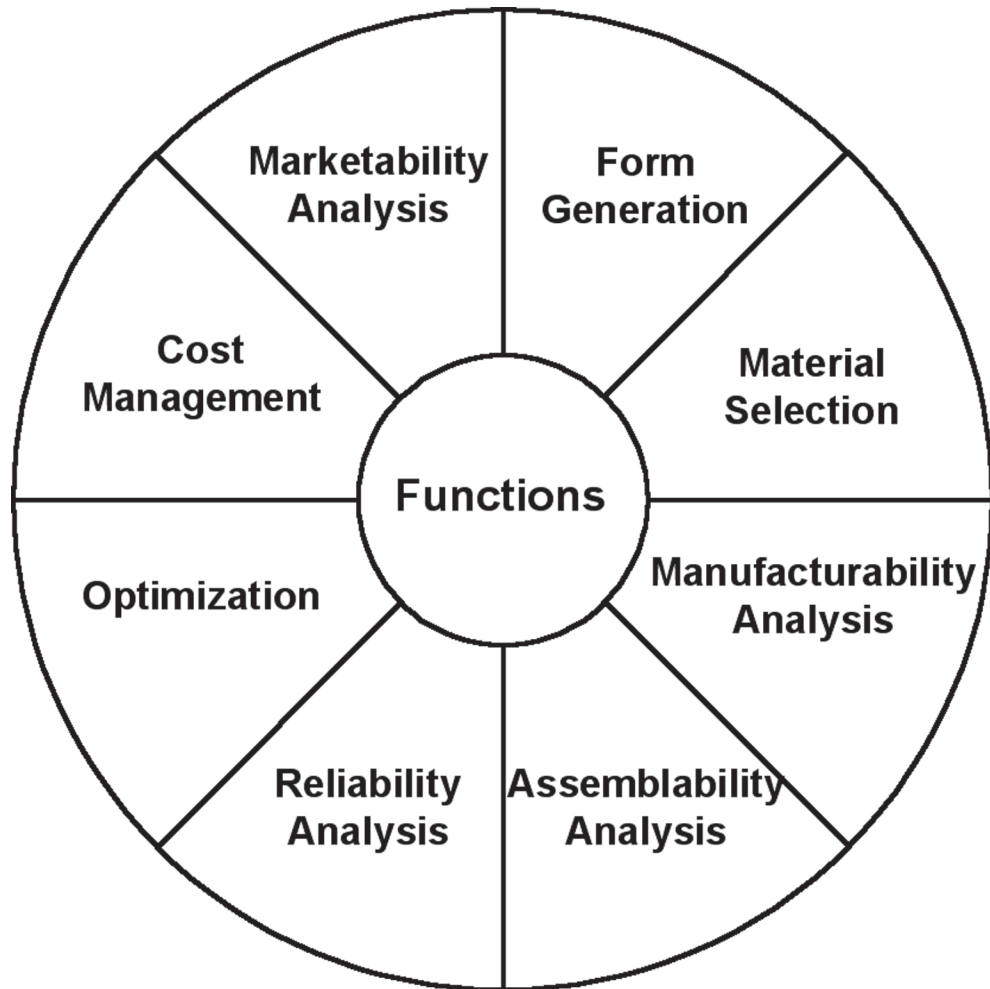


FIGURE 1. Function as the hub of design.

1. Current CAD systems increase the risks of downgrading synthesis during analysis. Mechanical designs need to consider material properties, tools selection, manufacturing, assembly processes, etc. They are much more complicated than pure electrical product designs from this perspective. The major reason for this is the difficulty of functional modularization and discretization of mechanical parts. Within electrical design, small modules such as resistors and capacitors are the lowest level to consider for designers. Computer-Aided Drafting is not enough for mechanical design. Integrated design tools capable of reasoning about materials, process planning, cost estimation, and other related issues are needed.
2. Current CAD systems are not “intelligent” enough to support error checking. Designers may violate rules of geometry, ergonomics, manufacturing, or assembly without awareness. For example, in a door lock design, a number of errors could be made: a dangling edge is created; the handle for human users is too short and edges are not rounded; or screw holes are too close to edge thus punch process may disrupt the edge. An intelligent system is needed to aid users and check these errors.
3. Current CAD systems lack mechanisms for “amateur” participation in a design process. In current CAD systems, basic geometric information of a part such as geometric shape, dimensions, and features is specified by designers directly and explicitly. Thus, this is largely dependent on engineering knowledge of designers. For customers, they usually lack this knowledge. Their requirement may just be “I need a structure here to place the shaft I just created” instead of “a 9 inch NPT threaded hole at the center of the frame”, or simply “I

need a tool to open a can”. A CAD system that cannot follow customers’ intentions will not establish effective human-machine communication channels. Customers’ specifications tend to be ambiguous.

Research on advancing of the conventional CAD paradigm, namely, Intelligent Computer-Aided Design (ICAD), has been on-going in the last 20 years. In general, ICAD aims at more intelligent use of computers to aid in design process.

A group of researchers^{19,21,22} developed an IDDL (Integrated Data Description Language) to code design knowledge in the IICAD (Intelligent, Integrated, and Interactive Computer-Aided Design) system. It uses the logic programming paradigm to express the design process for manipulating design objects, whereas the object oriented programming paradigm is used to express design objects. The PERSPECT system⁵ is a design tool that aims to support the effective utilization of experiential knowledge in numerical engineering design, in which design knowledge is generated by learning about their existing design domain (Domain Exploration) and “sharing” the learning activity between designers and computers (Shared Learning) so as to ensure the knowledge represented in computers is understandable to designers. CASECAD^{10,11} is a multimedia case-based design system that integrates traditional CAD and case-based reasoning. It stores and utilizes design cases in both textual and graphical modes. The main modules include case memory, case base manager, case-based reasoner, CAD package, and graphical user interface. The system IDEAL³ uses analogical reasoning to retrieve knowledge of a familiar problem or situation that is relevant to a given problem and transfers that knowledge to solve the current problem, which is cross-domain case-based reasoning. None of these systems are able to provide seamless transition from functionality to form.

IV. DESIGN MODULARITY

The market competition demands shorter design cycle for a new product. In order to reduce time, cost and risk for new product development, there is a need for new design paradigm based on functionality and modularization. Modular design has been adopted in a number of industries, such as electronic products and software development. Each module has certain functions and is self-contained, that is, each module’s functions do not depend on other ones. The designers need not design product from scratch each time. This approach reduces design time and encourages design automation.

A number of researchers have attempted to implement design modularization for mechanical products. Erixon⁷ presents a Design for Modularity technique — Modular Function Deployment (MFD), which consists of five major steps: Quality Function Deployment (QFD) analysis; modular concept generation; Module Indication Matrix (MIM) to identify possible modules; Modularity Evaluation Charts (MEC) for thorough evaluation; Module Indication Matrix to identify opportunities for manufacture and assembly.

Similarly, ‘Design Reuse’ aims to maximize the use of this concentration (of engineering creativity and expertise in design) by the reuse of successful past designs in part and in whole for new designs.¹⁷ The Engineering Design Conference ’98⁶ with design reuse as its main theme was held in UK, June 1998. Research on focused innovation, cognitive studies, computational perspective, use of standard components, design reuse tools, methods, etc. were presented.

Rather than product or part modularization, which is directly beneficial to designers and manufacturers as a phase of the whole production process, we focus on design modularization from CAD’s viewpoint, which emphasizes the methodology for easy-to-use CAD. We are proposing a method of functionality-based modular design for mechanical products. This approach intends to achieve functional modularization of design object so that designers can interact with computers comfortably and computers can aid designers to complete design effectively.

The advantages of design modularization are

1. Reduce the time of design. Instead of designing each product from basic geometric elements, modules are taken based on their functions. Design engineers can reuse the modules without changes or change some sections of the modules according to the new product requirements. The design history generated for re-designing the unchanged sections will be saved.
2. Reduce the costs of design. Besides the time spent on design, the resources for product analysis and evaluation are also reduced. Modularity helps to avoid errors and uncertainty during the design process.
3. Knowledge reuse. The design knowledge and experiences are stored physically as part of modules. The risk of defective design is reduced. The nature of design is to use old knowledge and acquire new knowledge about the relationship between functional space and form space. Knowledge accumulation and storage is critical in design activities.

4. Encourage concurrent engineering. The module is the media of communication among customers, market analysts, design engineers, manufacturing engineers, and other participants of design. A common language based on module is beneficial for understanding each other.

The only possible pitfall in applying design modularization is the possibility that a designer relies solely on modules to develop new products. Another pitfall could be when modules are not well configured. Accurate module or accurate knowledge about module is the premise of the system.

Modularity and flexibility are two conflicting aspects when modular design systems are developed. Modularity is helpful for routine design, while flexibility is essential for creative design. Both sides should be considered in a practical CAD system. It is an art to achieve a good balance between those two aspects. One compromise is to use hierarchical modularity and define lower level modules if necessary to maintain sufficient flexibility. We define *modus* to be the smallest module that is an independent functional unit that contains one or more features.

Functionality-based modular design is a promising method that facilitates Internet-based mechanical design and customization. Historically, the emergence of *feature-based design* has changed the use of lines and curves in mechanical design (which have no engineering meaning) to *features* that have an engineering implication. This notion increased the abstract level of design reasoning and somewhat simplified the work of mechanical design engineers. As illustrated in Figure 2, early designers used simple geometric shape to model the product from view 1. The emergence of features in design is showing in view 2. The same product can be modeled in different ways. The functionality-based modular design intends to help designers to model product from functionality viewpoint, as shown in view 3. Functionality is the focus of design for supply chain participants as well as customers. Functionality module as defined by *modus* will be the commonly used expression during a design process. The goal is to find a path such that design language and methods are meaningful for customers apart from engineering implication. It is clear that functionality-based design allows designers to deal directly with the intentions of design rather than the current indirect approach.

V. FORM GENERATOR IN PEGASUS DESIGNER SYSTEM

Pegasus designer system is a CAD system under development at the University of Pittsburgh, which aims

to assist supply chain participants and customers to participate in new product design over Internet. The goal is to develop a new type of distributed CAD system that supports product customization. Ease of use is the crucial requirement for this new design paradigm. *Pegasus* system is the fusion of CAD technology, Internet technology, and knowledge engineering, which facilitates information flow among customer, designer, manufacturer, and other stakeholders for product design. One of the functionalities of this system is the ability to allow a customer to generate mechanical product from concept and specifications to physical form, by the aid of knowledge engineering scheme. This task is accomplished by Form Generator of *Pegasus* system.

A. Functional Modular Representation

Although different design theories mentioned before have stated that design process is the mapping process between functional space and physical space, few concrete mapping methods have been explored. Relational approach¹⁴ searches for working principles to fulfill subfunctions. New design for a specific functionality can be derived by selecting the combined working principles and working structures. Case-Based Reasoning (CBR) has been used in various design fields, such as architecture, software engineering, structure engineering, etc.^{12,13}

The mapping process used here is not directly from functionality to component, which is employed in other approaches.^{9,16} Rather, mapping is from function to geometric module by the aid of knowledge base. The hypothesis here is that any function can be implemented by one or more features. We define *modus* as the basic design operation unit that embodies functionality instead of feature or basic geometric elements such as line, circle, etc. The smallest *modus*, which includes one or more features, implements one function. The hierarchical relationship among feature, *modus*, and component is shown in Figure 3.

Functionality is important in conceptual design stage, as it is the starting point to define a new design. In a top-down approach, conceptual or preliminary design is the process of functional decomposition. We disintegrate the functionality of products into *Fundamental Functionality Elements* (FFE) — for example, support, hold, transport, convey, conduct, etc. It should be realized that these FFEs are domain specific. Different types of products have their own terminology. Defining FFE enables the system to define functional module during design analysis and synthesis.

The element in functional space is FFE, while in physical space the element is *modus*. The relationship

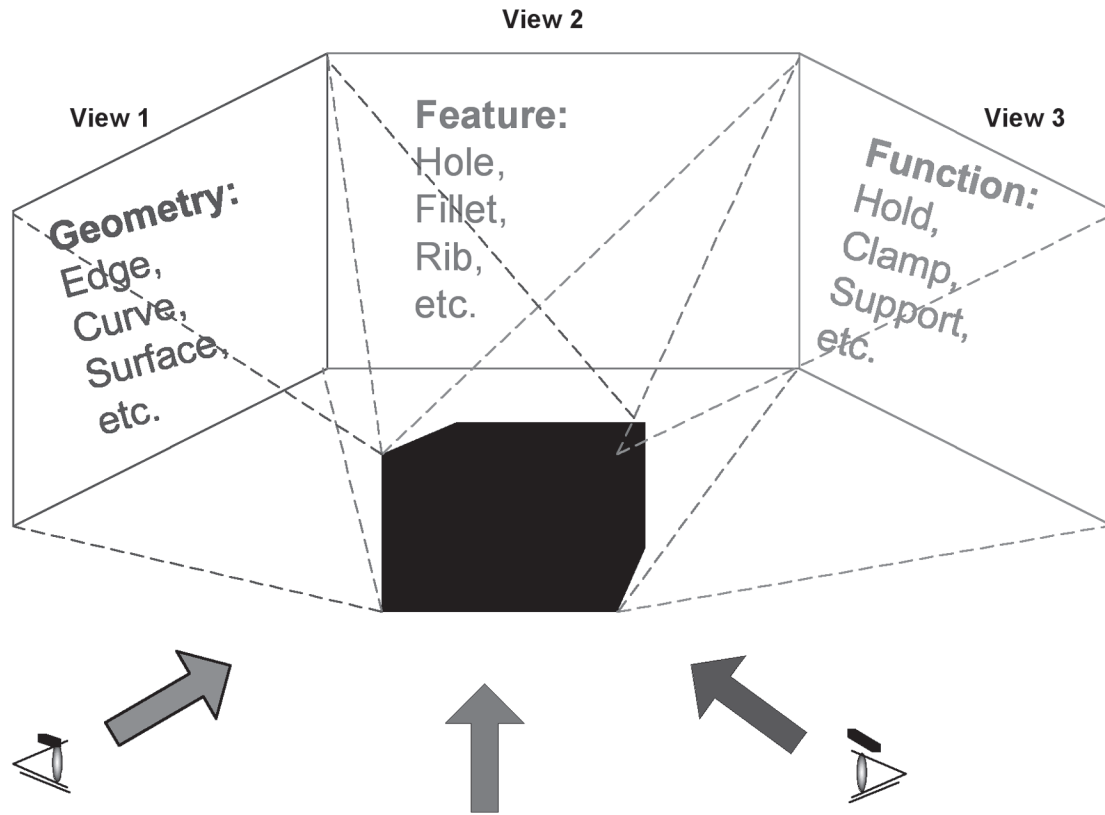


FIGURE 2. Different points of view for product design.

between FFE and modus is many-to-many, which is shown in Figure 4. A *Fundamental Modus* (FM) can be defined as a modus that is only related to one *Fundamental Functionality Element*, while a *Composite Modus* (CM) is related to more than one *Fundamental Functionality Element*. In the example of Figure 4, moduses m1, m3, m4, m5, are FMs and modus m2 is a CM. Because design is a particular activity of human being, human intelligence and knowledge contribute significantly during this process. The mapping from functionality to form will proceed appropriately under the guidance of knowledge base. These preferred shapes or forms are chosen according to past cases, experience and/or knowledge. The advantage of defining modus in the mapping is to increase the flexibility of conceptual design, comparing to component-based mapping.

Formally, we define F as the set of *functionalities*. f is an element of F ($f \in F$). Define M as the set of *modus*. m is an element of M ($m \in M$). The mapping from functionality to modus is defined as $p : F \rightarrow M$, which is the process of *modus retrieval*. The mapping from modus to functionality is defined as $p' : M \rightarrow F$, which is the process of *functionality retrieval*. These mappings are not one-to-one because of their inherent

problem-dependent properties. Further, we define C as the set of *contexts*, which captures the application environment of the target product. c is an element of C ($c \in C$). Then a more restricted mapping that tends to reduce the search space is $p_c : F \times C \rightarrow M$. Similarly, $p'_c : M \rightarrow F \times C$.

An abstract model of modus is built as follows: modus m can be defined as $m = \langle S_F, C, G, R, S_O \rangle$, where S_F is the set of functionalities that modus m is related to; C is the context that modus m is associated with; G is the geometric shape that modus m has; R is the material of which modus m is made; S_O is the set of other information that modus m has, which makes this model extendable.

The important relationships among moduses include functional relationship and spatial relationship. The functional relationship of moduses is the relationship of functionality sets associated with moduses. They have relationships of *union* (\cup), *intersection* (\cap), *difference* (\setminus), etc. The spatial relationship is the relationship among geometric entities. We define a World Coordinate System (WCS) located at the origin point of 3-dimensional Euclidean space. WCS is the reference of the whole product. The geometric

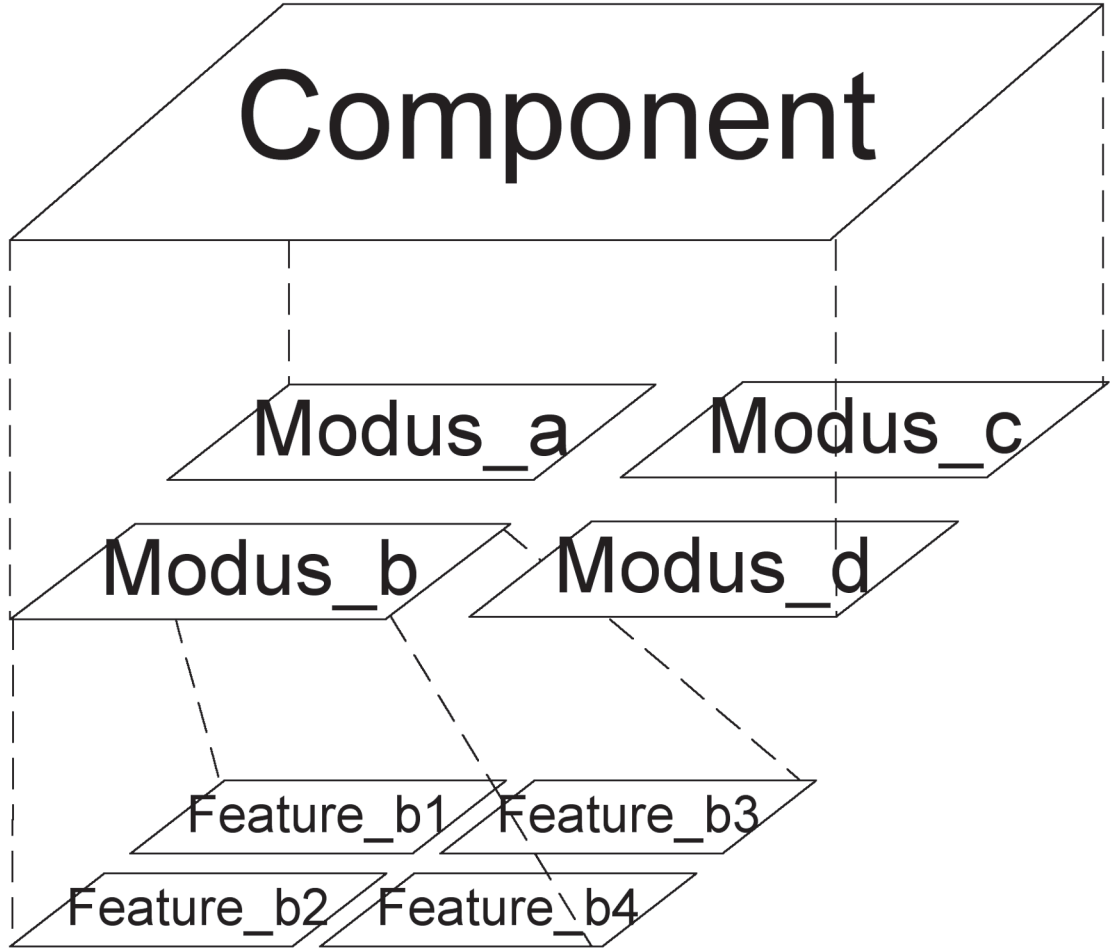


FIGURE 3. Relationship among feature, modus, and component.

entity of each modus has its own coordinate system, called Modus Coordinate System (MCS). The spatial relationship among moduses is expressed in terms of spatial relationship of MCS's that moduses are located in. For convenience of expression, we define a symbol of *reference* ($\langle TR \rangle$), in which TR is the *transformation* of coordinate systems. A *transformation* is the combination of a series of *translations* and *rotations*. TR is the transformation that one MCS or WCS needs to finish to get the new MCS. For example, $G_1 \langle TR | G_2$ means geometric entities of modus 2, G_2 , is positioned in MCS_2 that is referring to MCS_1 of geometric entities of modus 1, G_1 , after a coordinate transformation.

The flexibility of the modularity is enhanced by a special binary operation, *composition* (\oplus), between two moduses. For two moduses, $m_1 = \langle S_F^1, C_1, G_1, R_1, S_O^1 \rangle$, $m_2 = \langle S_F^2, C_2, G_2, R_2, S_O^2 \rangle$, define:

$$m_1 \oplus m_2 = m_3,$$

$$\text{where } m_3 = \langle S_F^1 \cup S_F^2, C_1, G_1 \langle TR_1 | G_2, R_1, S_O^1 \rangle.$$

$$m_2 \oplus m_1 = m_4,$$

$$\text{where } m_4 = \langle S_F^1 \cup S_F^2, C_2, G_2 \langle TR_2 | G_1, R_2, S_O^2 \rangle.$$

Along with a *composition*, a *transformation* should be defined such that the relative positions of two moduses can be determined in 3D space. An example of *composition* is shown in Figure 5. This operation is not permutable, i.e., $m_1 \oplus m_2 \neq m_2 \oplus m_1$.

\oplus can be looked as an operation of function and form union in the set of moduses. The result of the operation *composition* is a composite modus that has the functional and geometric information of both modus operands. After geometric entities of moduses are generated and necessary *compositions* are finished, each modus can then be working as an independent component. *Composition* combines two moduses, thus reduces the number of components. The design process

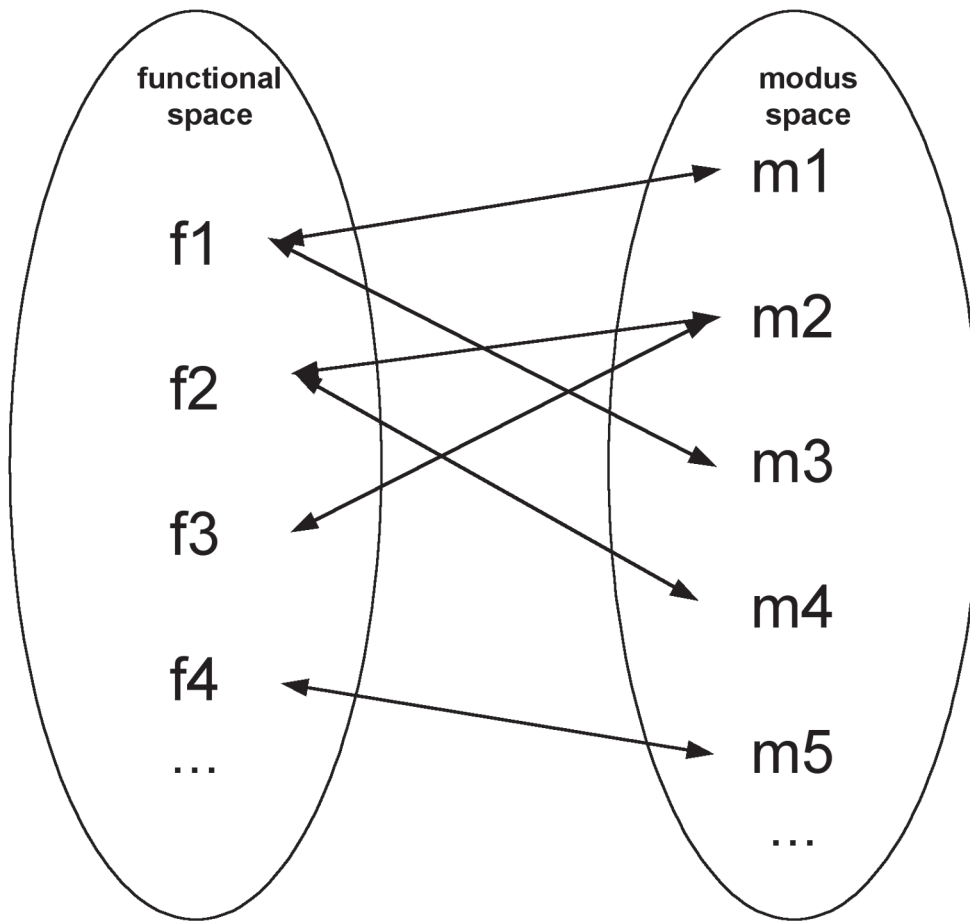


FIGURE 4. Mapping from functional space to physical modus space.

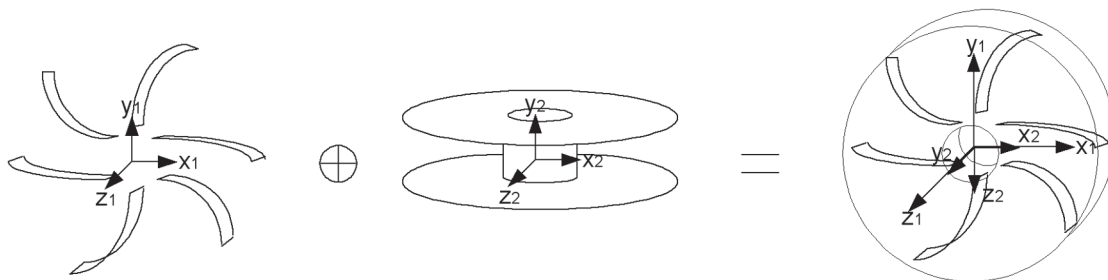


FIGURE 5. Modus composition operation.

can then be looked as the process of functional decomposition in functional space and modus composition in modus space. The structure of a product can be graphically modeled by a *hypergraph*, which is shown in Figure 6.

B. Functional Reasoning

The *modus retrieval* based on functionality, context, and other specification is accomplished by the aid of a reasoning engine. Both rules and cases are necessary for the reasoning. Scientific rules, such as conservation of energy, principle of hydraulics, etc., are important at the basic and abstract stage of reasoning. Engineering principles then are followed when reasoning comes into a more specific area.

A reasoning procedure example during a centrifugal pump design is shown in Figure 7. When the design process starts, the designer system asks the user the principle functional requirement of the target new product. According to scientific rules, the reasoning engine deduces that this device should be able to transfer kinematic energy of water to potential energy. Then a pump comes up because the knowledge base has the record that pump will accomplish the required job. There are different kinds of pumps available. After acquiring the parameters of working environment of the target pump, the system is able to suggest a cen-

trifugal pump to customers according to the rules of pump type selection. Then the system can help users to finish the functional decomposition and choose a series of *Fundamental Functionality Elements* needed during component-level analysis, according to both scientific rules and engineering rules. The *modus retrieval* is the following procedure that selects moduses, either *Fundamental Modus* or *Composite Modus*, which contain geometric, material and other information. At the modus level, users give detailed specifications to moduses, such as dimensions, materials, etc. Each modus then can become a separate component. If necessary, *composition* operations are conducted to reduce the number of components. In this example, blades, which propel water, and the impeller hub, which supports blades, can be integrated into one component. After all components are finished, a product assembly model can be built using spatial relationships among components.

The whole design process is aided by rule-based reasoning and case-based reasoning. Rules are applied for helping users' decision making. Cases are stored at the level of modus in the system.

C. Modeling Language

Historically, CAD format has presented a problematic barrier for CAD information exchange. To encourage

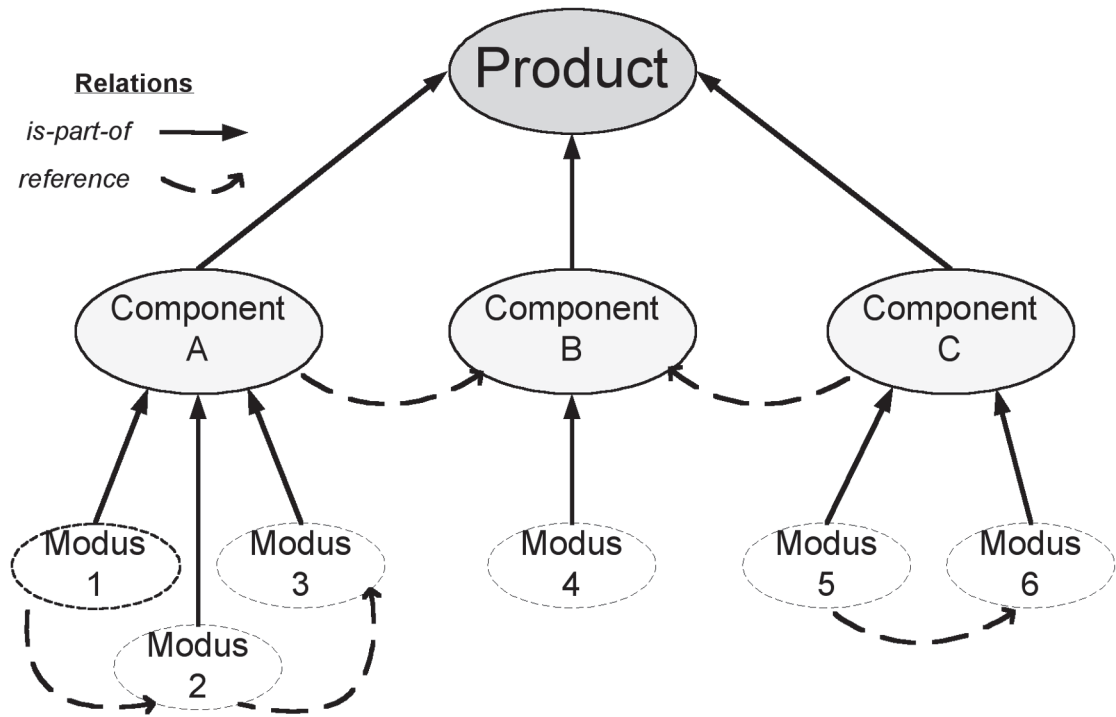


FIGURE 6. Hypergraph model of product.

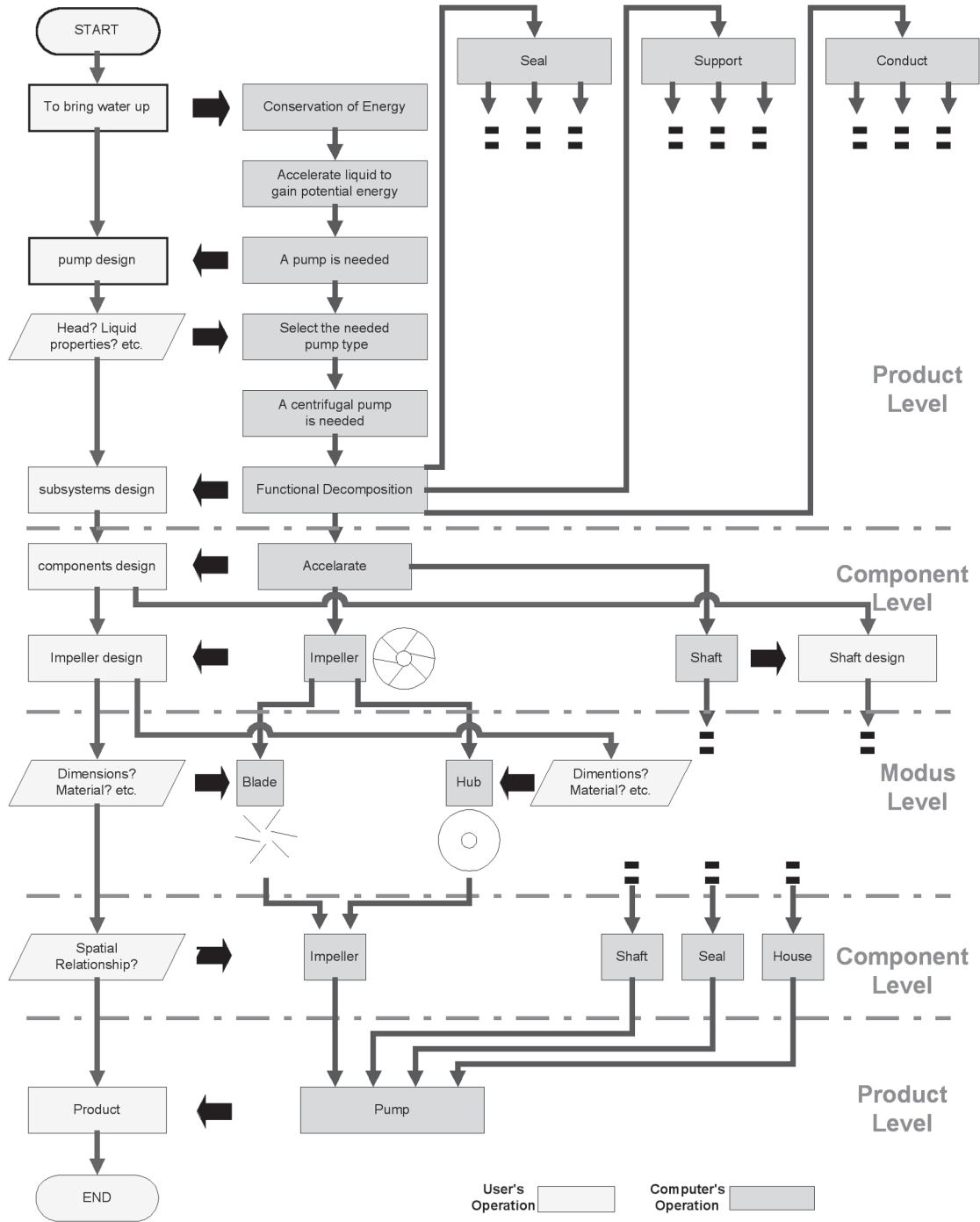


FIGURE 7. Flow diagram of functional reasoning in a centrifugal pump design.

future information flow and CAD application over Internet, a system independent data format is necessary and vital. With the emergence of Extensible Markup Language (XML),²⁵ the data exchange over Internet can have a uniform format. XML is a simple, flexible, and structured text format derived from Standard Generalized Markup Language (SGML) (ISO8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is playing an increasingly important role in the exchange of wide varieties of data over the Internet. It is becoming the backbone of Business-To-Business (B2B) electronic information exchange. An XML-compatible modeling language, Product Markup Language (PML), is developed in Pegasus. It is expected to result in a new CAD system-independent modeling language for mechanical product design.

PML has the following characteristics: (1) Extensible: By the nature of markup, PML can be extended for new information if necessary. When new concepts or notations are to be used, the modeler can extend its language scope by adding new elements in. It provides good scalability for the modeler. (2) Portable and interoperable: The language tends to separate system-independent content and system-dependent format of product information so that useful information about product will not be lost during data exchange and translation. PML can include more information in product files. It has the capability to include technical information, such as materials, tools selection, cutting path, etc., and managerial information, such as order number, cost, etc., as well as geometric information from different levels. (3) Object-oriented: The inherent hierarchical tree structure of the language enables good encapsulation so that modular transparency is guaranteed for the top-down approach of design. Products are modeled by PML, which describes the information about the product explicitly, such as geometries, functions, features, materials and contexts, etc. Theoretically all information about product can be modeled in PML. (4) Compatible to Information Infrastructure: XML is looked as the future of web technology. PML is compatible to the Web standard. Compatibility is indispensable when building an open system. Figure 8 shows the former pump example modeled in PML.

D. Form Generator Configuration

As a part of Pegasus system, Form Generator accomplishes the tasks of form generation from concept during conceptual design stage. To accommodate different levels of users, modular design is introduced in Generator.

As shown in Figure 9, Generator is comprised of Graphic User Interface, Modeler, Expert Shell, and associated libraries and knowledge base. Because moduses are

the elements that users are dealing with, Generator tends to be built as a flexible system that allows extension, that is, users are able to define their own moduses based on fundamental moduses and available geometries, materials, and other information. Also these libraries and knowledge base allow users to extend and update the information if required. Within the system, Expert Shell plays an important role to aid users to finish design based on logic and its knowledge. Modeler enables users to complete modus operation and product information integration. The user interface of Form Generator is shown in Figure 10.

VI. FUTURE WORK

Pegasus system is a pioneer web-based collaborative designer system. The ultimate goal is to develop an easy-to-use CAD system that supports mass customization in the future. The system will have the ability of finishing conceptual design, detailed design, design test and evaluation, performance simulation, and design optimization in real-time and on-line. Form Generator is the first step users will encounter when entering the system. Based on functionality, modular design is one of the pragmatic and rapid ways to accomplish conceptual design automation, considering both flexibility and speed.

The future work of Form Generator will be extending the capability of the system to support detailed design. Considering manufacturing, assembly, reliability, maintainability, ergonomics, and other related aspects, the system should be able to impose multidisciplinary constraints onto product, as well as customer's preferences. Those constraints should be captured and represented in appropriate ways so that those information will be transferred and be constructive for later stages of product design.

VII. CONCLUSION

A new conceptual design methodology, functionality-based modular design, is presented in this paper. It has the capability of high flexibility, fast speed in supporting mechanical product design and customization. A model of *modus* is developed to attain modularization to accommodate different system behavior requirements from users of CAD over the Internet. An XML compatible language, PML, which has good properties of interoperability, scalability, compatibility and extensibility, is used to model products. PML is able to encapsulate both geometric and nongeometric information in an attempt to solve the existing CAD file format problems. A module that supports functional modular conceptual design is built to implement the above developments.

```

<L_WATER:pump context:L_WATER="Low pressure, Water"
      pump_t="centrifugal">
  assembly_1
    <housing body_t="enclosed"
      inlet_n="1"
      outlet_n="1"
      material_t="steel">
      part_1
    </housing>
    <impeller vane_t="open"
      curve_t="volute"
      material_t="stainless_steel">
      part_2
    </impeller>
    <seal material_t="rubber">
      part_3
    </seal>
    <shaft material_t="stainless_steel">
      part_4
    </shaft>
    <bearing bearing_t="ball"
      row_n="2"
      housing_t="rotating">
      part_5
    </bearing>
    <bearing bearing_t="ball"
      row_n="2"
      housing_t="rotating">
      part_6
    </bearing>
</L_WATER:pump>

```

FIGURE 8. A pump in Product Markup Language.

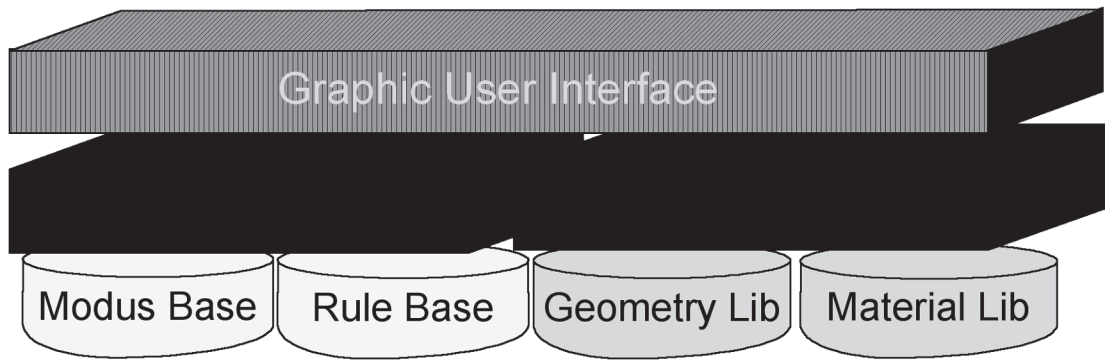


FIGURE 9. Architecture of Pegasus Form Generator.

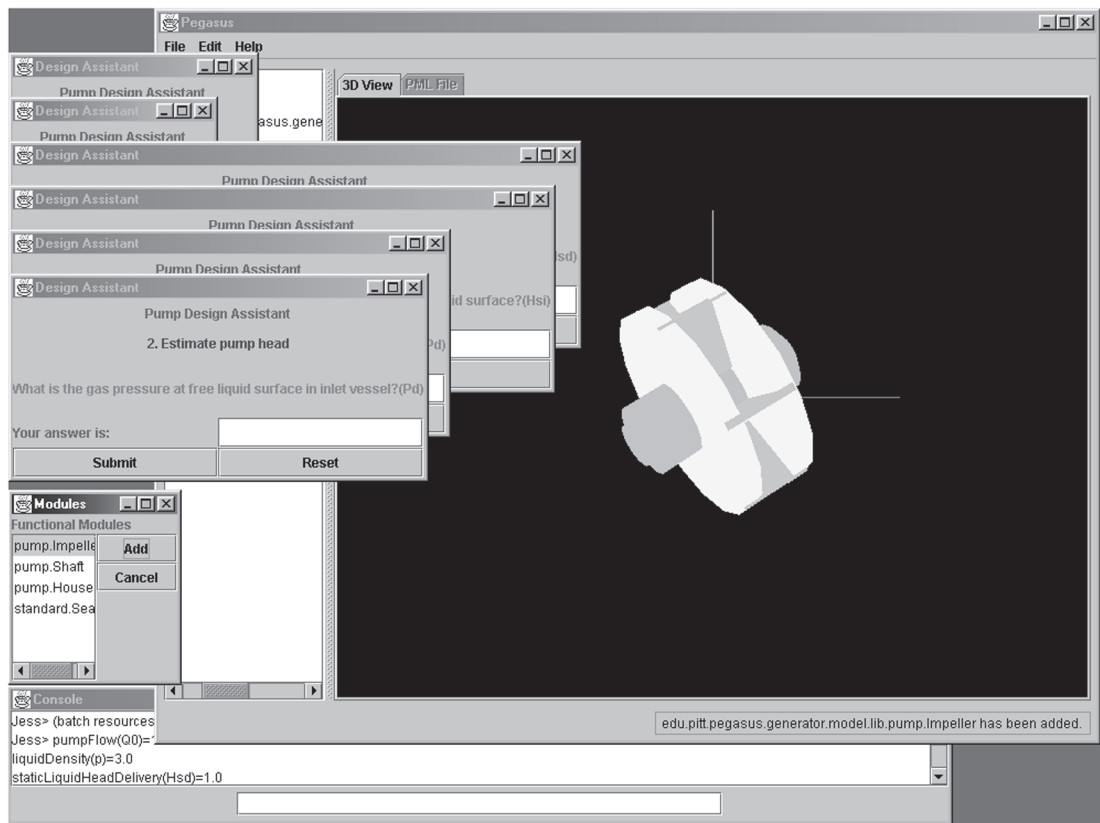


FIGURE 10. Graphic User Interface of Pegasus Form Generator.

REFERENCES

1. Alexander, C., *Notes On The Synthesis of Form*, Harvard University Press, Cambridge, Massachusetts, 1964.
2. Andreasen, M.M., Conceptual Design Capture. In *Proceedings of Engineering Design Conference '98 on Design Reuse*, S. Sivaloganathan and T.M.M. Shahin (ed.), Professional Engineering Publishing Limited, London, 1998.
3. Bhatta, S.R. and Goel, A.K., From Design Experiences to Generic Mechanisms: Model-based Learning in Analogical Design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, Vol.10, No.2 (April), pp.131–136, 1996.
4. Coyne, R.D., Rosenman, M.A., Radford, A.D., Balachandran, M., and Gero, J.S., *Knowledge-Based Design Systems*, Addison-Welsey, Reading, Massachusetts, 1990.
5. Duffy, S.M. and Duffy, A.H.B., Sharing the Learning Activity Using Intelligent CAD, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, Vol.10, No. 2 (April), pp.83-100, 1996.
6. Sivaloganathan, S. and Shahin, T.M.M. (eds.) *Proceedings of Engineering Design Conference '98 on Design Reuse*, Professional Engineering Publishing Limited, London, 1998.
7. Erixon, G., Design For Modularity, In *Design For X*, G.Q. Huang (ed.), pp. 356–379, Chapman & Hall, London, 1996.
8. Freeman, P. and Newell, A., A Model for Functional Reasoning in Design, *proc. 2nd IJCAI*, London, pp. 621–640, 1971.
9. Gorti, S.R. and Sriram, R.D., From Symbol to Form: a Framework for Conceptual Design, *Computer-Aided Design*, Vol. 28, No.11, pp.853–870, 1996
10. Maher, M.L., Balachandran, M.B., and Zhang, D.M., *Case-Based Reasoning in Design*, Lawrence Erlbaum Associates, Mahwah, NJ, pp. 141–164, 1995
11. Maher, M.L., CASECAD and CADSYN: Implementing Case Retrieval and Case Adaptation, in *Issues and Application of Case-Based Reasoning in Design*, M.L. Maher and P. Pu (eds.), Lawrence Erlbaum Associates, Mahwah, New Jersey, Chapter 7, pp.161–185, 1997.
12. Maher, M.L. and Garza, A.G.S., Case-Based Reasoning in Design, *IEEE Expert*, Vol. 12, No. 2 (March-April), pp. 34–39, 1997
13. Maher, M.L. and Pu, P. (eds.), *Issues and Applications of Case-Based Reasoning in Design*, Lawrence Erlbaum Associates, Mahwah, New Jersey, 1997.
14. Pahl, G. and Beitz, W., *Engineering Design: A Systematic Approach*, Springer-Verlag, London, 1996.
15. Qian, L. and Gero, J.S., Function-behavior-structure Paths and Their Role in Analogy-based Design, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, Vol. 10, No. 4 (September), pp. 289–312, 1996.
16. Rosenman, M. and Wang, F., CADOM: A Component Agent-based Design-Oriented Model for Collaborative Design, *Research in Engineering Design*, Vol.11, pp. 193–205, 1999.
17. Sivaloganathan, S. and Shahin, T.M.M., Design Reuse: An Overview, *Proc. Instn. Mech. Engrs.*, Vol.213, Part B, pp. 641–654, 1999.
18. Suh, N.P., *The Principles of Design*, Oxford University Press, Oxford, 1990.
19. Tomiyama, T., Object Oriented Programming Paradigm for Intelligent CAD Systems, in *Intelledge CAD System II: Implementational Issues*, V. Akman, P.J.W. ten Hagen and P.J. Veerkamp (eds.), pp. 3–16, Springer-Verlag, Berlin, 1989.
20. Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T., Supporting Conceptual Design Based on the Function-Behavior-State Modeler, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, Vol. 10, No. 4 (September), pp. 275–288, 1996.
21. Veerkamp, P., Akman, V., Bernus, P. and ten Hagen, P., IDDL: A Language for Intelligent Interactive Integrated CAD Systems, in *Intelledge CAD System II: Implementational Issues*, V. Akman, P.J.W. ten Hagen and P.J. Veerkamp (eds.), pp. 58–74, Springer-Verlag, Berlin, 1989.
22. Veth, B., An Integrated Data Description Language for Coding Design Knowledge, in *Intelledge CAD System I: Theoretical and Methodological Aspects*, P.J.W. ten Hagen and T. Tomiyama (eds.), pp. 295–313, Springer-Verlag, Berlin, 1987.
23. Yoshikawa, H., General Design Theory and a CAD System, in *Man-Machine Communication in CAD/CAM*, North-Holland, Amsterdam, 1981.
24. Yashikawa, H., Introduction to General Design Theory, in *Intelligent Manufacturing Systems I*, V.R. Milačić (ed.), pp. 3–17, Elsevier Science Publishers B.V., Amsterdam, 1988
25. WWW Consortium, <http://www.w3.org/XML/>

BIOGRAPHIES

Bart O. Nnaji has a B.S. in Physics from St. John's University with distinction; an M.S. and Ph.D. in Industrial and Systems Engineering from Virginia Tech, and obtained a certificate of Postdoctoral studies in Artificial Intelligence and Robotics from MIT. He was Professor of Department of Mechanical and Industrial Engineering at the University of Massachusetts at Amherst till 1996. He is currently the ALCOA Foundation Professor in Manufacturing Engineering at the University of Pittsburgh. Professor Nnaji has received 3 honorary doctorates from international universities. He received the 1998 Outstanding Young Manufacturing Engineer Award by the Society of Manufacturing Engineering; 1992 Outstanding Young Industrial Engineer Award; He is a Fellow of Nigerian Academy of Sciences; Fellow of the Institute of Industrial Engineers, and Fellow of the Society of Manufacturing Engineers. He was honored with the U.S. Secretary of State's *Distinguished Public Service Award* in 1995; *Distinguished Scientist Award* by the

World Bank — IMF in 1998; and the Nigerian President national honor — *Officer of the Order of Niger* (OON) in 2000. Professor Nnaji has served as principal or coprincipal investigator on over \$35 million research. He has published 5 books and over 100 technical articles. One of his books, *Computer Integrated Manufacturing and Engineering*, won the 1994 world best text book prize for Manufacturing Engineering. Professor Nnaji is the founding Editor-in-Chief of the *International Journal of Design and Manufacturing* and also serves as the Editor for the Design Department of *Institute of Industrial Engineers*

Transactions on Design and Manufacturing. Professor Nnaji served as Nigeria's Federal Minister of Science and Technology in 1993.

Yan Wang is a Ph.D. student in Department of Industrial Engineering, University of Pittsburgh. He received his MSEE degree from Chinese Academy of Sciences in 1998, BSEE from Tsinghua University, China, in 1996. His current research interests include 3D geometric modeling and collaborative knowledge-based CAD.