Int J Adv Manuf Technol (2006) 29: 137–150
DOI 10.1007/s00170-004-2213-5

ORIGINAL ARTICLE

Y.-E. Nahm · H. Ishikawa

A new 3D-CAD system for set-based parametric design

Received: 5 February 2004 / Accepted: 19 April 2004 / Published online: 5 April 2006

Abstract So far, there has been an extraordinary development of computer-aided tools intended to generate, present or communicate 3D models. But there has not been a comparable progress in the development of 3D-CAD systems intended to assist designers in representing and manipulating both geometric and non-geometric design information based on solid models, thus facilitating concurrent engineering (CE). Design objects continue to be produced by traditional means using the computer as little more than a drafting tool. In addition, the state-of-the-art 3D-CAD systems are incapable of encoding engineering uncertainties since only precise single-valued assignments are allowed for their modeling operations. Recently, set-based CE (SBCE) has been attracting public attention as an emerging CE paradigm. Such a set-based design (SBD) approach presents many possibilities in handling the uncertainties that are intrinsic at the early phases of design. This paper addresses a novel concept – set-based parametric design (SBPD) – which combines the SBD practice with the parametric modeling technique widely used in most 3D-CAD systems. A preference set-based design (PSD) model and a design information solid (DIS) model are proposed to incorporate the SBPD concept into the current 3D-CAD systems. Finally, a prototype system is implemented to illustrate the potential to achieve a SBPD practice.

Keywords Design information modeling and sharing · Intelligent CAD · Preference set-based design · Set-based concurrent engineering · Set-based parametric design

Y.-E. Nahm
Department of Mechanical Design Engineering,
Hanbat National University, SAN 16-1,
DuckMyoung-Dong, Yuseong-Gu, 305-719 Daejeon, South Korea
E-mail: nahm@hanbat.ac.kr
Tel.: +82-42-8211076
Fax: +82-42-8211153

H. Ishikawa (✉)
Department of Mechanical Engineering and Intelligent Systems,
The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, 182-8585 Tokyo, Japan
E-mail: ishikawa@mce.uec.ac.jp
Tel.: +81-424-435422
Fax: +81-424-843327

1 Introduction

Recently, a number of types of computer-aided design (CAD) systems have generally been used as designers' everyday tools, and have helped propel the concurrent engineering (CE) practice. The term CAD is widely used to describe any software that can assist product engineers in accomplishing a specified task. There are many types of CAD, but for our purposes, this paper discusses a CAD system capable of defining a design component with geometry, surfaces or solid models (i.e., a 3D-CAD system). Three-dimensional models have great importance not only in their traditional role as a means of communicating design information but also in externalizing the designer's thought process by allowing visualization of the design product [1, 2]. The early CAD systems based on boundary representations (B-reps), constructive solid geometry (CSG), or a hybrid of the two have often been criticized for their static modeling technique, and more and more replaced by parametric design systems enabling dynamic modeling and modification. Currently, there are a couple of variations of parametric modeling systems (e.g., variants programming, history-based constraint modeling, variational design, rule-based variants, and parametric feature-based design [3]) to alleviate the drawbacks of conventional CAD systems, which provide a more natural way to express design objects by capturing the designer's intent as well as geometric information [4, 5].

Nevertheless, most of the current commercial and academic CAD systems still have many drawbacks [1–9]. In this paper, the following two issues are highlighted as imperative functionalities that should be mounted on the future CAD system:

- The most crucial problem of current CAD systems is that they do not support the early stage of design (e.g., the conceptual design stage) [4–8]. The early design stage intrinsically contains multiple sources of uncertainties or imprecision in describing the design, and thus designers describe design objects based on many uncertain factors but with only a small quantity of information. During the design process, such vague and incomplete designs are gradually elaborated

to detailed descriptions. The parametric modeling technique involves a representation of an object, a set of parameters characterizing the object, and a list of constraints (equations or functions) applied to the object. However, the parametric model can typically represent only the nominal shape of the artifact at a single level of accuracy [5], by generally only allows precise single-valued assignments into geometry-related parameters. Usually, the parameters are too low-level to provide a good basis to support the early phase of the design process.
- Most CAD systems have mainly concentrated on geometric (graphical or shape) information rather than on the full range of design information required to define an engineering artifact completely [4, 8]. The lack of capability of handling non-geometric (or non-graphical) information results in an incomplete representation, since both facets of geometric and non-geometric design information are integral to the description of a design object. Thus, the geometric information is not sufficient to support all phases of the design process [5]. Engineers really want all of the information and graphics to be together. Also, an integrated data structure resulting from the integration of geometric and non-geometric design information can provide a basis for linking the current CAD systems with other engineering applications or CE processes. So far, much attention has been paid to the integration of geometric and non-geometric design information [1, 2, 8, 9]. In particular, an up-to-date engineering standard such as the standard for the exchange of product model data (STEP) is intended to allow designers to extract the needed information from a unified model to form their own domain models. However, since it is difficult to define a unified model in advance, it is desirable to develop an integrated design model, which is not given at the outset of the design process but emerges as the design progresses, by embedding various types of design information in a timely fashion [10].

First, this paper addresses the need to make persistent association and consistent management of geometric and non-geometric design information with the CAD geometry, and proposes an integrated CAD data structure, called a design information solid model (DIS-model).

In addition, in previous works [10–12], the authors have addressed new challenges to collaborative product design environments. The radical pace of changes in engineering design to a distributed, integrated, set-based and coordinated product design environment prompts CE teams to tackle new challenges. In particular, an emerging engineering practice in the collaborative product development process is set-based concurrent engineering (SBCE) paradigm [13–16]. Such a set-based design (SBD) approach presents many possibilities in representing and manipulating the engineering uncertainties that are intrinsic at the early phase of design. In the traditional design practice, a single design possibility is formulated, evaluated, and modified until a solution that meets the design objective is obtained. However, in this iterative process, there is no theoretical guarantee that the process will ever converge and produce an optimal solution [16]. By contrast, SBCE considers a broader range of design possibilities from the outset, communicates the possibilities among functional team members, and gradually narrows the set of possibilities to converge on a final solution, thus enabling more robust and optimized design than the traditional design practice. The change in engineering to the SBCE practice should involve a corresponding change in design method to a set-based process.

Second, this paper addresses the importance of SBD practices in engineering design, and proposes a SBD approach, called a preference set-based design model (PSD-model), which is incorporated into the parametric modeling technique of current CAD systems, thus enabling set-based parametric design (SBPD).

This paper is organized as follows. Some related research is reviewed in Sect. 2. Section 3 overviews the SBD practice. Section 4 addresses a novel concept, SBPD, and the DIS-model and PSD-model are proposed to incorporate the SBPD concept into current 3D-CAD systems. A prototype system is presented to illustrate the potential of SBPD in Sect. 5. Section 6 concludes the paper.

2 Related works and their limitations

2.1 CAD in a CE process

A brief literature review is presented below. Due to the broad base of the CAD/CAM/CAE topics, the authors encourage the reader to seek added information, and focus on the integration between CAD geometry and other engineering processes or applications in the CE field.

CE attempts to incorporate multiple functional aspects (e.g., function, geometry, manufacturing, marketing, etc.) from the early stages of the product development process. Then, multiple representations must be maintained to support the different perspectives and engineering activities. Cutkosky et al. propose a concurrent design system, called NextCut, which is able to work with multiple representations [17]. At the center are the shared models of hierarchical and object-oriented descriptions of multiple aspects where relationships between the parameters of feature-based solid models and other aspects are explicitly specified by a designer, and design changes are propagated to the related parameters through change notification messages. Xue and Dong present an integrated concurrent design system that can support various design tasks including function design, geometry representation, production process generation, cost estimation, etc. [18]. Geometric feature classes [19] are in advance defined to represent design elements, and an appropriate geometric feature is instantiated based on the descriptions of function design and used to construct a 3D model. Gorti et al. develop a knowledge representation model based on the SHARED object-oriented model, which incorporates both an evolving product and its associated design process [20]. The product (i.e., artifact) is represented by the combination of function, form, and behavior objects. The form object then contains geometry-related attributes including the 3D solid CAD file, position, and orientation. Gu presents a design modeling language called PML that represents product, solid, and features, and a communication method for linking feature-based solid models and various manufacturing applications [21]. Bradley and Maropoulos also present a product model that can be used from the early stages of product development, based on the concept of feature relations that can represent dimension, tolerance, and connectivity modeling [22].

In addition, the early focuses in the CE field were mainly on the parallelism or concurrency of entire lifecycle functions. In particular, a lot of research efforts have been made in the concurrence of upstream and downstream processes such as design and manufacturing [23–25]. Also, feature-based CAD models could be integrated with design for manufacturability (DFM) analysis [26]. While mainstream traditional solid modeling is based on quantitative information and mathematical representation [27], Gorti and Sriram propose an approach for mapping an evolving symbolic description of design into a geometric description. The key idea is the symbol-form mapping for geometry representation in conceptual design [28]. Ranta et al. also integrate the functional design and feature-based solid modeling [29]. Product information at the conceptual design phase is qualitative, imprecise, uncertain, or incomplete. Shu et al. propose a qualitative and imprecise solid modeling system to deal with qualitative and imprecise information of conceptual shapes [30]. More recently, agent technologies are applied to the solid modeling process [7, 31–33].

Even though rigorous research efforts have been made to introduce CAD geometry into the CE processes or applications from an earlier phase of product development, most tools enable only "shallow" or "superficial" integration. Design descriptions are first made by functional designs, and then the geometry-related data is passed to geometric modeling systems. Although the geometric information is associated with other aspects, it does not mean a complete integration of geometric and non-geometric information. It is interfacing rather than integration. Ideally, the best way to carry out the integration is to offer a product model in which all relevant product data are stored, and useful data for applications can be extracted. Sriram discusses the significance of current standards and knowledge-based standards over the networked environment for computer-supported collaborative engineering (CSCE) [34]. STEP, developed by the international organization of standardization (ISO), has promoted the development of standard data models for the exchange of data through the use of a neutral format that is independent of application. The STEP standard is a valid way to exchange model data. However, STEP data models are too big to analyze and maintain, and require a number of translators. Comparatively, modular applications are much easier to develop and understand. Motivated by this reason, Zhang et al. introduce a data-exchange framework for virtual enterprises to provide Internet-based services for STEP data translation [35]. STEP is also weak in representing model function, model behavior, a designer's ideas, and so on [36]. The STEP standard is intended to allow designers to extract their needed information, to form their own domain models. It seems, however, difficult to define all data in advance. Although it supports the rapid exchange of design information from the multiple aspects of the given problem, this framework is not intended to provide concurrent system modeling functionality [37]. Our DIS-model takes a different approach to achieve persistent association and consistent management of geometric and non-geometric design information with the CAD geometry (see Sect. 4.2).

2.2 Engineering design with uncertainties

In addition to the lack of capability of handling non-geometric information, the state-of-the-art CAD systems allow only precise single-valued assignments for their modeling. They cannot, therefore, handle uncertain quantities that are dominant in the early design phase, and cannot also capture designers' preferences on design possibilities in defining solid models.

Representing uncertainties is a critical topic that researchers have approached from many different directions, including a probabilistic-based approach, fuzzy set-based approach, interval set-based approach, and so on [38]. Probability is one mechanism to handle uncertainty or imprecision, but requires tremendous volumes of information to determine probability densities and conditional probabilities. Compared to probabilistic-based methods, a fuzzy set-based approach like method of imprecision ($M_O I$) [39–42] requires less computation and provides similar results [43]. Using a preference function is more expressive than using intervals alone in that they can represent a combination of preference and possibility in calculations. By the using the α-cut concept, the preference function results in a set of intervals and the standard interval arithmetic can be then applied to propagate the resulting intervals over the constraints. Even though the preference function can capture a designer's preference structure on a continuous set, the lack of a natural ordering for a discrete set precludes the use of the preference function. Also, $M_O I$ often makes incorrect propagations, because the conventional interval arithmetic does not consider the causal relationships among the variables describing physical engineering systems [44–46].

Finch and Ward present an interval set-based approach by proposing quantified relations (QRs), a class of predicate logic expressions, to explicitly represent causal relationships between variables in engineering systems, and the interval propagation theorem (IPT) algorithm to propagate intervals through QRs involving continuous and monotonic equations [44–46]. Compared to labeled interval calculus (LIC) [47, 48], this approach offers greater expressive power, clearer semantics, and closer ties to well-established ideas in computer science. However, unlike $M_O I$, it cannot explicitly represent the degree of desirability/preference of the designer. That is, design propagation by this approach generates only bounds on the membership of feasible sets of design variations. Since the approach provides only boundaries of the interval, the designer cannot directly determine which input values have contributed to any one particular value of the output (except at the boundaries) [39]. Therefore, this approach cannot provide any guidelines for design modification. Also, while the conflict-free computation among multiple design variables has been relatively well-studied in the existing interval set-based research, cooperation among multiple design agents where conflicts can occur is less well-understood. It cannot thus provide a compromise solution among multiple conflicting goals.

In order to overcome the problems mentioned above, the underlying idea of the proposed PSD-model is to combine the fuzzy set-based approach and the interval set-based approach; doing so could compensate for the limitations of both (see Sect. 4.3).

A more extensive review about engineering design with uncertainties can be found in our previous paper [38].

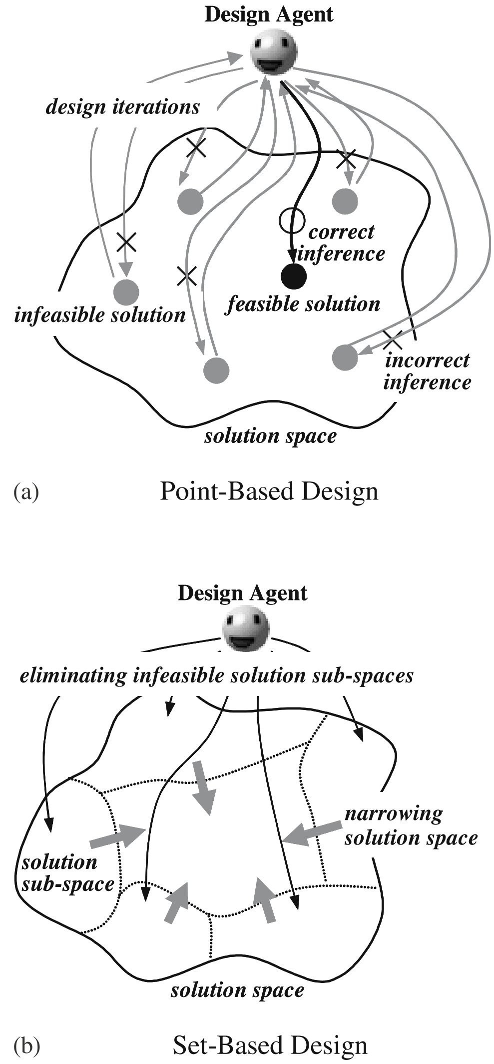


Fig. 1. Point-based versus set-based design

3 Point-based versus set-based design

The traditional design practice, whether concurrent or not, cannot see the forest for the trees. As shown in Fig. 1a, the traditional point-based design (PBD) practice quickly develops a "single solution" (i.e., a point in the solution space), critique it based on multiple disciplines or objectives, and then iteratively moves to some other point until it reaches a satisfactory solution. We cannot deny that this iterative design practice is indeed a common and widely accepted design approach in the engineering design community. However, this approach severely degrades efforts to design products and processes concurrently.

1. The PBD practice often uses single values to represent quantities describing engineering systems. However, the precise single values do not include information about uncertainty caused by many sources of variations [46].
2. Since single-solution proposals prompt responses that critique the design and suggest changes to accommodate another's considerations, this approach can lead to a large number of iterations if one picks the wrong starting point [16].
3. In the present CE environment, a design solution is iteratively refined through a series of decision-making processes. This means that any decision made by one member of the design team may invalidate previous decisions by others [14].
4. The CE philosophy tries to incorporate overlapping activities between entire lifecycle functions (e.g., concurrence of upstream and downstream processes). If an upstream function proposes only its best idea, it does not give other downstream functions a clear idea of the possibilities [14]. That is, the future outcome of an upstream activity is hard to predict. So, the earlier the downstream starts, the higher the risk of future changes, thus destroying concurrency or parallelism of the CE philosophy [49].

By contrast, SBD considers a broader range of design possibilities from the outset by dividing the solution space into relatively equal volumes (Fig. 1b). Then, designers explicitly communicate and contemplate sets of design alternatives. The sets are gradually narrowed through the elimination of inferior alternatives until the final solution remains. Some potential advantages that may be gained by adopting the SBD approach are summarized as follows:

1. SBD dramatically reduces the amount of back-tracking in the design process, and requires less frequent and less prolonged communication between functional teams, because team members can safely make any decisions within their area as long as they are valid for the entire set of possibilities.
2. Since SBD makes functional teams communicate about sets of solutions and regions of the design space, the feasible regions and additional trade-off results among alternatives help members outline sets of possibilities and understand the implications of choosing one alternative over another. Consequently, the communicating sets enable functional teams to understand the feasible regions of others.
3. In PBD, every change that part of an organization makes may invalidate all previous decisions. However, SBD enables reliable and efficient communication, since all communication is taken with the whole set of possible solutions, and the earlier communications remain valid during the set-narrowing process, and so on. The SBD practice is more generally referred to as SBCE, and overall concepts and principles of SBCE are given in [13–16].

4 SBPD: set-based parametric design

4.1 Overview

Engineering design can be described as the process of transforming a set of functional specifications and requirements into a complete description of a physical product or system (in design space) that meets those specifications and requirements (in performance space). When a design (or performance) space is defined by a set of design (or performance) parameters and the design (or performance) values, the process to generate one or more feasible solutions in the design space to accomplish the required performances is generally called parametric design. The parametric design at early stages plays an especially important role in any product design. As mentioned in the previous section, the conventional design practice usually utilizes point-to-point mapping from the design space to the performance space, and tries to find a single optimized solution that meets the design objective through large iterations of mapping processes.

However, the emphasis at early stages should be on the derivation of a wealth of robust design alternatives as opposed to a single optimized solution. A single point solution is not sufficient to support the preliminary design, since the design information at this stage is vague and uncertain, and design targets are constantly changing [50]. In order to cope with multiple sources of variations and ensure design robustness, we need to explore a large design space to get feasible sets of design possibilities, by a set-to-set mapping from design space to performance space, and vice versa. The point-to-point and set-to-set mapping processes may be compared to the design activities by experienced and inexperienced designers. An experienced designer utilizes the bread-first search strategy before working through any subproblem to its lowest level, while an inexperienced designer tends to identify a single sub-problem and work its solution out completely (i.e., depth-first strategy) [51].

The point-based practice is also predominant in current CAD systems. Solid modeling has been widely used in CAD and other engineering analysis systems due to its unambiguous, complete mathematical representation of real-world objects and the availability of much more information about the modeled object (such as mass properties). The parametric modeling technique further enables dynamic modeling and modification. However, since these modeling operations require the input of precise single values, it is difficult to identify and explore a broader range of design possibilities.

In this paper, our set-based design model is hybridized with the parametric modeling technique; thus it is dubbed "set-based parametric design" (SBPD). This capability would enhance the availability of current CAD systems from the earlier stages of design. Using our DIS-model, SBPD practice first defines a parametric model that contains a set of parameters to capture non-geometric information as well as geometric information and a list of constraints (equations or functions) applied to the model. Second, set-valued assignments are made into the design parameters with unique identifiers. The assigned values, called the preference set, represent both the continuous or discrete sets of possibilities and a designer's preference structure on the set elements. Third, the specified preference sets are propagated over the given constraints by following a sequence of processes of the PSD-model. The propagations result in the possibility distributions of the (intermediate or performance) parameters affected by the designer's preference sets. These possibilities are also stored in the DIS-model. Then, designers can promptly explore many design possibilities (i.e., combinations of parameter values) that lie in the design and performance preference sets initially specified by the designer.

4.2 Design information solid (DIS) model

The authors have proposed an integrated CAD data structure called a design information solid (DIS) model which enables persistent association and consistent management of geometric and non-geometric design information with the CAD geometry [8]. A plausible alternative to the integration of graphical and non-graphical design information is to link elements in the graphical data structure with an extensible set of non-graphical attributes by the use of a relational database management system (DBMS) [1]. However, the DIS-model adopts a different approach of embedding various types of design information into a general CAD data structure (i.e., B-reps) of the solid model as attribute data. Thus, the geometric and non-geometric design information is tightly bound to the B-reps data structure. The underlying idea of the DIS-model is that the data structure of a CAD system can be described as the vehicle for the storage and manipulation of design information.

The integrated CAD data structure for the DIS-model is shown in Fig. 2. Design information can be defined as attribute data of the B-reps of solid models. Then, DIS-model can encapsulate the full range of design information required to define completely an engineering artifact, including the qualitative information, such as design intention, function of parts and units, material, relationship between parts, tools necessary for assembling the parts, and so on, as well as geometric data. In Fig. 2,

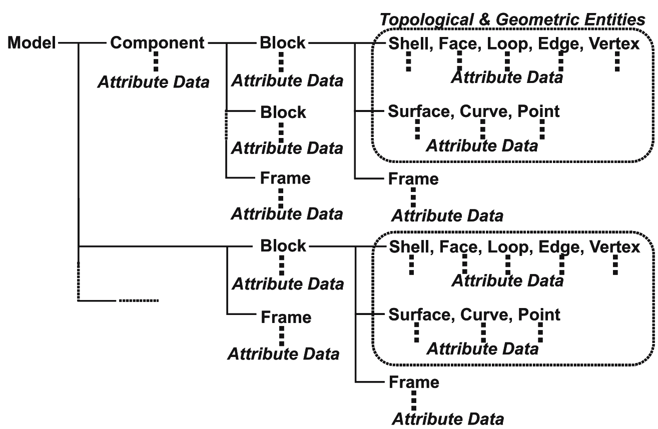


Fig. 2. DIS-model: design information solid model

a "block" is a solid that is composed of B-rep elements (i.e., topological and geometric entities) of a solid model. A block can have a set of attribute data. The main difference compared to typical B-reps is the existence of entities called the "component" and the "frame". Some product models are composed of one or more blocks with each attribute datum, while others may be composed of some product components that are composed of blocks. Therefore, a solid-modeling unit component is introduced to support the designer's thinking on the unit of the product. Each topological or geometric element of a B-rep solid may also contain attribute data for its block/component. Also, another different type of entity, a frame, is introduced to define a set of attribute data at any point of the component and block where the B-rep elements are not defined. A frame has the intrinsic attribute data of position (x, y, z) and direction (θ_x, θ_y, θ_z), and can be also used to set attribute data. There is no restriction for the amount or kind of attribute data of a DIS-model.

For SBPD, design parameters with unique identifiers as well as constraints are defined, where appropriate, as attribute data of a DIS-model. If an external computation is necessary, it is also possible to set a pointer to a function or method as attribute data. Moreover, when defining a DIS-model intended to be integrated with other engineering analyses, the nature of non-geometrical data stored could be tailored to suit specific applications. For example, a set of attribute data of a DIS-model could be defined for assemblability evaluation as shown in Fig. 3 [8]. The design information for the evaluation of assemblability is input into the elements, the block and frame, of the data structure of the DIS-model, through a screen menu or by clicking a solid itself. In this manner, the availability of the DIS-model for handling non-geometric data greatly enhances a CAD system's potential for the linkage to other engineering applications.

Also, much of the information on drawings is not explicit but is implicit, and there is often important design information that cannot be readily grasped from the drawings [1]. Thus, a DIS-model makes CAD systems more attractive to applications where shape information of the product is not prominent, by putting an appropriate interpretation about shape as attribute data.

A DIS-model lets designers add and associate geometric and non-geometric information to the CAD geometry, and store it with the CAD geometry. Consequently, a DIS-model enables the persistent association and consistent management of a full range of data elements. A DIS-model provides a useful means of capturing a complete product definition and sharing it among CE team members in a timely fashion.

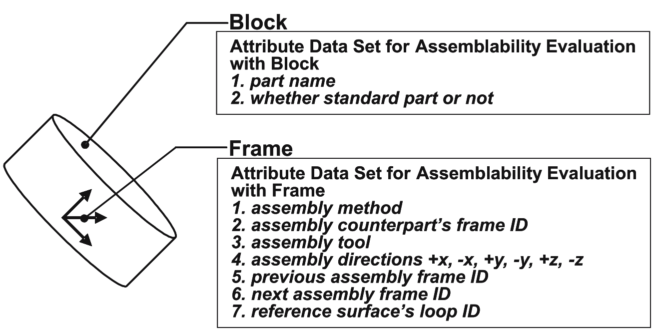


Fig. 3. An example of a DIS-model for assemblability evaluation

4.3 Preference set-based design (PSD) model

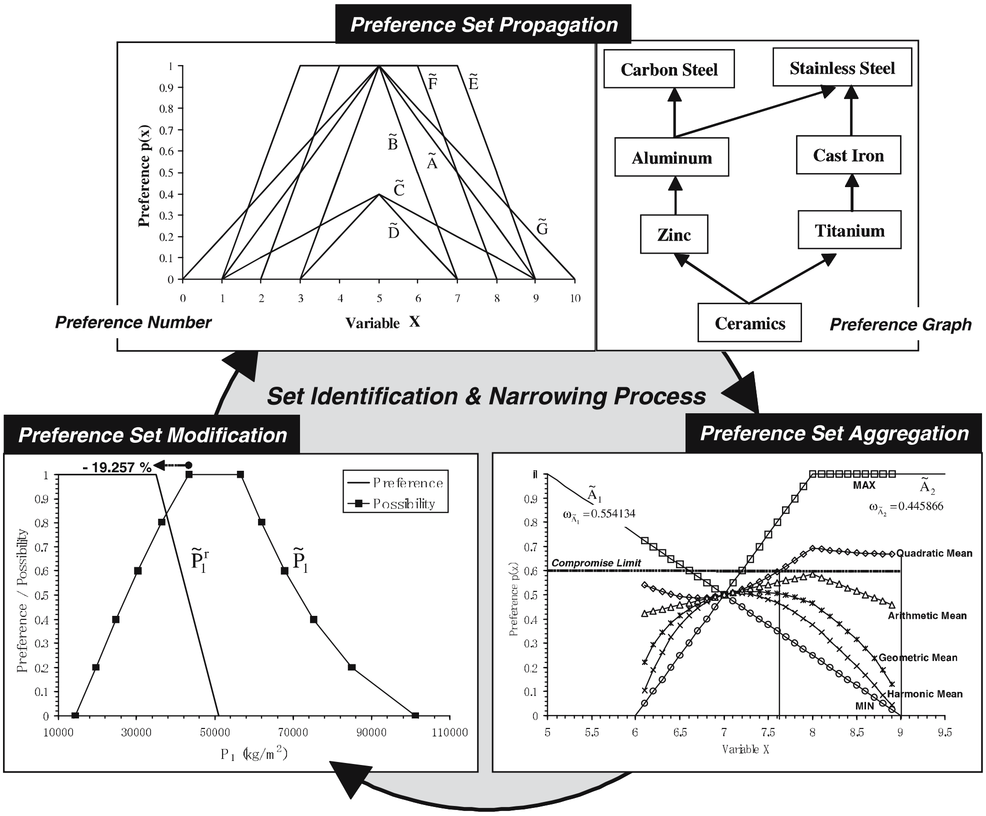
The PSD-model is a hybrid model that combines the strengths of $M_O I$ [39–42], QRs, and IPT [44–46], and employs a lot of novel ideas including the precision and stability (PS) measure as a new measure of uncertainty, the compromise limit to express the maximum degree of compensation permissible by a designer, the rational determination of weighting factors in aggregating different continuous sets, strength of preference (SOP) and closeness of match (COM) indexes for discrete set aggregation, a design modification process using PS measure, and so on [38].

As shown in Fig. 4, a PSD-model consists of set propagation, set Aggregation, and set modification methods. In our PSD-model, the continuous and discrete sets are represented by a preference set including a preference number (PN) and a preference graph (PG). Even though designers are uncertain about which single values to specify, they usually have a preference for certain values over others. First, to capture a designer's preference structure on a continuous set, we use both an interval set and a preference function defined on the interval. The preliminary information by pure interval-set representation is augmented with the designer's preference structure; thus, PN provides richer semantics. Even if a natural ordering can be defined over the continuous variables, there is no natural ordering over the discrete variables. Thus, designers cannot express their preference structures on the discrete variables by specifying preference functions. A PG is a qualitative and simple technique for representing the designers' preference structures by using graph theory. The PG representation is similar to the price-directed acyclic graph (DAG) of [52]. While the price DAG represents a designer's preferences over a selling (or buying) price range, the PG need not establish an overall price range. Also, it provides a more quantitative aggregation method for different PGs. The main advantage of this representation technique is that the designer need not establish a preference between every possible pair of possible assignments, but can begin by specifying only those preferences that it clearly knows initially [52]. As shown in Fig. 4, a designer may prefer stainless steel to either of aluminum or cast iron, and both zinc and titanium to ceramics, but has not determined the relative preference between aluminum and cast iron. In this way, the designer can generate a PG that reflects his or her preference among the elements of a discrete set.

A design problem can be described as a set of constraints including equality or inequality equations, qualitative constraints, computer-based procedures, and influence rules, etc., which are expressed by functional relations among design variables [53]. The proposed PSD-model uses the preference sets (i.e., PNs and PGs) to evaluate engineering constraints.

In general, (symbolic) discrete-set values cannot be directly applied to the numerical engineering constraint. However, the discrete-set values may be well interpreted by some intrinsic

Fig. 4. PSD-model: preference set-based design model



properties. For example, a material can be represented by its material properties such as Young's modulus, density, etc. In fact, these properties will be encoded into the constraints. Thus, we need a transformation process to make PGs propagate over constraints.

Let $X_1, X_2, \ldots, X_N$ be discrete-set variables. Each X_P has its own PG, G_P. Then, let C_P be an adjacency matrix for the G_P and let k be a positive integer. Then, the entry c_{ij} of C_p^k gives the number of k-stage dominances of i over j. That is, the dominance matrix D_P is $D_P = C_P + C_P^2 + \ldots + C_P^k$.

The sum of the entries in row i of the dominance matrix represents the total number of ways that i is dominant in one, two, $\ldots$, k stages [54]. One-stage and two-stage dominances are considered for each PG (i.e., $D_P = C_P + C_P^2$), and the resulting dominance is here called the strength of preference (SOP). For example, the dominance matrix (D_1) and SOPs of a material PG, as shown in Fig. 4, are determined as follows:

$$\begin{array}{ccccccc} C_1 & & C_1^2 & & D_1 & \text{SOP} \\ \begin{bmatrix} 0010000 \\ 0011000 \\ 0000010 \\ 0000100 \\ 0000001 \\ 0000001 \\ 0000000 \end{bmatrix} & + & \begin{bmatrix} 0000010 \\ 0000110 \\ 0000001 \\ 0000001 \\ 0000000 \\ 0000000 \\ 0000000 \end{bmatrix} & = & \begin{bmatrix} 0010010 \\ 0011110 \\ 0000011 \\ 0000101 \\ 0000001 \\ 0000001 \\ 0000000 \end{bmatrix} & \begin{array}{r} \text{Carbon steel} \rightarrow 2 \\ \text{Stainless steel} \rightarrow 4 \\ \text{Aluminum} \rightarrow 2 \\ \text{Cast iron} \rightarrow 2 \\ \text{Titanium} \rightarrow 1 \\ \text{Zinc} \rightarrow 1 \\ \text{Ceramics} \rightarrow 0 \end{array} \end{array} \quad (1)$$

As mentioned above, the sum of the entries in the ith row of the dominance matrix gives the numbers of ways that i is dominated. For example, from the first row of D_1, carbon steel is dominated in $0+0+1+0+0+1+0 = 2$ ways, while stainless steel is dominated in four ways, aluminum in two ways, cast iron in two ways, titanium in one way, and zinc in one way. In our context, the SOP shows which possible assignments are preferred to which other ones either directly or indirectly. Since the SOP is a relative preference of each element, it should always be normalized. Each SOP is then normalized with respect to the maximum SOP. In this case, the maximum SOP is equal to 4 (i.e., the SOP of stainless steel). This calculation normalizes the preferences to the interval [0, 1]. By using the normalized SOPs, a new PN about Young's modulus can be generated to encode the designer's preference of the material (see Fig. 5). Then, this PN can be propagated over the numerical engineering constraint. In this manner, symbolic discrete sets can be consistently encoded and manipulated in the proposed PSD-model.

Now, consider PNs $\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_N$ defined on interval sets $A_1, A_2, \ldots, A_N$ that will be assigned into the design parameters $X_1, X_2, \ldots, X_N$, respectively. Suppose any relation among N variables (i.e., any constraint) can be expressed by $G(X_1, X_2, \ldots, X_N) = G(\boldsymbol{X})$, where $\boldsymbol{X}$ is the vector of design variables. According to QRs [44–46], the PN is further quantified by preceding it with a logic quantifier (i.e., a universal or existential quantifier). Then, a quantified PN, $\tilde{A}_i$, can be ex-

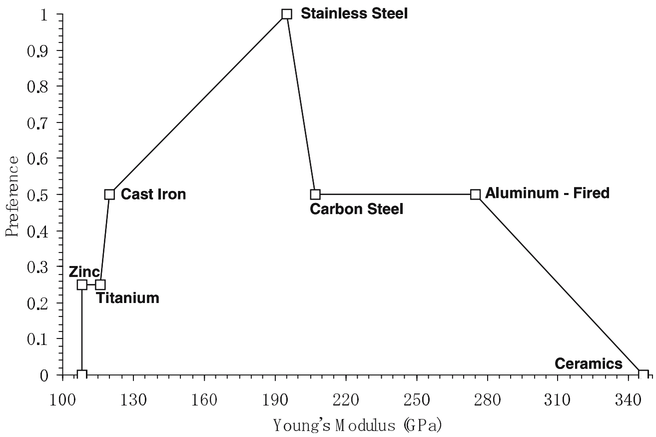


Fig. 5. Preference number of a material property inferred from the preference graph of material

pressed as

$$\tilde{A}_i, = Q\bar{A}_i,\ Q \in \{\forall, \exists\} \tag{2}$$

This seems a more natural representation for engineering constraints. Designers may require that the full range of some PNs should be taken into account and others be adjusted to achieve some desired performance. Then, the propagation process of PSD-model is summarized as follows:

Process 1

Instead of discretizing the support domain (i.e., A_i), PSD works with discrete preference values in a similar way to M_OI [39–42] and FWA [55], thus enabling PSD to utilize the two related concepts of α-cut representation and interval analysis. PSD first divides the range of preference function [0, 1] in the PN $\tilde{A}_i$ into a finite number of values, thus creating a set of α-cuts

$$_\alpha\tilde{A}_i = \{x | p_{\tilde{A}_i}(x) \geq \alpha\},\ \forall\alpha \in [0, 1] \tag{3}$$

where $p_{\tilde{A}_i}(x)$ is the preference function of the PN $\tilde{A}_i$, and x is an element of A_i. Then, the α-cut set at α is represented by the interval

$$_\alpha\tilde{A}_i = [\underline{x_\alpha}, \overline{x_\alpha}] \tag{4}$$

Here, the lower and upper bounds of $_\alpha\tilde{A}_i$ are denoted by $\underline{_\alpha\tilde{A}_i}(= \underline{[\underline{x_\alpha}, \overline{x_\alpha}]} = \underline{x_\alpha})$ and $\overline{_\alpha\tilde{A}_i}(= \overline{[\underline{x_\alpha}, \overline{x_\alpha}]} = \overline{x_\alpha})$, respectively. Therefore, the PN results in a set of closed intervals

$$\tilde{A}_i = \bigcup_{\alpha\in[0,1]} {}_\alpha\tilde{A}_i \tag{5}$$

When a preference function has non-convex shape, more than one interval may be made at α. These intervals should be treated sequentially in the following processes, and the results should be combined by union. The refinement in discretization depends on the degree of accuracy in approximation [55].

Process 2

Suppose a designer attempts to obtain the possible assignments to X_p in the given relation $G(X_1, X_2, \ldots, X_p, \ldots, X_N)$. $G(\boldsymbol{X})$ is rewritten by a function $G(\boldsymbol{X}) \equiv g(X_1, X_2, \ldots, X_p, \ldots, X_N) = 0$ such that the function $g(\boldsymbol{X})$ monotonically increases with respect to X_p. For example, a constraint $X_2 + X_3 = X_1 + X_4 - X_5$ can be rewritten by $X_2 + X_3 - X_1 - X_4 + X_5 = 0$ such that the constraint monotonically increases with respect to X_5. Then, PSD finds the increasing variable subset (I_g) and decreasing variable subset (D_g). I_g and D_g are the subsets of design variables for the function denoted by the subscript such that $g(\boldsymbol{X})$ is monotonically increasing and monotonically decreasing, respectively. In this example, $I_g = \{X_2, X_3\}$ and $D_g = \{X_1, X_4\}$. Furthermore, these subsets are partitioned by each element's quantifier, giving $I_g^\forall$, $I_g^\exists$, $D_g^\forall$, and $D_g^\exists$. Assuming the following PNs, $\forall\tilde{A}_1 \to X_1$, $\forall\tilde{A}_2 \to X_2$, $\exists\tilde{A}_3 \to X_3$, $\exists\tilde{A}_4 \to X_4$, and $\exists\tilde{A}_5 \to X_5$, we have $I_g^\forall = \{X_2\}$, $I_g^\exists = \{X_3\}$, $D_g^\forall = \{X_1\}$, and $D_g^\exists = \{X_4\}$.

Process 3

For each preference value α, PSD evaluates the function $X_p = f(X_1, X_2, \ldots, X_{p-1}, X_{p+1}, \ldots, X_N)$. While M_OI and FWA evaluate the function $f(\boldsymbol{X} \setminus X_p)$ by giving 2^N combinations for the $2N$ bounding values obtained from process 1, PSD calculates X_p with only two combinations based on IPT [44–46], thus significantly reducing computation. The underlying idea of IPT is that to make correct inferences for design constraints, the designer must assume worst-case values for uncontrollable variables and best-case values for controllable variables. Based on this idea, the PN $_\alpha\tilde{A}_p$ that can be assigned into X_p can be obtained by

$$_\alpha\tilde{A}_p = [f(\underline{I_g^\forall}, \overline{D_g^\forall}, \overline{I_g^\exists}, \underline{D_g^\exists}), f(\overline{I_g^\forall}, \underline{D_g^\forall}, \underline{I_g^\exists}, \overline{D_g^\exists})],$$
$$\text{if } \tilde{A}_p \text{ is universally quantified} \tag{6}$$
$$_\alpha\tilde{A}_p = [f(\overline{I_g^\forall}, \underline{D_g^\forall}, \underline{I_g^\exists}, \overline{D_g^\exists}), f(\underline{I_g^\forall}, \overline{D_g^\forall}, \overline{I_g^\exists}, \underline{D_g^\exists})],$$
$$\text{if } \tilde{A}_p \text{ is existentially quantified} \tag{7}$$

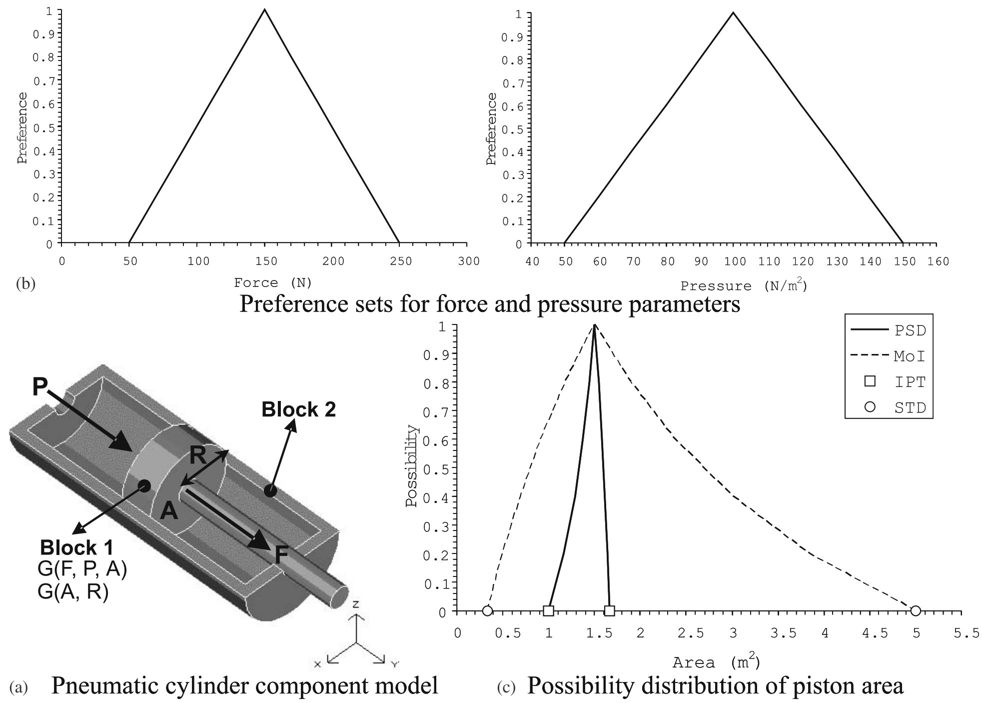
where, for instance, $\underline{I_g^\forall}$ indicates that the lower bounding values of universally quantified and monotonically increasing variables are applied to the function $f(\boldsymbol{X} \setminus X_p)$. Recalling the example above, the possible interval set of PN $_\alpha\tilde{A}_5$ is therefore obtained by the expression Eq. 7 and the function $X_5 = X_1 - X_2 - X_3 + X_4 : {}_\alpha\tilde{A}_5 = [\underline{_\alpha\tilde{A}_1} - {}_\alpha\tilde{A}_2 - \underline{_\alpha\tilde{A}_3} + {}_\alpha\tilde{A}_4, {}_\alpha\tilde{A}_1 - \underline{_\alpha\tilde{A}_2} - {}_\alpha\tilde{A}_3 + \underline{_\alpha\tilde{A}_4}]$.

Process 4

Repeat the process for other α's to obtain additional α-cut sets of $\tilde{A}_p$.

As shown in Fig. 6a, a designer may state a pneumatic cylinder design problem as: "for all piston areas (A) and for all pressures (P), there must be exist a force (F) satisfying an engineering constraint $G(F, P, A) : F = PA$." Also, another constraint, $G(A, R)$, may be defined for the parametric relationship between piston area (A) and radius (R). Then, the two constraints can

Fig. 6. An example to compare constraint propagation results by standard interval arithmetic (STD), M_OI, IPT, and PSD



be embedded into block 1 that is a data element of DIS-model. Based on the propagation process outlined above, the equation, $F = PA$, is rewritten so that $g(F, P, A) = 0$ is monotonically increasing with respect to A. That is, $g(F, P, A) : PA - F = 0$. Thus, $I_g = \{p\}$ and $D_g = \{F\}$. When the two PNs $\tilde{F}$ and $\tilde{P}$ are assigned into F and P as shown in Fig. 6b, the possible piston area, $\tilde{A}$, can be obtained by the relation $f(F, P, A) : A = F/P$. Since P and F are universally and existentially quantified, respectively, $I_g^{\forall} = \{P\}$ and $D_g^{\exists} = \{F\}$. Six α values are selected, viz., 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0. Since $\tilde{A}$ is universally quantified, the Eq. 6 is used to compute ${}_{\alpha}\tilde{A}$. For $\alpha = 0.0$, for example, the α-cut set ${}_{0.0}\tilde{A}$ is

$${}_{0.0}\tilde{A} = [\underline{F}/\underline{P}, \overline{F}/\overline{P}] = [50/50, 250/150] = [1.00, 1.666\ldots].$$

For other α values, the computation yields

$${}_{0.2}\tilde{A} = [70/60, 230/140] = [1.166\ldots, 1.641\ldots],$$
$${}_{0.4}\tilde{A} = [90/70, 210/130] = [1.285\ldots, 1.615\ldots],$$
$${}_{0.6}\tilde{A} = [110/80, 190/120] = [1.375, 1.583\ldots],$$
$${}_{0.8}\tilde{A} = [130/90, 170/110] = [1.444\ldots, 1.545\ldots], \text{ and}$$
$${}_{1.0}\tilde{A} = [150/100, 150/100] = [1.5, 1.5].$$

The solution by PSD is shown in Fig. 6c and is compared with the solutions by standard interval arithmetic (STD), IPT, and M_OI. STD and M_OI produce a wider interval than it should be. While IPT generates only bounds on the possibilities of feasible sets of design variations, PSD generates a full possibility distribution of feasible sets of design variations. The propagation method of a PSD model can also be operated on non-nominal and/or non-convex preference functions since these conditions correspond, respectively, to zero-interval and multiple-interval possibilities that are transparent to interval analysis.

So far, we have described how the PSD-model represents design possibilities and propagates the possibilities over engineering constraints to explore the feasible design space. The set aggregation and modification methods of the PSD-model are briefly introduced here, and full descriptions can be found in [38]. Since CE attempts to incorporate various product lifecycle functions from an earlier stage of design, this approach is intended to cause the designer, from the outset, to consider all elements of the product life cycle from conception to disposal. An excellent solution from one perspective may be a poor solution from another, making it suboptimal for the overall system. Due to the different perspectives, different set-valued assignments may be made into a single design parameter. In aggregating the different PNs, PSD enables the automatic selection of weighting factors and averaging operator (to be able to accommodate the different degrees of compensation in the continuum between non-compensating (MIN) and super-compensating (MAX)), by using the two new concepts such as the compromise limit and the precision and stability (PS) measure. The compromise limit indicates the maximum degree of compensation permissible by a designer. State-of-the-art measures of uncertainty (e.g., Shannon's entropy measure, γ-level measure, modified Shannon's entropy measure, etc.) often fail to give correct measures of the subnormal membership functions or differently shaped membership functions [56]. A PS measure is a different kind of uncertainty measure that can characterize both the information precision and information stability, and consistently produce reasonable measures regardless of the height and shape of preference functions. A more quantitative aggregation method is also provided for the aggregation of PGs. In addition, it is important to have a mechanism to support the designer in generating design modifications and determining the impact of a design modification. The modification method of a PSD-model is based on M_OI, but applies a different uncertainty measure – the PS measure – and employs quantitative dependency [53] to estimate the rate of change. Our PSD-model has been successfully applied to concurrent engineering design problems [38].

Most CAD systems are very powerful in managing exact numbers and measurements, but not for outlining vague ideas [7]. Due to the DIS-model and PSD-model, our SBPD practice can be regarded as a useful approach to obtain much benefit from CE by the simultaneous progress of design, modeling, and evaluation in a CAD system.

5 Implementation of an SBPD-based 3D-CAD system

Widely used solid modeler kernels include PARASOLID (EDS), ACIS (Spatial Technologies), Granite One (PTC), DESIGNBASE (RICOH), etc. Here, PARASOLID is used to construct a prototype system to implement the proposed SBPD practice. Figure 7 shows the main components of an SBPD-based 3D-CAD system. The operations by the designer and system are outlined as follows

Step 1 As described previously, the DIS-model lets designers add and associate geometric or non-geometric information to the CAD geometry, and store it with the CAD geometry. The designer creates a parametric solid model (1a) and defines a number of geometric and non-geometric parameters as well as constraints (1b). Those parameters and constraints are embedded into the appropriate data elements of the solid model as attribute data (1c).

Step 2 The DIS-model enables designers to collect various types of geometric and non-geometric design information in a timely fashion. The designer can search for, sort, and edit any information associated with the product geometry (2a & 2b).

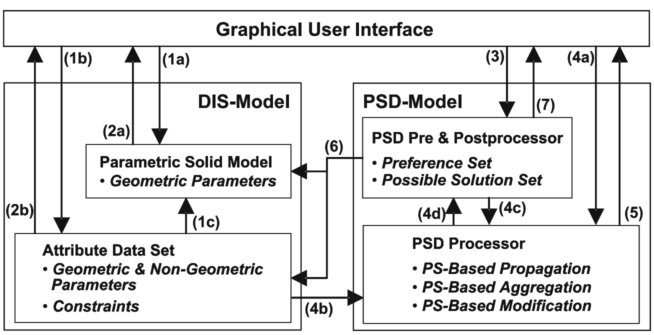


Fig. 7. Main components of an SBPD-based 3D-CAD system

Step 3 Once a set of parameters intended to be associated with the solid model are defined, preference set-valued assignments are made into the parameters by the designer, thereby evaluating the constraints.

Step 4 If the above steps are successfully completed, the designer can initiate the PSD processor (4a). The processor then propagates the designer's (input) preference sets (4c) over the given constraints (4b), and the resultant (output) solution set is passed to the PSD postprocessor (4d).

Step 5 When the processor cannot propagate the inputs or fails to obtain the correct outputs, it issues an error message to the designer with some reasons about the failure, and if possible, requests the designer to input data required to repair the error.

Step 6 From the preference sets or possibility distributions of each parameter, the preprocessor sets the most preferred or possible values (i.e., a single value at $\alpha = 1.0$) to the corresponding parameters of the DIS-model. According to the parameter values, the system displays the correct solid model.

Step 7 By changing parameter values within all input sets, the designer can perform the parametric study to explore a lot of design possibilities.

For a better understanding of these operations, a simple design example is presented. Figure 8 shows the generation process of a DIS-model for a shaft design example. The designer generates a parametric solid model in which a geometric parameter about the radius of shaft, r, is defined (a). This block has no detail geometry or dimension. By clicking the menu labeled "Attribute" or the solid itself, the designer can define other parameters that will be used to add geometric and non-geometric design information as attribute data. There are two types of attributes – a system attribute and a user attribute – in the hierarchical menu (b). On the one hand, system attribute is defined in a structured form. For example, the material is a general attribute for (mechanical) engineering design and the relevant properties (e.g., density, Young's modulus, Poisson's ratio, etc) can be retrieved with a search engine. As mentioned in Sect. 4.2, a system attribute can be also used to tailor various design information to suit specific applications (e.g., finite element analysis, thermal analysis, design for manufacturability, etc.). On the other hand, a user attribute is defined for the specific design problem. Assuming that the radius of the shaft, r, is related to the torque, T, and torsional stress, ts, in a design constraint, those non-geometric parameters, T and ts, also need to be defined and associated with r (d). Using the DIS-model, the designer can collect and refer to various types of design information (e).

Once the parameters and constraints required for the given design problem are identified, the designer can immediately initiate the PSD processes as described in Sect. 4.3 (see Fig. 9). Suppose a design constraint, $r = \sqrt[3]{\frac{16T}{0.9\pi ts}}$, is described with the constraint editor (c) and some input preference sets are specified by a designer (d). In the current implementation, nine kinds of preference functions are available, but there is no restriction on the preference shapes. Here, T and ts are specified by type 6 and

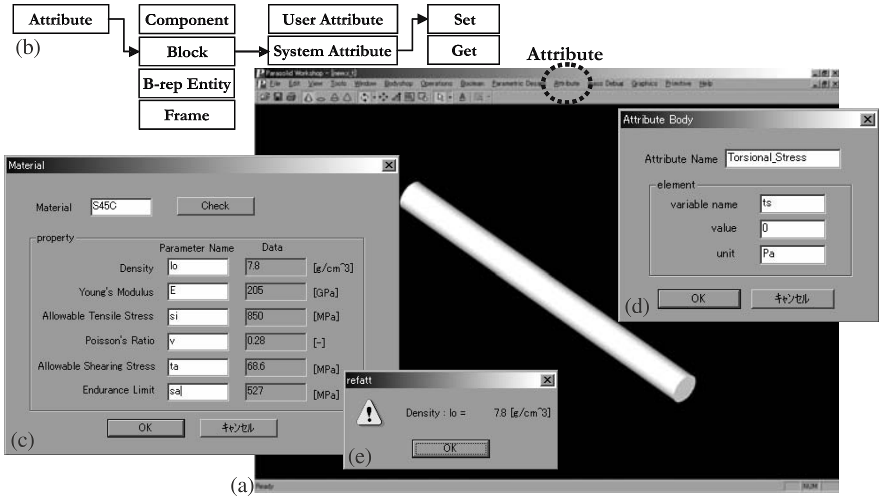


Fig. 8. Steps for DIS-model definition in SBPD

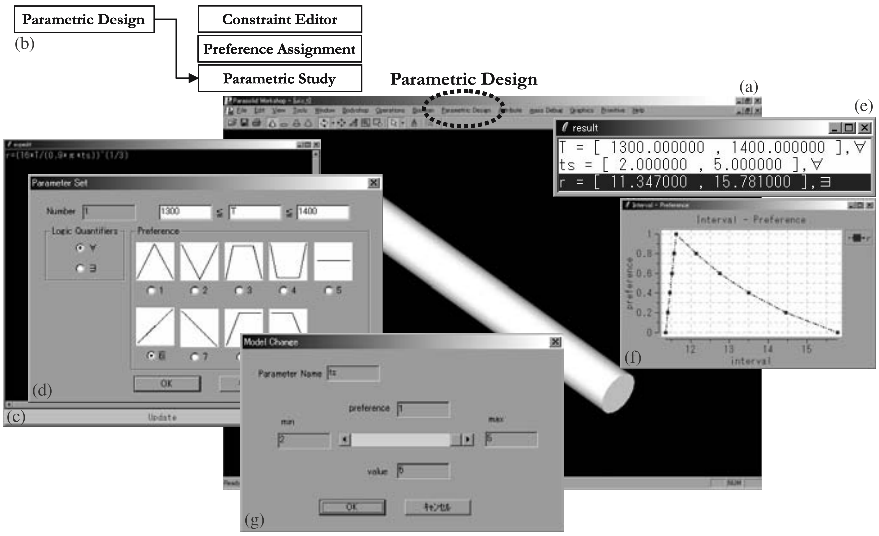


Fig. 9. Steps for PSD-model initiation in SBPD

7, respectively. Type 6 means that the degree of preference increases as the value increases, and Type 7 is the reverse of type 6. Both T and ts are universally quantified, and r is existentially quantified. To obtain the possible sets of r, the PSD processor then propagates the specified preference sets over the given constraint. If the computation is successfully completed, the result is provided to the designer (e & f). Note that the proposed system links the parametric aspects of a CAD model to slider bars so that the designers are in a simulation mode for exploring design possibilities (g). These possibilities are all possible what-if results based on the parameter values designers choose, and the current values are assigned into the corresponding parameters of the DIS-model.

As a more complex design problem, the multispeed (or multistage) gearbox design problem is illustrated using the proposed system (see Fig. 10). This gearbox model is composed of a com-

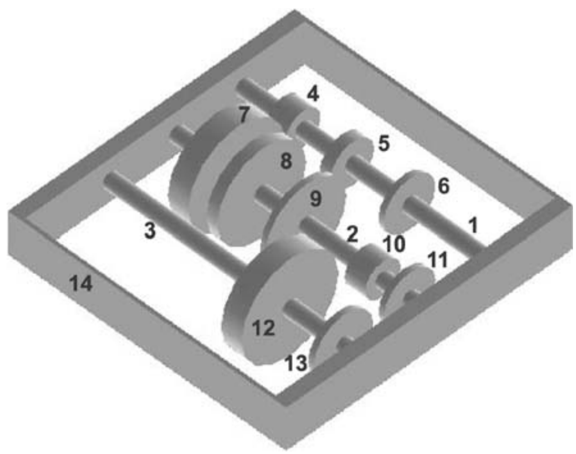


Fig. 10. A multispeed gearbox design model

ponent that contains 3 shaft blocks, 10 gear blocks and 1 housing block, and is similar to the punch picture (solid model with rough geometry) that is usually used at the conceptual design stage [8]. The main parameters that are defined as attribute data in the DIS-model are summarized in Table 1. In fact, the "parameter name" is used in the constraint description.

Figure 11 shows the SBPD process of the multispeed gearbox design problem, including the imposed design constraints (a), preference set-valued assignments (b), propagation results of the PSD model (c), and its graphical representation (d). The usability of the proposed system could be enhanced by the capability of user-friendly design modification and its prompt graphical display. Figure 12 shows two different design alternatives due to the change of material (b) and number of teeth (c). As mentioned above, slider bars permits varying any parameter. In this way, designers can see different design alternatives.

6 Conclusions and future work

This paper presents a vision of set-based parametric design (SBPD) that combines the set-based design practice and parametric modeling technique. A design information solid (DIS) model and a preference set-based design (PSD) model are pro-

Table 1. Main attribute data embedded into the solid model

Type	Attribute name	Parameter name	Embedded entity	Unit
Geometric data	Radius_Of_Shaft	$R1 - R3$	Face, Edge (Block 1–3)	mm
	Length_Of_Shaft	l	Face (Block 1–3)	mm
	Distance_Of_Shaft	$D1 - D3$	Face (Block 1–3)	mm
	Radius_Of_Gear	$r1 - r10$	Face, Edge (Block 4–13)	mm
	Width_Of_Gear	$b1 - b10$	Face (Block 4–13)	mm
	Distance_Of_Gear	$d1 - d10$	Face (Block 4–13)	mm
	Width_Of_Housing	$D4$	Face (Block 14)	mm
	Height_Of_Housing	$d11$	Face (Block 14)	mm
Non-geometric data	Transmitted_Power	L	Component	kW
	Input_Speed	$n1$	Component	rpm
	Min_Output_Speed	$n36$	Component	rpm
	Gear_Module	m	Block 4–13	mm
	Number_Of_Teeth	$z1 - z10$	Block 4–13	–
	Density	lo	Component	g/cm^3
	Young's_Modulus	E	Component	GPa
	Poisson_Ratio	v	Component	–
	Shear_Stress	ta	Component	Pa
	Endurance_Limit	sa	Component	Pa

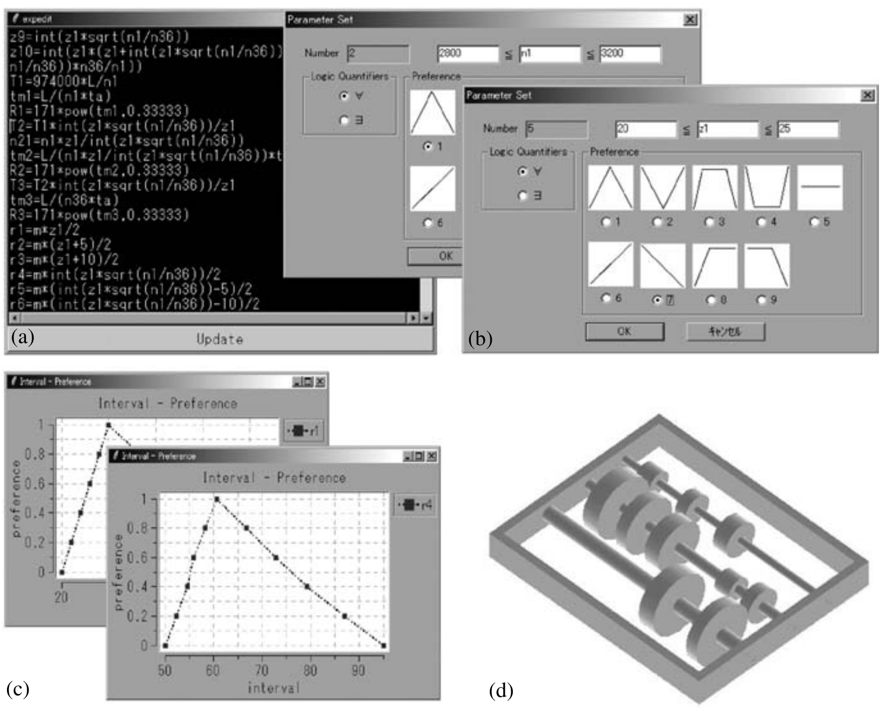


Fig. 11. An SBPD process of gearbox design

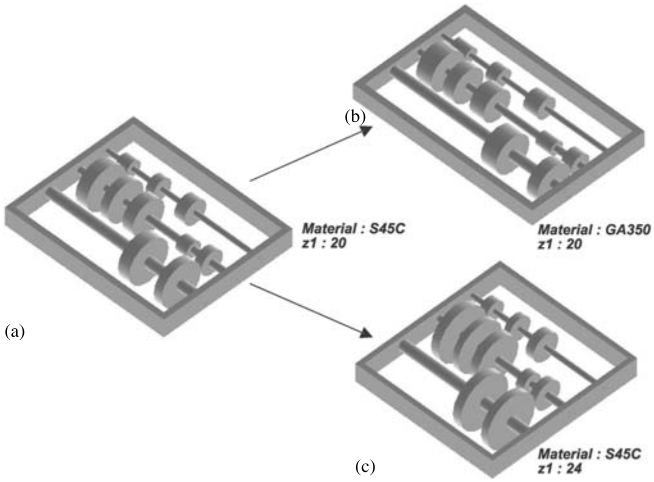


Fig. 12. Exploration of a set of design possibilities

posed to incorporate the SBPD approach into current 3D-CAD systems. A DIS-model can be regarded as an integrated CAD data structure that enables the persistent association and consistent management of geometric and non-geometric design information with the CAD geometry. A PSD-model is a new set-to-set mapping approach for the preliminary engineering design. A prototypical 3D-CAD system is constructed by combining the DIS-model and PSD-model, and the SBPD practice is successfully illustrated using the proposed system.

This paper has focused on the two drawbacks of current 3D-CAD systems: the lack of the ability to handle vague and incomplete designs and the absence of functionality for representing and managing non-geometric design information. However, there are a number of fundamental issues yet to be addressed. The advances in computer networks and information technology have brought engineering design into a new era. Much attention should be paid to more contemporary issues including (data-level, interface-level, and application-level) integrations of heterogeneous 3D-CAD systems, the collaboration and interoperability of legacy CAD systems (including 3D-CAD systems, finite element analysis tools, etc.), and so on. These systems are under development by the authors and will be fully discussed in the near future.

In addition, the proposed PSD-model captures the designer's preference structure and makes correct inferences about engineering constraints. However, it often does not produce a correct output interval for some combinations of input intervals. This would imply that no value of output could satisfy the given constraint. This problem is caused by the direct use of IPT. In the current implementation, the designer needs to employ a trial-and-error procedure to obtain a correct output interval. In some cases, such iterations seem not to be trivial. Therefore, a new mechanism is required to check the reasonableness of the input sets.

Acknowledgement The authors wish to thank Mr. Yasuhiro Kuba, our graduate student, for his programming efforts.

References

1. Anumba CJ (1996) Functional integration in CAD systems. Adv Eng Softw 25:103–109
2. Anumba CJ (2000) Integrated systems for construction: challenges for the millennium. In: Proceedings of the International Conference on Construction Information Technology, Hong Kong, pp 78–92
3. Monedero J (1997) Parametric design. A review and some experiences. In: Proceedings of the 15th Education and research in Computer Aided Architectural Design in Europe, Austria, pp 369–377
4. Paluri S, Gershenson JK (2001) Attribute-based design description system in design for manufacturability and assembly. Trans SDPS J Integr Des Process Sci 5(2):83–94
5. Pan Y, Geng W, Tong X (1996) An intelligent multi-blackboard CAD system. Artif Intell Eng 10:351–356
6. Kurumatani K, Tomiyama T, Yoshikawa H (1990) Qualitative representation of machine behavior for intelligent CAD system. Mech Mach Theory 25(3):325–334
7. Engeli M, Kurmann D (1996) Spatial objects and intelligent agents in a virtual environment. Automat Constr 5:140–150
8. Ishikawa H, Yuki H, Miyazaki S (2000) Design information modeling in 3-D CAD system for concurrent engineering and its application to evaluation of assemblability/disassemblability in conceptual design. Concurrent Eng: Res Appl 8(1):24–31
9. Gould LS (2001) New approaches to CAD integration. Internet draft, http://www.autofieldguide.com/archive.html
10. Nahm Y-E, Ishikawa H (2004) Integrated product and process modeling for collaborative design environment. Concurrent Eng: Res Appl 12(1):5–23
11. Nahm Y-E, Ishikawa H (2005) An Internet-based integrated product design environment. Part I: A hybrid agent and network architecture. Int J Adv Manuf Technol 27(3):211–224
12. Ishikawa H, Nahm Y-E (2003) Set-based design support by using multi-agent approach. In: Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Applications. Madeira Island, Portugal 2003, pp 117–122
13. Ward A, Liker JK, Cristiano JJ et al. (1995) The second Toyota paradox: how delaying decisions can make better cars faster. Sloan Manage Rev 36(3):43–61
14. Liker JK, Sobek, DK II, Ward AC et al. (1996) Involving suppliers in product development in the United States and Japan: evidence for set-based concurrent engineering. IEEE Trans Eng Manage 43(2): 165–178
15. Sobek DK II, Ward AC (1996) Principles from Toyota's set-based concurrent engineering process. In: Proceedings of the 1996 ASME Design Engineering Technical Conference, Irvine, pp 135–143
16. Sobek DK II, Ward AC, Liker JK (1999) Toyota's principles of set-based concurrent engineering. Sloan Manage Rev 40(2):67–83
17. Cutkosky MR, Tenenbaum JM, Brown DR (1992) Working with multiple representations in a concurrent design system. Trans ASME J Mech Des 114:515–524
18. Xue D, Dong Z (1994) Developing a quantitative intelligent system for implementing concurrent engineering design. J Intell Manuf 5:251–267
19. Xue D, Yadav S, Norrie DH (1999) Knowledge base and database representation for intelligent concurrent design. Comput-Aided Des 31:131–145
20. Gorti SR, Gupta A, Kim GJ et al. (1998) An object-oriented representation for product and design processes. Comput-Aided Des 30(7): 489–501
21. Gu P (1992) PML: product modeling language. Comput Ind 18(3): 265–277
22. Bradley HD, Maropoulos PG (1998) A relation-based product model for computer-supported early design assessment. J Mater Process Technol 76:88–95
23. Gayretli A, Abdalla HS (1999) An object-oriented constraints-based system for concurrent product development. Robot Comput-Integr Manuf 15(2):133–144
24. Chang K-H, Silva J, Bryant I (1999) Concurrent design and manufacturing for mechanical systems. Concurrent Eng: Res Appl 7(4):290–308

25. Silva J, Chang K-H (2002) Design parameterization for concurrent design and manufacturing of mechanical systems. Concurrent Eng: Res Appl 10(1):3–14
26. Gupta SK, Regli WC, Nau DS (1994) Integrating DMF with CAD through design critiquing. Concurrent Eng: Res Appl 2(2):85–95
27. Mantyla M (1988) An introduction to solid modeling. Computer Science Press, New York
28. Gorti SR, Sriram RD (1996) From symbol to form: a framework for conceptual design. Comput-Aided Des 28(11):853–870
29. Ranta M, Mantyla M, Umeda Y, Tomiyama T (1996) Integration of functional and feature-based product modeling – the IMS/GNOSIS experience. Comput-Aided Des 28(5):371–381
30. Shu H, Liu J, Zhong Y (2001) A preliminary study on qualitative and imprecise solid modeling for conceptual shape modeling. Eng Appl Artif Intell 14:255–263
31. Schmitt G, Engeli M, Kurmann D, Faltings B, Monier S (1996) Multi-agent interaction in a complex virtual design environment. AI Commun 9(2):74–78
32. Feijo B, Lethola N, Bento J, Scheer S (1996) Reactive design agents in solid modeling. In: Gero JS, Sudweeks F, eds. Proceedings of Artificial Intelligence in Design '96, pp 61–75
33. Bento J, Feijo B (1997) An agent-based paradigm for building intelligent CAD systems. Artif Intell Eng ineering 11:231–244
34. Sriram RD (2001) Standards for collaborative product development. In: Proceedings of the 8th ISPE International Conference on Concurrent Engineering: Research and Applications, 2001, pp 11–19
35. Zhang Y, Zhang C, Wang HP (2000) An Internet-based STEP data exchange framework for virtual enterprises. Comput Ind 41:51–63
36. Rosenman MA, Wang F (1999) CADOM: a component agent-based design-oriented model for collaborative design. Res Eng Des 11(4):193–205
37. Pahng F, Bae S, Wallace D (1998) Web-based collaborative design modeling and decision support. In: Proceedings of the 1998 ASME Design Engineering Technical Conferences, pp 21–22
38. Nahm Y-E, Ishikawa H (2005) Representing and aggregating engineering quantities with preference structure for set-based concurrent engineering. Concurrent Eng: Res Appl 13(2):123–133
39. Wood KL, Antonsson EK (1989) Computations with imprecise parameters in engineering design: background and theory. ASME J Mechanisms Transmissions Automat Des 111(4):616–625
40. Otto KN, Antonsson EK (1991) Trade-off strategies in engineering design. Res Eng Des 3(2):87–104
41. Antonsson EK, Otto KN (1995) Imprecision in engineering design. Trans ASME J Mech Des 117(B):25–32
42. Scott MJ, Kaiser RW, Dilligan M, Glaser RJ, Antonsson EK (1997) Managing uncertainty in preliminary aeroshell design analysis. In: Proceedings of the 1997 ASME Design Engineering Technical Conference, Sacramento, pp 31–41
43. Giachetti RE, Young RE (1997) Analysis of variability in the design of wood products under imprecision. In: Proceedings of the FUZZ-IEEE Conference, Barcelona, Spain, 1997, pp 33–40
44. Finch WW, Ward AC (1996) Quantified relations: a class of predicate logic design constraints among sets of manufacturing, operating and other variables. In: Proceedings of the 1996 ASME Design Engineering Technical Conference, Irvine, 1996, pp 98–107
45. Finch WW, Ward AC (1997) A set-based system for eliminating infeasible design in engineering problems dominated by uncertainty. In: Proceedings of the 1997 ASME Design Engineering Technical Conference, Sacramento, 1997, pp 55–66
46. Finch WW (1997) Predicate logic representations for design constraints on uncertainty supporting the set-based design paradigm. Doctoral dissertation, University of Michigan
47. Ward AC (1989) A theory of quantitative inference for artifact sets, applied to a mechanical design compiler. Doctoral dissertation, Massachusetts Institute of Technology
48. Ward AC, Lozano-Perez T, Seering WP (1990) Extending the constraint propagation of intervals. Artif Intell Eng Des Anal Manuf 4(1): 47–54
49. Terwiesch C, de Meyer A, Loch CH (2002) Exchanging preliminary information in concurrent engineering: alternative coordination strategies. Organ Sci 13(4):402–419
50. Lu SCY, Bukkapatnam STS, Ge P, Wang N (1999) Backward mapping methodology for design synthesis. In: Proceedings of the 1999 ASME Design Engineering Technical Conference, Las Vegas, Nevada 1999, pp 15–26
51. Kusiak A, Szczerbicki E (1992) A formal approach to specifications in conceptual design. Trans ASME J Mech Des 114(4):659–666
52. Parunak HVD, Ward AC, Sauter J (1999) The MarCon algorithm: a systematic market approach to distributed constraint problems. Artif Intell Eng Des Anal Manuf 13(3):217–234
53. Kusiak A, Wang J (1995) Dependency analysis in constraint negotiation. IEEE Trans Syst Man Cybern 25(9):1301–1313
54. Lial ML, Greenwell RN, Ritchey NP (2002) Finite mathematics. Addison-Wesley, Boston
55. Dong WM, Wong FS (1987) Fuzzy weighted averages and implementation of the extension principle. Fuzzy Sets Syst 21(2):183–199
56. Luoh L, Wang W-J (2000) A modified entropy for general fuzzy sets. Int J Fuzzy Syst 2(4):300–304