Current and Future Research in the Conceptual Design of Mechanical Products

Wynne Hsu and Irene M.Y. Woon
Department of Information Systems and Computer Science
National University of Singapore
Lower Kent Ridge Road
Singapore 119260
{whsu,iwoon}@iscs.nus.edu.sg

Abstract

Decisions made at the conceptual design stage have significant influence on factors such as costs, performance, reliability, safety and environmental impact of a product. However, knowledge of all the design requirements and constraints during this early phase of a product's life cycle is usually imprecise, approximate or unknown. Faced with such complexity, individual designers have restricted themselves to narrow, well-defined sub-tasks and as a result, progress in this area has been patchy and spasmodic. The purpose of this survey is to document the current state of research and development in this crucial design activity and in doing so, identify avenues of fruitful exploration. In this paper, we (1) introduce an easy-to-refer classification scheme, (2) provide a comparison of the advantages/disadvantages and limitations between the various techniques/tools and (3) suggest possible future research directions.

1 Introduction

Product design is an iterative, complex, decision-making engineering process. It usually starts with the identification of a need, proceeds through a sequence of activities to seek an optimal solution to the problem, and ends with a detailed description of the product. Generally, a design process consists of three phases. Phase 1 is product design specification where information about the product is collected and defined in precise yet neutral terms. Examples of terms used in a typical product design specification are performance, quality, reliability, safety, product life span, aesthetics, and ergonomics. Phase 2 is the conceptual design whose primary concern is the generation of physical solutions to meet the design specification. The final phase is the detailed design. In this phase, final decisions on dimensions, arrangement and shapes of individual components and materials are made with due consideration given to the manufacturing function. Figure 1 summarizes the three phases of the design process.

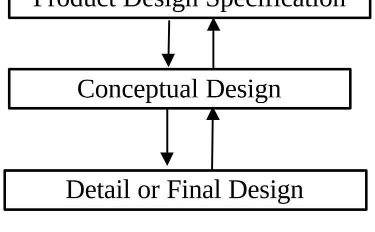


Figure 1. The Phases of Design

A study conducted by Lotter [Lotter] indicates that as much as 75% of the cost of a product is being committed during the design phase. More importantly, a poorly conceived design concept can never be compensated for by a good detailed design. Consequently, researchers have focused their attention and effort in developing tools and techniques that are able to support the various phases of design. Initially, effort was focused in providing support for the detailed design phase. Recently, increasing attention is being directed to support conceptual-level design activity.

To effectively support conceptual-level design activity, we need to resolve two inherent difficulties: (1) modeling the complex interactions between various facets of a product, and (2) reasoning about the generation and selection of feasible solutions. In the domain of VLSI design, these problems are relatively well defined. However, this is not so for the domain of mechanical products. The design of mechanical products is complex because they are, in general, multi-faceted. To describe a mechanical product, we need to express its function, its behavior, and its structure. Function is the perceived use of the device by the human being. Behavior is the sequence of states in which the device goes through to achieve the function. Structure refers to the physical components or forms that are utilized to achieve the behavior. Kuipers [Kuipers, 1984] illustrates this distinction with the example of a steam valve in a boiler. The function of the steam valve is to prevent an explosion, its behavior is that it opens when a certain pressure difference is detected and its structure is the physical layout and connection between the various physical components. Having expressed the various aspects of a mechanical product, we need to be able to understand the interactions between these different facets so as to be able to generate and select some feasible solutions. We refer to the latter as the reasoning problem while the former is referred to as the modeling problem.

A number of tools and techniques have been proposed in the literature to address the modeling and reasoning problems associated with the conceptual design of mechanical products. A lot of these tools/techniques are tailored to specific products or to specific aspects of the design activity. Each of them has their advantages and disadvantages. To aid researchers, it is necessary to keep stock of what tools/techniques have been proposed, their advantages and disadvantages, and their application domains. In this paper, we perform a preliminary survey of the modeling and reasoning tools/techniques that have been proposed in the literature to support conceptual design of mechanical products.

2 Classification Scheme

Due to the complex nature of design and the vast variety of mechanical products, the chance of being able to take an existing tools/techniques off the shelve and directly applying it to a new problem is very slim. Thus the usual classification schemes, along the lines of methodology-tools spectrum or processes involved in conceptual design (user requirements, component selection, component synthesis, transformation), are not very useful. Both schemes do not give a comprehensive view of the underlying representation or reasoning techniques used to support conceptual design. This is essential if the designer/researcher wants to customize some existing modeling and reasoning tools/techniques for their specific domains. In our survey, we extract the modeling representations and the reasoning techniques separately. The classification of the modeling representations is along the line of computer needs versus human needs. The classification of the reasoning techniques is divided into the types of reasoning performed and whether the technique requires large amount of data or is more procedural-oriented. Section 3 discusses the modeling representations. Section 4 presents the various reasoning techniques. Finally, the conclusions are summarized in Section 5.

3 Modeling Representations

One of the main difficulties in supporting conceptual design is the complexity involved in modeling the many facets of a mechanical product. In this section, we will explore the various representation schemes used in modeling the different facets of a design. The representations range from the formal specification methods such as languages to the highly visual representations such as images as shown in Figure 2. Computer-oriented modeling techniques refer to those whose primary concern is to ensure that computational reasoning can be carried out efficiently. On the other hand, human-oriented modeling techniques focus on providing conducive modeling environments that aid the creativity of human designer.

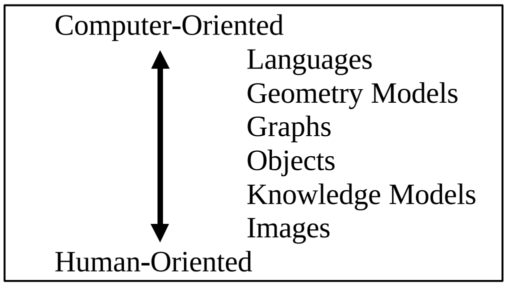


Figure 2. Spectrum of Modeling Representations

3.1 Language

Language represents an attempt at formalizing design. It is useful in expressing our understanding of designs unambiguously. In general, a language is defined by a grammar. A grammar is denoted by the quintuple *(T,N,S,P)* where *T* is the set of terminals, *N* is the set of non-terminals, *S* is the start symbol and *P* is the set of production rules. Table 1 [Neville, 1993] shows an example of part of the grammar (expressed in BNF specification language) for describing the positions and motions of each part of a fixed axes mechanism and their relationship between them. The terminals are expressed in bold fonts. The non-terminals are expressed in normal fonts. The start symbol is *Motion* and the production rules are listed in Table 1.

Motion ::= SimpleMotion | ComplexMotion

SimpleMotion ::= <Part, SM_Type, Axis, InitialPosition, Extent, Relations >

SM_Type ::= **Translate|Rotate|Screw|Translateand Rotate|Stationary|Hold**

Extent ::= AxisParameter **by** Amount

Amount ::= **Real | Constant | Variable | Infinity**

Table 1. Example of Part of a Grammar.

With this grammar, we can easily express the motion of any fixed axes mechanical device. For example, the motion of the rack when the lever is unlocked is described by the following predicate:

<**Rack**, **Translate**, O_1, {$x=c_1$, $\theta = 0$}, x **by** c_2, {$0\leq c_1+c_2\leq 4$}} where parts and axes are uniquely described by their names. Initial Part positions are expressed as constraints on motion parameters e.g. $\theta = \Pi$, $0\leq x \leq 4$. The extent of the motion is specified by the amount that the motion parameter changes e.g. a constant number, a variable. Relations between motions are expressed as constraints between motion parameters.

Due to its compact representations, grammar/language is an efficient means of structuring design knowledge. Indeed, many pieces of work have used language/grammar as the underlying representation for their design knowledge. For example, Rinderle and Finger [Rinderle,#116][Rinderle #117] used a graph-based language to describe behavioral specifications of design as well as the behavior of the components. Neville and Joskowicz [Neville, 1993] present a language for describing the behavior of fixed-axes mechanism e.g. couplers, indexers and dwells. Predicates and algebraic relations are used to describe the positions and motions of each part. Vescovi *et. al.* [Vescovi] developed a language, CFRL, for specifying the causal functionality of engineered devices. In terms of grammar, Carlson [Carlson, #141], Stiny [Stiny] and Heisserman [Heisserman] have looked into using shape and/or spatial grammars to express physical design forms. In particular, Mitchell [Mitchell] has combined shape grammars with simulated annealing to tackle the problem of free-form structural design. First, shape grammars are used to generate structural design possibilities. Then, stochastic optimization of the possible designs are achieved using simulated annealing. This allows the generation of large number of sound, efficient free-form solutions that otherwise would never have been imagined. A number of researchers [Reddy][Schmidt_GGREADA][Szykman][Brown,#148][Longnecker][Mullins] have also made use of grammars in engineering applications. Tyugu [Tyugu] also proposed an attribute model based on attribute grammar for representing implementation knowledge of design objects. Similarly, Andersson *et. al.* [Andersson, 1995] proposes a modeling language, CANDLE, which enables the use of engineering terminology to support early design phases of mechanisms and manipulator systems. In CANDLE, the basic taxonomies of engineering terminology are augmented with the physical and solution principles that are specific for the design of mechanisms and manipulator systems. In fact, the general approach adopted by researchers is that they would propose different special purpose languages to describe some aspects of design that they are interested in modeling. Language has been used to describe the functional, behavioral and structural aspects of a product.

A different approach taken is that instead of developing special-purpose languages for each application, effort is directed to developing a shareable design ontology. An ontology is a useful set of terms/concepts that are general enough to describe different types of knowledge in different domains but specific enough to do justice to the particular nature of the task at hand. Alberts [Alberts] proposed YMIR as an engineering design ontology. The “How Things Work” project at Stanford University [Iwasaki] aims to build a large-scale ontology of engineering knowledge. By having a common set of ontology, knowledge can be reused and shared. This allows better integration between the different phases of the product’s life cycle [Cutkosky, 1993].

3.2 Geometry Models

Geometry modeling focuses on representing the structural aspects of a product. The objective is to represent 2-dimensional or 3-dimensional geometric shapes in a computer [Requicha]. Popular representations of geometric shapes include: B-rep (boundary representation), CSG (constructive solid geometry), variational geometry and feature representations. Briefly, in a B-rep approach, a shape is represented by the boundary information such as faces, edges and vertices. The CSG approach models geometric shapes using a set of primitives such as a cube, cylinder or a prism. Complex shapes are built from the primitives through a set of operators (union, difference and intersection). Variational modeling allows a designer to use equations to model mechanical components analytically while in feature representation approach [Dixon, 1988][Libardi, 1986], a part is built from a set of primitive building blocks with the guarantee that this set of building blocks are manufacturable.

B-rep represents geometry in terms of its boundaries and topological relations. For example, the B-rep of a prism is given in Figure 3:

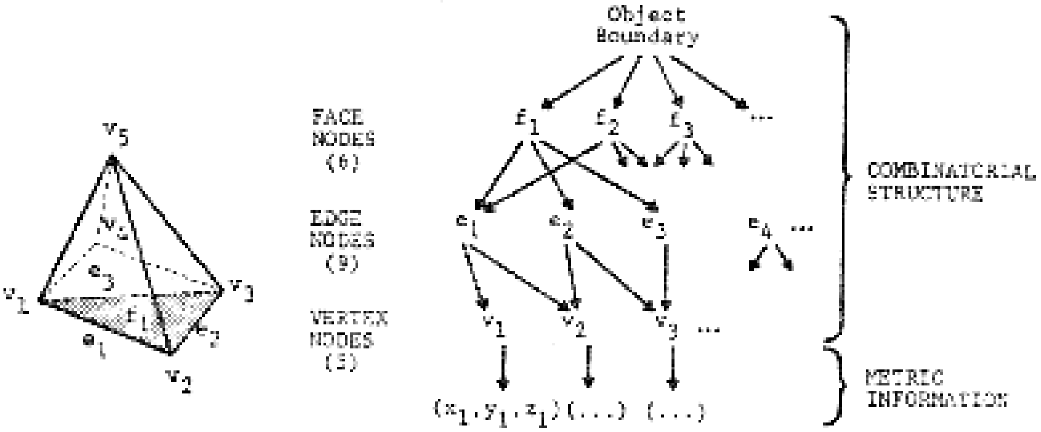


Figure 3. An example of Boundary Representation of a Prism.

The transformation from one topology to another can be achieved using Euler operators. Since Euler operators are sound [Weiler], the topological validity of the structure is guaranteed. The major limitation of B-rep is its inefficiency in performing geometric reasoning.

Besides B-rep, CSG is another geometry modeling technique that was widely accepted by both the research community and industry. For example, the primitives given in Figure 4 can be combined using set operations to form complex solids like that given in Figure 5.

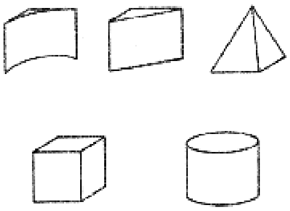


Figure 4. Some CSG Primitives.

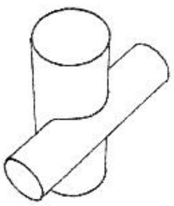


Figure 5. Complex Solid Example.

In spite of its promising start, CSG modeling faces several inherent limitations. The most serious limitation, in our opinion, is the non-uniqueness of the CSG representations. This non-uniqueness of representations makes recognition of shapes from CSG representation extremely difficult. Hence, this tends to dissuade researchers from relying solely on CSG representations alone. In addition, CSG representation does not guarantee that the solid it models is always a valid object. It is possible in CSG representation to model an invalid solid.

Variational Modeling is gaining popularity because it allows the evaluation of competing alternatives. The concept of using variational geometry in computer aided design started as early as 1981. Lin [Lin, 1981] in his thesis described the feasibility of using variational geometry to model geometric information. Light and Gossard expanded upon his work to allow modification of geometric models through variational geometry [Light, 1982].

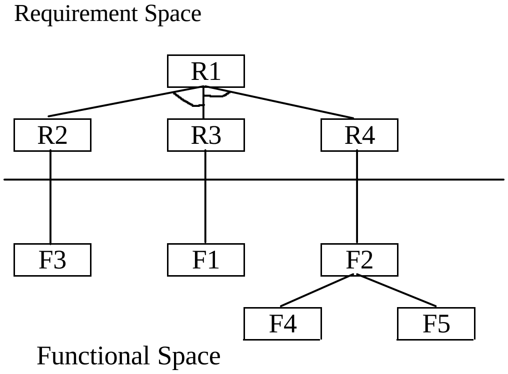
The notion of features was first proposed as form features [Dixon, 1988] [Nakajima] to bridge the gap between units of the designer's perception of forms and data in geometric models. Shapes are described as the way the designer understands them. A feature-based design approach allows a user to use mechanical features stored in a feature library in his design [Luby][Pratt][Roller][Shah]. It provides a means for building a complete CAD database with mechanical features right from the start of the design. However, this approach suffers from the difficulty of a limited number of available feature primitives. It is difficult to satisfy various design needs and in the event that the features interact with interact with one another, new features may arise that can cause complication with the analysis process. EDISON [Dyer] is an example of a system using feature-based modeling. It has a database of known mechanisms and is indexed by their functions, structures and situations in which they are used. Thus far, the majority of feature-based research focuses on using feature-based design for process planning [Cutkosky, 1988][Hayes, 1989] and feature recognition [Gadh, 1993]. Han and Requicha [] proposed a novel feature finder which automatically generates a part interpretation in terms of machining features. The feature finder strives to produce a desirable interpretation of the part as quickly as possible. Alternate interpretation could be generated if the initial interpretation was found not be unacceptable by a process planner.

Recently, the trend has been towards the integration of various representation schemes. Keirouz *et. al.* [Keirouz] proposed an integration of parametric, geometry, features, and variational modeling. With this integration, they showed that the system is able to handle geometry and "what if" questions arising in conceptual design.

In all the above approaches, the assumption is that the support of surface features is well-defined on prismatic objects. This is not the case for sculptured surface models and current methods often lead to data-explosion. Elsas and Vergeest proposed a displacement feature modelling approach [elsas]. In this approach, explicit modelling of protrusions and depressions is done in free-form B-spline surfaces that can achieve real-time response and with unprecendented flexibility.

3.3 Graphs

Graphs and trees are popular representations in the conceptual design stage. They have been used to model all aspects of a product -- function, behavior, and structure. [Malmqvist, 1994] demonstrates how graphs can be used to model the functions of structural systems in mechanics, electronics, hydraulics e.g. hole punch, washing machine. Nodes of graphs are lumped elements which correspond to the different physical properties (capacitance, transformers) and these nodes are connected by edges (bonds) e.g. force, velocity. The power flow direction and causality of bonds are specified. [Murthy, 1987] manipulate a graph of models to modify a given prototype of some structural engineering system e.g. design of beams. A model describes the behavior of the system under certain explicit assumptions. The models form the nodes in a graph and the edges represent sets of assumptions that must be added or relaxed to go between adjacent models. Graph/trees have also been used to model the physical representations of the design components and their layout [Joshi, 1988][Chung, 1989]. Besides modeling structural, behavioral and functional aspects of the product, graph and trees have also been used to model requirements and constraints [Kusiak, 1995#185]. Kusiak and Szczerbicki [Kusiak, 1992] use tree models in the specification stage of conceptual design to represent the functions and requirements of mechanical systems, with an incidence matrix to represent the interaction between requirements and functions. Figure 6 shows the requirement and functional tree for the design of a shaft coupling.



R1:Design a shaft coupling
R2:Nature of the coupling is rigid
R3:Coupling is able to transmit torque
R4:Nature of the coupling is flexible

F1:Transmit energy
F2:Compensate offset of the shaft
F3:Connect two parts of the shaft rigidly
F4:Compensate offset applying a sliding element
F5:Compensate offset without applying a sliding element

Figure 6. An Example of Graph Model.

An arc between the nodes of a tree represents a conjunction. A node without an arc represents a disjunction. There are therefore two sets of requirements that satisfy R1: {R2, R3} and {R3, R4}.

One main advantage of using graphs to model different aspects of design is that graph theory is a developed field of study. By using the graph model, we are implicitly tapping into the rich resources of the many existing graph algorithms with well-founded theoretical basis. The drawback of using graph models is that it lacks the concepts of classes and inheritance. Such concepts are useful in conceptual design.

3.4 Objects

An increasingly popular modeling representation is the object. An object is an entity that combines its data structure and its behavior into one. The advantages of object representation are abstraction (focus on what it does before deciding how to implement it), encapsulation (separating external aspects of an object which are accessible to other objects from the internal implementation details which are hidden from the other objects), polymorphism (do not consider how many implementations of given operation exist) and inheritance (of both data structure and behavior which allows sharing without redundancy).
Figure 7 [Marefat, 1993] shows a component with two slots with its corresponding object representation given in Figure 8.

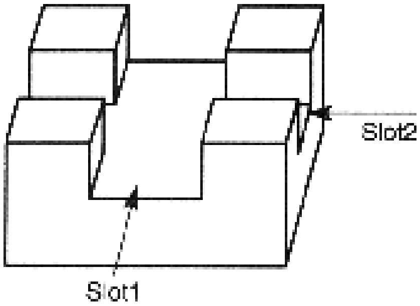


Figure 7. A Component with 2 Slots.

Instance	aSlot

```
Class             Slot
Superclass        Gen-Feature
Initialization
   name           slot1                  {id for the feature}
   length         10.0(1.0 0.0 0.0)      {magnitude and direction}
   width          2.0(0.0 0.1 0.0)       {magnitude and direction}
   depth          1.0(0.0 0.0 0.1)       {magnitude and direction}
   faceCollection (F131 F132 F130)       {id of the faces that make the feature}
   appCollection  ((0.0 1.0 0.0))        {collection of approach directions}
   feedCollection ((0.0 0.0 1.0))        {collection of feed directions}
        :
interactions      ((anInteraction))      {list of instances of Interactions}
```

Figure 8. The Object Representation of Figure 7.

Objects have been used to model many different kinds of entities. Martin [Martin, 1993] proposed an object-oriented tree representation to model metal fatigue and fracture. In his object-oriented tree, each node is a “class”. The root object represents the most general case of fatigue and fracture. Each class has a slot to represent the associated constraints and relationships. A similar approach has been taken by Ohki [Ohki, 1994] to use object-oriented structure to represent constraints (law of physics) and physical objects (diode). In the domain of ship design, Masaharu Yoshioka et. al. [Yoshioka] uses objects to represent the physical objects knowledge and the design process knowledge. Kolb and Bailey [Kolb, 1993] use object-oriented techniques for modeling preliminary designs in the domain of aircraft engine design. Types of objects modeled include: components (physical elements of a design), sub-models (properties of a design as a whole such as total weight, total cost), programs (external analysis codes for evaluating the design components), modules (simple design analyses), links (specifying constraints between objects). A novice approach to geometric reasoning using object-oriented approach was proposed by Nacaneethakrishnan *et. al.* [Nacaneethakrishnan] whereby geometry is abstracted in terms of form features and the spatial relationships between features are represented using intermediate geometry language (IGL). Object algebra is then used to perform geometric reasoning. Kusiak [Kusiak, 1991] uses a hybrid of object-oriented representation and production rules in his CONDES system. The object-oriented representation is used to model design synthesis while the production rules are used to guide the process. Bento et.al [Bento] present a hybrid framework to integrate first-order login into the object-oriented paradigm for representing engineering design knowledge. The login component will be used for representing knowledge that is expected to be subjected to frequent changes throughout the design process, while the objects are used to describe other pieces of knowledge whose structure is less likely to change.

3.5 Knowledge Models

As mentioned earlier, conceptual design is an engineering activity performed early in a product life cycle, where complete and exact information and knowledge of requirements, constraints etc. is difficult to obtain. This highly skilled task is very complex and requires a mixture not only of different sources of knowledge (e.g. costing, performance, environmental issues) but also different types of knowledge (e.g. physical, mathematical, experiential) [Krause, 1995]. To represent such vast sources and types of knowledge, new models have been introduced. We call them knowledge models. The more common forms of knowledge models includes: frames, rules cases and metamodels. Knowledge models are needed to facilitate high-level reasoning. For example, a rule that applies the generic knowledge associated with intersections to learn how a slot’s edge entities are affected (refer to Figure 7) is given in Figure 9.

```
If feature is SLOTA and
        If interactingFeature is SlotB and
```

```
If typeofInteraction is intersecting then
    Send the message intersectingWIth: SlotB to SlotA
    To get edge entities of SlotA based on typeof Interaction
```

Figure 9. An Example of Rule Representation

Besides rule representation, frame representation is also widely used. In the paper by Tong and Gomory [Tong, 1993], he used a frame-based structure to model parts of standard kitchen appliances and light sources. The rule based paradigm is adopted by Rao [Rao, #158] to give advice on which alternative should be chosen in the design of ball bearings. There are at present a number of tools which couple knowledge based systems with conventional systems. Krause and Schlingheider [Krause, 1995] gives a comprehensive overview of such tools e.g. ICAD, MEDUSA-ENGIN, CONNEX. Increasingly, these tools are addressing the problematic areas of development and design [Taleb-Bendiab, 1993][Bracewell, 1995]. Navinchandra [Navinchandra] looks into the use of case representations to support conceptual design activities. In another work by Hsu *et. al.* [Hsu], case representations have also been used to capture assembly-oriented design concepts. The exploitation of a designer’s sense of scale to derive a feasible design is demonstrated by Li [Li, 1993] where this sense of scale is formally represented in a knowledge base and scaling is used to increase the efficiency of the design process. Recent development has been towards the concept of metamodels. A metamodel is a qualitative model of causal relationships among all the concepts used for representing the design object [Kiriyama #127]. The metamodel reflects the designer’s mental model about the structure and behavior of the design object. Metamodel mechanism include the primary model (description of the requirement given by the designer) and aspect models (qualitative and quantitative, focusing on specific aspects of the design object).

Though there have been many successful applications that are built upon knowledge models presented here, a number of issues still remain unresolved. Some of these issues include: the verification of the correctness of knowledge models, the handling of incomplete knowledge, the resolution of inherent contradictions that are present in knowledge models and the incremental addition of new knowledge to existing knowledge models.

3.6. Images

Perhaps the closest to human’s way of thinking and reasoning is through the use of visual thinking models. Visual thinking has its beginning since 1969 [Arnheim, 1969]. It did not gain a high profile in design research until Mckim [McKim, 1980] demonstrated through experimental studies that visual thinking is vital to all branches of design practice. Freehand sketching is good for accelerating discussions and for comparing different solutions [Ullman, 1989]. For example, Figure 10 shows a hand-sketched diagram that aids the designer in visualizing design concepts.

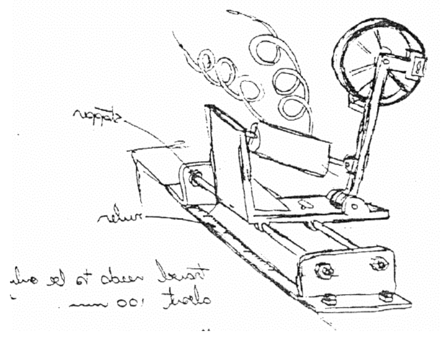


Figure 10. Example of a Hand-Sketched Diagram

Hand-sketched diagram is also good in allowing different views of the sketch so as to obtain a good spatial image of the design solution [Ehrlenspiel]. In 1990, Radcliffe and Lee [Radcliffe] proposed a model for the process of visual thinking that overcomes the barrier between the cognitive processes and the physical domain. Sittas [Sittas] further explored the issues involved in supporting the creation and manipulation of 3D geometry during the conceptual design sketching activity.

3.7 Future Research Directions

It is clear that many different forms of knowledge are required to execute the design process and that this knowledge cannot be captured by any single existing modeling technique [Krause,1995]. Languages are computationally efficient and unambiguous and they are a powerful way of structuring knowledge. However, they fail to account for the relation between shape and function [Cagan, #150] and they are not able to represent the complex reasoning demanded by the conceptual design activity. Geometry models allow designers to represent physical forms in computer format. However, they cannot represent the intricate knowledge in processing and selecting one physical form over another. Knowledge models aim to capture the human designer' reasoning process. Yet, they are unable to fully model the real world constraints and to simulate commonsense reasoning. The advantages of the object-oriented approach are clear and many researchers have adopted object-oriented models in their applications. Unfortunately, up to now, there is still a lack of common consensus as to when to classify an entity as an object and when to classify the entity as an attribute. This ambiguity makes communications between different groups of people and different phases of design activities difficult.

In spite of the tremendous efforts in deriving a suitable modeling technique for conceptual design, there is still a long way to go. Currently, the modeling technique is not able to support the seamless integration between the different phases of design. In addition, the modeling techniques tend to be quite restrictive. The issue of incomplete and abstract knowledge has not been addressed fully. As advances are made in new areas such as hypermedia information processing, it may be worthwhile to explore how these ideas may be incorporated to resolve some of the outstanding issues in design representations.

4 Reasoning Techniques

The second difficulty in supporting conceptual design activity is the difficulty of generating and selecting appropriate means of mapping the user's requirements into some physical structure that can realize the given set of requirements. Many existing mechanical design systems derived this mapping based on a refinement approach [Kusiak, 1990]. The system starts with a rough design that is successively refined at each step. At each refinement stage, constraints that guide the design process is generated. Frequently, the constraints generated cannot be satisfied by existing prototypes in the library and modification of the prototypes is required. However, such modifications is impossible unless one is able to model the function, behavior, and the structure of the product and then apply suitable transformations from one plane of abstraction to another to arrive at candidate sets of solutions. In the previous section, we introduces various ways of representing these models. In this section, we focus on the transformation process.

Theoretically, there are three pairs of mappings to be considered in order to realize the transformation process. They are: function ↔ structure, behavior ↔ structure, and function ↔

behavior mappings. Practically, it is hard to distinguish between function and behavior of a product. For instance, in value engineering, there is no difference between function and behavior. Though some researchers [Sticklen] [Hunt, 1993] argue that knowledge of the function of the device is important as it allows focus on a particular subsystem, thus reducing the complexity of the model to be analyzed, other researchers [Goel, 1992][Huhns, 1992][Kleer, 1984] view function to mean the designer's intended purpose for the product and classify unintended uses for the product as behavior. Therefore, they concentrate on defining all the behaviors of the device, since one or more of these behaviors will represent the device function. In this paper, we adopt the second position. The reasoning techniques are, therefore, classified under the headings of realizing: function → structure, structure → function, behavior → structure, or structure → behavior mappings. In addition, for practical reasons, it is helpful to know whether a particular reasoning technique requires large amount of data (data-intensive) or whether it requires prior knowledge about the domain (knowledge-intensive). Knowing this fact can aid a designer/researcher in selecting the right reasoning technique for his/her chosen domain. Table 2 summarizes the state-of-the-art reasoning approaches along the 2 dimensional space.

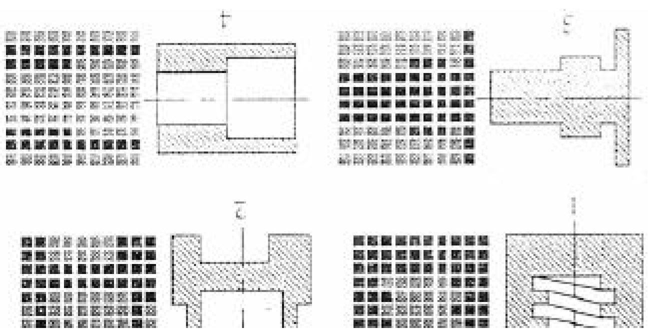
	Data-Intensive	Knowledge-Intensive
Function → Form	Neural Networks Case-Based Reasoning	Knowledge-Based Optimization Value Engineering
Form → Function	Machine Learning	Knowledge-Based
Behavior → Form	Case-Based Reasoning	
Form → Behavior	Qualitative Reasoning	

Table 2: Classification of Reasoning Techniques

4.1 Neural Networks

Artificial neural net models were first introduced in the hope of achieving human-like performance [Lippmann]. Biological neurons transmit signals over neural pathways. Each neuron received signals from other neurons through special connections called synapses. Some inputs tend to excite the neuron; others tend to inhibit it. When the cumulative effect exceeds a threshold, the neuron fires and a signal is sent to other neurons. An artificial neuron receives a set of inputs. Each input is multiplied by a weight analogous to a synaptic strength. The sum of all weighted inputs determines the degree of firing called the activation level. The input signal is further processed by an activation function to produce the output signal, which is transmitted along. A neural network is represented by a set of nodes and arrows. A node corresponds to a neuron, and an arrow corresponds to a connection between neurons. In general, neural networks are good for classification tasks and for performing associative memory retrieval. As a result, many neural networks applications in engineering design is geared towards either classifying the designs into families of design problems [Kumara] or to find the nearest values for the design parameters [Hung]. One way of applying neural networks to classify designs [Kumara] proceeds as follows:

The design problems are represented by a grid of 10x10 pixels that capture the shape of objects (vector size of 100). If a pixel is black (takes a value 1), it represents the presence of the corresponding functional requirement. If it is white (takes a value 0), it represents the absence of



the corresponding functional requirement (see Figure 11). This binary vector is then passed to an adaptive resonance network to classify it into different design families.

Figure 11. 10x10 Pixels of the Shape of Objects.

Grierson [Grierson] proposed a coupling of neural network with genetic algorithms to arrive at an alternate best concept solution through evolution and artificial learning in the domain of bridge structure examples. Taura et.al. also proposed the use of generic algorithm as part of the shape feature generating process model to aid in representing free-form shape features. The common limitations of neural networks and genetic algorithms are that the design must be specified by a limited list of design attributes, this implies that the reasoning carried out is superficial based solely on the similarity of the attributes and their values. In addition, neural networks often require a large set of training data. This proves to be impractical for real-life applications.

Closely linked to neural networks are the machine learning techniques. Recent work in application of machine learning to engineering design can be found in [Archiszewski, 1994], [#28] and [Rao, #29]. Archiszewski is concerned with the adequacy of domain representation (irrelevant attributes, insufficient descriptors) which will allow reasoning by two methods – data-driven constructive induction, and hypothesis-driven constructive induction. O'Rorke focuses on the abductive form of explanation-based learning. Case studies involving explaining physical processes, explaining decision, explaining signals are given. Rao [Rao, #29] concentrates on using machine learning techniques to learn bi-directional models that can provide design synthesis support and hence reduce design iterations. This approach is aimed at parameterized domains (domains with fixed structure) so that all the variables in the design stage are well defined.

Another related area is genetic algorithms.

4.2 Case Based Reasoning

Much of design consists of re-design, in the adaptation of a previous design to a new context, or in the design iteration cycle. Case-based reasoning applies past experience stored in a computerized form towards solving problem in similar contexts. It involves three stages: the representation of cases, the matching and retrieval of similar cases, and the adaptation of the retrieved cases. Figure 12 shows the basic idea of the three stages of case-based reasoning for a redesign problem.

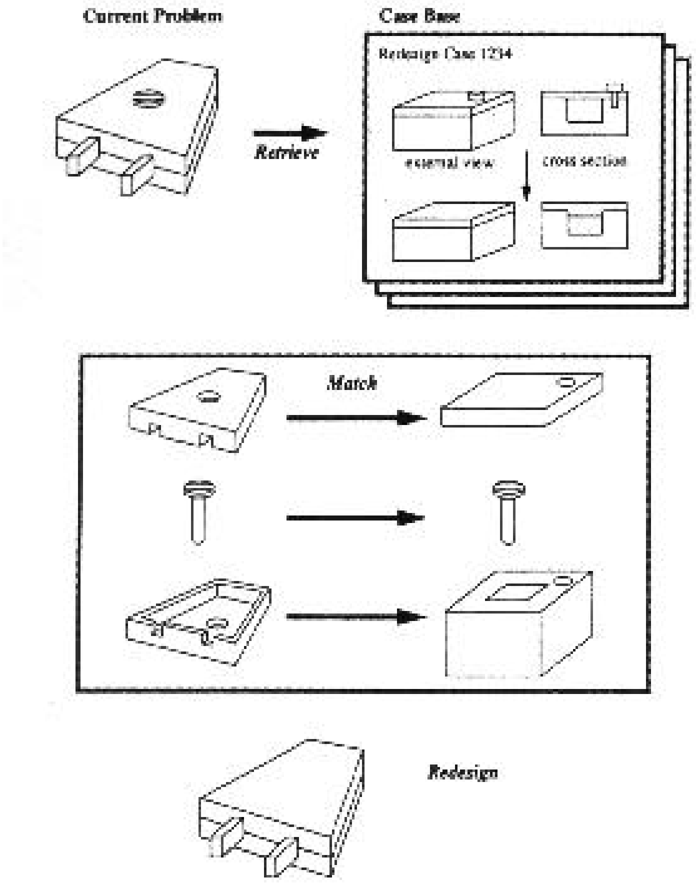


Figure 12. Three Stages of Case-Based Reasoning for Redesign

Case-based reasoning has been successfully applied where the structure and content of design information can be encoded symbolically and manipulated using artificial intelligence techniques. KRITIK [Goel, 1992] solves the function-structure design task in the domain of physical devices. Knowledge of previously encountered designs are organized as a design case which contains the functions it can deliver, and a pointer to the structure-behavior-function model for the design that explains how the structure of the design delivers its functions. The cases are indexed by the functions delivered by the stored designs. In CADET [Sycara, 1992], each case involves 4 different representations: Object-Attribute-Value tuples, functional block diagrams, causal graphs and configuration spaces. Thus, all three levels of abstraction are represented and reasoning using the causal graph enables the structure-function transformation. If no case matches the current specification, transformations are applied to it until it resembles some case in the case database. In a similar approach, Li et. al. employs a library of mechanical devices to aid in the design synthesis process [Li, 1996]. In addition, Irgens [Irgens] has extended the scope of case library to include design intent, design data, and customer feedback, so as to provide a complete integrated historic advice for product prototyping.

Case-based reasoning techniques favor classes of domains where the number of primitive components are large as is the number of possible interactions between them as the computational cost of retrieval and adaptation would be less than generating the solution from primitive components. On the other hand, case-based reasoning cases are stored over a long period of time and for that large number of cases, this may not be practical. A consensus exists among AI researchers that reuse of the process of design rather than the product of the design might be more useful. Researchers like Mostow [Mostow, 1992] and Banares-Alcantara et. al. [Banares-Alcantara, 1995] are experimenting with applying case-based reasoning to design plans. To address the issue of large number of cases, Murakami and Nakajima proposed a computerized method of retrieving mechanism concepts from a library by specifying a required behavior using qualitative configuration space as a retrieval index. During retrieval, only mechanism concepts which realize specific kinematic behavior are retrieved. This effectively reduces the huge search space required.

4.3 Knowledge-Based Reasoning Techniques

Knowledge bases are used to capture procedural design knowledge as well as product or domain knowledge. We investigate types of reasoning techniques appropriate to interpretations of design descriptions: abductive, deductive, constraint-based, and non-monotonic reasoning.
Abductive reasoning says that:

> The surprising fact C is observed;
> But if A were true, C would be a matter of course.
> Hence there is reason to suspect that A is true.

In other words, abductive reasoning (goal directed) tries to derive the premises of a stated conclusion.

On the other hand, deductive reasoning says that:

> Suppose if A is true, then C would be a matter of course,
> Now, we observe the fact A.
> We can conclude that C is true.

Hence, deductive search (data driven) moves to arrive at some conclusion, given the initial facts.

An example of an abductive search strategy is given in Tong and Gromory [Tong, 1993] in the design of small electromechanical appliances. Rao [Rao, #158] shows the use of deductive search strategy in selecting the appropriate ball bearings' design for a set of input parameters e.g. load type, bearing speed, environment of use, etc. Arpaia et al. [Arpaia, 1995] makes use of both patterns of reasoning in the design of measurement systems, in mapping from the logical attributes to the physical components of the instrument. Typically, abductive and deductive reasoning will face the problem of scaling-up. Deductive reasoning will produce the correct results only if the premises from which the conclusions derive and that the link between the premise and the conclusion are correct. Abductive reasoning concludes from observed symptoms to presumed cause and as such, relies on the validity of the premise-conclusion relationship.

A refinement to the idea of goal-directed search is the distinction between constraints and objectives. A constraint is a statement about a design, the truthfulness of which does not depend on any tradeoffs with goals. For example, the manufacturing cost of the product of around $100 is a constraint whereas a manufacturing cost objective is to have the product manufactured at the lowest cost. In many instances, it may be possible to translate an objective into constraints e.g. the objective "minimize manufacturing cost" could be stated as manufacturing cost should be less than or equal to $100. Kolb and Bailey [Kolb, 1993] specifies constraints between objects derived from analyzing the design of an aircraft engine, and employing constraint propagation technique to integrate and perform mathematical analyses of the resulting solution which is the set of design parameters that satisfies all constraints. Oh *et. al* [Oh, 1995#184] give an example of how a constraint based approach may result in the improved design of a video cassette tape. In another attempt, Vujosevic et. al. [Vujosevic, #187] use a reason maintenance system to perform goal-directed search. An assumption-based truth maintenance system and multiple worlds are used to discover and store information about feasible designs and to avoid further consideration of infeasible design alternatives.

The systems of reasoning presented thus far assume a consistent fact case and knowledge base. However, conceptual design is invariably carried out under conditions of incomplete knowledge. In order for the design to progress in a rational manner, assumptions are made for the value of some parameters. Later on in the design process as more information becomes available, assumptions and information inferred by them retracted. This field of study is known as non-monotonic reasoning. Research into this for conceptual design is almost nonexistent, with the exception of work by Smith and Boulanger [Smith, 1994] where assumptions (defined as defaults and preferences) are accommodated within the rules and models for the preliminary design of a bridge.

4.4 Optimization

Various optimization techniques have been applied to the problem of engineering design. In general, design problems are represented as follows. Let the continuous variables be x and the discrete variables by y. The parameters which are normally specified as fixed values are represented by theta. The design goal (or goals) can be expressed as the objective function $F(x, y, \theta)$. This function is a scalar for a single criterion optimization, and a vector of functions for a multi-objective optimization. Equations and inequality constraints can be represented as vectors of functions, h and g, that must satisfy,

$$h(x, y, \theta) = 0$$

$$g(x, y, \theta) \leq 0.$$

Many techniques have been proposed to solve optimization problem. A survey of the state-of-the-art optimization techniques in structural design can be found in Koski [Koski]. The focus of the survey is primarily based on the Pareto optimality concept. Briefly, Koski classified the multi-criteria structural design process into three phases. The first phase is the problem formulation where the criteria, constraints and design variables are chosen. The second phase is the generation of Pareto optimal solutions. The final phase describes the decision-making procedure employed to select the best compromise solution. In another paper by Levary [Levary], he draws attention to the interaction between operation research techniques and engineering design. Specific applications of operation research methods are discussed with respect to the following engineering disciplines: computer engineering, communication system engineering, aerospace engineering, chemical engineering, structural engineering and electrical engineering.

Optimization approaches are not always easy to apply because the data required by the algorithms may not be available, their scope of applicability restricted and algorithms are not able to provide optimal solutions because of the problem’s complexity.

4.5 Value Engineering

Value Engineering is a process applied to achieve focused conceptual design goals. It was first described in the literature by Bytheway [Bytheway], Ruggles [Ruggles], and Miles [Miles, 1982]. In Value Engineering, a designer begins with a description of the basic function of the design.

The functions are then decomposed until each function can be mapped to a component that will accomplish it. Using the chair as an example [Sturges], a possible functional specification is shown in Figure 13. By emphasizing different functions (for example, choosing to support the knee instead of bottom) leads to different class of design (see Figure 14).

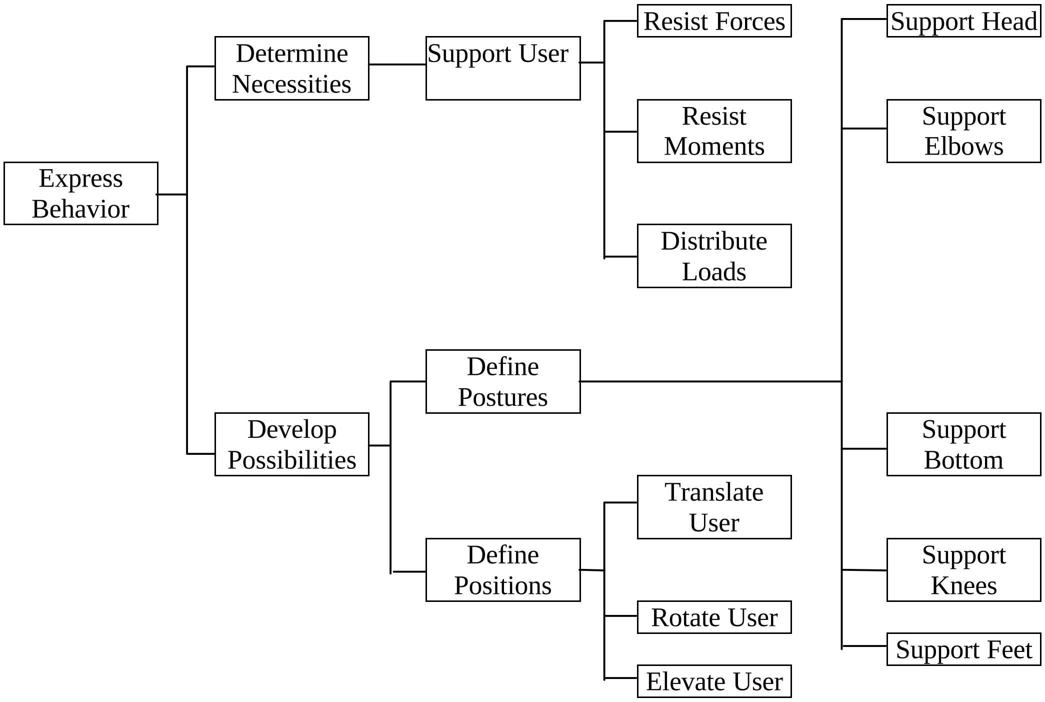


Figure 13. Functional Specification for a Chair

Figure 14. Two Designs of A Chair.

However, in spite of its value in achieving focused conceptual design goals, there are very few computer-aided tools to support the process of value analysis. This may be due to the high-level and often subjective set of verbs and nouns used in value engineering process.

4.6 Qualitative Reasoning

Qualitative Reasoning (QR) is defined as the identification of feasible design spaces using symbols and intervals of continuous variables. This allows formal simplified representations about a domain that maintains enough resolution to distinguish and explain the important features of behavior while leaving out the irrelevant details. For example, we are interested in whether water in the pan is hot or cold rather than its exact numerical value. Qualitative reasoning is therefore particularly pertinent in early design phases when little quantitative information is available. The device-centered ontology proposed by De Kleer and Brown [Kleer, 1984] deals with the problem of deriving function or behavior of system given its structural descriptions and some initial conditions. All possible behaviors are determined by generate-and-test or constraintsatisfaction technique. Other researchers [Faltings, #171] [Kurumatani, 1990] have applied the process-centered ontology [Forbus, 1984] to represent and reason about the states and behaviors of mechanical devices that handles kinematics and dynamics of mechanical devices.

EDISON [Hodges, 1992] is a project which aims to construct an engineering design invention system by employing qualitative reasoning in the domain of mechanics. Work on this project has considered how functional knowledge can be integrated with qualitative reasoning. Many other researchers [Roddis, 1991], [Schwartz, #164], [Bozzo, 1992] have proposed the use of qualitative reasoning to structural engineering design problems. Fruchter [Fruchter, #71] applies qualitative reasoning at different structural abstractions (structural, process and structure parameter) level to select design modifications arising from performance problems of lateral load resisting frame structures. Murthy [Murthy, 1987] uses a graph of models approach in PROMPT to analyze a prototype based on first principles and derive its behavior from its structure. Li et. al. [Li, 1996] recently propose a combination of qualitative and heuristic approach to the conceptual design of mechanisms. The basic idea is to represent and classify a library of mechanical devices qualitatively and then employs best-first heuristic searches to generate a set of feasible design alternatives from a given specification. Figure 15 shows an example of qualitative description for a mechanical device.

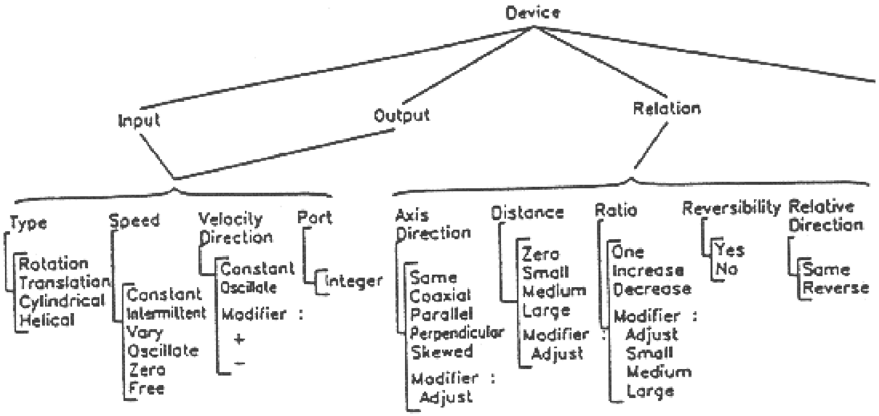


Figure 15. An Example of Qualitative Description of a Mechanical Device.

One of the limitations of qualitative reasoning is the generation of spurious behaviors that is a result of ambiguity. However, ambiguities are seen as 'strong points' by De Kleer and Brown; they use it as a means to explore alternative interpretation of the same system. Ways of resolving ambiguity include maintaining information about partial ordering relations of parameters [Simmons, #165], incorporating heuristics [Kalagnanam, 94][Schwartz, 1995] and maintaining a semi-qualitative model [Parsons, 1991][Shen, 1992].

There is not enough resolution in qualitative representations to reason effectively about space, shape and spatial events. The ability to address this shortcoming will have important implications on the field computer aided design. Some researchers working in this field are Gelsey [Gelsey, 1987], Faltings [Faltings, 1989], Forbus [Forbus, 1991].

4.7 New Trends in Conceptual Design

With the rapid increased in popularity of the World Wide Web, new research places more and more emphasis on the need to support collaborative design. Pahng et. al. proposed a framework for modeling and evaluating product design problems in a computer network-oriented design environment. In product design, many inter-related design decisions are made to meet potentially competing objectives. These decisions may span many disciplines. Thus there is a need for an integrative framework that enables designers to rapidly construct performance models of complex problems and for information sharing on the internet. Approaching the same problem from a different angle, Ndumu and Tah examined the use of agents to assist the design effort. An agent is a self-contained program capable of controlling its own decision-making and acting based on its perception of its environment, in pursuit of one or more objectives. Using two examples from construction supply chain provisioning and building design, the authors demonstrate the advantages that an agent-based approach brings to collaborative design. Yet another approach taken to aid in the integrated design environment is proposed by Tichkiewitch and Veron. They present two models (product model and data model) and two exchange modes (a formal mode and an informat mode) to facilitate co-operative work between partners of the life-cycle of the product. Lu et. al. focused on the integration of various design and analysis models into a cohesive set to aid the collaborative negotiation process demanded by the design activity.

A separate yet distinct shift in the focus of design research is toward the support for virtual prototyping. Virtual prototyping refers to the analysis of a product without actually making a physical prototype of the part. Such analysis may be performed via the aid of expert system agents that reside in a distributed fashion on the internet. These agents require CAD information at different level of abstractions. Gadh and Sonthi [] look into the different levels of geometric abstraction for achieving such virtual prototyping. Koch and Raczynski looks into using the virtual reality techniques as a supplement of the conventional CAD and rapid prototyping methods to achieve virtual prototyping. A digital mock-up is designed in virtual reality and is referred to as virtual prototype. Drews and Weyrich focus on discussing the interaction and functional simulation of virtual prototyping. Dani and Gadh use the virtual reality environment to allow the creation of concept shape designs rapidly. Cartwright presents a modeling approach that allows the user to experiment, explore or make changes to the virtual prototypes.

4.8 Future Research Directions

While much progress has been made in the reasoning techniques to support conceptual design activity, there remains a large gap in transferring these techniques to the real-life design applications. This is because many of the techniques face the problem of scaling-up. In addition, some of these techniques make certain simplifying assumptions that are unrealistic for real-world applications. For example, in the case of using neural networks to map functions to forms, the implicit assumption is that there exists a one-to-one mapping from functions to forms. This is certainly not the case since one function can be realized by many different forms and one form can satisfy multiple functions. Thus, a large part of conceptual design activity still depends largely on the creative abilities of the human designer. As such, it is our belief that the future of reasoning techniques lie in two directions: (1) addressing the problem of scaling up by integrating datamining techniques to automatically uncover interesting domain knowledge that is useful for the conceptual design activities; and (2) exploring a closer mode of interaction with human designer to take advantage of the creative abilities of the human designer.

5 Conclusion

In this paper, we have performed a preliminary survey of the tools and techniques that have been proposed to aid in the conceptual design of mechanical products. In particular, we focus on the modeling and reasoning techniques underlying the each proposed tools/methodology. The survey results show that in spite of the great advances in both the modeling and reasoning techniques, much remains to be done. We hope this survey will serve to motivate researchers to look closely at the underlying modeling and reasoning techniques for conceptual design of mechanical products, and perhaps to derive an integrated framework for the next generation of computer-aided design tools.