Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

P.J. Veerkamp, T. Kiriyama, D. Xue, T. Tomiyama

# Representation and implementation of design knowledge for intelligent CAD
## Theoretical aspects

Computer Science/Department of Interactive Systems Report CS-R9070 December

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

# Representation and Implementation of Design Knowledge for Intelligent CAD
## Theoretical Aspects

Paul Veerkamp
*Department of Interactive Systems*
*Centre for Mathematics and Computer Science*
*P.O.Box 4079, 1009 AB Amsterdam, The Netherlands*

Takashi Kiriyama, Deyi Xue, and Tetsuo Tomiyama
*Department of Precision Machinery Engineering*
*Faculty of Engineering, The University of Tokyo*
*Hongo 7-3-1, Bunkyo-ku, Tokyo 113, Japan*

*Abstract:* This paper is the first part of two combined papers about design knowledge representation and its implementation. In this paper we focus on theoretical aspects of the subject. This paper is organized in the following way. First, we describe a design process as step-wise refinement along three design stages, i.e. conceptual, fundamental, and detailed design. Secondly, as a general model of design activities in these design stages, we propose metamodel. Thirdly, based on the model, we discuss design knowledge representation. Since a design object is constrained by the physical laws of the real world, knowledge about a physical world, i.e ontological knowledge, is indispensable to design a new artifact. We present metamodel mechanism which allows an ICAD system to support a designer with ontological knowledge. And a physical feature library is proposed as a knowledge base of physical phenomena.

*CR Categories and Subject Descriptors:* D.3, F.4.1, J.6.
*Key Words & Phrases:* intelligent CAD, knowledge representation, ontological knowledge, qualitative physics, metamodel, aspect-model

## 1. Introduction

Recent development of ICAD (intelligent CAD) has desperately diverged into various directions [15, 22]. For example, there is a strong belief that anything called expertise can be fragmentally coded and this may lead to *design expert systems* [6]. Since design is an activity that deals with physical objects in the real world, it might be interesting to codify what regulates physical entities. From a point of view of CAD systems for mechanical engineering, management of geometric constraints is, among other things, considered of crucial importance [2, 14]. Despite these serious efforts, however, the definition of ICAD is still unclear.

The IIICAD project, initiated by ten Hagen and Tomiyama [17], is an attempt to build an intelligent, integrated, interactive CAD system and can be contrasted with other approaches in that it aims at building a framework or basis to implement a wider range of design knowledge. The IIICAD system is interested in representing knowledge on both design processes and objects rather than just design objects [20].

Report CS-R9070
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

This paper presents a mechanism to describe both design processes and design objects in such a way that it can be used for creative design. We call it the *metamodel evolution mechanism*. A metamodel is a context-free description of a design object which is manipulated during the course of design. In this paper we demonstrate how a metamodel evolves during the design process. Furthermore, we introduce ontological knowledge as a means to represent knowledge about physical phenomena. It is used to understand and envision behavior of an design object. This kind of knowledge is indispensable for a system which aids a designer in performing creative design.

This paper is organized in the following way. It is the first part of a combination of two papers, and it focuses on theoretical aspects of design knowledge representation. The second paper deals with implementational aspects [21]. In §2 we introduce a general design process model. This model is capable of representing a wide range of stages of the design process, from conceptual design to detailed design. The usage of this model to represent ontological knowledge is described in §3. We show how ontological knowledge can be employed to describe a creative design process. In the second part we describe a knowledge representation language, which is used to implement such a process.

## 2. Design Process Representation

In this section we present a formalization of the design process described in three steps. First, in §2.1 we subdivide the entire design process into several design stages. Each of these stages corresponds to a certain phase of the design process. During these phases, the design object model evolves from an initial specification to a manufacturable description of the product. At each stage of the design process, a designer performs activities which are particular for that stage. Also, at each stage the design object model is manipulated in different ways. In this section, we give along with the theory of design, its implications for the design process and design object models. One of our assumptions is that an intelligent CAD system is used by the designer to assist him in performing a design job. We accordingly show which role an ICAD system plays being an integral part of the design process.

Secondly, we derive a design process model from the theory. Each stage of the design process is mapped to the model. Such a model is presented in §2.2 to provide a design process representation. In the model, design is described as a stepwise refinement process where the design object model is transferred from one state to another. It allows a designer to manipulate a design object in several ways during a sequence of actions. Each action corresponds to a design step belonging to one of the design stages.

Thirdly, in §2.3 we show how such a design step is taken. The goal of a design step is to obtain more information about the design object model. In other words, during a design step the design object is extended in order to get a more detailed description. After each design step the design object model is evaluated to check whether it still meets the initial specifications.

### 2.1. Design process theory

Designing is an activity which is based on both knowledge and experience. Examination of designers at work shows that each individual designer tackles a design problem in different way. If you ask a designer why he actually designs in the way he does, he gives an unsatisfiable answer. He does not really know [11, 18]. However, out of observations from literature it became clear that there are three distinguishable successive stages in the design process: *conceptual design, fundamental design,* and *detailed design* [8, 10, 16]. Traditionally, these stages are particularly distinguished in mechanical engineering due to differences in the way drawings are used.

These design stages have in common that they operate on a design object representation. But, each stage has its own demands on representational issues. An overview on these different stages and the corresponding design object representations is depicted in Fig. 1. The implication is that we need a design object model which allows for the representation of properties characteristic for each of the three stages. What kind of characteristics these are is shown in §2.1.1 through §2.1.3.

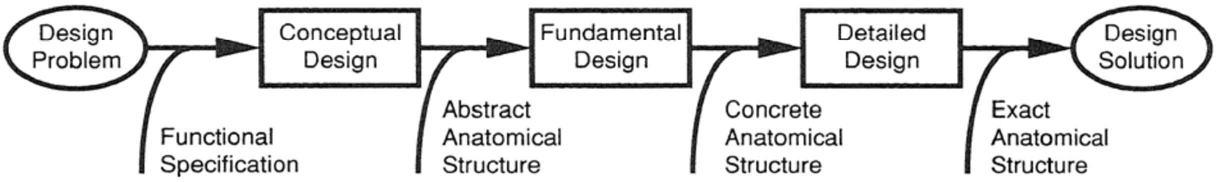



**Fig. 1. Different stages of design with corresponding design object representations**

**2.1.1. Conceptual design.** Design starts with a need, the statement of a design problem. A design problem does not necessarily have to be an entirely new problem; it might have been solved by previous designs. Design is thus often a matter of improving existing designs. The necessity for these improvements may be caused by several reasons, e.g. changed requirements, a disappointing performance, excessive costs, etc. In another case, the design problem may be of a total new kind. The former is called *routine* design, the latter *creative* design. The way both categories of design problems are solved is basically the same. The difference lies in the amount of time spent in one of the four stages. However, creative design demands more from an ICAD system than routine design, since for routine design existing design schemes are known and can be applied. For creative design new design schemes have to be developed.

In the early stage of design the problem is analyzed by the human designer, and the output of this analysis consists, among others, of:

- a precise statement of the problem in terms of function and behavior,
- limitations placed upon the resulting product, e.g. spatial requirements, cost constraints, international standards etc.
- the amount of quality that should be worked to.

The last item is in most cases the bottleneck; how to produce reasonable quality at the lowest possible costs. The analysis of the design problem is carried out without use of our system. The result of this phase is called a *functional specification*.

The functional specification of a design problem is used as a starting point for the design process performed by an ICAD system. A designer supplies a functional specification to the system, and the system translates it to an initial design object model. The statement of the design problem is thus transformed into an *abstract anatomical structure*, which describes the problem in terms of broad solutions. The broad solutions are represented in the form of design *schemes*. The transformation from a design problem specification to an abstract anatomical structure is accomplished through an interaction between the designer and the design system. The schemes specify the kind of dialogue which will be held, in other words, they denote the design process knowledge. In this phase the most important decisions are taken and it makes the greatest demands on the designer. It is the designer who decides what kind of design scheme is executed.

**2.1.2. Fundamental design.** During the course of fundamental design the abstract anatomical structure is converted to a more concrete anatomical structure. It is a description of something we can actually make; a rough decomposition of the artifact is created. The principal shape of the design object is fixed. A primary solution for the major components of the decomposition is chosen.

To find such a partial solution a designer uses experience obtained during previous design sessions. It is often the case that a certain part of the design object has been designed before, or that it is similar to a part which has been designed before. For this purpose the designer possesses a collection of *prototype* solutions which are applied as standard components for parts of a new design [12]. Such a collection of possible design solutions is part of an ICAD system. A library of standard components allows the designer to use a certain component as a prototype for a part of a new design. The prototype may be modified by the designer according to his wishes. If a suitable prototype is absent, then a designer is allowed to choose a prototype which resembles closest to the desired component and to change it completely. The design knowledge which is necessary to choose the right prototypes, and to manipulate them, is also denoted by means of design schemes.

This phase of the design process consists of several steps. Initially, there is hardly any structure in the model, most of the parts are unknown. Then during several steps the model is gradually structured. When the last phase of fundamental design is reached, the entire structure of the design object model is determined. Now, the only thing which needs to be done is to refine the model and work out details.

**2.1.3. Detailed design.** Prior to this phase of the design process, a complete structure of the design object model is defined. The major parts of the design object are determined and described by a decomposed model. The purpose of detailed design is to produce an exact description of the anatomical structure. Dimensions and tolerances are set, all constraints are satisfied and all parts are integrated into one coherent model. Therefore, all attributes of the model receive a definite value, and all relations among the various parts are defined. The model is verified with the initial specifications, and it is evaluated to check whether the requirements are met.

The design is focused on specific parts of the design object model without worrying about global issues. Local optimizations are achieved which result in small changes. The main issue at this phase, is to allow a designer to concentrate on a certain part of the design object. The part is highlighted and it is modeled in its own context, i.e. special conditions which only apply for this particular part are now valid. After being modeled such a part is replaced in the whole, and checked whether it still fits. An ICAD system allows the designer to generate such models on specific parts of the design object model. It is done by certain design schemes.

During this stage of design, the designer consults various experts, to obtain some information on various aspects of the design. These experts perform domain specific calculations, or evaluate the design object model in a certain context. They add new information to the design object model. Sometimes the designer asks several experts for information at the same time. Each expert adds data from its own field of expertise to the design object description. Some experts may add contradictory information to the design object description, since they have different background knowledge. It is the designer's task to maintain the consistency of the design object model.

Those experts can also be consulted in an ICAD system: they are called *aspect* models. Aspect models allow the designer to pose a question on a specific domain to the system. The system incorporates knowledge about that domain of expertise. The knowledge is represented by means of schemes.

### 2.2. Design process model

In this section we give a general model of the design process applicable to the three stages of the design process presented in the previous section. The *general design model* is employed to define a system's architecture and to develop language constructs for design representation language [19]. The model is used to build a framework which guides the design process as it is executed by the system in order to understand the designer's demands. Knowledge about the three stages of the design process is embedded in the framework. Thus, the system is always informed about the current stage of the design process and the current state of the design object [1]. In other words, the designer decides how to perform the design process. An ICAD system is an intelligent aid to the designer to assist him in achieving his goal by giving him the right tools for each specific stage of the design process.

The knowledge how to perform the design process and which tools are applicable at a certain instant is denoted by means of *scenarios*. These scenarios represent the design schemes introduced in the previous section. They describe what kind of actions must be carried out at a specific phase of the design process. For each phase of the design process there is a different set of scenarios. The framework in which the scenarios are executed is based upon the general design model presented in this section.

**2.2.1. From specification to solution.** We use the design process theory presented in §2.1 as a basis for giving a formalization of design processes and design knowledge. The theory is influenced by the Extended General Design Theory of Tomiyama and Yoshikawa [16], which is based on axiomatic set theory. It models design as a mapping from the function space where the design object specifications are described in terms of functions, onto the attribute space where the design solutions are described in terms of attributes. Roughly speaking, one starts with a functional specification of the design object and ends up with a manufacturable description. In the remainder of this section we present a general design model which is derived from the theory. The overall outlook of the general design model is depicted in Fig. 2.

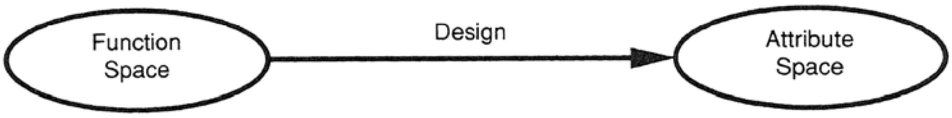



**Fig. 2. Design process model**

The basic ideas behind the general design model are as follows:

- From the given functional specifications a candidate for the design solution is selected and refined in a stepwise manner until a complete solution is obtained, rather than by trying to get the solution directly from the specifications. The latter is not possible in a non-trivial design problem, since it involves a very complex object with a multitude of parts.
- The design process is regarded as an evolutionary process which transfers the model of the design object from one state to another, gradually obtaining a more detailed description. The number of attributes grows as the design process proceeds and a growing number of the functional specifications is met.
- To evaluate the current state of the design object model, various interpretations of the design object model need to be derived in order to see whether the object satisfies the specifications or not.

We call those interpretations of the design object model *contexts* and they can be regarded as interpretations of the design object observed from certain points of view. Contexts allow a designer to model the current state of the design object in a certain environment, i.e. they represent an aspect model. More information about the design object is obtained through these contexts and hence the number of attribute grows. Contexts are created by means of *scenarios,* which contain design knowledge and data necessary to build an aspect model. Scenarios perform the reasoning about a context and lead the dialogue with the designer.

**2.2.2. Stepwise refinement of metamodel.** Considering the general design model, the system starts from the functional specification of a design object and continues the design process until a design solution is obtained. During this process the design object model is refined in a stepwise manner. The central description of the design object is called a *metamodel.*

A metamodel is the design object model which is used during the design process as a central model from which aspect models are derived. A metamodel is a context-free description of the design object. It contains all entities the design object is composed of, and it includes the relationships and dependencies among these entities. Data which is used in one of the contexts derived from the metamodel is stored apart from the metamodel. For instance, geometric data of the design object is not to be found in the metamodel, but in a geometric aspect model. An example of a metamodel is shown in Fig. 3. It contains four entities and several relationships among these entities. Note that there are relationships which affect only a single entity. Other relationships affect several entities. We present the metamodel in detail in §3.

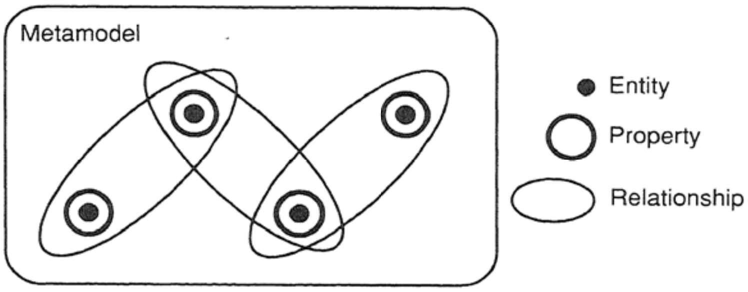



**Fig. 3. Example of metamodel**

The stepwise refinement process shown in Fig. 4 behaves as follows: at a certain stage of the design process the metamodel $M_{i-1}$ is the current, incomplete description of the design object. In order to get a more detailed description an aspect model is derived from the metamodel. Through this aspect model some new information about the design object is obtained. After this refinement the new information from the aspect model is merged into the metamodel $M_{i-1}$. If the merge is successful, i.e. the new information is consistent with the current $M_{i-1}$, then the result of the merge is a new state of the metamodel, $M_i$. This process is continued, obtaining $M_{i+1}$, etc., until the design object model is a complete and satisfactory description of the desired artifact. Here 'complete' means that the description has enough information to manufacture the design object.

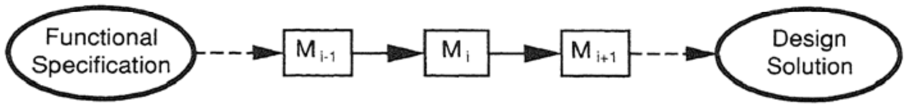
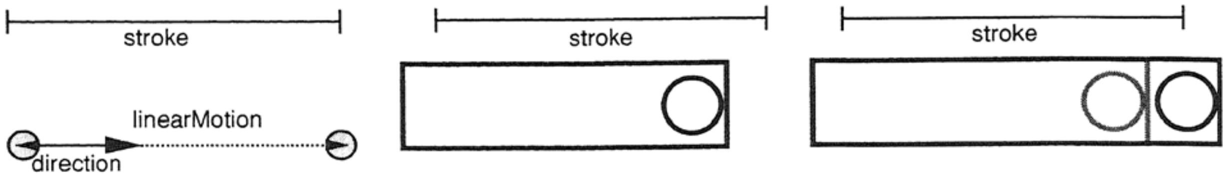



**Fig. 4. Stepwise refinement of the metamodel**

In Fig. 5 an example of this process is depicted. It shows several stages of the design of an linear motion mechanism. In Fig. 5.(a) the state of design is at the conceptual design phase. The metamodel consists of an abstract anatomical description of the design object. Some states later the design is arrived at the fundamental design phase (see Fig. 5.(b)). The metamodel is a concrete anatomical description of the design object without detail, i.e there are some inconsistencies between the geometric and the kinematic representation. The stroke achieved by the geometric representation is not the desired stroke. The detail is present in Fig. 5.(c) when the design is at the detailed phase. Here, the metamodel consists of an exact anatomical description which is almost complete. Inconsistencies caused by results from different aspect models are removed.

**2.2.3. Multiple representations.** The design process as described above deals with the ideal situation in which the stepwise refinement process is a linear process from functional specifications straight to the design solution. It can be regarded as a sketch of the design process in retrospect. In practice, it is merely a process of trial and error, rather than the straightforward process shown in Fig. 4. The designer might not be satisfied with a certain state of the design and wants to redo it from a certain point. But he keeps in mind the things which were useful and which were not, and the redesign will therefore be more efficient. In another occasion the




a. Conceptual Phase b. Fundamental Phase c. Detailed Phase

**Fig. 5. Example of stepwise refinement**

designer might like to regard the design object from different points of view at the same time, i.e. he wants to create multiple aspect models concurrently in order to compare the outcome from different experts.

During the course of the design, an occasion frequently occurs that the designer is not satisfied with the current state of design. Instead of redesigning everything from scratch, the designer wants to preserve part of the results. The designer restarts the design from a previous design state which still met his demands. The implication for the general design model is that it must be possible to withdraw the current metamodel and perform some *backtracking* to a previous one. In consequence, each individual state of the metamodel must be maintained as the design process proceeds. On top of that, when the design is continued from a prior metamodel, it must be prevented from taking the direction which led to the unwanted result. Thus not only the metamodel states, but also the design process history must be maintained.

In Fig. 6 an example of the backtracking process is shown. For some reasons, the metamodel $M_j$ does not fulfill the designer's requirements, so he decides to redesign from a prior state. In this case, he backtracks to the previous metamodel $M_{i-1}$. The design is restarted from this state taking a different direction. The design now proceeds to metamodel $M_i$, and so forth.

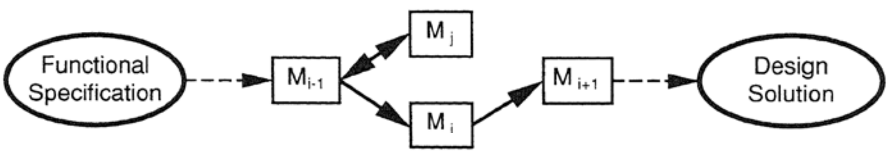



**Fig. 6. Backtracking to a previous metamodel**

## 2.3. Metamodel evolution scheme

The general design model as we have introduced it so far, gives a global outline of the design process, without going into details. We now focus on how to transfer from one state of the metamodel to the next; i.e. how do we perform a design step? Here we elaborate on the concept of a *context*. Through the creation of a context the designer provides an interpretation of metamodel in a certain environment, i.e. he generates an aspect model. For each design step there exists an associated scenario, which creates a context in which the design object is modeled. The scenario contains the design knowledge which is applicable for its context, and it knows how to derive new data from the context and the current state of the metamodel. In a context new information about a design object is obtained. The current metamodel together with the new information form the next metamodel. This section deals with the mechanism which transfers one state of the metamodel to the next one.

**2.3.1. Contexts.** The design process is a continuous manipulation of the design object model. A certain aspect of the artifact is highlighted and some properties about the design object are changed. This highlighting is done by contexts. By applying a context, the attention is focussed on a specific part of the design object model. A context embodies a part of the metamodel with the addition of *aspect-data,* design procedures and design rules. Aspect-data is information about the design object specific for a certain context, e.g. geometric information. It describes the context dependent properties of the design object. An example of a context is shown in Fig. 7. A context is an interpretation of the metamodel focusing on a certain aspect of the design object. It is used to derive new properties or to update uncertain or unknown properties about the metamodel in order to get a more detailed description. The new properties about the design object, which are derived in a context, are merged with the original metamodel when the modeling is completed.

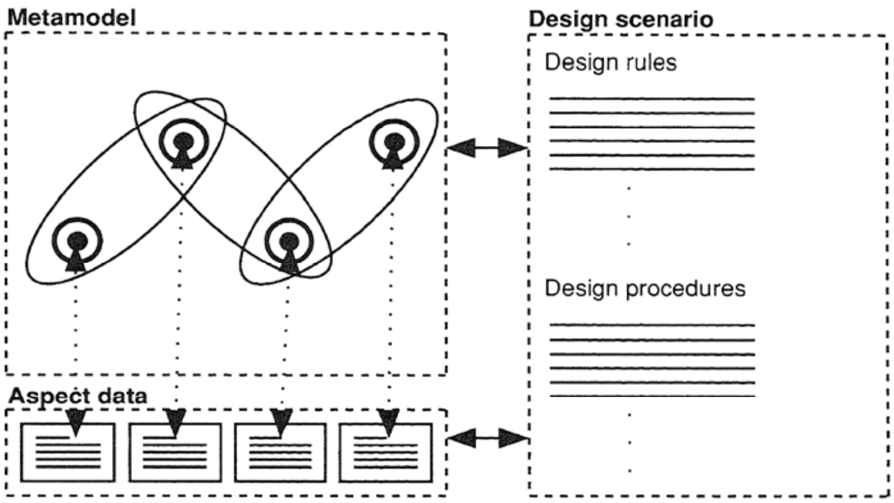



**Fig. 7. Example of context**

A context is created by the execution of a scenario. A design scenario consists of design procedures and design rules which describe the procedural and declarative design knowledge respectively. The design knowledge is applied to the metamodel and to the aspect-data. A context consists of (a part of) the metamodel together with knowledge about the design object being valid in that particular context. The designer can interact with a context and change its contents. After the session the contents of the context being evaluated is mapped to the next state of the metamodel. The mapping mechanism is presented in the section below.

An example of such a context is a geometric representation of the design object. A part of the metamodel is collected and processed in a context with geometrical knowledge in order to generate an image of that part of the design object. In other words, there exists a scenario which creates a geometric aspect model. The aspect-data of the model consist of the geometric attributes of the design object. The design procedures in the scenario describe geometrical functions, and the design rules describe geometric knowledge. Such a context describes the mapping of the metamodel onto a description suitable for the generation of a geometric model.

**2.3.2. Metamodel evolution.** *Metamodel evolution* is the mechanism which is applied to perform a design step. It transfers a metamodel from its current state to the next, according to the design process model introduced in §2.2.2. The mechanism is driven by scenarios. The designer selects a design scenario which is appropriate for the current state of the metamodel $M_i$. The scenario is executed in a context $c_i$ and performs in dialogue with the designer some action on the context. The execution of a scenario continues as long as new information can be obtained in the context. The acquisition of new information is accomplished through the design procedures and the design rules in a scenario.

The contents of a context is evaluated, when the execution of its scenario is completed. The evaluation checks the context $c_i$ for consistency with the metamodel $M_i$, i.e. there are no facts that contradict each other and all constraints over the design object model are met. The metamodel $M_i$ is transferred to $M_{i+1}$, if the evaluation succeeds. In case of failure, all results of $c_i$ are discarded and the process will restart from $M_i$ (see Fig. 8). This backtracking is performed in dialogue with the designer, so that the next attempt can be made more successful.

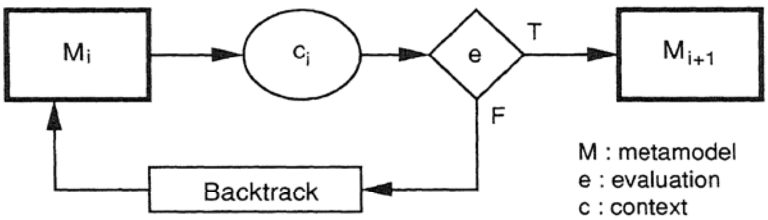



**Fig. 8. Metamodel evolution**

A consecutive application of the metamodel evolution mechanism enables the designer to perform a design job. Hence, the design process consists of the execution of a series of design scenarios. For each state of the metamodel, there exists an appropriate scenario.

## 3. Metamodel Mechanism and Ontological Knowledge

### 3.1. The necessity of ontological knowledge

The metamodel plays three roles in the design process. The first role is to act as a central model to integrate aspect models. Different aspect models can be derived from the metamodel depending on contexts. The second role is to represent a qualitative model of the design object. During conceptual design, the metamodel models the design object by its primary behavior and structure. The third role is to serve as a working space where the design object model evolves by stepwise refinement.

Among these three roles of the metamodel, the first and second roles are characterized by their need for knowledge about the physical world. First, aspect models derived from the same design object are not independent of each other. In order to maintain consistency among aspect models, their relationships must be known. The relationships among aspect models are derived from constraints of the real world. Therefore, to reason about these relationships we need knowledge about physical laws. Secondly, knowledge about available structural components and physical phenomena is indispensable to achieve conceptual design. At this stage, the design process maps a functional specification to an abstract anatomical structure. The designer breaks down the specification into behaviors of the design object, and determines structures which embody the behavior. Knowledge about structural components and physical phenomena is used by the designer to perform this mapping.

The necessity of knowledge about the physical world becomes evident when design involves more creative activities. *Creative design* contrasts with *routine design* in that it requires the designer's ability to generate new scenarios for a design solution. A new mechanism is found, and new aspect models are built. Since models of the design object must be consistent with the laws of the real world, modeling requires the designer's knowledge about physical laws.

Human knowledge for understanding and predicting physical phenomena is called *ontological knowledge* in artificial intelligence [7]. It implies that there exists a physical world independently of any particular intentional perception. For an ICAD system, knowledge about the physical world stored in the system is the system's ontology. The boundary of the ontology confines the ability to assist a designer in creative design. In other word, the intelligence of an ICAD system relies on the ontological knowledge it can use. Therefore, the more an ICAD system is intended to model, analyze, and predict physical phenomena, the richer its knowledge about physics must be. In this sense, ontological knowledge is necessary for an ICAD system to support the designer in creative design. It is obvious that one cannot describe the physical world strictly free from any intentions. Thus it is hopeless to make a thoroughly ontological model. Nevertheless, the designer has knowledge about physics in the domain of design which he can predict the behavior of the design object with. Such knowledge forms the ontology of the domain. In the remainder of this section we concentrate on the use of ontological knowledge.

### 3.2. Metamodel mechanism

The first role of the metamodel, e.g. maintaining relationships among models, is realized by the *metamodel mechanism*. In the metamodel mechanism, the metamodel is a model representing qualitative relationships among aspect models. Before we explain the metamodel mechanism, let us examine the nature of relationships among aspect models.

An aspect model assumes a set of definitions of concepts for representing properties of a design object. We call the definitions a *background theory* of the aspect model. For instance, algebraic geometry is a background theory of a solid model. It defines concepts such as vertex, edge, face, and solid object. Strength of materials is a background theory of a distorsion model, containing concepts such as stress and rigidness. An aspect model is a selective representation of the design object filtered by its background theory. Fig. 9 illustrates a design object, a background theory, and an aspect model.

An aspect model is generated by interpreting a metamodel in the context of its background theory. Thus relationships between aspect models essentially originate from relationships between their background theories. Hence knowledge for integrating aspect models can be represented as relationships among concepts defined in their background theories. In other words, knowledge about relationships among aspect models is represented on the level of background theories, rather than on the level of aspect models. There is an advantage of representing

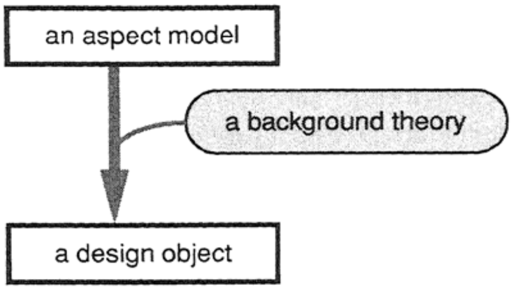



**Fig. 9. A design object, a background theory, and an aspect model**

knowledge on the level of background theory. Since the knowledge can be independent of a particular representation of the aspect model, we can use a coherent, universal definition of concepts to represent relationships among aspect models. On the other hand, if the knowledge is represented on the level of aspect models, the knowledge must be revised when representation of an aspect model is changed. Furthermore, when we add a new aspect model, we must know about internal representations of the related models in order to write relationships among them. Such drawbacks result in an inflexible knowledge representation. Therefore, by means of the metamodel mechanism, we represent knowledge for integration of aspect models on the level of background theories (see Fig. 10).

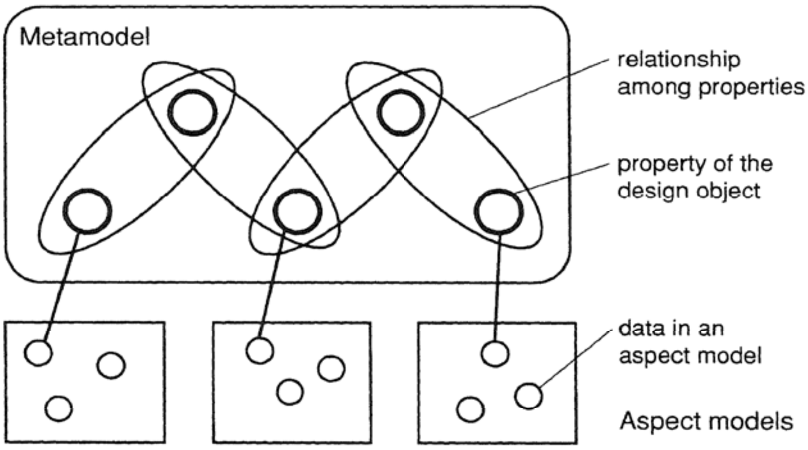



**Fig. 10. Metamodel mechanism**

The metamodel mechanism integrates aspect models as follows (see Fig. 11). Each aspect model is represented in the metamodel by entities, properties of entities, and relationships among entities. Here, an *entity* represents either an individual object such as a pin, a part of an object such as a slot, or a physical phenomenon such as the motion of a pin. A property characterizes an entity by its type, e.g. solidObject(aPin). And a relationship correlates arbitrary number of entities such as contact(pinFace, slotFace4, position1). Besides the representation of the aspect model itself, the metamodel represents the dependencies among aspect models. When an aspect model is modified, validity of relationships is evaluated by the metamodel mechanism. If an inconsistency is found, relevant aspect models are adjusted to remove it.

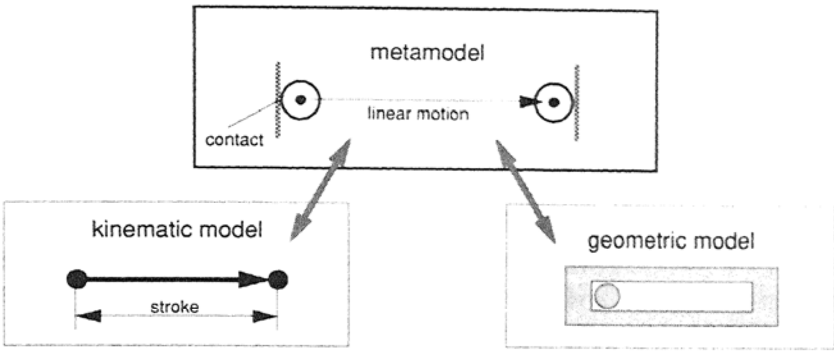



**Fig. 11. Relationship between the geometeric and kinematic models**

Let us illustrate the use of physical laws by an example on a robot. Suppose that the motion of the robot is modeled by a kinematic model. If the physical law correlating motion and force is known, forces caused by the motion can be predicted from the kinematic model. Thus the effects of the motion can be modeled in a distorsion model. If the kinematic model represents that an arm of the robot revolves round a shaft, then it can be predicted that the shaft will bend as a result of a centrifugal force. This reasoning can be performed by using a physical law about the relationship between rotation and centrifugal force.

In the example above, the essential knowledge is that rotation causes a centrifugal force. To find this fact, we rather need qualitative laws about relationships among physical phenomena than quantitatively precise equations. The knowledge about the physical world necessary for the metamodel mechanism is essentially qualitative relationships among physical phenomena. Reasoning about qualitative relationships among physical phenomena is intensively studied in the field of qualitative physics [3]. Results from this research are useful for representing ontological knowledge in the metamodel mechanism. In order to correlate aspect model, however, we must have a scheme to represent various physical phenomena uniformly. Process-oriented qualitative physics [5] can be used to represent such a uniform scheme to implement the metamodel mechanism.

### 3.3. Physical features

The second role of the metamodel is to represent the design object during the course of conceptual design. If an ICAD system has knowledge about physical phenomena of a certain domain, it can assist the designer to create new artifacts in that domain. Design Catalogue [13], for instance, attempts to assist the designer with tables of known physical phenomena categorized according to their influences and conditions. The designer looks up suitable physical phenomena in the tables. Though Design Catalogue's goal is to aid the designer during conceptual design, it is not intended to act as an automated tool on a computer system. Our goal is to assist a designer in the same way, but with the aid of an ICAD system. Another research in the same direction is the CYC enterprise [9], which is motivated by the importance of a common sense knowledge base for reasoning about entities of the real world. It is an example of equipping a computer system with an ontological knowledge base. For an ICAD system, such a knowledge base is indispensable in order to assist the designer to find the right physical phenomena.

We are building a knowledge base about engineering common sense in the field of mechanical engineering. An entry of the database is called a *physical feature*. Each physical feature represents certain physical phenomena and structural components. Examples of physical feature include a pulley and a wedge. A pulley is applied to invert an input force, producing an output force in an opposite direction. Its structural components are a wheel and a wire, and forces applied to the wire are physical phenomena. A wedge is also used to change directions of forces. In this case, a pair of angular walls and a wedge placed between them are structural components. Physical phenomena are forces applied to the wedge and the walls. The wedge receives a force, which is magnified and redirected toward the walls.

A physical feature is used by a designer to describe the qualitative behavior of a design object. A designer models behavior of the design object by selecting appropriate physical features from the database. The word *feature* is originally used in the field of CAD to designate an element of geometric model which is recognized by a designer as a unit for modeling a shape [4]. A physical feature is an extended concept of the original idea. The result of designing with physical feature is a qualitative model about behavior and structural elements of the design object.

In practice, a physical feature database is estimated to be very large, therefore collecting physical features is not an easy job. It is, however, worthwhile since knowledge about engineering is not yet represented in sufficient quantity to assist the designer to do creative design. In fact it is still unclear how we can efficiently construct a large physical feature database. Therefore we are currently concentrating on two particular domains, viz. kinematics and fundamental physical laws, to build prototypical physical features.

## 4. Conclusions

In this paper we presented an analysis of the design process, and a formalization according to the analysis. Although the design process consists of several distinguishable stages, it was possible to build a general design process model which covers all stages. It describes activities performed during the design process by means of the metamodel. We proposed the metamodel as the central model of the design object. It is a series of models of the design object which obtains data through stepwise refinement. Aspect models are derived from the metamodel in order to evaluate the design object from specific viewpoints.

There are three roles of the metamodel in an ICAD system, i.e. i) it is a central model to integrate aspect models, ii) it is a model of qualitative behavior and structure, and iii) it is a working space where models evolve. The first and second roles require a CAD system to possess ontological knowledge about the physical world. In order to achieve the first role, we proposed the metamodel mechanism. In the metamodel mechanism, the metamodel represents relationships among aspect models which are reasoned by using qualitative physical laws. The physical feature database is a knowledge base to aid the designer in the second role. We aim at supporting creative design with these tools. In [21] we describe an implementation of the metamodel mechanism.

## References


1. V. Akman, P.J.W. ten Hagen, J. Rogier, and P.J. Veerkamp, ''Knowledge Engineering in Design,'' *Knowledge-Based Systems*, vol. 1, no. 2, pp. 67-77, 1988.
2. F. Arbab and J.M. Wing, ''Geometric Reasoning: A New Paradigm for Processing Geometric Information,'' in *Design Theory for CAD, Proceedings of the IFIP W.G. 5.2 Working Conference 1985 (Tokyo)*, ed. Hiroyuki Yoshikawa, and Earnest A. Warman, pp. 145-165, North-Holland, Amsterdam, 1987.
3. H.G. Bobrow, ''Qualitative Reasoning about Physical Systems: An Introduction,'' *Artificial Intelligence*, vol. 24, no. 3, pp. 1-6, North-Holland, Amsterdam, 1984.
4. J.J. Cunningham and J.R. Dixon, ''Designing with Features: The Origin of Features,'' in *proceedings of ASME Computers in Engineering 1988*, pp. 237-243, 1988.
5. K. Forbus, ''Qualitative Process Theory,'' *Aritificial Intelligence*, vol. 24, no. 3, pp. 85-168, North-Holland, Amsterdam, 1984.
6. J.S. Gero, *Expert Systems in Computer-Aided Design, Proceedings of the IFIP W.G. 5.2 Working Conference 1987 (Sydney)*, North-Holland, Amsterdam, 1987.
7. P. Hayes, ''The Second Naive Physics Manifesto,'' in *Formal Theories of the Commonsence World*, ed. J.R. Hobbs, and R.C. Moore, pp. 1-36, Ablex Publishing, Norwood, NJ, 1985.
8. V. Hubka, *Principles of Engineering Design*, Butterworth & Co Ltd, 1982.
9. D.B. Lenat and R.V. Guha, *Building Large Knowledge-Based Systems*, Addison-Wesley, Reading, MA, 1989.
10. J. Mostow, ''Toward Better Models of the Design Process,'' *AI Magazine*, vol. 6, no. 1, pp. 44-57, 1985.
11. J. Rogier, P.J. Veerkamp, and P.J.W. ten Hagen, ''An Environment for Knowledge Representation for Design,'' in *Proceedings of Civil Engineering Expert Systems*, Madrid, 1989.
12. J. Rogier, ''A Component Class for Design Objects,'' in *Intelligent CAD Systems III – Practical Experience and Evaluation*, ed. P.J.W. ten Hagen, and P.J. Veerkamp, Springer-Verlag, Berlin, 1990.
13. K. Roth, *WDK 13, Proceedings of the 1987 International Conference on Engineering Design, (ICED 87)*, 1987.
14. H. Suzuki, H. Ando, and F. Kimura, ''Synthesizing Product Shapes with Geometric Design Constraints and Reasoning,'' in *Intelligent CAD, II*, ed. H. Yoshikawa, and T. Holden, North-Holland, Amsterdam, 1990. Forthcoming
15. P.J.W. ten Hagen and Tetsuo Tomiyama, *Intelligent CAD Systems I: Theoretical and Methodological Aspects*, Springer-Verlag, Berlin, 1987.
16. T. Tomiyama and H. Yoshikawa, ''Extended General Design Theory,'' *Proceedings of the IFIP WG 5.2 Working Conference on Design Theory for CAD*, pp. 95-125, North-Holland, Amsterdam, 1987.

17. T. Tomiyama and P.J.W. ten Hagen, ''The Concept of Intelligent Integrated Interactive CAD Systems,'' CWI-Report CS-R8717, 1987.

18. T. Tomiyama, T. Kiriyama, H. Takeda, D. Xue, and H. Yoshikawa, ''Metamodel: A Key to Intelligent CAD Systems,'' *Research in Engineering Design*, vol. 1, no. 1, pp. 19-34, 1989.

19. P.J. Veerkamp, ''Multiple Worlds in an Intelligent CAD System,'' in *Intelligent CAD, I*, ed. H. Yoshikawa, and D. Gossard, pp. 77-88, North-Holland, Amsterdam, 1989.

20. B. Veth, ''An Integrated Data Description Language for Coding Design Knowledge,'' in *Intelligent CAD Systems I – Theoretical and Methodological Aspects*, ed. P.J.W. ten Hagen, and T. Tomiyama, pp. 295-313, Springer-Verlag, Berlin, 1987.

21. D. Xue, T. Kiriyama, P.J. Veerkamp, and T. Tomiyama, ''Representation and Implementation of Design Knowledge for Intelligent CAD – Implementational Aspects,'' *in this volume*, Compiegne, 1990.

22. H. Yoshikawa and D.C. Gossard, *Intelligent CAD, I, Proceedings of the First IFIP Working Group 5.2 Workshop on Intelligent CAD*, North-Holland, Amsterdam, 1989.