

# FEATURES IN KNOWLEDGE INTENSIVE CAD: ROLES & TYPES

David C. Brown

*AI in Design Group, Computer Science Department, WPI, Worcester, MA 01609, USA.*

**Abstract:** In this paper we examine the roles that features can play in KIC systems. We also examine the definition of the term “feature”, and attempt to shed some light on the reason for the profusion of types of features that have been discussed in the literature. We revisit the general definition of the term, and attempt a definition that uses concepts from Artificial Intelligence.

**Key words:** feature, function, structure, behavior.

## 1. INTRODUCTION

This paper first reports on a discussion between members of the Features working group at KIC-5, the 5th IFIP Workshop on Knowledge Intensive CAD, and then presents an analysis of how features can be defined in a comprehensive and all-inclusive manner.

## 2. THE ROLES OF FEATURES IN KNOWLEDGE INTENSIVE CAD

In general, features can be said to provide “implicit guidance”. This is because their availability provides a set of concepts and a vocabulary that can affect the designer’s or design system’s view of all aspects of the product’s life cycle. Features are an additional type of knowledge that can be used when designing. They can affect designs, they can affect designing and they can affect reasoning. If defined well, and comprehensively, they can provide an ontology, although this would require more completeness, detail

and information (about relationships between features, for example) than is normally provided.

## 2.1 Constraining effects

Features can provide constraining effects on the designer, as they can:

- a) Standardise designs;
- b) Constrain the design space;
- c) Move a design problem towards a configuration problem.

By providing an agent (designer, or design system) with features this ensures that every design produced will have some or all of those features in common. If the set of features is comprehensive enough then the designs might be dominated by those features and can be said to be “standardised”.

The set of features reduces the need for the agent to produce those ingredients of the design, and prevents those areas of the design space that do not include one or more features from ever being examined. Hence the design space is constrained, reducing the amount of reasoning (i.e., search) required.

If the features available are predominantly geometric or structural then the resulting design might be thought of as a configuration of those features. Hence appropriate features can change a design problem to become more like a configuration problem.

## 2.2 Relaxation effects

Features can provide relaxation effects on the designer, as they can:

- a) Stimulate creativity in a novice;
- b) Encourage design space exploration if they are parametric features.

One of the characteristics that differentiates a novice from an expert is that a novice does a lot more searching through the possibilities, because they have no reason to prefer one over another, while an expert, because of knowledge compiled from experience, can quickly prune inappropriate and unproductive areas of the search space. The novice, faced with so much choice, tends to be content to find a solution at all. If features are available, the novice can avoid some of the unproductive searching, and can focus on using the features provided, allowing them to feel more relaxed and confident (as the features are in some sense “pre-approved”), trying combinations that they might not have tried otherwise. Hence features can stimulate creativity in a novice designer.

For more expert designers, Parametric features can lead to design space exploration, as by varying parameters, many more detailed designs can be examined. The ability to easily vary designs within prescribed limits allows the designer to relax their tendency to fixate on known solutions.

### **2.3 Correctness effects**

Features can provide correctness effects on the designer, as they can:

- a) Contain knowledge about what is ‘possible’ in lifecycle processes;
- b) Enable detection of problems and inconsistencies;
- c) Interact with KIC systems to help with feature mapping.

Different types of features, such as design or manufacturing, have associated knowledge (sometime explicit, sometimes not) about different lifecycle processes. As mentioned above, features are in some sense “pre-approved”, providing a potential assurance of correctness, or at least a reduced possibility of problems.

This associated knowledge helps direct the designer when looking for potential manufacturing (or other) problems. Feature interactions can be explicitly sought, which focuses the hunt for problems and inconsistencies.

The knowledge associated with features can be exploited by KIC systems during feature mapping (e.g., functional features mapping to manufacturing features, or strength features mapping to manufacturing features).

### **2.4 Description effects**

Features can provide description effects on the designer, as they can:

- a) Standardise documentation;
- b) Shorten design description;
- c) Allow description with abstractions;
- d) Act as anchor points for Design Rationale.

Just in the same way that use of features can standardise designs, they can standardize documentation of the design, and even of the design process. In addition, use of standard ingredients has the possibility of shortening the design description, as the details of the features can be “common knowledge” and in some cases do not need to be included in the design description.

Feature families allow description with abstractions. Thus referring to a more abstract feature could convey a family of designs.

And finally, features act as anchor points for Design Rationale. Features are defined because they have some utility, and their rationale should be well understood. Hence the rationale for a feature can be inherited into the rationale for the design itself.

### 3. PUBLISHED DEFINITIONS

Unfortunately there have been many published definitions of the term “feature” [Salomons 1995; Shah & Mantyla 1995]. This has produced ambiguity and has allowed a confusing number of different types of features, including:

- Form Features: related to the geometric form.
- Precision Features: concerning deviations from nominal dimensions.
- Technological Features: related to performance and operation.
- Material Features: material composition, treatment, conditions.
- Assembly Features: concerning the assembly process.
- Pattern Features: patterns of similar entities.
- Connection Features: geometric constraints.
- Property Features: properties not explicitly related to geometry.
- Application Features: related to process planning requirements.
- Manufacturing features: related to particular processes, such as machining.

Faced with such variety, in order to provide a generic definition, writers have gradually been moving towards very general definitions that are in fact only minor variations of “*anything about the thing being designed that’s of interest*” [Dixon 1991].

For example, Vandenbrande & Requicha [1993] define features as regions of an object that are meaningful for a specific activity or application. Shah [1992; 1995] stresses that features represent the engineering meaning of the geometry of a part or assembly.

Note that these definitions, along with most definitions of the types mentioned above, relate features back to “regions” or to geometry. That is, they relate to physical structure, regardless of the fact that the *reason* for declaring something a feature varies depending on the type identified (e.g., Assembly, or Connection).

#### 4. FUNCTIONAL FEATURES

In addition to the above types, Functional features have also been identified. These concern function, purpose or behaviors. Functional features have been increasingly mentioned in the literature, as CAD has moved away from being purely geometric. McGinnis & Ullman [1992] write that:

“Functional features include both the purpose of the design object such as support, stability, or strength and the behavior that the design object performs like lifting, gripping, or rotating. The form features embody the physical characteristics of design objects in a design while the functional features explain what purpose the design objects achieve individually and what behaviors they exhibit in the overall design.”

Interestingly, this suggests that while a functional feature will have implications for structure/form it is not necessarily about the form.

#### 5. DESCRIBING “ANYTHING”

Given the generic definition of a feature as “anything of interest”, we first discuss how to describe “anything”. Later we consider how to describe “of interest”.

The term “anything” refers to all or any realisable portion of what is being designed. The design is represented by a description, which traditionally has been geometric. However, a richer and more complete description of an object can be made using a three level **Structure-Behavior-Function** (SBF) representation, such as that introduced by Sembugamoorthy & Chandrasekaran [1986].

In that early work, the structure level describes the components and their connections, the behavior level describes behaviors in terms state changes of the structure, and the function level names behaviors that have external utility. This has been refined in a variety of ways in subsequent research [Umeda & Tomiyama 1997; Chandrasekaran 1994].

So, given an SBF representation, the term “anything” can refer to all or any realizable portion of an SBF description.

During designing, the SBF description may be *incomplete*, or *abstract*. SBF representations allow for layers of abstraction at the structural and behavioral levels, and for designs to consist of a configuration of non-decomposable elements. Both structures and behaviors can be configurations.

This is completely consistent with what we see during designing. Design decisions can be very abstract (e.g., “it will be pump-like”) or very concrete

(e.g., “3.25 inches in diameter”), depending on the design phase. Design decisions can be made about structure:

e.g., shape, components, configuration, material, surface finish, etc.;

about behavior:

e.g., how many states, state-state changes, time taken, flow rates, etc.;

or about function:

e.g., what useful behaviors will be available in what environments under what situations.

There are now many variations of the SBF representation [Umeda & Tomiyama 1997; Chandrasekaran 1994]. In most functional reasoning research, the description of structure is not the main focus of the work, and so the representation normally ‘bottoms out’ at an abstract description of the types of components and the way they are connected, with no geometric detail. However, nothing would prevent this level of detail from being included.

The structural description,  $S$ , is considered to include what’s needed to describe the physical design, including geometry, spatial relationships, components, and properties. Structural properties,  $P_S$ , include material, surface area, surface finish, color, mass, etc.

Refinements and extensions to SBF that can be found in the literature include:

- a goal state or control relation that the function is intended to achieve;
  - a listing of the properties of the designed object that enable the function;
  - a listing of the properties of the environment that enable the function (including sequences of inputs, or changes to the environment’s state);
  - the relationships between the properties that enable the function;
- the purpose of the function, described in terms of the environment (i.e., not just the local behavior involved, but its larger impact—perhaps its design rationale [Lee 1997]).

## 6. FUNCTION

Function is best thought of in terms of the way that the designed object interacts with an environment when placed in it. Chandrasekaran & Josephson [1996] define a function of an object in terms of the effect it has on its environment. The environment can use behaviors, structure or structural properties, and can provide ‘inputs’ or stimuli that trigger the function to enable, prevent or maintain the interaction [Keuneke 1991].

For example, a clock functions when placed in an environment that provides it with power and allows the time display to be visible. Note that in

this example a subset of the object's behaviors is being used for a function: the 'ticking' sound is normally not used.

A knife functions when placed in an environment which applies a force on the knife causing a particular portion of the structure (the 'sharp' edge) to be in contact with another material.

In this last case, the function is provided by a very specific piece of the knife's geometry, and the material properties that prevent it from bending, breaking or changing its shape. In the clock case, the function is provided by a combination of behaviors and the structure that supports them.

In these examples above we are considering the "intended function" of the designed object: i.e., the design intent. This intended function is concerned with the process referred to as "intended use". Designed objects can be used in other ways (i.e., analogical use) by taking advantage of other substructures, subsets of behaviors, or properties (e.g., using a shoe as a hammer, or using a ticking clock as a baby pacifier).

## 7. PROCESSES AND GOALS

Returning to the clarification of the term "anything of interest", the definitions above from Vandenbrande & Requicha [1993] and Shah [1992; 1995] imply that something is of interest if it is "meaningful for a specific activity" and that this activity must contribute to "engineering meaning".

Hence any activity that engineers should be concerned with while designing can qualify as such an "activity".

Shah [1995] appears to confirm this Concurrent Engineering view when he writes that:

"The *engineering significance* may involve formalizing the function that the feature serves, or how it can be produced, or what actions must be taken when performing engineering analysis or evaluation, or how the feature 'behaves' in various situations."

Rosen et al. [1991] define a feature as "a model of the form and intent of some aspect of a design which is of direct interest in a CIM viewpoint". Their "CIM viewpoint" includes Design, Assembly, Manufacturing, Marketing, Analysis, and Service: i.e., any activity with which engineers should be concerned.

Consider all the **processes**,  $P = \{p_1, p_2, \dots, p_n\}$ , in which the designed object can be involved: e.g., intended use, being described, assembling, packaging, diagnosis, simulation, recycling, manufacturing, designing, maintenance, etc.

In order to be “meaningful”, and therefore of “interest”, we need to consider why someone might be paying attention to one of these processes while designing. The answer is that the designer  $D$  has **goals**,  $G_D = \{g_{D1}, g_{D2}, \dots, g_{Dm}\}$ , and that she or he is trying to satisfy these goals.

A goal,  $g_{Di}$ , could be one of: reduce complexity; reduce resources used; reduce cost; increase reliability; provide ease of use; satisfy requirements; allow access; etc.

The *context* for each goal,  $g_{Di}$ , is a process,  $p_j$ . Each process has a set of compatible goals: i.e., “meaningful for a specific activity”. For example: one can simplify packing, or simplify describing the design; one can reduce the cost of manufacturing, or of designing; one can increase reliability during intended use, or during assembly; one can allow access for maintenance, or for recycling. Calculations, simulations, estimates, measurements or heuristic hypotheses may be made to determine if a goal is affected or met.

## 8. DEFINING “INTEREST”

A designer,  $D$ , has at least one **point of view** (POV). A point of view,  $POV_{Dij}$ , is described by a *goal/process pair*,  $(g_{Di}, p_j)$ .

Interest is relative to a POV. That is, someone has an **interest** if a design decision or set of decisions might affect the POV’s goal in the context of its associated process. The effect might be positive or negative.

For example:

- choice of a very hard material might increase the cost of the manufacturing process;
- choice of a large fan as a component might reduce “ease of use” during the intended use, due to noise and moving air;
- choice of acute angles at edges might reduce “ease of use” during the intended use, due to the possibility of a person being injured;
- choice of acute angles at edges might reduce “ease of use” during packaging, due to the possibility of packaging material being damaged;
- choice of friction instead of electromagnetic force as a form of resistance might increase the ability to use the component with electrical devices;
- choice of lack of symmetry might increase handling time; and
- choice of snap fits might decrease assembly time.

What is of interest to a person depends on the design stage and may be determined by their job/role, by their experiences, or by the information obtained by computer simulation, mental simulation, or by physical modeling of the design.

In most design scenarios, the designer needs to adopt a variety of points of view in order to consider life-cycle issues. Their interest changes as they

scan through the stages of the life cycle and, as a consequence, different processes become relevant. *Consequently, the relevant features and feature types vary dynamically over time.*

## 9. DEFINING A FEATURE

The generic “anything of interest” definition above can be modified now that we know more about both “anything” and “of interest”. A portion of an SBF description represents “anything”, while “interest” concerns a POV.

*Definition:*

A **feature** is described by any portion of an SBF description of the (partially) designed object that may affect the goal of a particular POV.

Note that this clearly means that a feature is dependent on at least one process. Rosen et al. [1991] argue, similarly, that a feature contains “relevancy”, where “relevancy identifies which CIM viewpoint a feature is defined in”.

This definition is consistent with the important notion that a feature is a “view” of the design and that features cannot be listed or “limited a priori” [Shah 1995].

Given this definition, a feature can be described by the S, B or F portions of the description. It corresponds to something in the object being designed, but it *isn't always directly about structure.*

Hence, for example, a feature can be:

- a structural property: color, size, finish, mass, reflectance, etc.;
- a portion of the surface;
- any portion of the components of the designed object;
- a configuration;
- a set of relationships;
- a behavior, or behaviors; or
- a function, or sub-function.

It may even be a property of a behavior, or of a function. In addition, as SBF is a layered representation, features can be at any level of detail.

## 10. NON-GEOMETRIC FEATURES

Many features that fit the definition above refer to “a region of an object”, but there are a variety of other cases:

- Color can be a feature, as it may of interest because it enables the designed object to function during intended use. Color is an element of the Structural properties,  $P_S$ .
- Mass can be a feature, as it may of interest because it can affect goals in a number of processes, including handling, packaging, and intended use. Mass is an element of the Structural properties,  $P_S$ .
- A clock's ticking can be a feature, as it may of interest because it enables the easy diagnosis of a broken clock mechanism or a lack of power. The "ticking" is a behavior, but not one that is part of the intended function.
- A clock's hands moving can be a feature, because it enables the designed object to function during intended use. In this case, the 'moving' behavior is part of the intended function.

None of these cases refer to a region of the object, and the features aren't geometric. Of course, as function depends on behavior and/or structure, and behavior also depends on structure, these can be traced back to structure, or to structural properties. But reference back to a "region" isn't always possible. Color for example occupies a region, but is not itself a region or a "shape" [Shah 1995]—it's just a different 'type'.

## 11. ABSTRACT FEATURES

During early stages of the design process, decisions can be made that will cause some impact on the goal of a POV, and can be seen as doing so (i.e., a feature can be detected). However, at that time only an abstract description of the structure is possible. To insist that features always refer to a region of the object cannot be correct if we take this literally, as early in the design process (especially in non-routine situations) it is possible for no "form" to exist, and for no description of structure to have been committed to.

Shah [1991] recognizes this situation and defines an "abstract feature" as:

"Entities that cannot be evaluated or physically realized until all variables have been specified or derived from the model."

This should not be confused with Shah's definition of abstract features as higher level classes in a feature taxonomy [1995], even though in some cases the two may be equivalent. One key difference is that the definition we have developed above has the advantage that it allows abstract features to be of types other than structural.

## 12. FUNCTIONAL FEATURES, REVISITED

Given our new definition of a feature we have a choice of what a functional feature might be. It could be a feature that is of type function (i.e., referring to the F portion of the SBF description). It could also be a feature of any type that impacts intended use: e.g., a behavior that prevents the design from functioning. However, this seems less satisfactory.

In an interesting way, *every* feature is functional. By definition a feature is of interest because it has an effect on a process, and is detected by its effect on a goal. That process is forming an “environment” for that feature, and is interacting with it. This is consistent with the definition of a function of an object as the effect it has on its environment: i.e., by definition, a feature of the designed object is functioning in the environment of each process where that feature affects a goal.

This is not the “intended function” of the designed object, except in some special cases. However, a good designer who is viewing the object from a number of points of view corresponding to the phases of the life-cycle, will ensure that the designed object functions as intended when involved with all the corresponding processes. For example, it should be easy to pack, easy to maintain, and easy to disassemble. Paying attention to a variety of dynamically detected and selected features during design makes this possible: i.e., it will function well for those processes.

## 13. CONCLUSIONS

In this paper we have examined the various roles of features in KIC systems, as well as the definition of the term “feature”, and “functional feature” in particular. The goals of the paper are to shed some light on the reason for the profusion of types of features that have been discussed in the literature, and to revisit the general definition of the term using concepts from Artificial Intelligence.

By separating structure, behavior and function, and by defining function, a variety of types of feature can easily be defined. The distinctions introduced will allow the development of more flexible use of features in knowledge-based design systems and in designer support systems.

We feel that it is inevitable that knowledge and intelligent use of knowledge will continue to become an increasingly useful component of CAD tools. These tools will know what a designed object is for, and how it works. Computerised simulations of behavior will be augmented by a deeper understanding of how the object interacts with its environment, so that whether it provides its intended function can be checked at design time.

Features, as deliverers of functionality, will play a significant role in this process.

## ACKNOWLEDGEMENT

The author would like to thank the other members of the Features Working Group at the IFIP KIC-5 Workshop: Mike Wozny, Xiu-Tian Yan, Harald Otto, Michiko Matsuda, and Udo Kannengiesser.

## REFERENCES

- B. Chandrasekaran, "Functional Representation and Causal Processes", *Advances in Computers*, Vol. 38, Academic Press, 1994, pp. 73-143.
- B. Chandrasekaran and J. R. Josephson, "Representing Function as Effect: Assigning Functions to Objects in Context and out", *Proc. AAAI-96 Workshop on Modelling and Reasoning about Function*, Portland, OR, August 1996.
- J. R. Dixon, personal communication, 1991.
- A.M. Keuneke, "Device Representation: The Significance of Functional Knowledge", *IEEE Expert*, Vol. 6, No. 2, Apr. 1991, pp. 22-25.
- J. Lee, "Design Rationale Systems: Understanding the Issues", *IEEE Expert*, Vol. 12, No. 3, 1997, pp. 78-85.
- B. D. McGinnis & D. G. Ullman, "The Evolution of Commitments in the Design of a Component", *Journal of Mechanical Design*, March 1992, Vol. 114, pp. 1-7.   
<<http://www.engr.orst.edu/~ullman/evolut1.htm>>
- D. W. Rosen, J. R. Dixon & X. Dong, "A Methodology for Conversions of Feature-Based Representations", *Design Theory and Methodology*, DE-Vol. 31, ASME, 1991, pp. 45-51.
- O. W. Salomons, *Computer Support in the Design of Mechanical Products*, Ph.D. Thesis, Department of Mechanical Engineering, Universiteit Twente, 1995.   
<<http://www.pt.wb.utwente.nl/staff/otto/thesis/>>
- V. Sembugamoorthy & B. Chandrasekaran, "Functional Representation of Devices and Compilation of Diagnostic Problem Solving Systems". In: *Experience, Memory, and Reasoning*, (Eds.) J. L. Kolodner & C. K. Riesbeck, Lawrence Erlbaum Associates, Publishers, 1986, pp. 47-73.
- J. J. Shah, "Conceptual development of form features and feature modellers", *Research in Engineering Design*, Vol. 2, 1991, 93-108.
- J. J. Shah, "Features in Design and Manufacturing", In: *Intelligent Design and Manufacturing*, A. Kusiak (Ed.), John Wiley & Sons, 1992, pp. 39-72.
- J. J. Shah & M. Mantyla, *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*. J. Wiley & Sons, Inc., 1995.
- Y. Umeda & T. Tomiyama, "Functional Reasoning in Design", *IEEE Expert, special issue on AI in Design*, (Eds.) Brown & Birmingham, Vol. 12, No. 2, March/April 1997.
- J. H. Vandenbrande & A. A. G. Requicha, "Spatial reasoning for the automatic recognition of machinable features in solid models", *IEEE Pattern Analysis and Machine Intelligence*, Vol. 15, No. 12, pp. 1269-1285, December 1993.   
<<http://www-lmr.usc.edu/~jan/Papers/FRec1/NewFRecWeb.html>>