## References

[1] S. Finger and J. R. Dixon, A Review of Research in Mechanical Engineering Design, Part II, *Research in Engineering Design*, 1(2):121–137, 1989.

[2] E. L. Gursoz and F. B. Prinz, *Corner-Based Representation of Non-manifold Surface Boundaries in Geometric Modeling*, Technical Report, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA, 1989.

[3] J. M Pinilla, S. Finger and F. B. Prinz, Shape Feature Description and Recognition Using an Augmented Topology Graph Grammar, In *NSF Engineering Design Research Conference*, pages 285–300, University of Massachusetts, Amherst MA, June 11-14, 1989.

[4] S. A. Safier and S. Finger, Parsing Features in Solid Geometric Models, Submitted to *9th European Conference on Artificial Intelligence*, Stockholm Sweden, August 1990.

[5] K. J. Weiler, Topological Structures for Geometric Modeling, Ph.D. Thesis, Rensselaer Polytec Institute, 1986.

while a manufacturer might define a hole by its location, radius, and manufacturing process (e.g. a punched hole). Both the designer and manufacturer use the label *hole*. While the features labeled as holes are similar, they are not identical. The difference of perspective for characterizing the concept *hole* necessitates differing feature definitions.

## 4 Feature Compilers and Parsers

Pinilla [3] has created a graph grammar based on the non-manifold modeller of Gursoz and Prinz. The grammar's domain is the graphs representing an design's topology augmented with geometric information. This graph is called the augmented topology graph (ATG). Both non-manifold and manifold objects can be represented with the ATG. The ability to represent both manifold and non-manifold objects is essential in describing partial designs or referring to *non-existent* elements such as center lines or symmetry planes.

We define grammars that generate families of features based on the ATG; for example, we can generate instances of through holes using a grammar. We use this grammar in two ways. One is to provide the designer with feature abstractions and feature instances with which to compose the design. This work is preliminary and many issues remain open.

The other way in which we use grammars is to recognize features in the design. The feature extractor finds the complete set of feature instances derivable from the productions of an augmented topology graph grammar given the grammar and an augmented topology graph representing the geometry of a design. Because both the geometric model and the feature definitions are represented as graphs, the problem of feature extraction is the problem of finding isomorphic subgraphs, an NP-complete problem.

We propose three levels of abstraction for recognizing features. At the lowest level is a non-manifold solid modeller. This level provides complete information for representing a design, including all topological and geometric data about the model. The intermediate level of abstraction is augmented topology graph of Pinilla. This level captures the geometric relationships from the input grammar and maintains the non-manifold representation. The most abstract level represents manifold features and any manifold portion of non-manifold features such as geometric and topological relations between faces. Any non-manifold portion of these feature is represented at the intermediate level. By providing multiple levels of abstraction, we reduce the search space and concentrate the search on the areas most likely to match particular features.

## 5 Conclusion

We have presented a method for creating higher-level entities by which geometric information can be organized and manipulated; however, we are left with the problem of creating the physical interface by which the designer composes a design from these higher-level entities. We still must find a natural and intuitive way to create, manipulate, and edit three-dimensional designs using a mouse and keyboard.

requirements for the representation of designs: one is the need for higher-level abstractions of geometry and behavior with which designs can be composed; the other is the need for a representation that supports the many different points of view from which a design can be interpreted during its life-cycle. Perspectives support the mechanical design process by providing analysis and synthesis tools to the design effort. They are integrated around a shared, domain-neutral representation of the design in which each tool in the environment can extract and reason about elements of the design.

Our approach is to describe features using a graph grammar. Because the designed object is an element in the language generated by this grammar, features can be recognized by parsing the feature against the graph of the object. We provide a representational link between the low-level geometric representation and the high-level design abstractions by formalizing a language to express classes of high-level objects in terms of the low-level ones. Given this language, we are then able to extract and label high-level elements from the neutral low-level geometric representation.

## 2 Geometric Representation

The standard geometric modellers used in CAD systems were created to represent solid objects in $R^3$ space. These models are not able to represent incomplete objects. The non-manifold geometric modeling systems created by Weiler [5] and by Gursoz and Prinz [2] address this issue. These representations build upon the boundary representations, but they are able to represent the more complex adjacency patterns such as dangling edges or nested cones that can occur in non-manifold objects. Because one-, two-, and three-dimensional objects can be represented consistently in non-manifold representations, they are highly suited to design systems. Using a non-manifold representation, the design can include a center line of a hole, a parting plane for a mold, and internal boundaries for a finite-element mesh, as well as the enclosing shell of the designed object.

## 3 Features in Mechanical Design

An area of active investigation in mechanical design is the use of features derivable from the geometry, that is, *form features* [1]. Two approaches to using features in design have been explored. One is to restrict the designer to a set of labeled, primitive features to use in the design. A restricted set of features limits the expressiveness of the design system and allows only one interpretation of the design. This approach has the advantage that feature extraction is relatively straightforward because the features on the design can be tracked as they are added.

The other approach is to use a neutral geometric representation of the design and to extract the locations of the features when required. We have taken the second approach because in addition to providing abstractions to the designer, we must support multiple perspectives that each require a different grouping of primitive elements into features. When experts view an object, they perceive it in terms of their own expertise. For example, manufacturers see features that affect the processes used to fabricate a part, while structural engineers see sources of stresses and other features that tend to reduce the life of a part. Features can be geometric, such as slots or chamfers; they can be quantitative, such as distances between holes; they can be functional, such as alignment; or they can be qualitative, such as a rough, but unspecified, surface finish.

For example, several different features might all be labeled *hole*. From a functional point of view, a designer might specify a hole only by it’s centerline, radius, and purpose (e.g. alignment)

Visual Interfaces to Geometry
CHI '90

Seattle, April 1-2 1990

# Feature-Based Representations for Mechanical Designs

Susan Finger, Yung-Cheng Chao, Christopher Young, and Scott Safier

**Abstract**

As part of a research project on concurrent engineering, we are creating a workstation for mechanical designers. The goal is to create a system in which knowledge of downstream activities is infused into the design process so that designs can be generated rapidly and correctly. The design space can be viewed as a multi-dimensional space in which each dimension is a different life-cycle objective such as fabrication, testing, serviceability, reliability, *etc*. These dimensions are called *perspectives* because each dimension can be thought of as a different way of looking at the design. At any stage, a design can be viewed from the design perspective, the manufacturing perspective, from the reliability perspective, from the serviceability perspective, *etc*.

To date, most of the research on this project has focused on the issues of design representation, control, and analysis. However, to create a system that would be usable by a designer, we have begun to address the issues of the designer interface. Among the issues of interest to us are enabling the designer to:

- compose designs using high-level features as well as geometric primitives
- edit, modify, and retrieve designs by attributes
- configure the design environment for the current task
- control the flow of comments and suggestions from the down-stream perspectives to the designer

This short paper discusses the first two issues relating to the geometric interface. We present an approach for defining high-level geometric entities using grammars. These high-level entities, called features, not only give the designer more powerful primitives with which to compose designs, they also provide a means for organizing, re-interpreting, and displaying information about the geometry of designs.

## 1 Introduction

Our research in feature-based design systems for mechanical designers is motivated by the realization that geometric models represent the design in greater detail than can be utilized by designers, process planners, assembly planners, or by the rule-based systems that emulate these activities.

In a large multi-disciplinary project, we are creating a design system in which a mechanical designer can generate an initial design by combining high-level features that are linked to topological, geometrical, and behavioral attributes. This system is based on two beliefs about the