Virtual Prototyping through Feature Processing

J. Ovtcharova, A. S. Vieira

Fraunhofer–Institut für Graphische Datenverarbeitung (IGD)
Wilhelminenstraße 7, 64283 Darmstadt, Germany
Telephone: +49 6151 155 207/246
Telefax: +49 6151 155 299
e–mail: jivka/vieira@igd.fhg.de

Abstract

For the integration of different product development tasks to be supported within a virtual prototyping process it is necessary to take account of semantic information, alongside the geometric shape information. The need of semantically endowed primitives leads to the use of features. In this paper we focus on one crucial important aspect in which virtual prototyping systems should evolve: the *feature–based design of product prototypes*. This will offer possibilities to users to define *high–level semantic information* in the prototype model in a way which allows a high degree of correspondence between virtual prototype and physical product. Moreover, features should be used as *processing entities* for design, as well as for the information integration of design with different downstream phases of virtual prototyping such as analysis, manufacturing simulation, cost and quality estimation.

Key Words

Virtual prototyping, semantic product modeling, features, feature processing, feature–based design module

J. Rix et al. (eds.), *Virtual Prototyping*

"Virtual presence is the synthesis of a perceptual consistency that convinces users that either they are immersed in another space or that computer generated artifacts exist in the physical space (illusion)".

W. Chapin, T. Lacey, L. Leifer (Chapin, 1994)

1 INTRODUCTION

The virtual prototyping world gives us an unusual opportunity to compress time and to reduce production costs. The traditional limitations of the physical prototyping such as weight, measures and material treatment are avoided. The ability to use computer simulation implies that we have appropriate models of product prototypes. Appropriation means not only models which describe the geometry, but describe the complete product semantics in terms which designers understand and which reflect the design intent to downstream applications. As a consequence, the illusion of a prototype presence in a virtual space requires a comprehend and semantic correct computer–based model which fully complements a prototype existing in the physical space.

Some essential requirements to the computer–based representation of virtual prototypes are as follows:

A/ *Computer–Aided Engineering (CAE) system point of view:* The development of virtual prototypes requires an engineering environment based on an integrated product development process and an integrated product data model. One of the central components of a CAE system for supporting virtual prototyping is a suitable design module which enable the development of prototypes corresponding to the customer needs and which can be developed easy, fast and cheap.

B/ *Design processing point of view:* First, supporting the intuitive and creative work of the designer in different design phases, such as conceptual, embodiment and detail design is one of the most important requirements for early and fast prototyping. The models which are generated in the design process have to be processed in downstream phases of the prototype development, such as analysis, manufacturing simulation, cost and quality estimation. Second, using data at different levels of abstraction is needed to build comprehensive semantic models (Ovtcharova, 1992). Third, a mechanism to check the consistency of the semantic model data based on feature constraints is essential (Vieira, 1994). Fourth, facilities for applying these constraint concepts for easy and fast redesign based on parametrics and variational design methods and tools must be provided. Fifth, easy and fast configuration by using pre–defined and user–defined modeling entities has to be supported.

C/ *Graphical presentation point of view:* New modeling and presentation techniques, such as virtual reality and graphical simulation are of great importance.

In this paper, we focus our attention only on the second topic, the design processing for virtual prototyping. Furthermore, to satisfy the requirements discussed above, we concentrate our research on the design processing using the feature paradigm.

Features have been identified as meaningful engineering aids for modeling products using high–level semantic data. Thus, a principal impetus for feature research was the observation that the successful computerization of product development needs augmented models that are capable of representing both geometric and non–geometric information. The main advantages of features include: 1) a vocabulary which is more natural for expressing the product prototype than are pure geometric models, 2) the possibility for using features as a basis for modeling product information in different phases of prototyping such as design, analysis, process planning and manufacturing, and 3) an increase in the designer's productivity and cost effectiveness. A vast amount of research activities has recently been focused particularly on the problems involved when integrating previously separate prototyping phases into one system that covers all major aspects of prototyping. Feature modeling is now regarded as a key technology in achieving high level of efficiency, automation and information integration in this broad area.

2 THE NOTION OF A FEATURE

The feature definition as adapted in our work deals with the assignment of features into feature types, as presented in (Ovtcharova, 1994). Here, we distinguish between *generic* and *application* features. Generic features, such as form, material, and precision features are used to describe properties of products in a general way. For example, *form features* are defined as *shape characteristics* of product parts which conform to some *shape configuration* of the product, like depressions (negative volume form features) and protrusions (positive volume form features), but have no *specific functional meaning*. Application features, such as design features and machining features, can be defined by using application–specific data over the definition of form features in the concrete application area. Such a feature object hierarchy is quite useful in the following sense: first, grouping features into classes with respect to given properties leads to development of *unified mechanisms* for modeling different products possessing these properties. Second, feature classification leads to a *common terminology* and development of product data exchange formats.

Design features are the set of application features used by the designer to model the a concrete product prototype. They encapsulate information about the semantic (or functional) meaning as well as the shape of a part which can facilitate communication of the design intent to downstream product prototyping phases.

Pre–Defined Design Feature 'Screw Hole'	
Design Feature:	Screw Hole
Form Feature:	Passage
Cross Section:	Circular
Thread:	fine
Tolerance:	H12
Rated Size:	M8
Material:	default
Feature Constraints:	d = local high of the part $r = f(\textit{engineering requirement}) \pm \varepsilon_{thin.}$
Feature Methods:	CreateInstanceObject, ModifyInstanceObject, ValidateInstanceObject, DeleteInstanceObject

Passage with Circular Cross Section	
Form Feature:	Passage
Cross Section:	Circular
Form Feature Elements:	Side Face
Location Parameters:	location point (c), normal direction (n), reference direction (x)
Dimensional Parameters:	radius (r), depth (d)

Solid Representation	
Solid Primitive:	'cylinder'

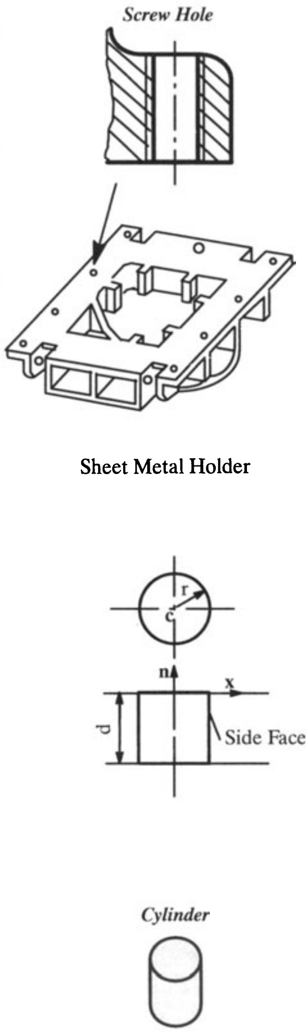

Figure 1 An example of a screw hole definition

We define two main types of design features (Ovtcharova, 1993): (1) *pre–defined* design features which are always available and define the fixed set of most used design features, and (2) *user–defined* design features which complete the previous feature set and are dynamically created during the design session. The pre–defined design features are more often used design features, such as cylindrical holes, rectangular pockets, simple slots. Each pre–defined design feature is implicitly described by a name, the type of the corresponding form feature (depression, protrusion), a list of design parameters, constraints to express semantic information and a set of methods which are necessary for creating and manipulating the feature (Figure 1). If the set of pre–defined features is not sufficient, the designer can specify his own set of features, which will be included in a user–defined design feature library. This library is created dynamically (during the interactive design session) and complete the previous library with more complex design features.

In the design phase of the prototype development, the creation of semantically correct feature–based models not only increase the quality of the design but also influences the performance of the downstream phases. The certainty of the semantically correction of the model data, is also an important factor to achieve an integrated development process. Thus, resulting in faster and cheaper prototype development. Product semantics are expressed by different kinds of *features* and *constraints*. Therefore, we state that constraints are one of the major product information, besides features, used by designers to express their design intent. Constraints attached to features describe the consistent states that features can hold, and express the behaviour of the features in the prototype part model, as well as the behaviour of the model itself. For all these reasons constraints are used as crucial elements to preserve the semantic validity of feature–based models (Vieira, 1994).

In our approach, feature constraints are expressed on the basis of a full parametric definition of features. Parameters might be dimensions, locations, and functional or technological parameters without any direct geometric meaning. Parameters allow easier manipulation of feature entities, quick design changes, powerful engineering analysis of the prototype model, and improve the quality of the data transferred during the design process and to downstream applications. In association with parameters, constraints raise a potent design method to solve the design change problem, and to capture and maintain the design intent.

In our work, *three categories* of constraints exist. Two of them are based on the theory behind the parametric and variational design methodologies (Weissbarth, 1994), (Chung, 1990), (Roller, 1994) for solving geometry. The third category of constraints includes all the constraints which do not need to be solved, but only tested for validity.

In this paper, we deliberately omitted most of the definition and representation aspects of features and constraints which are described in details in our previous publications (Ovtcharova, 1993), (Ovtcharova, 1992).

3 THE BENEFIT OF USING FEATURES FOR VIRTUAL PROTOTYPING

The intention of features is to provide the designer with a set of meaningful engineering aides, thus supporting him in his creative work at a higher level of abstraction. There is a common agreement that design by features has the potential to *improve* the quality of the design, to *speed up* the design process, to *reduce* the costs and to *shorten* the time–to–market. It also improve the link of the different phases of prototyping, such as design, analysis, manufacturing simulation, cost and quality estimation. The advantage of using features for virtual prototyping can be characterized by the definition of feature data at different levels of abstraction, the maintenance of constraints for preserving semantically correct feature–based models, and the development of an architecture for feature–based design adaptable to users and applications.

The purpose of this paper is to suggest essential directions for feature–based design for supporting virtual prototyping. Using a series of assumptions regarding conceptional, implementational, and applicational issues, we articulate the direction of design by features based on evidence through summaries of previous work and examples of design applications.

The following, presents essential challenges for research and development in feature–based design for virtual prototyping.

Challenge 1: Design of a methodology for developing comprehensive semantic models

For the integration of different virtual prototyping tasks it is necessary to take into account comprehensive semantically correct prototype models *through the entire prototyping cycle*. Features are used to define *comprehensive semantic models* of prototypes using data at different levels of abstraction in a way which allows a high degree of correspondence between the virtual prototype and the physical product.

Challenge 2: Support for configurable prototype models

Feature–based models offer facilities for the configuration of prototype models by using *pre–defined* and *user–defined* features. Tools and methods for graphical–interactive generation of features can be developed to support the designer in creating his own sets of features. User–defined design features will complete the set of standard (or pre–defined) features delivered with the system and can extend dynamically during the design session as required by applications and user needs.

Challenge 3: Support for easy and fast redesign

Design feature models offer facilities for easy and fast redesign based on parametrics and variational design methods which are used by feature constraint entities. Design features can be described by a *set of parameters and constraints* required to build the basis for consistency maintenance of features during the whole design process. The association of values to the parameters and the satisfaction of the constraints will offer useful and semantic correct feature representations.

Challenge 4: Development of an integrated feature processing environment

Features support the intuitive and creative working of the designer in different design phases, such as conceptual, embodiment and detail design. The models based on features can be processed in downstream phases of the prototype development, such as analysis, manufacturing simulation, cost and quality estimation. To take full advantage of features, an *integrated environment* supporting the design and the downstream phases has to be developed.

Challenge 5: Development of an open, modular and extensible system architecture

Feature–based design tools must have an open and modular architecture which allows them to be integrated in different CAE system environments. *CAE system architecture* can be developed by a complete separation of the functionality of the feature modelers from the underlying solid modelers. Thus, possibilities are offered to the users in modifying their feature modeler via integrating new feature modeling facilities configurating the feature modelers to new applications without changing the solid modelers.

4 FEATURE PROCESSING USING SINFONIA

For supporting virtual prototyping through feature processing we introduce SINFONIA, a module for *feature–based design* which is configurable to users and applications within diverse CAD environments, particularly in the area of mechanical engineering (Ovtcharova, 1994). The module has an open and modular architecture allowing the modification of existing functionalities and integration of new modeling facilities and application tasks.

The main modules of SINFONIA are the *Feature Modeler* and the *Design Feature Manager* (Figure 2). The Feature Modeler is responsible for the instantiation and modification of features and the creation of the feature–based model. The Design Feature Manager allows feature data and design processes to be managed in a uniform way. The CAD system environment in which SINFONIA is integrated consists of the following modules: the *User Interface System* and the *Application* modules (offering tools for interaction of the user with application specific part models and for communication with external systems and applications, such as NC modules, etc.), the *Solid Modeler* (responsible for creating the shape representation of the feature–based model), the *Consistency Manager* (providing services to handle all kinds of different constraints within the design environment) and the *Product Database* which includes all services for storing and retrieving various product data.

The central idea in SINFONIA is that the design features defined and instanced from the feature library are used to model a product part in a concrete application context. Their description derives specific meaning from the view of the function of the product part, including shape data as well as non–shape data. Moreover, SINFONIA supports the users to *define their own specific design features*, thereby allowing to represent new types of products. This is accomplished by a design that foresees the interactive specification of design features by the user. The starting point of the design process are *pre–defined design features* delivered with the module. During the design process it is possible to combine pre–defined design features or to define new features and store them as *user–defined design features*. By repea–ting this process more and more complex design features are immediately available during the design process.

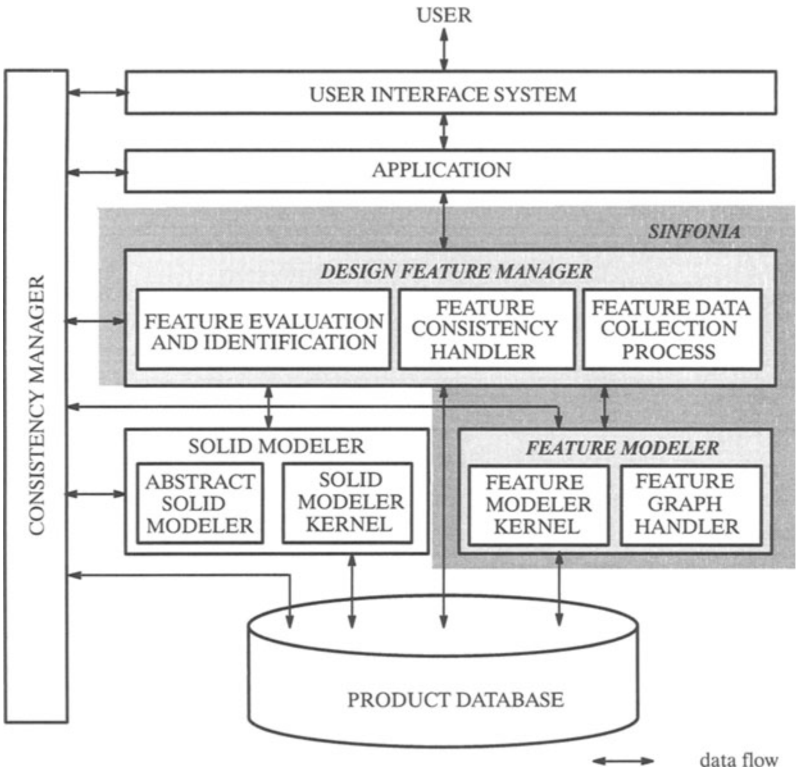

Figure 2 Feature–based design module within an open CAD environment

Figure 3 shows the feature processing pipeline for the creation and modification of pre–defined design feature objects. The design process will be activated by the application or the user interface level (1). This is done by selecting the correspond modelling operation such as create, copy, or modify the feature object. To illustrate this process, the creation and modification of a cylindrical blind hole feature is presented.

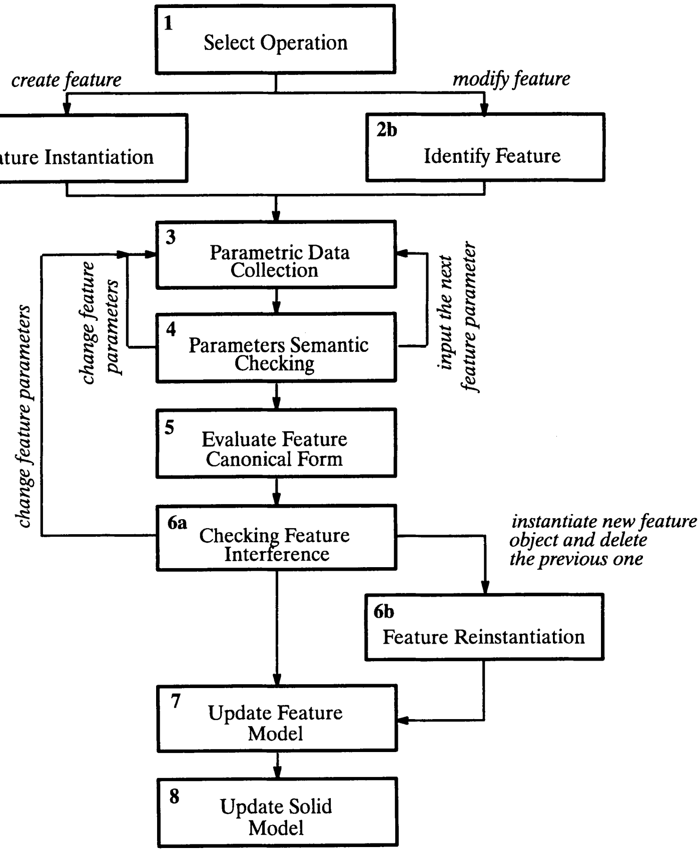

Figure 3 Feature processing pipeline for pre–defined design features

To insert a new design feature in the part model, an instantiation of the feature object will be performed (2a). A set of default parameter values are parsed to this instance. For example, the instantiation of the cylindrical blind hole specifies automatically a form feature of type depression and a cross section of type circular.

In the case of a feature modification, a shape entity is selected and the feature is identified from the feature graph representation (2b). This process of identification is handled by the feature evaluation and identification module.

The next step in this process consists of the complete collection and consistency checking of the semantic feature data (3,4), for example, the parameters defining the feature.

For our cylindrical blind hole, parametric data such as location (lcs) , radius (r), depth (d) and material (mat) must be supplied. These data will be checked (without interference on the semantic of other features of the part model), as shown in Figure 4a and Figure 4b, through geometric analysis techniques of the part model. The results of this geometric analysis can be used as input for the constraint solving process. The data can be collected and checked either interactively while designing or, non-interactively after all parameters are collected. If a semantic error occurs, the process returns for a correction of the inputted data or is aborted.

After, the canonical form of the design feature is evaluated (5). This intermediate shape representation is used for the second phase of the consistency check: the detection of feature interferences (6a). Figure 4b shows a simple situation where only consistency checking on the parametric data is not sufficient in order to detect this interference. Even if the part model is described parametrically, a check on the parametric model representation can often be too complex. Moreover, it is some times not solvable.

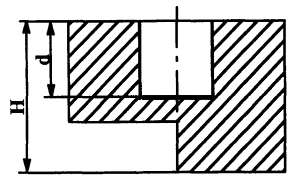

Figure 4a Creation of a semantically correct blind hole

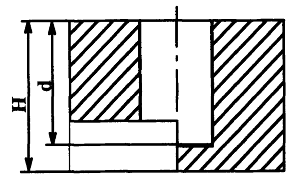

Figure 4b Modification of the cylindrical blind hole

The parametric check (d<H) is not enough to detect the semantic inconsistency of the blind hole

At this stage of the design process, the following situation can occur:

- no interference is found and the process continues;
- interferences with other features where found but they do not change the semantic of the features in the part model;
- interferences where found and the user aborts the process; and finally,
- interferences allow a change of the object class and the process continues.

This last situation is also illustrated on Figure 4b. The user had the creation of a blind hole in mind. However, the depth entered was to big and an interference was found. In this situation the system can analyze if the entered data fits semantically in some other kind of design feature available in the modeler. In our example the suggestion could be the reinstantiation of a cylindrical through hole (6b).

Finally, the design feature data are represented in the feature model. This representation consists of the creation of a node within the feature model and the insertion in the hierarchical graph structure (7). This process is performed by the feature modeler and the feature graph handler. The insertion process detects all dependencies of this feature object to all other features within the graph. The corresponding solid model representation will be updated (8).

5 CONCLUSION AND FUTURE WORK

In this paper, we focused our attention on the design processing aspect of the whole virtual prototyping process. Some essential requirements for the computer–based representation of virtual prototypes using the feature paradigm have been identified as follows: First, to support the intuitive and creative work of the designer in different design phases, such as conceptual, embodiment and detail design. The models generated in the design phases have to be processed in downstream phases of the prototype development, such as analysis, manufacturing simulation, cost and quality estimation. Second, to support different levels of abstraction to build comprehensive semantic models. Third, a mechanism is needed to check the consistency of the semantic model based on feature constraints. Fourth, facilities have to be provided for applying constraints for easy and fast redesign based on parametrics and variational design methods and tools. Fifth, easy and fast configuration by using pre–defined and user–defined modeling entities is needed.

To satisfy these requirements, an environment for integrated feature processing has to be developed. The system architecture of this environment has to be open, modular and extensible to allows the integration into different CAE systems.

As part of such an environment, we proposed SINFONIA, a feature–based design module. SINFONIA has an open and modular architecture that allows to modify and extend existing functionalities, and to integrate new modeling facilities and application tasks. Two main modules, the *Feature Modeler* and the *Design Feature Manager* have been presented in details. The Feature Modeler is responsible for the instantiation of features and the creation of the feature–based model. The Design Feature Manager allows feature data and design processes to be managed in a uniform way. Using SINFONIA, the users can work with standard pre–defined design features delivered with the module, or to define dynamically their own specific design features during the design session. Furthermore, SINFONIA allows the interactive definition of constraints concerning the product semantics. The CAD system environment in which SINFONIA is integrated has been outlined and the most important modules have been briefly described. At this stage of our investigation, we do not provide an overall implementation, but demonstrated that a sound conceptual foundation is necessary, achievable and realizable. Currently, an object–oriented version (C++ programming language) of the both modules, the Feature Modeler and the Design Feature Manager is in development.

6 REFERENCES

Chapin, W., Lacey, T. and Leifer, L. (1994) DesignSpace: A Manual Interaction Environment for Computer Aided Design, *Proceedings of the CHI'94 Conference Companion,* Boston, Massachusetts, USA, April 24–28, pp.33–34.

Chung, J.C. and Schussel, M.D. (1990) Technical Evaluation of Variational and Parametric Design, *Computers in Engineering,* Vol. 1, pp.289–298.

Ovtcharova, J. and Jasnoch, U. (1993) Towards a Consistency Management in a Feature–Based Design, *Proceedings of the 1993 ASME International Computers in Engineering Conference,* San Diego, California, USA, August 8–12, pp. 129–143.

Ovtcharova, J., Haßinger, S., Vieira, A.S., Jasnoch, U. and Rix, J. (1994) Sinfonia – An Open Feature–Based Design Module, *Proceedings of the 14th ASME International Computers in Engineering (CIE94) Conference,* Minneapolis, Minnesota, USA, September 11–14, Vol.1, pp.29–43.

Ovtcharova, J., Pahl, G. and Rix, J. (1992) A Proposal for Feature Classification in Feature–Based Design, *Computer & Graphics*, Vol.16, No.2, pp.187–195.

Roller, R. (1994) Foundation of Parametric Modeling, *Parametric and Variational Design*, B.G. Teubner, Stuttgart, pp.63–71.

Vieira, A.S., Ovtcharova, J. and Jasnoch, U. (1994) Consistency Management Aspects in Sinfonia, *Proceedings of the Workshop on Graphics and Modeling in Science & Technology,* Coimbra, Portugal, June 27–28.

Weissbarth, T. (1994) Associative Design. The Variational and Parametric Design Approach in the CAD/CAM/CAE Product EMS of Intergraph, *Parametric and Variational Design*, B.G. Teubner, Stuttgart, pp.73–80.

7 BIOGRAPHY

Dr. Jivka Ovtcharova graduated from the Department of Heat–Power Processes and Control of Nuclear Stations, Moscow Power Institute, Russia in 1982. From 1982 to 1987, she worked at the Bulgarian Academy of Sciences, Sofia. In 1987, she joined the Technical University of Darmstadt as a researcher. Jivka Ovtcharova received her doctor's degree in computer sciences from the Technical University of Sofia in 1992. She is currently a project ma–

nager at the Fraunhofer Institute for Computer Graphics, Darmstadt, Germany. Her current work is focused on the development of feature–based design modules adaptable to diverse CAD environments, in particular for supporting rapid prototyping and virtual prototyping technologies.

Ana Sofia Vieira graduated from the Department of Mathematics, University of Coimbra, Portugal in 1993. Since March 1992 she has been working as a guest researcher at the Fraunhofer Institute for Computer Graphics on the field of Cooperative & Hypermedia Systems. Since May 1993 she is working on the concept and development of a Consistency Mechanism for Feature–based Design Systems, based on the combination of three design methodologies: feature–based design, constraint–based design, and parametric design. Her research interests include feature–based modeling, algebraic and algorithmic constraint solving techniques and object–oriented analysis.