8

Dynamic and semantic modeling - a new approach to model products

K.-P. Greipel
Siemens Nixdorf Informationssysteme AG
Otto-Hahn-Ring 6
D-81739 München
Tel. ++49 89 636 49852
Fax ++49 89 636 47413

J. Colpaert
Siemens Nixdorf Informationssysteme AG
Siemenslaan 1
B-8020 Oostkamp
Tel. ++32 50 83 2739
Fax ++32 50 83 2488

Abstract

We present a new modeling approach that transfers the power of object orientation from the programmer to the end user. The approach is non-geometric and independent of the application. The keywords are *semantic* (supporting conceptual reasoning), *dynamic* (supporting creativity and flexibility), *relations* (supporting constraints) and *structure* (reducing complexity). The main innovation is its dynamic object technology: the end user can extend applications at runtime with her/his problem-specific notions, constraints and structures. Coupled with modern intelligent user interface techniques this technology realizes a new big step towards intuitive graphical interactive example-based programming. This opens the horizon for a new generation of easy-to-customize CAD systems that exceed the possibilities of current parametric and feature-based shape modeling and introduces more logic into modeling. New system can be realized that support early design and branch-specific product modeling.

Keywords

dynamic object technology, early design phase, semantic product modeling

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

1 INTRODUCTION

Current CAD tools are essentially geometry-oriented. Although concepts as driving dimensions (parametrics) and regions of interest (features) have been introduced in industrial CAD systems, the modeling focus is still the shape of a product. The core of a product description still is geometry: points, curves, surfaces and solids are the building blocks of current CAD systems. In advanced system some relations are imposed to geometry, for example: construction rules (parallel, orthogonal, aligned, etc.), variant dimensions, feature-to-part structures. Product performance aspects and product manufacturing are derived from these geometric descriptions that already detail the product to the final shape.

The support during early design phases remains completely unresolved. Continuous reengineering and the use of better and better tools have optimized manufacturing since the begin of industries by factors up to thousand whereas the product design process has essentially remained the same. Production industries are increasingly recognizing that the design process has to be optimized to reach time-to-market requests. Thus investigations of companies and scientific experts say:

- 2/3 of product development time are spent in the design phase
- 80% of the cost of a product is determined in the early design phase
- only 20% of the product knowledge is reached in the early design phase.

CAD as electronically drawing board with 3D extrusions (at the high end side) cannot help to solve these problems.

2 NEW MODELING PARADIGM

We propose a new modeling approach to solve these problems. This approach does not regard geometry as center of a CAD system any longer. It propagates a dynamic logical data scheme called dynamic object technology as new core. This innovative scheme is capable to describe the semantics, relations, structures and multiple representations of a product. The shape aspect is merely handled as one specific view that is nevertheless of greatest importance as natural interaction environment, for visual and physical verifications and for shape design reasons.

2.1 Example

In the following we consider a simple example. It illustrates how additional views like functional specification view, principle solution view, structure view and layout view can enhance the early design process. We consider the functional requirement to be 'combine two plates'. A possible solution is to 'connect both plates with bolt and nut'. This results in the following structure: two 'plates' with subfeature 'countersink hole' and a 'bolt' with 'nut'. The layout of all features and parts is shown in the picture below. In addition these components may have many interdependencies. So the bolt and its nut may be read from a discrete standard part table according to the thickness of the two plates. The diameters of the subfeature countersink hole are again dependent on the diameters of bolt, bolt head and nut. If the designer has modeled with these relations, then a simple editing of say the

thickness of one of the plates will result in an automatic consistent update of all other components.

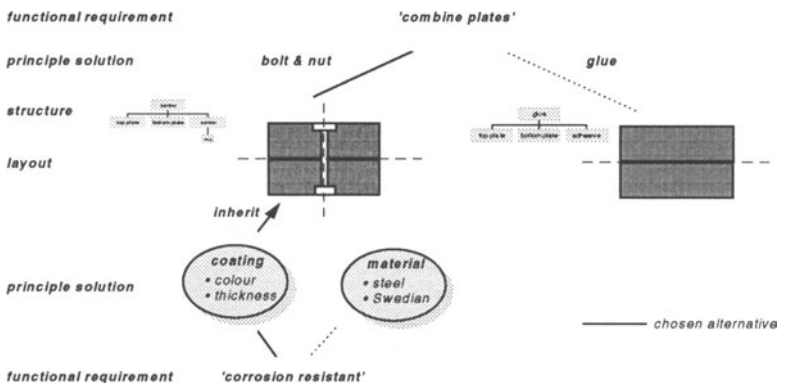

Figure 1 A simple design example capturing full design intent.

An alternative solution may be to 'glue' the two standard plates leading to a complete different structure and bill of material consisting of two plates and the 'adhesive'.

Lets assume the designer decides for the first solution for ease of maintenance. In a concurrent engineering environment a new additional requirement considering the corrosion resistance of the bottom plate may arise. Again the designer may consider two solutions: a special material or a special coating for the bottom plate. A completely new notion 'corrosion resistance' will be added to the design model with attributes that have never been considered before.

This very simple example already reveals

- how the additional semantic modeling domains function, principle solution, structure and layout help to capture design intent and help to approve design concept as early as possible
- how dynamic modeling helps to react to new requirements without loosing already designed elements.

2.2 Semantic modeling

Semantic modeling complements the geometric view of products with views needed to support conceptual reasoning and early product knowledge. In the early design stages a functional requirement is fulfilled with a coarse technical solution. These are described in a structural way (the core components of solution), with relations (what are the dependencies between the components) and layout (how do these components fit together). This

conceptual product description can already be used to derive most of the later manufacturing, maintenance and disposal aspects of the product. The layout also serves as a geometric base on which the detailed shape of the product can be modeled.

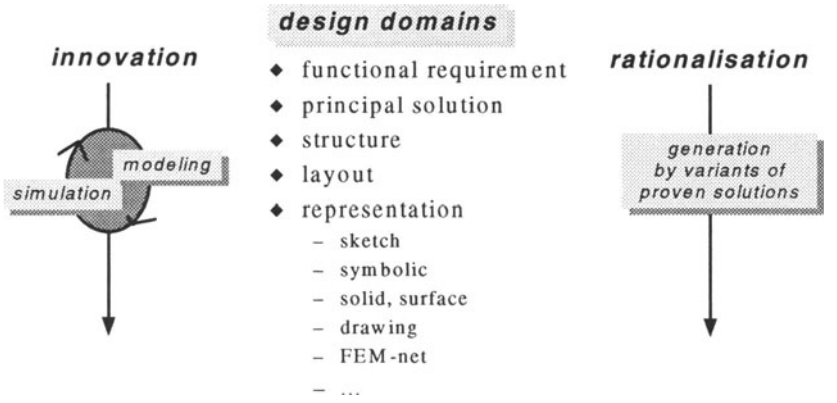

Figure 2 New design domains for semantic modeling.

This modeling approach supports innovative design (with early simulation cycles on not fully detailed models) and reuse of proven solutions (generation by variants of proven solutions).

2.3 Dynamic modeling

In current CAD systems the software engineer determines the objects and methods by which a product can be modeled. The end user can only design in the restricted views (in modern systems: sketch, 3D feature, assembly, drawing) and elementary functionality (in modern systems: parametric geometry operations) programmed in the CAD system. The designer has no possibility at all to add her/his problem-specific views, object types and methods to the system. Dynamic modeling frees the designer's creativity and provides the flexibility needed in concurrent engineering environments. The designer can customize the system to her/his specific needs at runtime. She/he can learn the system to communicate with the user using the semantic notions that are best suited to describe a problem and its solution. The base for this revolutionary modeling approach is the innovative dynamic object technology.

3 DYNAMIC OBJECT TECHNOLOGY

The current object-oriented database systems and programming languages operate with a very rigid relation between the class definition and the instance object of a given class. The class definition exactly describes how each instance object will look like. In CAD systems,

this is a severe handicap for the creative mind of their users and the flexibility of concurrent development processes. To overwhelm this handicap we propose to open object oriented techniques to support end user creativity and flexibility and to bring object technology to end users. This will be provided by the following characteristic of the dynamic object technology:

- dynamic instances
  The content of an object instance can be extended without having to extend the class definition. An object type definition must be like a template. Instances may contain more information than what the type prescribes. When one starts designing an object, one is only aware of the key characteristics it has to satisfy. This can be seen as functional specification. During the design process this specification is enriched to a technical solution within an object instance.
- multiple inheritance
  Multiple inheritance is provided at the object instance level, not only at the object type level. This means that a specific technical solution can fulfill several orthogonal functional requirements (for example regard our example above where the bottom plate is both of type 'plate' and 'corrosion resistant'). In most programming languages one can only inherit from class A and B if an intermediate class that inherits from both A and B has been defined.
- introspection
  The system as well as the models designed in the system are fully transparent. The end user can query system state and data model. Typical questions of the user are: give me all operations I can perform on this object; how has this object been designed (browse the object history)? what other users of the system are using this object type?
- deferred generalization
  Design of technical solution often happens inductively: the designer realizes after constructing an object that she/he has defined a reusable solution. So the technology must be able to derive the class definition (specification) from an object (example solution) that is already there. This behavior has been called deferred generalization. In conventional object oriented systems, one has to define the class before one can create the first instance.
- persistent actions
  As we have seen with objects above, dynamic extensibility is the key to more creative and flexible design. This in addition also implies that at runtime also new commands acting on these objects can be easily defined. In most cases these commands combine as parameters objects of different classes. We therefore introduced the notion of an 'action' that is independent of an assignment to a specific class as methods do. These actions are independent entities that take objects as parameters.

4 OBJECT LAYERS

The innovative object technology is realized in three layers. The core is an object repository. A second layer called dynamic object driver provides the runtime extension method. The last layer called constraint manager is responsible for maintaining consistent relations. The technical concept is based on two advanced techniques exceeding current oodb techniques:

- a general identification scheme guaranteeing unique reproducibility of dependent objects according to whatever changes of their determining environment
- an elaborated basket scheme which collects objects and actions at runtime and builds new types out of them.

The identification scheme manages the associatively and consistency of the data model. The basket scheme delivers the kernel functionality of the dynamic object technology described above. It supplies dynamic instances, multiple inheritance, introspection, deferred generalization and persistent actions.

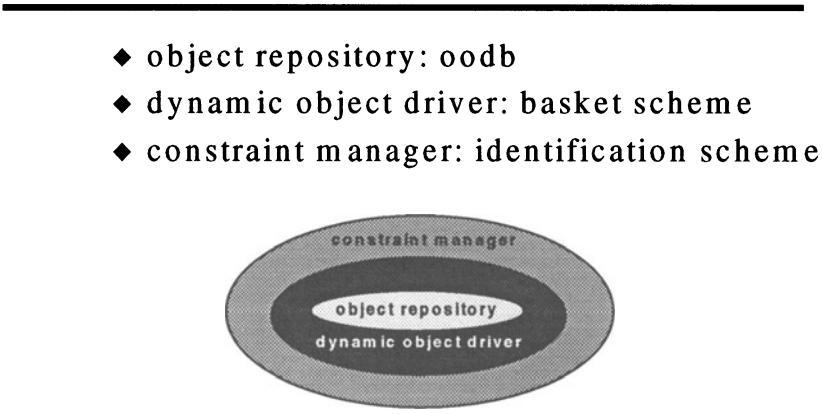

Figure 3 Object layers enabling dynamic and semantic modeling.

5 MODELING ARCHITECTURE

Introducing the described concept into the environment of a CAD system we identified three kinds of different knowledge for a modeling application. These different knowledge types again can be assigned to different software architectural layers.

5.1 IT knowledge

In the 1st layer the information technology knowledge that is common to all CAD is provided in a predefined runtime kernel. It especially handles data management. So the first layer realizes the identification scheme and the basket scheme. The basket scheme presents several predefined generic data structures to the user, e.g.: object and action, user object and user action, layer and group. These predefined structures are again connected to generic user interface elements so that new CAD applications can be built very quickly.

5.2 Domain knowledge

The 2nd layer realizes the basic notions and the basic operators of a CAD application. These are programmed by using the generic objects and actions of the first layer. Typically the geometric functionality of a CAD application is provided in the second layer, e.g.: points, curves, surfaces and solids. This functionality can be realized by integrating industrial standard libraries for geometric computing.

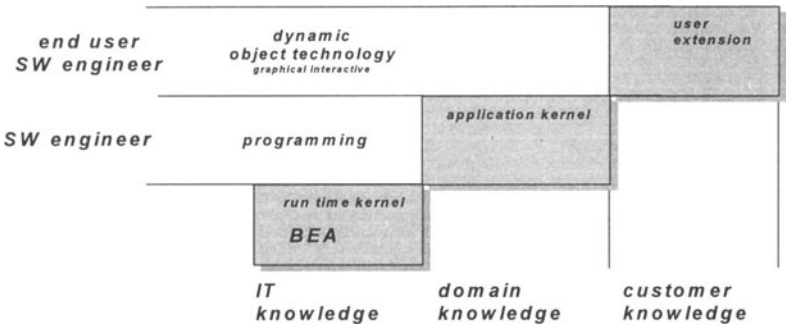

Figure 4 A modeling architecture mapping specialization of knowledge.

5.3 Customer knowledge

The 3rd layer realizes domain-specific and company-specific knowledge. The dynamic extension techniques of the 1st layer and the basic objects and action of the 2nd layer are used to build high level applications with advanced domain know how. Typical examples are sheet metal, driving gears, conveyor equipment. The main advantage of the dynamic object technology is that the domain specialist himself can teach the system the special semantics. There is no more translation to be done between the domain engineers' language and the software engineers' language needed to realize specific applications.

6 STATE AND PROSPECT

The core ideas of dynamic and semantic modeling have already been proven: first advanced prototypes show the promising potential of this new modeling approach. A new CAD direction overwhelming current geometric thinking and capturing full design intent can be developed.

BIOGRAPHY

K.-P. Greipel Doctorate in mathematics at University of Munich. Assistant professor. At Siemens Nixdorf since 84. Senior developer and project manager. Design of innovative CAD systems for electronics and mechanics. Central consultant for logistics and production. Manager of international large scale projects.

J. Colpaert Electronic engineering degree from the KIH Ostende. Engineering degree in computer science from the University Leuven. Key member of start up team of the electronic imaging systems department at AGFA-Gevaert. Designer of laser printer architectures. Manager of the printer software department. Group leader for innovative CA-technologies at Siemens Nixdorf since 84.