PERGAMON

Robotics and Computer Integrated Manufacturing 18 (2002) 275–281

Robotics and Computer Integrated Manufacturing

www.elsevier.com/locate/rcim

A feature-based database evolution approach in the design process

Raimundo R.M. da Cunha*, A. Dias

Mechanical Engineering Department, Federal University of Santa Catarina, P.O. Box 246, 88040-900 Trindade, Brazil

Abstract

Design data are assigned in geometric and non-geometric form in order to meet design requirements. These data and information must be encapsulated in a data structure that has significance for design applications in each design process phase. The main goal of this research is to find design data groups that represent each mechanical design phase, which will be called *phase's design signature*. In addition, current data should be an evolution of the geometric and non-geometric information of the previous design phase. In this paper, the purpose is to identify and model a set of design features that encapsulate the design data and their transformations which occurred during the mechanical design phases. This database must capture the designer's intents that can be modeled and implemented using feature-based model in the conventional CAD systems, object-oriented modeling, and Java classes. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Design data; Feature; Object-oriented modeling

1. Introduction

Design information of generic product comprises geometric and non-geometric data types. These two kinds of data may be encapsulated in a single data structure—design objects or design features—that may have different meanings for specific application context.

This paper uses the design features as units of information, which can exchange and share design information among different design phases. Then, each design phase consists of a set of features, whose identification and classification of information and data are specific and typical of the corresponding phase.

In this research feature is defined as:

> Composition of design objects, which encapsulate own attributes and methods of each design phase, called phase's design signature.

This proposal intends to use feature modeling to store information and data analysis of each mechanical design phase, resulting in a "design signature" for each design phase. It means that object-oriented modeling of design feature classes can semantically capture the data transformation and their evolution through the design phases. Besides, the set of features that represents the current design phase should be an evolution of the geometric and non-geometric data of the previous phase.

Currently, the design process methodology that is commonly accepted by more important researchers divides the design process into phases [1] or models [2], which capture the modifications and transformations in each design phase [3]. Therefore, in this paper, it was decided to use models of features to capture the designer's intents in each design phase, such that it may be possible to show the evolution of the entire product design process. For systematization and easiness of integration among design phases and other subsequent stages of the product life cycle, it is intended that the product design data modeling considers its evolution based on the product design process system with well-defined design phases, such as informational, conceptual, preliminary or embodiment, and detailed design.

Therefore, according to that pointed out previously, this research will focus on two main issues, namely:

- Data modeling and feature-based computational data structure for the mechanical design with CAD systems; and
- Computer support to the evolution and transformation of mechanical product design data during the design process phases.

*Corresponding author. Tel.: +55-48-3319899; fax: +55-48-234-1519.

PII: S0736-5845(02)00018-2

The *phase's design feature* can be implemented as a computer interface to instance specific features of each design phase, and to edit the necessary non-geometric information giving it an application meaning. Such features have to converge to feature instances used by conventional CAD system environment in order to create the product-detailed design.

CAD system will be used as a client-platform for the implementation of computer support tools to the mechanical design process. It will provide computational resources or may have such resources integrated/implemented into it, such as standardized components database, feature libraries, product functions, solution principle database, and expert rules.

In Section 2 of this paper, a bibliographical overview will be introduced. Some basic concepts will be discussed, such as the difference between design information and design data, and the closeness between features and the object concept. Some researches about feature modeling, object-oriented modeling and the approaches for capturing the evolution and transformation of design process will be also analyzed.

Section 3 introduces the proposed approach, which considers product documentation in all the phases of mechanical design process; and then the classes of the object-oriented data structure of the phase's design feature are introduced.

Finally, the last section exemplifies, in a schematic manner, the phase's design feature in the mechanical design. Also, some final considerations are given.

2. Bibliographic overview

2.1. The design information and the design data

Stair [4] affirms that it is possible to separate data and information. Information is a set of data with relation and rules well defined and associated to an application context. Data are only values without context, because the data are always available, so it does not need an application context. The information is a result or an output of the transformation process based on data, which are inputs. The information depends on adding knowledge to the data, which are made through relations and rules of the domain application, such as that shown in Fig. 1(a). In conclusion, information is a data with application context.

In design activity, design information refers to nouns, which have semantics related to it. In an object-oriented modeling context, information is associated with definition of classes, whereas the design data refer to values, quantities, dimensions, etc. The design data is a more real, exact, specific and punctual fact. It is commonly associated with object instance.

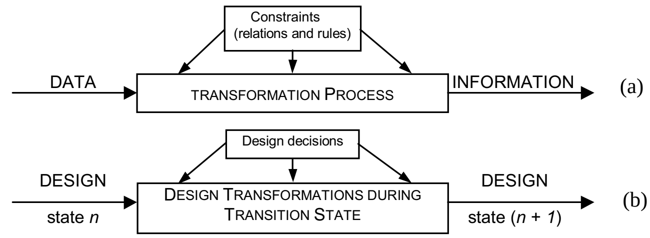

Fig. 1. Transformation process from data to information.

Table 1
Differential vision between design information and design data

Data	x	Information
Is value		Is noun
Is changeable and mutable during a process		Is defined and fixed during a process
Has many representations		Has same meaning and semantic
Is independent of application context		Is dependent of application context
Has a single and only view		Has many views at each application
Has object instance meaning		Has class meaning

There is another difference between design information and design data. Design data can assume different representations, but the design information associated with each representation is the same. It establishes the main design information characteristic based on its unchangeable or immutable nature. The information does not change or transform it into a process design, because the semantics is fixed during a process, so the information should not change. Design data changes during process design, which stores the geometric and non-geometric values. These changes and transformations occur throughout the design phases, when the designer specifies the constraints and takes decision. Fig. 1(b) also shows this transition by defining design states.

When the focus of design is specific to information related to the product, a better product model can be achieved, since isolated information favors product modeling.

By exemplifying the difference between data and information design, consider the "cost of a product". In this case, one may interpret that the "cost" is information related to the product, and the "cost value" is a design data.

Table 1 shows the differences between design data and design information.

Therefore, it is important to have computational algorithms, tools and data structures to save intelligently a complete framework design information constituted by data attributes, behavior, and semantics of the application domain. The goal is to save and retrieve the object instance when needed, which are the design features instantiated in the CAD model.

2.2. Design feature modeling versus object modeling

The research efforts in 1990s recommended that data modeling be made based on the object-oriented model [5]. Thus, most CAD system companies have developed their programs based on object-oriented paradigm. They have provided resources and tools that implement these principles. This choice still prevails and continues to be valid currently and for future modeling.

The development of CAD systems has shown a convergence between the feature-based CAD systems and the object-oriented modeling. There are some researches that show this tendency [5–7]. The computational object-oriented modeling context has shown that object definition encompasses the same role of feature modeling definition in the CAD system technology. Also, in the case of mechanical component design and product development, feature idea is framed perfectly with the object-oriented modeling and programming. But the term object is wide and very subjective, since objects can model everything in the real or abstract idea world. It is also worth pointing out that "object" is a strange term used in the designer's usual terminology. Then, the term feature has been used to replace the object idea in an Engineering context. So, the feature concept is an abstraction that includes the idea of objects inside the design context, such as that shown in Fig. 2.

Some definitions of object and feature concepts listed below are also very similar and demonstrate the closeness between objects (*computer modeling*) and features (*CAD entities modeling*) in the design context.

According to Warman [5], "features or design elements are treated as objects that have underlying code and can handle messages".

For Shah [8], "features are generic forms that the engineers associate certain properties or attributes and useful knowledge in reasoning processes on the product, in other words, the features can be seen as primitives of Engineering".

For McGinnis and Ullman [9], "feature is any particular or specific characteristic of a design object that contains or relates information about that object. It is verbally represented in the form of a noun or noun clause".

For Salomons [10], "features can be treated as design objects, belonging to a general class, which inherits properties of other classes".

Therefore, from the sequence of feature definitions and taking into account the object-oriented modeling theory, it is concluded that

$$\text{Object} \equiv \text{Feature} \Leftrightarrow (\text{Geometry, Topology}) \cup \text{Semantic}. \tag{1}$$

In Eq. (1), the geometry and topology represent the physical model portion, which is independent, exact and quantitative, whereas the semantics represents the abstract model portion, which is context-dependent, subjective and qualitative.

While this abstraction advance of the CAD entities modeling appeared initially in the area of mechanical engineering, it was first noticed with commercial implementations in the area of architecture, engineering and construction (AEC), such as that cited in Day [7] and Achten and Leeuwen [3]. Presently, object-oriented modeling using features have been growing and spreading in the mechanical design and product development areas, and it is the current tendency of the CAD market.

So, it may also be said that there is a consensus of several authors whose product model based on features together with other modeling technologies (such as virtual reality, object-oriented modeling, etc.), consolidates them as appropriate formalisms for the representation of the designer's intents and data evolution during the design phases.

2.3. Product data and information modeling in the design process phases

This paper assumes that the role of information in design activities is to give a meaning or semantics to the design data, according to McGinnis and Ullman's [9]

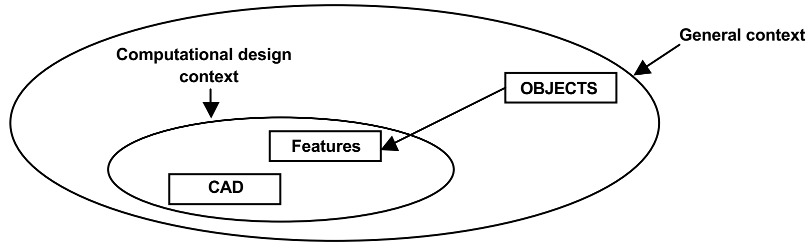

Fig. 2. Convergence between general and computational design context.

vision. They affirmed that, when designing a component, some constraints on its form and function are given at the initial design phases. Other constraints arise from the designer's knowledge of the domain, and other constraints are derived from each design decision made. They conclude that the component refinement evolution depends on a set of relationships and interplay among design constraints. The authors also assign that the final product design is the result of the actions in design state by means of a set of changes called transition state. The design state is due to the design decision and the state transition or transformations process is evidenced by changes on attributes of design constraints—relations and rules of domain application—such as that shown in Fig. 1(b).

McGinnis and Ullman [9] identified two types of protocol structures to characterize the relationship between constraints and design features. They are exemplified below:

The "instantiation protocol" of the features is

- ⟨form feature⟩-⟨instantiation⟩ and ⟨functional feature⟩-⟨instantiation⟩

The instantiation protocol refers to the values specified for the feature.

Ex.: (operation time is 40 s) or (hole height $\geqslant$ 10 mm)

The "relationship protocol" occurs among features:

- ⟨dependent feature—relationship—feature independent(s)⟩

The relationship protocol establishes that one or more features restrict a single feature. They identified ten possibilities of protocols corresponding to this pattern.

For example:

- ⟨form feature⟩-⟨form relationship⟩-⟨feature independent(s)⟩

Ex.: (hole position—in the middle of—support base)

- ⟨form feature⟩-⟨functional relationship⟩-⟨feature independent(s)⟩

Ex.: (central hole—supports—principal axis).

The work developed by McGinnis and Ullman [9] are fundamental ideas for the understanding the relationships among constraints and design features on initial design phases, mainly during informational (clarification task) and conceptual phases. Also, the protocol structures above show a certain relationship pattern that may be used for standardization of the interrelationships between constraints and design features, considering form and functional aspects.

Au and Yuen [11] also developed research with the same goal as McGinnis and Ullman [9]. Au and Yuen propose a hierarchical structure of the vocabulary between objects, features, and relations (constraints), as shown in Fig. 3. This structure was represented through a feature graph. By analyzing and comparing this graph with other structures proposed in other researches [9,11,14], it could be concluded that the feature graph might be used to represent graphically the protocol established by McGinnis and Ullman [9]. It meets and satisfies the consistent data structure to relate constraints and features.

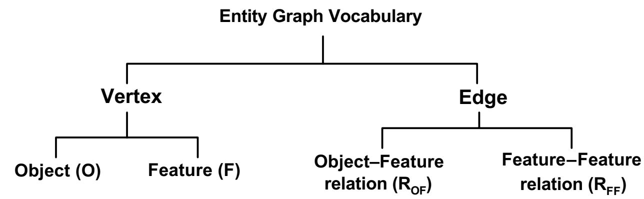

Fig. 3. Hierarchical structure of the entity graph vocabulary (Au and Yuen [11]).

A representation of the evolution and transformation of the design data based on feature model was also proposed by Achten and Leeuwen [3]. They accomplished a case study of the architectural design domain. Based on the drawings that represented the design phases, they made an analysis following the steps mentioned below:

- In each phase, they identified the information in the features in order to be aware of the information type to be treated in the problem;
- If the elements are new, then they made the definition and the type registration of the simple and complex features.

Ex.: Example of a Sketch feature type

complex ConceptualFeature.sketch.Sketch {
TypeDate {12/03/2002}
TypeAuthor {RR}
TypeDescr {"Store the general shapes and rough dimensions of the design sketch"}
Spec ConceptualFeature.sketch.Sketch contains[0..?];
Spec User.value.Dimension dimension[0..?];
Spec User.value.Function function;
Has ConceptualFeature.structure.AssociatedFeatures listFeatures;
Spec User.value.NumberOfFeatures numberOfFeatures;
}

- In the case of features already defined, they made the instantiation of the corresponding objects, filling out the respective attributes.

Ex.: Instantiation of a Sketch steelStructure feature, given by the authors

```
ConceptualFeature.sketch.Sketch steelStructure = {
        contains[1] = bar1
        dimension[1] = 20
        contains[2] = bar2
        dimension[2] = 10
        function = {FunctionSupport}
        listFeatures = {slab1, slab2}
        numberOfFeatures = 2
}
```

The types and the instances of the features are defined and represented textually through a language of specific definition, called feature type definition language (FTDL), just as shown above in examples. A tool developed in that work and described in other researches supports this language.

As Achten and Leeuwen [3] describe the transformations of the product object using a set of predefined objects, it will use the objects that encapsulate the design information, such that they will serve as a mean to visualize the evolution during the design phases; and to constitute the phase's design feature.

Regli and Cicirello [12] also proposed a design signature represented by a graphical structure, which consisted of a hypergraph, where the vertices represented the design attributes and the entities in the boundary model that the attributes refer to.

Generally, these researches are based on product model, which works as an interface among different product viewpoints. For Van Der Net, mentioned by Maziero et al. [14], the product model should satisfy three basic needs:

- In an integration vision of the product life cycle—it is to capture and to register the intentions, wishes and the designer's reasoning;
- In a design vision—it is to create a consistent product description to aid their design, as well as subsequent activities of the life cycle;
- In a production vision—it allows to analyze the product manufacturability, simultaneously to the design development.

These topics above favor concurrent engineering (CE), and integrated product development. But the idea of a global product model constituted for several views of the product needs two basic requirements, namely:

- A consistent, extensible and reusable data structure, which needs standards of objects definition, terminology, procedures and computational architecture [7];
- A consensual design methodology to orient the process design.

These two requirements will be discussed in the next section.

3. Design phase features

3.1. Design methodology by using CAD systems for product documentation

Reusing, sharing and exchanging of data and information in the design process need a consistent computational and methodological background. So, it is necessary to follow a design methodology with well-defined phases.

This paper will use the design methodology defined by Pahl and Beitz [1], as shown in Fig. 4(a) and (b). Many other researchers have used the same sequence of phases to describe the design process of the product.

It has been observed that the product documentation is made only during the detailed design, as shown in Fig. 4(a). In this paper, it will be considered that product documentation has to be made during the whole design process, as shown in Fig. 4(b) and it is recommended by Cunha [13]. So, much information related to the product

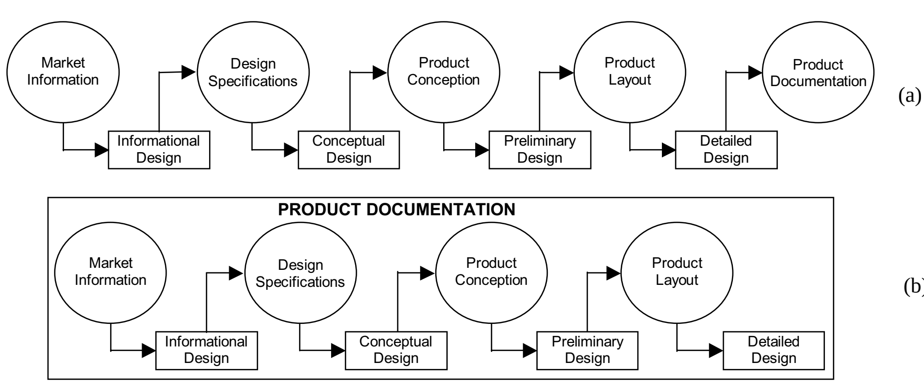

Fig. 4. Design phases and product documentation using CAD systems.

Fig. 5. Feature classification on mechanical design process.

conception has to be encapsulated in a new data structure. The question is what kind of data structure has to be used in order to fulfill all the information concerned in all the design phases of a product? And how will the new data structure be used and integrated to CAD systems?

3.2. Object-oriented data structure of design

The features are modeled as objects that encapsulate the geometric and topological data and their semantics (identity, state, and behavior) with specific attributes of each design phase. In this paper, the feature concept will be used to define the new data structure to document each design process phase.

Fig. 5 shows a hierarchical proposal of features based on the design process, which focus on the evolution of the mechanical design during the process phases.

The hierarchical structure assumes a generalization/specialization relationship characteristic, that is, each feature class is an evolution of the previous class and inherits its attributes and methods.

Each feature is a class that assumes a whole/part relationships' characteristic; design phase feature constitutes for many other design features that are specific to the design context of the product modeling.

The superclass Feature class has generic attributes for feature identification, and manipulation methods, such as instantiation, and insertion/deletion of the feature in a CAD model. The goal is that other classifications should get their attributes by using the inheritance mechanism. It favors data reusability and data sharing among different contexts or views of the product.

Other classes of features—Informational, Conceptual, Preliminary, and Detailed—are the specialization of the superclass Feature. They will be defined by the composition of other features, such that they represent and characterize the corresponding design phase.

3.3. The data design evolutions and transformations

Some examples of identification of features that characterize the design process phases will be presented. Object-based computer modeling suggests compositions of objects that become easier to spread various applications to this data structure, such as that shown in Fig. 5.

The data structure of the informational and conceptual phases should be able to instance features like tables that feed database and vice versa. Most of the data in those phases are organized in databases, in order to evaluate user specifications, design restrictions, capabilities and tools to manufacture and evaluate cost. No geometric shapes are present in these phases, although some sketches of the product may appear in the end of the conceptual phase. Some information begins to be linked to the graphic forms, which can be instantiated later as basic features of the CAD system.

In the informational design, Informational_Feature class can represent the data structure that has the following form:

```
Informational_Feature {
        Constraint_List
        Material_List
        ClientRequirement_Table
        DesignRequirement_Table
        ...
}
```

The information in this phase is characterized as having a textual and descriptive nature. The Informational_Feature gets attributes such as the list of design constraints, a rough bill of materials (BOM) of its component parts, client requirement table, and design requirement table. The main goal is to obtain the first version of the design specifications.

Following the design methodology, the Conceptual_Feature class is where the Informational_Feature becomes a structure of functions. So, these can be composed of the following:

```
Conceptual_Feature {
        GlobalFunction_List
        PartialFunction_List
        ElementaryFunction_List
        DesignIcon_List
        ...
}
```

In this phase, the information is built by the list of product functions: global, partial and elementary functions. The main goal here is to obtain a product conception or an alternative solution to the design problem.

In the preliminary design phase, the geometric information is still not definitive. But the structure of functionality of product takes form. Some geometry and its relationships begin to become visible. All the information captured in this phase will be defined as the Preliminary_Feature class. This feature can be composed of the following:

```
Preliminary_Feature {
        ProductSketch_Shape
        HolePosition&Direction
        HoleDirection&Direction
        MainProduct_Regions
        ...
}
```

The Preliminary_Feature class encompasses general information, such as a sketch of product shape, positions, locations of parts, and subsystems. The main goal here is to obtain a product layout that fulfills each design requirement, user specification and design constraints introduced in the conceptual and informational phases.

Once the embodiment of the product is obtained, it is ready to be detailed. The Detailed_Feature class composes the product data structure, where each part and its details have been defined in order to be manufactured. This paper suggests that it is possible to use a conventional CAD system that provides customization tools. So, such a class has to encompass data that can be used by a CAD system, in order to make the designer task fast and time saving. The Detailed_Feature may have the following form:

```
Detailed_Feature {
        BasicFeatures_List
        Compound_Features
        ...
}
```

The data and information present in this class should make it easy to get a final and detailed product with all the information necessary to create the geometric and non-geometric CAD model, and to link the product development with subsequent activities.

An application to manage the proposed data structure will be implemented using the Java programming language, and the integration with the CAD system should occur during the detailed design phase. All the design information can be stored by using the object persistence mechanism provided by Java.

4. Conclusion

The classes of the phase's design features represented in this paper are only a schematic representation of the kinds of attributes present in the respective classes. This paper introduced a proposal for a wider data structure of product design process. The next step of this research is to model and define more precisely the design data and information, such that product design data can be captured in each phase. The main challenge is to structure each feature class so that it can link non-geometric data to geometric CAD model to represent the data needed in the product design evolution.

Special attention should be given to the modeling of the design data and design information, which will decide the success or failure of this proposal for efficient exchange and sharing of data among the different mechanical design phases. It has to be integrated to appropriate computational tools such as CAD systems with database capabilities. These issues are prevalent in current researches and it will be incorporated into the next CAD system generation.

References

[1] Pahl G, Beitz W. Engineering design—a systematic approach. 2a. ed. Springer: London, 1996. p. 544.
[2] Ohsuga S. Toward intelligent CAD systems. Comput Aided Des 1989;21(5):316–37.
[3] Achten H, van Leeuwen J. A feature-based description for design processes: a case study. 1998, URL: http://ds.calibre.bwk.tue.nl/research/publications, October, 2000.
[4] Stair RM. Principles of information systems—a managerial approach. 2nd ed. Thompson Publishing Inc., 1996.
[5] Warman EA. Object oriented programming and CAD. J Eng Des 1990;1(1):37–46.
[6] Kumar V, Burns D, Dutta D, Hoffmann C. A framework for object modeling. Comput Aided Des 1999;31(9):541–56.
[7] Day M. Objects in CAD, CADD & CADdesk–Web Resource, 2000, URL: www. edaltd.co.uk.
[8] Shah JJ. Assessment of features technology. Comput Aided Des 1991;23(5):331–43.
[9] McGinnis BD, Ullman DG. The evolution of commitments in the design of a component. J Mech Des 1992;114, URL: www.engr.orst.edu/~ullman.
[10] Salomons OW, van Houten FJAM, Kals HJJ. Review of research in feature-based design. J Manuf Syst 1993;12(2):27.
[11] Au CK, Yuen MMF. A semantic feature language for sculptured object modeling. Comput Aided Des 2000;32:63–74.
[12] Regli WC, Cicirello VA. Managing digital libraries for computer-aided design. Comput Aided Des 1999;32:119–32.
[13] da Cunha RRM. Feature-based exchange of data and information on mechanical design phases applying CAD systems (original in Portuguese), Florianópolis-SC–South of Brazil, November-2000 (Qualifying), Mechanical Engineering Department, Federal University of Santa Catarina, 2000. 110pp.
[14] Maziero NL, Ferreira JCE, Pacheco FS, Prim MF. A feature-based object-oriented expert system to model and support product design. J Braz Soc Mech Sci 2000;XXII(4):523–43.