Product Development with Multiple-View Feature Modelling

W. F. BRONSVOORT, A. NOORT, J. van den BERG and G.F.M. HOEK

Computer Graphics and CAD/CAM Group

Faculty of Information Technology and Systems

Delft University of Technology

The Netherlands

W.F.Bronsvoort/A.Noort@its.tudelft.nl

Abstract: Multiple-view feature modelling can support applications from various phases of product development, by providing an own view on a product for each of these applications. Each view contains a feature model of the product specific for the application. Current approaches to multiple-view feature modelling only deal with form features. Hence, only the later product developments phases, in which the geometry has to be fully specified, are supported. In addition, they only deal with single parts, not with assemblies. A new approach to multiple-view feature modelling is presented here that overcomes these shortcomings. It supports conceptual design, assembly design, part detail design and part manufacturing planning, by providing a view for each of them. The views and the operations that can be applied on them, and the ways the views are kept consistent, are described.

Key words: Feature modelling, Feature views, Consistency maintenance, Conceptual design, Assembly design, Part detail design, Part manufacturing planning

1. INTRODUCTION

Contemporary product development requires dealing with resources economically, and aims at higher quality of products. To realize this, concepts like concurrent engineering and feature modelling have been introduced. Concurrent engineering aims to design better products in less time, by using Design For X (Huang 1996) and simultaneous engineering

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35637-2_11](https://doi.org/10.1007/978-0-387-35637-2_11)

R. Soenen et al. (eds.), *Feature Based Product Life-Cycle Modelling*

© IFIP International Federation for Information Processing 2003

(Bullinger and Warschat 1995). Feature modelling aims at providing ways to specify more design intent in a product model.

In feature modelling, the model of the product to be developed is built from features, which can be described as entities with some attributes and properties that are useful in the reasoning process of one or more applications (Shah and Mäntylä 1995).

Multiple-view feature modelling is a concept that combines concurrent engineering and feature modelling. It supports applications from various phases of product development, by providing an own interpretation of, or view on, the product for each of these applications. Each view contains a feature model of the product specific for the application, which is built from instances of generic features that are stored in the feature library of the view (Bronsvoort et al. 1997, De Martino et al. 1998, Hoffmann and Joan-Arinyo 2000).

The main shortcomings of the current approaches to multiple-view feature modelling are that they only deal with form features, i.e. regions of the product geometry that have some functional meaning, and hence support only the later product development phases in which the geometry has to be fully specified, and that they only deal with single parts, not with assemblies. In this paper, a new approach to multiple-view feature modelling is described that overcomes these shortcomings, by also providing support for conceptual and assembly design.

Section 2 describes some existing approaches to multiple-view form feature modelling. Section 3 presents the new enhanced multiple-view feature modelling approach. Sections 4, 5 and 6 describe the way conceptual design, assembly design, part detail design and part manufacturing planning are supported by the new approach. Section 7 presents the way the feature models of different views are kept consistent, e.g. when the feature model of one of them is changed, and Section 8 gives some conclusions.

2. MULTIPLE-VIEW FORM FEATURE MODELLING

Several approaches to managing multiple form feature views exist. With feature conversion, or to be more precise feature model conversion, a new feature model for some application is derived from an existing feature model for another application, and changes made in the feature model for one application are propagated to the feature models for other applications.

In one-way feature conversion, typically, the feature models of analysis and planning views are derived from the feature model of a design view. If the analysis or the planning results in the need to change the model of the

product, then these changes have to be made in the feature model of the design view, and the feature models of the other views have to be updated from that model.

Cunningham and Dixon (1988) propose a multiple-view system architecture with one-way feature conversion. In this proposal, the designer uses design by features to build the feature model of the design view, called the primary view, and feature conversion is used to derive the feature models of, for example, the finite-element view and the process planning view, called secondary views (see *Figure 1*).

Figure 1. Architecture with primary view and secondary views (after Cunningham and Dixon (1988)).

De Martino et al. (1994, 1998) describe a system for multiple-view feature modelling including one-way feature conversion. The feature models of the views are integrated by an intermediate model that contains, among other things, a representation of the product by means of so-called shape features, i.e. generic protrusions and depressions. The intermediate model is built by a shape-feature recogniser. A new feature model for a view is created from this model by an application-feature converter.

Suh and Wozny (1997) propose a multiple-view feature modelling system with one-way feature conversion using a graph-based intermediate model. The feature model for an application is recognised in the intermediate model by a hybrid feature recognition method that uses a graph-based technique to search features in the intermediate model, given the constraint graph of a feature class, and a hint-based technique to reduce the time that is needed to find a feature.

Bouras et al. (1999) describe a one-way feature conversion approach to create feature models of a product that are suitable for analysis applications. First, a morphological analysis is performed on the shape of the product to create a set of shape features. Subsequently features corresponding to a

chosen viewpoint are generated from these shape features. Finally, the feature model generated in this way is simplified and idealised to obtain the so-called working elements, which can be used in an analysis application.

However, support for feature conversion in a single direction only is far from ideal, because it requires an adjustment to a product model to be made in a feature model that may significantly differ from the feature model in which the need for the adjustment occurs. To allow a product model to be adjusted in the feature model of the view in which the need for it occurs, multiple-way feature conversion has been developed.

Bronsvort and Jansen (1993) propose multiple-way feature conversion as the ideal approach to multiple feature views. Later on, de Kraker et al. (1995, 1997), Bronsvort et al. (1997), Dohmen (1998) and de Kraker (1998) describe its implementation in a prototype multiple-view feature modelling system. They use a geometric-reasoning-based feature recognition algorithm that derives the feature model of a new view from a non-manifold intermediate cellular representation (Bidarra et al. 1998), and a special kind of coplanar constraints, called link constraints (Dohmen et al. 1996), to propagate changes made in the feature model of one view to the feature models of the other views.

Jha and Gurumoorthy (2000a, 2000b) describe two algorithms that can be used in multiple-way feature conversion. The first algorithm extracts a feature model from a geometric model, based on information whether a particular face of an instance of a feature class creates a new face in the feature model. The second algorithm propagates adjustments in one feature model to the other feature models, based on a special tree data structure that incorporates all feature models for a product.

Hoffmann and Joan-Arinyo (2000) propose an architecture in which the models for different applications are distributed over different CAx systems, and a master model is used to associate the models. This architecture effectively performs multiple-way feature conversion between feature views, at least if the models in these systems are feature models. Each of the systems is said to be a client of the master model server (see *Figure 2*).

The approaches to multiple feature views described above, which all use form features, share a number of shortcomings.

First, all approaches focus on the later product development phases, in which the geometry of the product has to be fully specified. They use the geometry of the product as a basis, and, therefore, cannot be applied in the early product development phases, such as conceptual design.

Second, all approaches deal with a single part only. Real products, however, rarely consist of a single part. Dealing with products that consist of multiple parts, i.e. assemblies, does not only involve the separate parts, but also the relations between these parts.

Figure 2. Clients manage their own private model, but can deposit some of their information in the model that is managed by the master model server (after Hoffmann and Joan-Arinyo (2000)).

To support the early product development phases and products with multiple parts, the enhanced multiple-view feature modelling approach has been developed.

3. ENHANCED MULTIPLE-VIEW FEATURE MODELLING

The enhanced multiple-view feature modelling approach can support applications from product development phases in which the geometry of the product has not been fully specified and phases that deal with multiple parts and the relations between them. Views have been developed to support the conceptual design phase, the assembly design phase, the part detail design phase and the part manufacturing planning phase. Some preliminary ideas on enhanced multiple-view feature modelling have been presented in (Noort and Bronsvoort 2000); here the approach is presented in much more detail.

The conceptual design view supports the conceptual design phase. Several approaches to conceptual design exist; three frequently used ones are shortly described. First, in the case-based reasoning approach, a conceptual design is based on a design for a similar product that is chosen from a set of existing designs, and is adapted to fit the requirements for the current product (Maher and Pu 1997). Second, in the sketching approach, a conceptual design is created from sketches that are made by the designer in some drawing application and converted into 3D geometry by the modelling system (Lipson and Shpitalni 2000). Third, in the configuration design approach, a conceptual design is created by specifying the components and

the way these are connected in the product (Wielinga and Schreiber 1997). The configuration design approach will be supported here.

The assembly design view supports the assembly design phase. Many approaches to assembly design exist. Some support all phases of product design that involve multiple parts, and provide several tools to analyse, for example, whether the components can be assembled (Zha et al. 2001), others focus on a specific aspect, such as the mate and contact relations between components (Whitney and Mantripragada 1998), or ways to represent complex assemblies (Heissermann and Mattikalli 1998). The assembly design approach that is supported here focuses on the detail design of the connections between the components (van Holland and Bronsvort 2000).

A part detail design view supports the detail design phase of a part. In this phase, the details of the geometry of the part are specified.

A part manufacturing planning view supports the planning of the way a part can be manufactured, taking into account the capabilities of the available manufacturing equipment.

The views are elaborated in Sections 4 to 6.

Each view has its own feature model of the product, with instances of feature classes from a view-specific library. Each class contains a set of feature constraints, which represent generic requirements from the development phase related to the view. Because only instances of the feature classes from the feature library of a view can be used to build a feature model for that view, each feature model of a view satisfies the generic requirements from the development phase related to that view. In addition to the constraints in feature instances, the feature model can also contain model constraints that represent specific requirements.

The views can be divided into a group of views that deal with all components of a product and the relations between them, i.e. the aggregate-oriented views, and a group of views that deal with a single part only, i.e. the part-oriented views. The conceptual design view and the assembly design view are the aggregate-oriented views, and the part detail design views and the part manufacturing planning views are the part-oriented views (see *Figure 3*). The part-oriented views on a part should represent the same part. The aggregate-oriented views should represent the same product, and represent the parts from the part-oriented views as (sub)components that are related. If the views satisfy these requirements, they are said to be consistent.

Consistency maintenance is elaborated in Section 7.

Figure 3. The relations between the part-oriented views and the aggregate-oriented views.

4. CONCEPTUAL DESIGN VIEW

The feature model of the conceptual design view is built from components and interfaces between components. In this way, the conceptual design view can support configuration design. The complete geometry of the components does not have to be specified.

A component in the conceptual design view consists of a base shape with shape concepts and references attached to it, and has attributes such as weight, volume, material type, and a description of its function. Each component fulfils a specific function within the product, which is specified by the designer by means of the shape concepts and the function description.

The base shape is used to position the shape concepts and the references of the component, and gives an impression of the geometry of the component. It is built from simple volumetric shape elements with block, vblock, cylinder or cone shape and one or more constraints to attach the element to the rest of the base shape. In addition, geometric constraints can be used to further position the shape elements with respect to each other.

Shape concepts are more or less abstract entities that represent a specific function of a component and have a shape and a type. The shape can be block, cylinder, cone, vblock or unknown. The type specifies the nature of the shape, i.e. whether it adds material to or removes material from the component, and the way the shape concept is positioned on the base shape or another shape concept. It can be aligned protrusion (Figure 4 (a)), isolated protrusion (Figure 4(b)), isolated depression (Figure 4(c) top), passage (Figure 4(c) bottom), open depression (Figure 4(d)), or unknown. The values for the dimensions of the shape and the faces of the other entities in

the model that are used to position the shape concept can also be specified, but this is not required. In addition, the function of the shape concept may be specified by means of a description.

Figure 4. Different types of shape concepts: aligned protrusion (a), isolated protrusion (b), isolated depression and passage (c), and open depressions (d).

References are used to position other model elements, such as shape concepts, and to relate interfaces to the component. They can be points, lines and faces.

Constraints can be used to restrict, for example, the value of the volume attribute of a component or the relative position of two shape concepts, and thus represent functional information. Different types of constraints are available, such as algebraic constraints that restrict attributes of components and shape concepts to be, for example, equal to some value or in some interval, and geometric constraints that restrict the position and orientation of shape concepts or references.

Interfaces between components in the conceptual design view represent the freedom between the components, i.e. how they can move relative to each other. Interfaces are specified with respect to references on components.

Pre-defined types of interfaces, such as hinge, are provided to the designer to be used in the model (see Figure 5). By default, interfaces only specify the type of freedom left. Optionally, the direction of the freedom may also be fixed by using the position and orientation of references on which the interface is specified. This approach again allows the designer to exactly specify what is required, but leave irrelevant details unspecified.

To show the feature model to the designer, geometry and relational graph cameras can be used.

Figure 5. Icons for different types of interfaces: fixed interface (a), prismatic interface (b), hinge interface (c), cylindrical interface (d) and spherical interface (e).

Geometry cameras show the base shape of the components, and can also show the shape concepts on the components and the references (see Figure 6(a)). Whether, and how, a shape concept is shown in a geometry camera depends on the amount of information that has been specified for the shape concept; it is not visualised if the type has been specified as unknown, its geometry is visualised if type, shape and all other parameters have been specified, and it is visualised by an icon based on the type and/or shape otherwise. Whether references are shown in a geometry camera, depends on the fact whether these have been selected for visualisation by the designer or not.

Figure 6. A geometry camera (a) and a graph camera (b) on the feature model of the conceptual design of a water pump.

Graph cameras show the components and the interfaces between them. A component is represented by a node that contains an image of the component. An interface is represented by a node that contains an icon for the interface (see Figure 5), and is connected to the nodes that represent the components that are related by the interface. An example of a graph camera is shown in Figure 6(b).

The model operations that can be performed on the feature model of the conceptual design view are adding, removing and editing components and interfaces. Editing components, for example, may involve changing the base

shape, the functional description, and adding, removing or editing shape concepts, references or constraints on the component. The values for the parameters of these model operations can be specified via a modelling panel or via selection in a camera.

An example of the use of the conceptual design view for the development of a modern version of the historical high-wheel bicycle (*Figure 7(a)*) is given here. The model of the conceptual design view represents the components of the product, with their important aspects only, and the interfaces between the components. *Figure 7(b+c)* show some cameras on the model.

Figure 7. The historical high-wheel bicycle (a), and some cameras on the model of a modern version of it: the base and the shape concepts of the front fork component in a geometry camera (b), and the interfaces between the components in a graph camera (c).

5. ASSEMBLY DESIGN VIEW

The feature model of the assembly design view consists of components and connection features between them, which together form assemblies. The assembly feature model can contain multiple unconnected assemblies and components, but no unconnected connection features.

Components are the elements of the assembly feature model that are combined, and on which the assembly information is defined. A component

is centred around a reference frame, i.e. all elements of a component are (in)directly specified relative to this frame. The components have the form features of the connection features that have been specified on them attached to them (see below), but no other form features.

A component is either a single component, which is based on a part, or a compound component, which is based on an assembly. A single component represents a part in the assembly model. A compound component encapsulates an assembly for further assembly modelling purposes, by hiding its internal structure of components and connection features, and dealing with the boundary of the assembly only.

Connection features are functional relations that represent assembly information, such as the internal freedom of the connection, the types of the form features needed for the connection on the components, and the way the connection can be established (van Holland and Bronsvoort 2000). Constraints are used to specify part of this assembly information. For example, in a dove-tail connection feature, contact constraints are used between faces of the dove-tail rib form feature and the dove-tail slot form feature to specify that the rib should stay in the slot. Other examples of connection features are a rib-slot and a pin-hole connection feature.

To show the feature model of the assembly design view, geometry cameras, relational graph cameras and hierarchical graph cameras may be created (Bronsvoort et al. 2002).

The geometry that is displayed in a geometry camera on a component consists of the so-called reference geometry, i.e. the geometry of the part or assembly related to that component, and in addition the geometry of the form features of the connection features on that component. The reference geometry is included to give a better insight into the complete geometry of the component, but is visualised without shading to emphasise that it is not part of the feature model for the component in the assembly design view. It is also used to enable a more intuitive way of creating form features of the connection features on the component: these need not to be specified with respect to the reference frame or other form features, but can also be specified with respect to the faces of the reference geometry.

Relational graph cameras display the connections between the components in an assembly (see *Figure 8(a)*).

Hierarchical graph cameras display the hierarchy of an assembly with its components (see *Figure 8(b)*), and optionally subcomponents of compound components (see *Figure 8(c)*). They use nodes containing an image of the component to represent components, and represent the hierarchy between compound components and their subcomponents by edges between their nodes.

Figure 8. Graph cameras showing a relational graph of an assembly (a), a hierarchical graph of the assembly (b), and the hierarchical graph including the subcomponents of one compound component (c).

The model operations that can be performed on the feature model of the assembly design view are creating and removing components, adding a connection feature between components, editing the parameters of a connection feature, and removing a connection feature. The values for the parameters of these model operations can be specified via a modelling panel or via selection in a geometry or graph camera.

Adding a connection feature between components requires the appropriate form features, e.g. a dove-tail rib and a dove-tail slot for a dove-tail connection, on these components. If these form features do not yet exist in the assembly design view, they have to be created. This results in the need for the feature models of the other views to be updated in order to keep all views consistent (see Section 7). Removing a connection feature also removes the related form features, at least if they are not used by any other connection feature: form features only exist in the assembly design view if they are related to a connection feature, otherwise they are removed.

An example of the use of the assembly design view for the high-wheel bicycle from the previous section is given here. The example shows the connection of the rear wheel to the rear fork with an additional axis component that is connected to the rear fork with a fixed pen-hole connection and to the wheel using a rotating pen-hole connection (see Figure 9).

Figure 9. The rear wheel of the bicycle is connected to the rear fork using an axis component.

6. PART DETAIL DESIGN AND MANUFACTURING PLANNING VIEWS

The form feature models of a part detail design view and a part manufacturing planning view are built from form features and additional model constraints (Bidarra and Bronsvoot 2000).

In a part detail design view or manufacturing planning view, the form features contain class-specific design information that is captured by means of feature elements and feature constraints. Feature elements are shapes and user-defined variables. Feature constraints can be, for example, a geometric distance face-face constraint, a dimension constraint, which specifies a dimension to be within a given range, and an on-boundary constraint, which specifies a feature face to be on the boundary of the part. Examples of form features are block protrusion and rounded blind slot.

Model constraints are used to specify additional functional information that is related to only one feature instance, and functional information that is related to several feature instances. They can be of the same types as the feature constraints.

To show the shape of the form feature model of a part detail design view or a manufacturing planning view, geometry cameras are used that combine several visualisation techniques (Bronsvoot et al. 2002). These cameras provide functional insight into the feature model by visualising all sorts of engineering information. This includes all features of a specific class to show all regions of the geometry with a similar function, intersections between features to show all regions of the geometry that have been specified more than once, and information on the fact whether a feature face

is on the boundary of the part or not to show all regions that can be formed by removing material.

The way this engineering information is visualised is defined by the user: several line visualisation and shading techniques may be used for it, in various combinations. An example of a geometry camera showing the feature faces of some selected features shaded, and the faces of the rest of the model with dashed-hidden-line visualisation, is given in *Figure 10*.

Figure 10. A geometry camera that combines several visualisation techniques to provide functional insight into a feature model.

The model operations that can be performed on the feature model of the view are adding a new form feature or model constraint, editing the parameters of an existing feature or constraint, and removing an existing feature or constraint.

After each operation, the validity of the model is checked. The model is no longer valid if one or more constraints are not satisfied. If an invalid situation is found, the operation is held up until the invalid situation has been removed by the user by additional, corrective model operations (Bidarra and Bronsvort 2000).

The values for the parameters of these model operations can be specified via a modelling panel or via selection in a geometry camera.

An example of the use of the part detail design and manufacturing planning view for a part of the high-wheel bicycle of Section 4 is given here. The example shows the detail geometry of the part that is used to fix the saddle to the frame, and the model of the manufacturing planning view of this part that can be used to determine the way the part can be manufactured (see *Figure 11*).

Figure 11. The model of the detail design view of the part that is used to fix the saddle (a), and the model of the part manufacturing planning view that is derived from this model (b).

7. CONSISTENCY MAINTENANCE

Consistency maintenance is used to update the feature model of a view in order to make it consistent with the feature model of another view. Because of the different natures of the views in the enhanced multiple-view feature modelling system, the feature model of a view cannot always be derived from the feature model of an arbitrary other view, and different techniques are needed for different pairs of views.

To support the use of the views described in the previous sections, consistency maintenance needs to be performed only for the source-target pairs that are indicated by an edge in *Figure 3*.

These pairs of views are a minimal set of pairs of views to maintain the consistency of the feature models of all views. This is important because the number of consistency maintenance steps that need to be performed should be minimised to reduce the number of consistency maintenance techniques required, and to optimise the performance.

Model elements in different views that should be kept consistent, can be related by a designer by specifying associations between them. For example, associations can be specified between a component in the conceptual design view and a component in the assembly design view, between a component in the conceptual design view and one or more parts in their detail design view, and between a shape concept of a conceptual component and a form feature of an associated assembly component or part.

Consistency maintenance between the conceptual design view and the assembly design view consists of keeping the components in the conceptual design view consistent with the associated components in the assembly design view, and keeping the interfaces in the conceptual design view

consistent with the associated connections in the assembly design view. Notice that some components in the conceptual design view may be associated to a compound component in the assembly design view, and as a result some (single) components in the assembly design view are only via a compound component associated to a component in the conceptual design view.

A component in the conceptual design view is consistent with a component in the assembly design view if each shape concept of the conceptual component that is associated to a form feature of the assembly component is consistent with that form feature. Notice that a shape concept in the conceptual design view does not have to be related to a form feature in the assembly design view, and vice-versa.

A shape concept is consistent with a form feature if the shape, type and dimensions of the form feature satisfy the constraints that have been specified on the corresponding attributes of the shape concept.

If a shape concept is inconsistent with the associated form feature, the designer has to adjust the shape concept or the form feature, or to remove the association.

For each interface in the conceptual design view, a connection feature in the assembly design view has to exist. An interface is consistent with a connection feature if the types of freedom that are reduced by the interface and the connection feature are equal, and the direction of this reduced freedom in the interface, if it has been specified, corresponds to the direction in the connection feature.

If an interface is inconsistent with the associated connection feature, the designer has to adjust the interface or the connection feature, or to remove the association.

Consistency maintenance between the conceptual design view and the part detail design views consists of keeping the components in the conceptual design view consistent with the associated parts in their detail design view. This involves keeping the shape concepts of the components consistent with the associated form features (see above), and keeping the attributes of the components consistent with the physical properties of the associated parts. Notice that the feature models of the detail design views on parts may contain additional form features that are not related to any shape concept.

The attributes of a component are consistent with the physical properties of the associated parts if, for example, the sum of the values of the volume attributes of the parts is within the range prescribed for the volume attribute of the component.

If the attributes of the component are inconsistent with the associated parts, the designer has to adjust the parts, or change the requirements on the component, e.g. the range for a value attribute.

Consistency maintenance between the assembly design view and the part detail design views consists of keeping each single component in the feature model of the assembly design view consistent with the feature model of the detail design view on the associated part.

This is the case if the faces of the form on the component that are intended to be on the boundary of the component, are on the boundary of the feature model of the associated part. Notice that faces of form features in the feature model of the detail design view on a part that are on the boundary of the part do not have to exist among the faces of the features in the feature model of the assembly design view.

Automatic feature recognition is used to make the feature model of a part in a detail design view that is inconsistent with its associated component in the assembly design view consistent again (de Kraker et al. 1995, 1997, Bronsvort et al. 1997). Feature linking is used to keep the feature models of the parts in the detail design views consistent with the associated components in the assembly design view during simple adjustment of the feature model of either. The system automatically creates links between the faces of the features from different views that are co-planar and on the boundary of the part or the component. A link consists of a geometric co-planar constraint (Dohmen et al. 1996).

Consistency maintenance between the detail design view and the manufacturing planning view of a part consists of keeping the feature model of the part detail design view consistent with the feature model of the part manufacturing view.

This is the case if they represent the same geometry.

Again automatic feature recognition is used to make the feature model of a part detail design view that is inconsistent with the feature model of the manufacturing planning view on the same part consistent again, and vice versa. Similarly, feature linking is again used to keep the feature model of the part detail design view consistent with the feature model of the part manufacturing planning view during simple adjustment of the feature model of either (see above).

An example of consistency maintenance of the feature models of different views on the high-wheel bicycle of Section 4 is given here. The example shows that the feature models of two part detail design views and the assembly design view are kept consistent after a change in the feature model of one detail design view; see *Figure 12*.

Figure 12. A change in the detail design view of a leg of the front fork, is propagated to the assembly design view of the associated component, and subsequently to the assembly design view of the connected component and the detail design view of the associated part.

8. CONCLUSIONS

This paper presented the enhanced multiple-view feature modelling approach, which is able to support a wider range of product development phases than previous multiple-view feature modelling approaches. In contrast to these approaches, the new approach also supports development phases in which the geometry of the product has not been fully specified, and product development phases that deal with multiple parts and relations between them.

Further, it allows the feature model of any view to be adjusted, while keeping all views consistent. The techniques that are used for this have been described.

A prototype system has been implemented with a conceptual design view, an assembly design view, part detail design views and part manufacturing planning views.

The feature models of the various views, which represent only those aspects of a product that are relevant in the associated product development phase, have been found to considerably increase the insight of the designer into the product model. The designer is not disturbed by aspects of the model that are irrelevant during that phase, but can concentrate on the aspects that are relevant.

Altogether, it can be concluded that the enhanced multiple-view feature modelling approach is capable of supporting a wider range of product development phases, and thus to better support product development.

REFERENCES

- Bidarra, R. and Bronsvoort, W. F. (2000). Semantic feature modelling, *Computer-Aided Design*, Vol. 32, No. 3, pp. 201-225.
- Bidarra, R., de Kraker, K. J. and Bronsvoort, W. F. (1998). Representation and management of feature information in a cellular model, *Computer-Aided Design*, Vol. 30, No. 4, pp. 301-313.
- Bouras, A., Belaziz, M., Tehari, A., Wahu, G. and Brun, J. M. (1999). A strategy to support application's dependent features interpretation, *Journal of Intelligent Manufacturing*, Vol. 10, No. 1, 41-47.
- Bronsvoort, W. F., Bidarra, R., Dohmen, M., van Holland, W. and de Kraker, K. J. (1997). Multiple-view feature modelling and conversion, In Strasser, W., Klein, R. and Rau, R. (eds.): *Geometric Modeling: Theory and Practice - The State of the Art*, pp. 159-174, Springer-Verlag.
- Bronsvoort, W. F., Bidarra, R. and Noort, A. (2002). Feature model visualization, To be published in: *Computer Graphics Forum*, Vol. 21, No. 4.
- Bronsvoort, W. F. and Jansen, F. W. (1993). Feature modelling and conversion - Key concepts to concurrent engineering, *Computers in Industry*, Vol. 21, No. 1, pp. 61-86.
- Bullinger, H. J. and Warschat, J. (1995). *Concurrent Simultaneous Engineering Systems*, Springer-Verlag.
- Cunningham, J. J. and Dixon, J. R. (1988). Designing with features: the origin of features, In Tipnis, V. A. and Patton, E. M. (eds.): *Proceedings of the 1988 ASME International Computers in Engineering Conference and Exhibition*, Vol. 1, pp. 237-243.
- De Martino, T., Falcidieno, B., Giannini, F., Haßinger, S. and Ovtcharova, J. (1994). Feature-based modelling by integrating design and recognition approaches, *Computer-Aided Design*, Vol. 26, No. 8, pp. 646-653.
- De Martino, T., Falcidieno, B. and Haßinger, S. (1998). Design and engineering process integration through a multiple view intermediate modeller in a distributed object-oriented system environment, *Computer-Aided Design*, Vol. 30, No. 6, pp. 437-452.
- Dohmen, M. (1998). *Constraint-based Feature Validation*, PhD Thesis, Delft University of Technology.
- Dohmen, M., de Kraker, K. J. and Bronsvoort, W. F. (1996). Feature validation in a multiple-view modeling system, In McCarthy, J. (ed.): *CD-ROM Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*.
- Heissermann, J. and Mattikalli, R. (1998). Representing relationships in hierarchical assemblies, In *CD-ROM Proceedings of the 1998 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*.
- Hoffmann, C. M. and Joan-Arinyo, R. (2000). Distributed maintenance of multiple product views, *Computer-Aided Design*, Vol. 32, No. 7, pp. 421-431.
- van Holland, W. and Bronsvoort, W.F. (2000). Assembly features in modeling and planning, *Robotics and Computer Integrated Manufacturing*, Vol. 16, No. 4, pp. 277-294.
- Huang, G. Q. (1996). *Design for X*, Chapman and Hall.
- Jha, K. and Gurumoorthy, B. (2000a). Multiple feature interpretations across domains, *Computers in Industry*, Vol. 42, No. 1, pp. 13-32.
- Jha, K. and Gurumoorthy, B. (2000b). Automatic propagation of feature modification across domains, *Computer-Aided Design*, Vol. 32, No. 12, pp. 691-706.
- de Kraker, K. J. (1998). *Feature Conversion for Concurrent Engineering*, PhD Thesis, Delft University of Technology.

- de Kraker, K. J., Dohmen, M. and Bronsvort, W. F. (1995). Multiple-way feature conversion to support concurrent engineering, In Hoffmann, C. M. and Rossignac, J. (eds.): *Solid Modeling '95, Third Symposium on Solid Modeling and Applications*, pp. 105-114.
- de Kraker, K. J., Dohmen, M. and Bronsvort, W. F. (1997). Maintaining multiple views in feature modeling, In Hoffmann, C. M. and Bronsvort, W. F. (eds.): *Solid Modeling '97, Fourth Symposium on Solid Modeling and Applications*, pp. 123-130.
- Lipson, H. and Shpitalni, M. (2000). Conceptual design and analysis by sketching, *AI-EDAM - Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 14, No. 5, pp. 391-401.
- Maier, M. and Pu, P. (1997). *Issues and Applications in Case-Based Reasoning in Design*, Lawrence Erlbaum Associates.
- Noort, A. and Bronsvort, W. F. (2000). Enhanced multiple-view feature modelling, In Brunet, P., Hoffmann, C. and Roller, D. (eds.): *CAD Tools and Algorithms for Product Design*, pp. 80-94, Springer-Verlag.
- Shah, J. J. and Mäntylä, M. (1995). *Parametric and Feature-based CAD/CAM*, John Wiley and Sons, Inc.
- Suh, Y. S. and Wozny, M. J. (1997). Interactive feature extraction for a form feature conversion system, In Hoffmann, C. M. and Bronsvort W. F. (eds.): *Solid Modeling '97, Fourth Symposium on Solid Modeling and Applications*, pp. 111-122.
- Whitney, D. E. and Mantripragada, R. (1998). The Datum Flow Chain: A systematic approach to assembly design and modeling, In *CD-ROM Proceedings of the 1998 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*.
- Wielinga, B. and Schreiber, G. (1997). Configuration-design problem solving, *IEEE Expert*, Vol. 12, No. 2, pp. 49-56.
- Zha, X. F., Du, H. J. and Qiu, J. H. (2001). Knowledge-based approach and system for assembly oriented design, part i: the approach, *Engineering Applications of Artificial Intelligence*, Vol. 14, No. 1, pp. 61-75.