Feature-Based High Level Design Tools
A Classification

Henri H. Achten, Jos P. van Leeuwen
Eindhoven University of Technology, The Netherlands

Key words: Features, Feature-Based modelling, architectural design, design process, design support

Abstract: The VR-DIS project aims to provide design support in the early design stage using a Virtual Reality environment. The initial brief of the design system is based on an analysis of a design case. The paper describes the process of analysis and extraction of design knowledge and design concepts in terms of Features. It is demonstrated how the analysis has lead to a classification of design concepts. This classification forms one of the main specifications for the VR-based design aid system that is being developed in the VR-DIS programme. The paper concludes by discussing the particular approach used in the case analysis and discusses future work in the VR-DIS research programme.

1.   DESIGN INFORMATION FOR ARCHITECTURAL DESIGN

Design information is used throughout every phase of the design process. It develops along with the design from high level abstract concepts such as symmetry, co-ordination, spaces, function, etc. to low level concrete concepts such as dimensioned window frames, light sockets, and finishes. Traditional product modelling has been successfully applied to the phase in the design process when the early, formative phase of design has been concluded. For an adequate support of early design phases, they have proven to be too rigid and set. In particular, product modelling approaches lack flexible definition of object types nor can these be extended when required.

Design information at the start of a design process is quite different from design information at the phase of preliminary design. Design, as a problem-solving process, involves activities of searching information, analysing, manipulating, and

1

structuring information, generating new information, and evaluating and communicating information. These are not sequential activities, but take place in cycles (Markus 1969), (Maver 1970), Roozenburg and Eekels (1995). (Lawson 1990) argues that designers tend to switch in an ad hoc manner between different activities, resulting in concurrency of activities with no predictable sequence. The dynamic nature of design should be supported by design aid systems.

In the Design Systems group at Eindhoven University of Technology, a research programme has been initiated called VR-DIS, meaning Design Information System and Distributed Interactive Simulation in Virtual Reality. The goals and projects of this programme have been reported in (de Vries et al. 1997). The possible advantages of the VR-DIS programme compared to conventional CAD systems are discussed in (Achten et al. 1998c). They propose that VR technology shows the best performance in the early design stage, using tools to create and evaluate (abstract) design models based on a three dimensional dynamic representation, and that it has the most potential in those areas where traditional CAAD has a poor performance.

In order to establish such a design support system, it is necessary to develop a functional brief. The performance specification should be formulated in terms that can easily be transformed computationally. Feature-Based modelling, an approach currently under investigation and development in the group, provides such a formalism. Research in the VR-DIS project on the possible uses of Feature-Based modelling in a VR environment concern conjoint measurement of user behaviour (Dijkstra et al. 1996; 1998; 1999; Dijkstra and Timmermans 1997a; 1997b; 1998), Case-Based Reasoning (Groot et al 1999; Mallory-Hill 1998; Achten et al. 1998d), and user interface (Coomans 1999).

The development of a design support tool in VR needs to be based on an understanding of the required tools and design information needs. For this purpose, a design case drawn from practice is studied. The case describes in terms of Features the steps in the design process represented by drawings. For this purpose, a description technique derived from previous research (Achten 1997a; 1997b; Achten et al. 1998b; Achten and van Leeuwen 1998) is extended and used for analysis of the drawings that are made during the design process. Each drawing is analysed and described in terms of Features. The transitions between drawings are also described in terms of Features. This description provides a formal basis for developing design tools that can be used in early design.

1.1 Feature-based modelling in the research context

Feature-Based modelling (FBM) originates from areas of Mechanical Engineering. The background and history of these techniques have been discussed and summarised in early papers by (Cunningham 1988), (Shah 1991; 1994), and (Bronsvoort 1993; 1996). FBM has been reviewed for its relevance to architectural design in (Van Leeuwen et al. 1996; 1997). The main conclusions from the latter reviews are that concepts of FBM are very relevant for modelling architectural information in a broader sense. In the VR-DIS programme, the following definition of the term Feature is employed (Van Leeuwen 1998a):

"A Feature is a collection of high-level information, possibly emerging during design, defining a set of characteristics or concepts with a semantic meaning to a particular view in the life-cycle of a building."

This definition reflects four important aspects of Feature modelling in the architectural context:

1. A Feature has high-level information with semantic meaning.
2. Both physical and non-physical characteristics and concepts can be defined.
3. Definition and use of emerging Features during design is supported.
4. A Feature relates to a particular view in the life-cycle of a building.

(Van Leeuwen 1998a) provides a Feature modelling framework for the development of information modelling systems for support of architectural design. The framework defines how Features are to be modelled. Feature models are flexible in that they support alteration of specific Feature Types during the design process. They are extensible through support of defining new Feature Types and Feature Instances. Also, it is possible to define relations between Feature Instances that have not yet been foreseen at the Feature Type level. This dynamic character of Feature modelling seems to be in accordance with the dynamic nature of design.

For Feature modelling, a tool has been developed in the group to define Feature Types and Feature Instances and to manipulate them. In time, Feature manipulation of the design is envisioned to be an integral part of the VR environment. Work by (Coomans and Achten 1998; Coomans and Timmermans 1998; Coomans 1999) is aimed towards this development. For now, the Windows-based Feature tool is used for Feature definition. Features can be represented in a graphical way (Van Leeuwen 1998a) or in a textual way, the Feature Type Definition Language (Van Leeuwen 1998b). In this paper, we will be using the textual representation.

2.   THE CASE STUDY

A case study has been done on an actual building design, which is analysed in terms of Features (Achten and van Leeuwen 1998a). It concerns an actual design executed by an architect's office. In the office, AutoCAD is used from the very start of the design work. Drawings during the design process are made as new copies rather than changing or revising old drawings. In this way, key phases of the design process are available for analysis.

The case study is based on 30 drawings made during the design process. Each single drawing is taken as a step in the design process for which a Feature model can be established. The transition from one step to the next therefore, represents the design decisions taken from that phase to the next.

− We define as a phase: one single drawing. Notation: phase n, $n \in N$.
− We define as a step: the transition from phase n to phase n+1.
− We define the brief as phase 0.

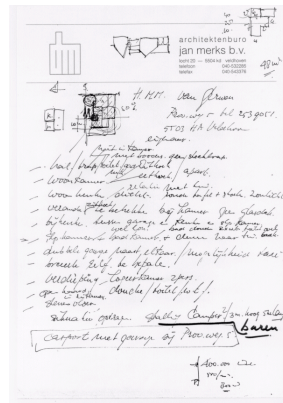The following figures represent the first six phases in the design case.
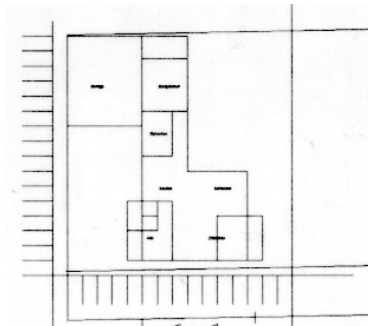
Figure 1. Phase 0 of the design case
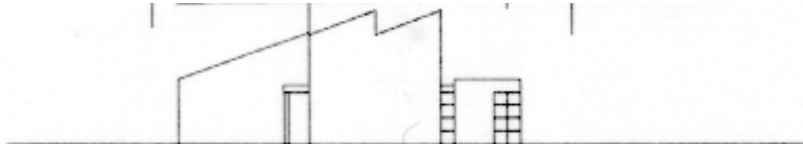

Figure 2. Phase 1 of the design case
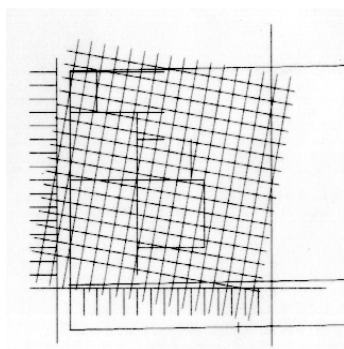

Figure 3. Phase 2 of the design case

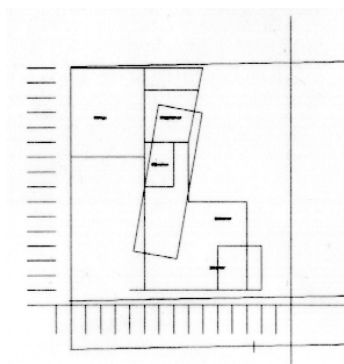*Figure 4.* Phase 3 of the design case

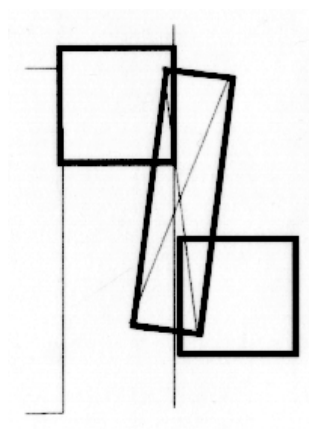*Figure 5.* Phase 4 of design case

Figure 6. Phase 5 of design case

The analysis of the design case is carried out as follows:

1. In phase n, identify the elements that have to be defined as Features. These elements are concepts used in the design, and typically are nouns in the text or graphic units in the drawings.
2. If the elements are new, define a complex Feature Type or simple Feature Type for each new element. An element is defined as a complex Feature Type when it cannot be defined as a simple Feature Type (string, integer, real, Boolean, or enumeration). If the element in question already has been defined as a Feature Type in any previous phase, record which changes have taken place and determine whether these should be included at the Type level or the Instance level.
3. In the case of new elements, establish Feature Instances based on the new Feature Types defined in step 2. In the case of existing Feature Types, record changes in Feature Instances. Proceed to phase n+1.

2.1 Example of Feature-Modelling in the Case

The brief of phase 0 provides a lexicon of design elements that play a role in the building design. In the example we will focus on the concept of space. Spaces such as "hall," "toilet," and "wardrobe" are elements of the lexicon. First, Feature Types have to be defined, after which Feature Instances can be made.

2.1.1 Feature Types

Rather than defining a Feature Type for each kind of space (such as a Feature Type Hall for the hall, Toilet for the toilet, etc.), we will define a Feature Type space of which the various spaces in the brief are instances.

The text in the brief notes the following aspects of a space:
− Function (such as "bedroom" and "bathroom").
− Contained elements (such as "stair" and "toilet").
− Visual relationship (such as "kitchen closed with respect to living").
− Access relationship (such as "doors to garden and bathroom").
− Daylighting (such as "daylighting in kitchen").
− Adjacency (such as "scullery between garage, kitchen and bathroom").
− Rooftype (such as "no glass roof").
− Number of persons (such as "guest room for two persons").

Determining which aspects are to be included in the definition of the Feature Type Space and which aspects are to be defined in other Feature Types is not straightforward. If the aspect concerns only the space itself, and does not refer to other elements, then it can be included in the type. Following this guideline, constraint-like relations such as visual relationship are better defined outside the Feature Type. Function, contained elements, daylighting, rooftype, and number of persons are within the particular space and therefore are included in the type definition. The Feature Type space is defined accordingly, and results in the following:

```
complex BuildingElement.space.Space {
   Has BuildingElement.space.Space contains[0..?];
   Spec User.value.Daylighting daylightIsUsed;
   Spec User.value.Function function;
   Spec BuildingElement.structure.Rooftype kindOfRoof;
   Spec User.value.NumberOfPersons numberOfPersons;
}
```

The first line identifies the Feature Type class, which is 'complex' in this case. The text 'BuildingElement.space.Space' is the Feature identification in the context of a Feature Type library.

The next five lines define the aspects of the Feature Type Space as identified above. They are the contained elements, daylighting, function, rooftype, and number of persons respectively. Each line has a three-part structure: relation, FeatureID, and role. Four of the relations are specifications since they further define the space. The "contains" relation is a decomposition since the contained spaces are part of the space. The FeatureIDs refer to Feature Types that are related to the Feature Type Space. Their definitions follow next. The role describes the role of the Feature in the definition. The numbers in brackets (for example '[0..?]') indicate cardinality of the relation: how many instances of this role are allowed or required in a Feature Instance.

In order to complete the Feature Type definition of space in this phase, the Feature Types User.value.Daylighting, Function, Rooftype, and NumberOfPersons must be defined as well.

```
string BuildingElement.structure.Rooftype {
}
boolean User.value.Daylighting {
   TypeDefault {True}
}
string User.value.Function {
}
integer User.value.NumberOfPersons {
   TypeDefault {1}
   TypeUnit {"person"}
}
```

The Feature Type Space is not only used to make Feature Instances of spaces such as hall and kitchen, but also in all other cases where in the brief of phase 0 a space is mentioned. This is the case for example, in the access relationship between spaces. The corresponding Feature Type is Constraint.access.Space_Space:

```
complex Constraint.access.Space_Space {
   Spec BuildingElement.space.Space access[2..2];
}
```

2.1.2 Feature instances

The Feature Types have to be instantiated into the particular Features Instances used in the design. In the case of Feature Type Space, this means making instances of spaces such as hall, living, kitchen, veranda, etc. We will provide an example of the Feature Instance living.

```
BuildingElement.space.Space Living = {
    contains[1] = Dining
    contains[0] = Sitting
    function = FunctionLiving
}
BuildingElement.space.Space Dining = {
function = FunctionDining
}
BuildingElement.space.Space Sitting = {
    function = FunctionSitting
}
User.value.Function FunctionDining = {
    Value {"Dining"}
}
User.value.Function FunctionLiving = {
    Value {"Living"}
}
User.value.Function FunctionSitting = {
    Value {"Sitting"}
}
```

In subsequent phases the Feature model is established on the basis of the previous phase, which enables to track changes during the design process.

2.2 Feature-Based Description of Phase 0

In this phase, the brief is translated into a Feature model. The brief consists of elements of the design (spaces and objects such as bathtub, toilets, etc.) and of relations between elements. The relations can be viewed as constraints. Some of these constraints concern spatial relations. These can be expressed as: A _adjacent_ B, A _in_ B, A _above_ B, and their logical opposites A _NOTadjacent_ B, A _NOTin_ B, A _NOTabove_ B, with A and B Feature Types.

Other constraints concern access from one element to another. These access constraints can be expressed as: A _access_ B and the opposite A _NOTaccess_ B. Since this relation is reciprocal, only one _access_ constraint has to be defined.

The third type of constraint is expressed in the statement 'kitchen is closed with respect to the living,' meaning that there is no visual relation between the kitchen and living space. The constraints A _visual_ B and the opposite A _NOTvisual_ B are reciprocal, and can be defined in the same manner as the _access_ relation.

The 'fireplace in living' constraint can be expressed in two ways: as a Fireplace_in_Living instance of a Heating_in_Space constraint Feature Type, or by establishing an association-relation in the Feature Instance Living with the Feature instance Fireplace. We have chosen here for the first option, since heating elements are bound to occur in more spaces.

The tables below show the Feature Types and Feature Instances for elements and constraints respectively.

Table 1. Feature Types and Feature Instances of elements in phase 0

Feature Type (super type)	Feature Instance
Space	Hall, Toilet, Wardrobe, Living, Sitting, Dining, Kitchen, Veranda, Scullery, Garage, Bedroom, Bathroom, Shower, GuestRoom
Door (ElementInWall)	DoorBathroom_Bedroom, DoorBathroom_Garden
Floor	FloorLiving
Material	MaterialFloorcovering, MaterialRoofVeranda
Roof	RoofVeranda
Stair	Stair
Garden	Garden
Chair (Furniture)	Chair
Table (Furniture)	Table
Fireplace (Heating)	FireplaceLiving
Bathtub (Sanitary)	BathtubBathroom
ToiletPot (Sanitary)	ToiletPotHall, ToiletPotGuestroom, ToiletPotBathroom
WashBasin (Sanitary)	WashBasin1_Bathroom, WashBasin2_Bathroom, WashBasinGuestroom
Daylighting	DaylightingBedroom, DaylightingKitchen, DaylightingVeranda
Function	FunctionBedroom, FunctionHall, FunctionDining, FunctionKitchen, FunctionSitting, FunctionLiving
Storey	StoreyGroundFloor, StoreyFirstFloor
NumberOfPersons	NumberOfPersonsGuestroom

Table 2. Feature Types and Feature Instances of constraints in phase 0

Feature Type (constraint type)	Feature instance
Space_adjacent_Space (spatial)	Kitchen_adjacent_Living, Veranda_adjacent_Living, Scullery_adjacent_Garage, Scullery_adjacent_Bedroom, Scullery_adjacent_Kitchen
Space_adjacent_Garden (spatial)	Kitchen_adjacent_Garden, Bedroom_adjacent_Garden
Storey_above_Storey (spatial)	Storey1_above_Storey0

Feature Type (constraint type)	Feature instance
Furniture_NOTin_Space (spatial)	Furniture_NOTin_Kitchen
Heating_in_Space (spatial)	Fireplace_in_Living
Stair_NOTin_Space (spatial)	Stair_NOTin_Living
Space_Space (Access)	Bedroom_access_Bathroom
Space_Garden (Access)	Bedroom_access_Garden,
	Kitchen_access_Garden
Space_NOTvisual_Space (visual)	Kitchen_NOTvisual_Living

2.3   Feature-Based Description of Phase 1

In phase 1, the brief is transformed to a set of spaces in a drawing. Significant changes with respect to phase 0 consist of assigning shapes and their dimensions to spaces, and locating them in the site by means of a grid.

The notion of shape can be implemented as part of the existing Feature Type Space or by defining a new Feature Type for shape, which is associated with the Feature Type Space. When considering phase 1 only, the first option would suffice. However, the notion of shape is very basic in architectural design, and many other kinds of shapes may occur later in the design, each with their own intrinsic properties. Therefore, we have chosen to define a supertype 2DShape of which Rectangle is a subtype. For the definition of a rectangle and its position in a drawing, the dimensions and position need to be defined as Feature Types as well. Therefore, Length, Point, and Coordinate also are new Feature Types:

```
complex Geometry.shape.2DShape {
   TypeDescr {"General shape definition with point of reference"}
   Spec Geometry.topology.Point referencePoint[1..1];
}
complex Geometry.shape.Rectangle(Geometry.shape.2DShape) {
   TypeDescr {"Rectangular shape with dimensions and reference point"}
   Spec Geometry.dimension.Length length[1..1];
   Spec Geometry.dimension.Length width[1..1];
}
real Geometry.dimension.Length {
   TypeDescr {"Linear dimension in m"}
   TypeDefault {1}
   TypeUnit {"m"}
}
complex Geometry.topology.Point {
   TypeDescr {"Point in orthogonal axial system with x, y, z co-ordinates"}
   Spec Geometry.topology.Coordinate xcoordinate[1..1];
   Spec Geometry.topology.Coordinate ycoordinate[1..1];
   Spec Geometry.topology.Coordinate zcoordinate[1..1];
}
real Geometry.topology.Coordinate {
   TypeDescr {"Coordinate along an axis in an axial co-ordination system"}
   TypeDefault {0}
```

```
    TypeUnit {"m"}
}
```

The Feature Type 2DShape is associated with the existing Feature Type Space (bold line shows addition to old Feature Type):

```
complex BuildingElement.space.Space {
TypeDescr {"Space element within which activities can take place"}
    Spec BuildingElement.space.Space contains[0..?];
    Spec User.value.Daylighting daylightIsUsed[1..1];
    Spec User.value.Function function;
    Has BuildingElement.structure.Rooftype kindOfRoof;
    Spec User.value.NumberOfPersons numberOfPersons;
    Assoc Geometry.shape.2DShape shape;
}
```

The shapes are drawn in a grid which is used for co-ordination. The grid can be defined by stating its module and a point of origin relative to which co-ordinates are established (this also accommodates the use of multiple grids). Both the origin of the grid and the position of elements in the grid require a set of co-ordinates. We define therefore, on the instance level, a Feature Instance Origin (instance of Geometry.topology.Point) relative to which measures can be taken and grids positioned. The left-bottom corner of Grid_1 is placed on the Origin.

```
complex Geometry.Topology.Grid {
    Descr {"Origin and module of a grid"}
    Spec Geometry.topology.Point originOfGrid[1..1];
    Spec Geometry.dimension.Length moduleOfGrid[1..1];
}
```

The positive x-axis is oriented horizontally and to the right of the Origin. The positive y-axis is oriented vertically and above the Origin, as is customary in architectural design. For the Feature Type Rectangle, the reference point is defined as the most left-bottom corner of the rectangle, width and length being measured in the orientation of the positive x and y axis.

These Feature Types and Feature Instances suffice to describe the state of phase 1. The Feature Instance Kitchen, for example, is changed because of the additional association to the Feature Type Space of which it is an instance, and the definition of its location and dimensions in the associated Rectangle. In phase 1, the Kitchen has dimensions 3.60 m x 3.60 m, and is located on co-ordinates (6.0, 6.0, 0):

```
BuildingElement.space.Space Kitchen = {
    Descr {"Kitchen"}
    daylightIsUsed = DaylightingKitchen;
    function = FunctionKitchen;
    shape = Rectangle_Kitchen;
}
Geometry.shape.Rectangle Rectangle_Kitchen = {
    Descr {"Rectangular shape for kitchen"}
```

```
        length = Length_Kitchen;
        referencePoint = ReferencePoint_Kitchen;
        width = Width_Kitchen;
    }
    Geometry.dimension.Length Length_Kitchen = {
        Value {3,6}
    }
    Geometry.dimension.Length Width_Kitchen = {
        Value {3,6}
    }
    Geometry.topology.Coordinate Coordinate_X_Kitchen = {
        Value {6}
    }
    Geometry.topology.Coordinate Coordinate_Y_Kitchen = {
        Value {6}
    }
    Geometry.topology.Coordinate Coordinate_Z_Kitchen = {
        Value {0}
    }
```

In phase 1 not all spaces mentioned in phase 0 are present, and there are four spaces that have not been assigned a function name by the architect. The older instances of spaces that are included in phase 1 change in the same manner as the kitchen example. Space_0 through space_4 are instantiated directly according to the new Feature Type definition.

3.   A CLASSIFICATION OF DESIGN CONCEPTS

The description of the design process in the case studies provides a new way to look at design processes. In particular, changes from one phase to the next can be expressed in terms of changes in the Feature model. In this way, design actions can be matched to Feature model alterations.

The changes in the Feature model are based on the case study. The descriptions of the Feature model alterations are very specific for the case. Therefore, it is necessary to classify them into more general descriptions of design actions and associated changes in the Feature model. The classification provides the proper vocabulary for establishing the functional brief for the VR-DIS system. The following table presents the classification and the definition of the terms for changes in the Feature model.

Table 3. Design actions and changes in the Feature model

Design action	Changes in the Feature model
Generalisation	When a group of objects share common properties, define the specific objects as Feature Types, and define a Feature Type (super type) of which they are sub types. The super type functions as generalisation.

Design action	Changes in the Feature model
Concept identification	Terms in the brief that are relations or spatial-, material-, and functional elements, are defined as Feature Types.
Element creation	Terms in the brief that are actual parts in the design ("hall", "floor", "fireplace") can be instantiated directly on the basis of the corresponding Feature Types.
Constraint creation	Terms in the brief that are relations in the design can be instantiated on the basis of constraint Feature Types.
Concept extension	Adding an association relation to a Feature Type in order to include more characteristics.
Shaping	Giving shape to the spatial elements involves element creation of the Feature Types Shape and of Feature Types position and dimension.
Assignment	On the Instance level make an association relation between Feature Instances.
Move	Move means that the co-ordinates that define position have been changed in a Feature Instance.
Substitution	Substitution means that an existing association between Feature Instances is broken and that one of the Feature Instances is replaced.

4.   DISCUSSION

In the case, nine design actions have been identified and described in terms of changes in the Feature model. They are either on Feature Type level only (generalisation, concept identification, and concept extension), on the Feature Instance level only (element creation, constraint creation, assignment, move, and substitution), or a mix of both levels (shaping).

In the current approach, the Feature model was constructed on the basis of the case itself without using predefined Features. The 'dynamics' of the design process can be described in a clear way. In a design support environment however, it is not likely that each design task will start with an 'empty' Feature model, and that the whole has to be constructed through the design. Pre-defined Feature Types will probably be used extensively. The Feature Types that are stock and trade in the building and construction industry can be defined in libraries beforehand and used in any design (so-called "generic Feature Types"). Feature Types that are specific for a design project can be easily created or modified (so-called "specific Feature Types").

The classification can be used to define the design functionality of the VR-DIS system in terms of changes in the Feature model. The operations generalisation, concept identification, concept extension, element creation, constraint creation,

14

assignment, move, substitution, and shaping are defined as changes in the Feature model. Since the functionality of the VR-DIS system is going to be based on Features, this description will facilitate development of tools that support the design actions.

In some of the design actions changes are made in the Feature Type. In a design system using Feature models, this raises the problem of consistency in the model. The changes in the Feature Type have to be propagated through all the relevant instances, and it has to be checked whether this creates inconsistencies or not. The theory of Feature-based modelling as developed does not provide this test, but leaves this responsibility to the user of the Feature data.

Two prototypes have been developed (De Vries and Jessurun 1998a; 1998b) that solve the constraints defined as Features in the model (in phase 0). Furthermore, they keep the constraints solved in real-time, so that changes in the VR-environment show immediate feedback in the behaviour of the spaces.

In the case study, geometry is defined in complex Feature Types for points, dimensions, and shapes. The theoretical framework for Feature-Based modelling now includes geometric Feature Types, which can be used for modelling shapes.

The Feature-Based modelling approach used in the research is aimed to support the early stages of design. The stages in design have been defined in Roozenburg and Eekels (1995) with the "basic design cycle". The basic design cycle encompasses all the phases a design process will go through at least once. It consists of a number of activities, each leading to a result. The cycle starts with a function, which is analysed, resulting in criteria; upon which synthesis takes place, resulting in a provisional design; that is simulated, resulting in expected properties; which are evaluated, resulting in an outcome; upon which a decision is taken whether or not to continue (Roozenburg and Eekels 1995, p. 88-92).

It is clear that in each activity, information of the design will be put to different uses. An underlying Feature model therefore, can have various uses. For example, the statement "scullery between garage, kitchen and bathroom" is information resulting from the analysis activity. In a provisional design, it may appear that the scullery is located elsewhere, which is another piece of information that is in contradiction with the previous one. The first piece of information states a constraint, and the second a state of affairs. Future work will have to focus on these different uses and requirements of the Feature model.

5.   ACKNOWLEDGEMENTS

6. REFERENCES

Achten, H.H. 1997a, "Generic Representations - Typical Design Without the Use of Types", Junge (ed.), *Proceedings of the 7th International Conference on Computer Aided Architectural Design Futures*, 117-133, (Kluwer, Dordrecht).

Achten, H.H. 1997b, *Generic Representations - An Approach for Modelling Procedural and Declarative Knowledge of Building Types in Architectural Design*, Ph.D. Thesis, Eindhoven University of Technology, Bouwstenen 46.

Achten, H.H., and J.P. van Leeuwen, 1998a, "A Feature-based Description Technique for Design Processes: a Case Study", Timmermans (ed.), Proceedings of the 4th conference on Design and Decision Support Systems in Architecture and Urban Planning.

Achten, H.H., and R.M. Oxman, and Th. Bax, 1998b, "Typological Knowledge Acquisition Through a Schema of Generic Representations", Gero and Sudweeks (eds.), Artificial Intelligence in Design '98, 191-208. (Kluwer, Dordrecht).

Achten, H.H., and Jessurun, J., and Leeuwen, J.P. van, and Vries, B. de, 1998c, Virtual Reality Design Information System/Distributed Interactive Simulation, Design Systems Report 1998/1, Eindhoven.

Achten, H.H. and Mallory-Hill, S.M. and Vries, B. de, 1998d, Workshop Case Based Reasoning and Design, Design Systems Report 1998/2, Eindhoven.

Bronsvoort, W.F. and Jansen, F.W. 1993, "Feature Modelling and Conversion, Key Concepts to Concurrent Engineering", *Computers in Industry*, 21(1), 61-86.

Bronsvoort, W.F. and Dohmen, M. and Bidarra, R. and Van Holland, W. and De Kraker, K.J. 1996, "Feature Modelling for Concurrent Engineering", Proceedings of the International Symposium on Tools and Methods for Concurrent Engineering '96, Budapest, 46-55.

Coomans, M.K.D. and Achten, H.H. 1998, "Mixed Task Domain Representation in VR-DIS", Proceedings of the 3rd Asia-Pacific Conference on Computer Human Interaction, APCHI98, Hayama-machi, Kanagawa, 1998.

Coomans, M.K.D. and Timmermans, H.J.P. 1998, "A VR User Interface for Design by Features", Timmermans (ed.) Proceedings of the 4th Conference on Design and Decision Support Systems in Architecture and Urban Planning, Maastricht.

Coomans, M.K.D. 1999, "A Virtual Reality User Interface for a Design Information System", *CCAI : the Journal for the Integrated Study of Artificial Intelligence, Cognitive Science and Applied Epistemology*, Rijks Universiteit Gent.

Cunningham, J.J., and J.R. Dixon 1988, "Designing with Features: the Origin of Features", Proceedings ASME Computers in Engineering Conference, San Francisco.

Dijkstra, J., Roelen, W.A.H. and Timmermans, H.J.P. 1996, "Conjoint Measurement in Virtual Environments: a Framework", Timmermans (ed.), 3rd Design and Decision Support Systems in Architecture and Urban Planning Conference, Vol I: Architecture Proceedings, 59-71.

Dijkstra, J. and Timmermans, H.J.P. 1997a, "Exploring the Possibilities of Conjoint Measurement as a Decision-Making Tool for Virtual Wayfinding Environments", Proceedings of The Second Conference on Computer Aided Architectural Design Research in Asia, 61-71.

Dijkstra, J. and Timmermans, H.J.P. 1997b, "The Application of Conjoint Measurement as a Dynamic Decision Making Tool in a Virtual Reality Environment", *Proceedings of the 7th International Conference on Computer Aided Architectural Design Futures*, 757-770. (Kluwer Dordrecht).

Dijkstra, J., Timmermans, H.J.P. and Roelen, W.A.H. 1998, "Eye Tracking as a User Behavior Registration Tool in Virtual Environments", Proceedings of The Third Conference on Computer Aided Architectural Design Research in Asia, 57-66.

Dijkstra, J. and Timmermans, H.J.P. 1998, "Conjoint Analysis and Virtual Reality – A Review", Timmermans (ed.), 4th Design and Decision Support Systems in Architecture and Urban Planning Conference.

Dijkstra, J., Timmermans, H.J.P. and Roelen, W.A.H. 1999, "Conjoint Analysis and Virtual Reality – Exploring the Possibilities", *Design Systems Reports*, Design System Report 1999/1, TUE.

Groot, E. de and Mallory-Hill, S.M. and Zutphen, R.H.M. and Vries, B. de, 1999, "A Decision Support System for Preliminary Design", Proceedings of the 8th International Conference on Durability of Building Materials and Components, Vancouver (forthcoming).

Leeuwen, J.P. van and Wagter, H. and Oxman, R.M. 1996, "Information Modelling for Design Support". Proceedings of the 3rd Design and Decision Support Systems in Architecture and Urban Planning Conference, Spa.

Leeuwen, J.P. van, and H. Wagter, 1997, "Architectural Design-by-Features". Proceedings of CAAD Futures '97, 97-115. (Kluwer, Dordrecht).

Leeuwen, J.P. van, and H. Wagter, 1998, "A Features Framework for Architectural Information: a Case Study". Gero and Sudweeks (ed.), Artificial Intelligence in Design '98. (Kluwer, Dordrecht).

Mallory-Hill, S.M., 1998, "Building a Case-Based Design Assistant for Workplace Environment Design", Timmermans (ed.), 4th Design and Decision Support Systems in Architecture and Urban Planning Conference.

Markus, Th.A. 1969, "The Role of Building Performance Measurement and Appraisal in Design Method", Broadbent and Ward (eds.), *Design methods in architecture*, (Lund Humphries, London).

Maver, Th.W. 1970, "Appraisal in the Building Design Process", Moore (ed.), *Emerging Methods in Environmental Design and Planning*, MIT Press, Cambridge, Massachusetts.

Roozenburg, N.F.M. and Eekels, J. 1995, *Product Design: Fundamentals and Methods* (John Wiley & Sons, Chichester).

Shah, J.J. and Mäntylä, M. and Nau, D.S. (eds.) 1994, *Advances in Feature Based Manufacturing* (Elsevier Science, Amsterdam).

Shah, J.J 1991, "Assessment of Features Technology", *Computer-Aided Design*, 23(5), 331-343.

Vries, B. de and Leeuwen, J.P. van and Achten, H.H. 1997, "Design Studio of the Future", Proceedings of the CIB W78 Conference Information Technology Support for Construction Reengineering, Cairns, Australia, 139-148.

Vries, B. de and Jessurun, A.J. 1998a, "An Experimental Design System for the Very Early Design Stage", Timmermans (ed.) Proceedings of the 4th Conference on Design and Decision Support Systems in Architecture and Urban Planning, Maastricht.

Vries, B. de and Jessurun, A.J. 1998b, "Features and Constraints in Architectural Design", Proceedings of DETC'98, 1998 ASME Design Engineering Technical Conference, Atlanta, Georgia.