

Towards generic representations of designs formalised as features

Henri Achten and Jos van Leeuwen

Eindhoven University of Technology
Department of Architecture, Building and Planning
Eindhoven
The Netherlands

ABSTRACT

Feature-Based Modelling (FBM) is an information modelling technique that allows the formalisation of design concepts and using these formal definitions in design modelling. The dynamic nature of design and design information calls for a specialised approach to FBM that takes into account flexibility and extensibility of Feature Models of designs. Research work in Eindhoven has led to a FBM framework and implementation that can be used to support design. Feature models of a design process has demonstrated the feasibility of using this information modelling technique. To develop the work on FBM in design, three tracks are initiated: Feature model descriptions of design processes, automated generic representation recognition in graphic representations, and Feature models of generic representations. The paper shows the status of the work in the first two tracks, and present the results of the research work.

1 INTRODUCTION

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 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 (de Vries and Achten 1998). 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 a design support system in VR, it is necessary to have an information model that can capture both the data of the design in the system, and that is flexible and extensible enough to capture the changing nature of design. The information model can also be used to define the functionality of the design support system. Feature-Based modelling (FBM), an approach under investigation and development in the Design Systems group (van Leeuwen 1996; 1997; 1998; 1999), provides such a formalism for design information. Capturing the design decisions in a design process is studied on the basis of generic representations, a methodology for analysing graphic representations in design (Achten 1997a; 1997b; 1998). In order to understand design processes for the development of the design support system, three tracks are developed:

1. A Feature model description of a concrete design case (Achten and van Leeuwen 1998; Achten and van Leeuwen 1999). This work provides insight in the dynamics of design from the perspective of Feature models.
2. Automated recognition of generic representations (Achten 2000). This work provides a means to understand the content of graphic representations in design.
3. Feature model description of generic representations. This work provides a formal description of generic representations in design.

In this paper, we describe the first tracks that have been established so far in the research of the past two years.

2 GENERIC REPRESENTATIONS AND FEATURE-BASED MODELING

Generic representations and Feature-Based models form the main foundation of the work. Both approaches are introduced briefly below.

2.1 Generic representations

Generic representations are a means to understand the content of graphic representations. Graphic representations vary to a great extent in their appearance. A survey of historical examples (Achten 1997a, p. 15-21) yields the following observations:

- Under the assumption of a particular convention of depiction (plan, perspective, section, etc.) a graphic representation does not need extensive textual elaboration in order to produce a correct interpretation. This means that the constituent graphic elements of the graphic representation provide strong clues about the interpretation,

and that these elements do not change very much over a long period of time (Medieval drawings for example, are still intelligible).

- In a graphic representation, the identified elements are not the most basic elements (vertices, lines, planes, etc.) but aggregates of these elements with a particular interpretation. For example, a closed polygonal shape with constant thickness and particular hatching indicates a wall, a closed filled-in circle indicates a column, or a set of lines and circles indicates a vault system.
- An architectural graphic representation that makes sense in a particular convention of depiction and encoding presents a feasible and well-balanced whole. In a well-constructed graphic representation a number of design conflicts between elements are solved.

On the basis of the observations stated above, it is proposed that it is possible to classify graphic representations on the basis of their constituent elements to determine what design decisions are involved in that particular graphic representation. A constituent element is called a graphic unit. It is a set of graphic entities that has a specific meaning such as grid, axis, or shape. This combination of form and interpretation is crucial to the concept. Each interpretation has different implications in a design context.

Graphic representations that have the same graphic units fall under the same generic representation, however varied their appearance may be. Graphic units and generic representations have been identified in an analysis of over 200 graphic representations (Achten 1997a).

24 Graphic units have been identified, such as (1) *simple contour*, (2) *contour*, (3) *measurement device*, and (4) *specified form*. It has been found that up to four graphic units constitute generic representations. 50 Generic representations have been identified, such as (15) contour in grid, (28) element vocabulary in grid, and (41) schematic subdivision in grid and refinement grid (see Achten 1997a; 1997b; 1998) for a more elaborate discussion of graphic units and generic representations).

2.2 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 (Bronsvort 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 (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.

2.3 Feature models of design processes

To understand the dynamics of design, and the required design support, 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 and van Leeuwen 1998; Achten and van Leeuwen 1999) 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, and they have been classified. This description provides a formal basis for developing design tools that can be used in early design.

The case study has been described in detail in the papers mentioned above. We will give an example of the Feature Type space of which the definition (e.g. living room, kitchen, garage) changes during the design process. In the first step of the case study, the Feature Type space is described as follows:

```

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 it is used in the first step of the case study. They are the contained spaces, 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.

The Feature Type Space has instances in the design. In the first step of the case study, the living room for example, is described as follows:

```

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 the second step of the case study, geometry is added to the concept of space as the architect makes a first schematic of the spaces that are in the brief of his assignment.

This means that the Feature Type Space is extended with geometry:

```

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;
}

```

If we look at another space, kitchen, then the change in the Feature instance is as follows:

```
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}
}
```

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.

2.4 A feature-based classification of design concepts

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 following table presents the classification and the definition of the terms for changes in the Feature model. 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).

Table 1: **Design actions and changes in the Feature model**

DESIGN ACTION	CHANGES IN 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.
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.

The FBM 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, p. 88-92) with the “basic design cycle”. 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.

3 DERIVATION OF GENERIC REPRESENTATIONS IN A DRAWING

The analysis of the case study described above is based on finding graphic units and generic representations in the drawings that document the steps of the design process, and make Feature models of these descriptions. At the moment, this analysis is done by hand by the researcher. We are aiming to automate this process for the following reasons:

1. To derive a more rigorous description of the analytical process.
2. To further test the consistency of the description of generic representations and graphic units.
3. To make faster analysis of design cases.
4. To acquire Feature-Based descriptions of design cases.

At the moment, we have developed a user-assisted approach to identify graphic units in a graphic representation, and are working on an architecture for automated graphic unit recognition. The user-assisted approach essentially is a decision tree that by questions leads the designer to identify graphic units. The architecture for automated graphic unit recognition should provide the basis for developing a system that can replace the user-assisted approach. The third step that would be required, description of the identified graphic units and generic representations in Feature models is not addressed yet.

3.1 A decision-tree for identifying graphic units

The decision-tree is a question-answer mechanism that leads to identification of a graphic unit in a drawing. The nodes are not bifurcal in order to decrease the number of questions that need to be answered. The process starts when the designer prompts the system for case retrieval. Each pass through the tree identifies one graphic unit (Figure 1).

Each node in the decision-tree either is a question or an identified graphic unit. The questions are identified with capital letters A, B, C, etc. The graphic units are identified with their numbers. At each decision point, a schematic drawing clarifies the decision that has to be taken. Textual clarification is also available, but for brevity's sake this is not included. Below follows the list of questions in the decision-tree.

A. Is it a graphic or symbolic element?

- If graphic element, go to B.
- If symbolic element, graphic unit is (7) function symbols.

B. Is it a closed shape or a set of one or more lines?

- If a closed shape, go to C.
- If a set of one or more lines, go to D.

C. Is it a single shape or multiple shapes?

- If multiple shapes, go to E.

- If single shape, go to F.

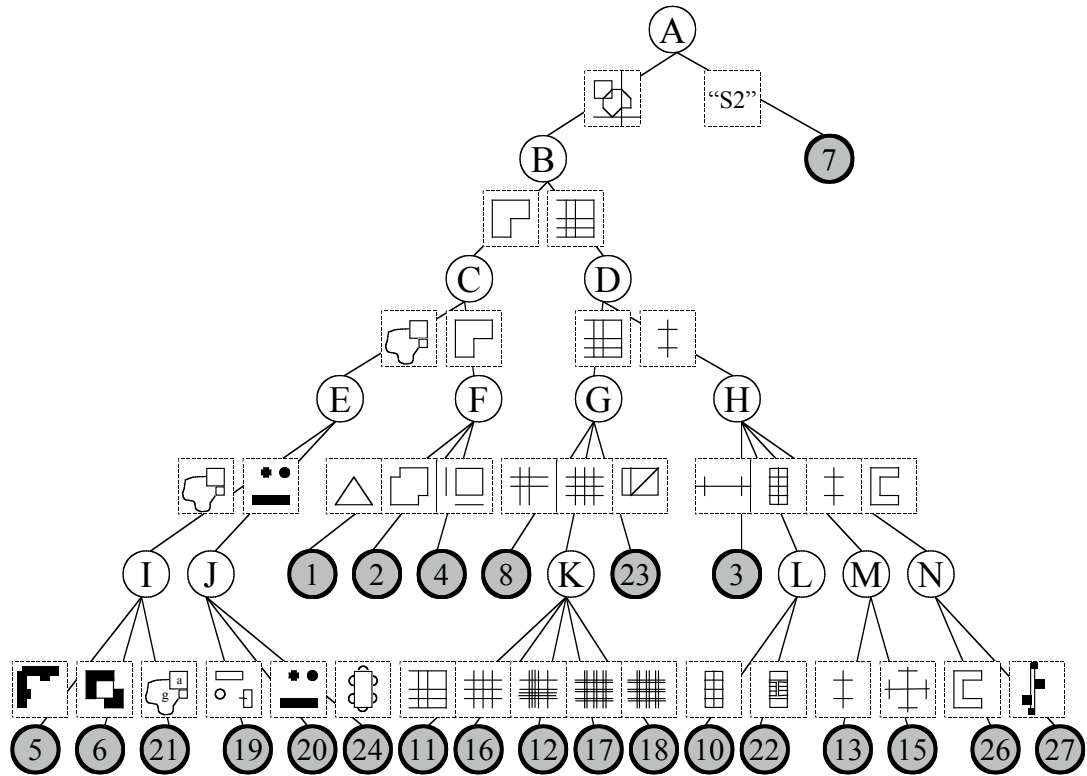


Figure 1: The decision-tree for identifying graphic units in a graphic representation

D. Is it a coordinating system or not?

- If coordinating system, go to G.
- If not a coordinating system, go to H.

E. Is it spaces or building elements?

- If spaces, go to I.
- If building elements, go to J.

F. Is it a Platonic shape, a more irregular shape, or a shape that is specified?

- If Platonic shape, graphic unit is (1) simple contour.
- If irregular shape, graphic unit is (2) contour.
- If specified shape, graphic unit is (4) specified form.

G. Is it a zone, grid, or proportion system?

- If zone, graphic unit is (8) zone.
- If grid, go to K.
- If proportion system, graphic unit is (23) proportion system.

H. Is it for measurement, subdivision, composition, or circulation?

- If measurement, graphic unit is (3) measurement device.
- If subdivision, go to L.

- If composition, go to M.
- If circulation, go to N.

I. Is the contour elaborated and showing the structure, or are the contours complementary, or do the spaces have functional indications?

- If contour is elaborated, graphic unit is (5) elaborated structural contour.
- If the contours are complementary, graphic unit is (6) complementary contours.
- If the spaces have functional indications, graphic unit is (21) functional space.

J. Is it a set of elements, structural elements, or rules for elements?

- If set of elements, graphic unit is (19) element vocabulary.
- If structural elements, graphic unit is (20) structural element vocabulary.
- If rules for elements, graphic unit is (24) combinatorial element vocabulary.

K. Is it a modular field, grid, refinement grid, tartan grid, or structural tartan grid?

- If modular field, graphic unit is (11) modular field.
- If grid, graphic unit is (16) grid.
- If refinement grid, graphic unit is (12) refinement grid.
- If tartan grid, graphic unit is (17) tartan grid.
- If structural tartan grid, graphic unit is (18) structural tartan grid.

L. Is it a schematic subdivision or a concrete partitioning?

- If schematic, graphic unit is (10) schematic subdivision.
- If concrete, graphic unit is (22) partitioning system.

M. Is it a schematic axial system, or a concrete axial system?

- If schematic, graphic unit is (13) schematic axial system.
- If concrete, graphic unit is (15) axial system.

N. Is it a schematic circulation scheme, or a concrete circulation system?

- If schematic, graphic unit is (26) circulation scheme.
- If concrete, graphic unit is (27) circulation.

The questions first aim to isolate groups of graphic representations that have a meaningful resemblance (shapes, sets of lines, spaces, building elements, etc.), and then differentiate on the basis of their interpretation (grid, axial system, schematic subdivision, etc.) to derive the specific graphic unit. This method is not specific enough for automated graphic recognition, but as a question-answer mechanism for a designer it is adequate.

The decision-tree leads in a maximum of five questions to a graphic unit. This process needs to be reiterated until all the elements of the drawing have been identified, although a user may decide to have only part of the drawing interpreted.

3.2 Automated generic representation recognition in drawings

Generic representations are defined on the basis of graphic units in a graphic representation. Graphic units are sets of graphic entities with a particular interpretation. Graphic entities are the primitives that constitute a drawing. For example: the graphic entity line, co-ordinated orthogonal at a modular distance, defines the graphic unit called *grid*. Another example: a set of filled-in crosses, circles, and lines depicting furniture elements, define the graphic unit called *element vocabulary*.

Graphic units have varying degrees of complexity. The *grid* example can be defined rather straightforward on the basis of the graphic entity line. The *element vocabulary* however, takes more steps of interpretation. For this purpose, an architecture of several layers and mechanisms is proposed in which automated generic representation recognition can be implemented (see Figure 2).

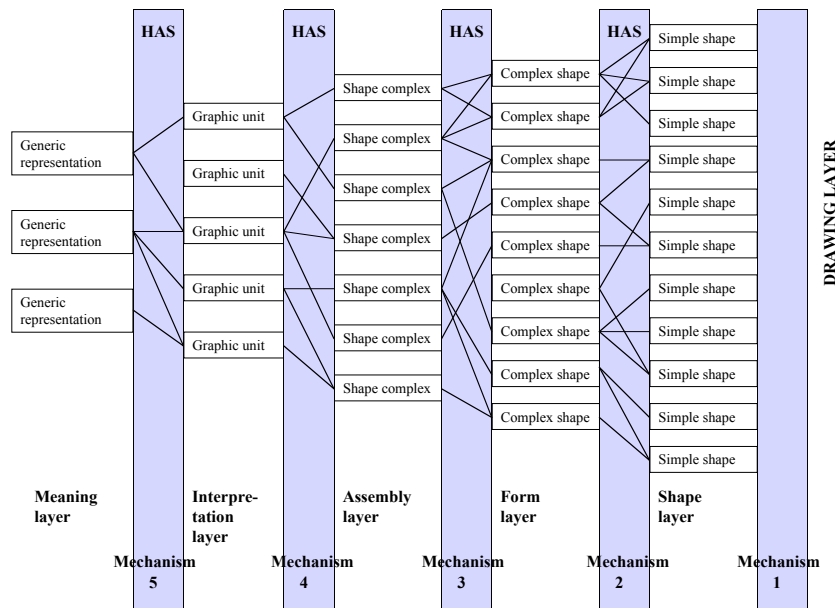


Figure 2: **An architecture for automated generic representation recognition**

In the architecture, six layers of interpretation are defined: drawing layer, shape layer, form layer, assembly layer, interpretation layer, and meaning layer. Five mechanisms, numbered 1 to 5, mediate information between the layers.

Two main movements of information are distinguished. The first move, from the drawing layer to the meaning layer in a step by step fashion through all layers using the mechanisms between the layers, constitutes recognition of one or more generic representations in the drawing. The second move, from the meaning layer to the drawing layer, constitutes drawing according to the definition of one or more generic representations.

- **Drawing layer:** the drawing layer is the current graphic representation that the designer is working on. In a CAD environment, the elements typically are vector representations.
- **Mechanism 1:** from the drawing layer graphic primitives need to be derived for the system to reason with. Shape recognition algorithms have been researched widely in the area of computer vision, but a good selection of these needs to be made in the context of the present work.

- **Shape layer:** the shape layer contains the graphic primitives that are recognised from the drawing. The primitives mostly are lines and dots.
- **Mechanism 2:** from the shape layer, complex shapes are constructed as sets of the graphic primitives. Mechanism 1 and 2 in current day shape recognition systems often are integrated already. For clarity, we have distinguished between the two for now.
- **Form layer:** the form layer contains the complex shapes that make up forms such as circle, squares, rectangles, etc.
- **Mechanism 3:** from the form layer, shape complexes are constructed on the basis of complex shapes. An issue in this mechanism is the phenomenon of emergence, where sets of shapes can yield new shapes because of intersections.
- **Assembly layer:** the assembly layer consists of shape complexes that are more elaborate shapes.
- **Mechanism 4:** from the assembly layer, the shape complexes are identified as graphic units. Ambiguity and multiple interpretations need to be dealt with in this mechanism.
- **Interpretation layer:** the interpretation layer consists of the graphic units that are identified in the graphic representation.
- **Mechanism 5:** from the interpretation layer, generic representations are identified as sets of graphic units. Full and partial matching need to be dealt with in this mechanism.
- **Meaning layer:** the meaning layer consists of the generic representations that have been identified in the graphic representation.

Recognition of generic representations in a graphic representation is not purely a movement from the drawing layer to the meaning layer. Any transition from drawing layer to shape layer, shape layer to form layer, form layer to assembly layer, etc. is informed by information from the next layer, since the process essentially is about search and interpretation. The existing set of generic representations and graphic units informs the mechanisms that look for shapes, complex shapes, and shape complexes. The implementation strategy must reflect this characteristic of the process. For this purpose, the decision tree described in the previous section can be complementary to this process. A computational approach to implement the work still has to be chosen.

4 CONCLUSION

In the present paper an overview was given of a FBM approach to describe design processes. Two of the three tracks of inquiry have been addressed: graphic unit and generic representation recognition in graphic representations, and Feature-Based models of design processes. The third track still needs development: Feature-Based models of generic representations and graphic units in the architecture presented in section 3.2. The work provides insight into the dynamics of design processes, and offers a formalism to describe functional requirements of a design aid system.

5 REFERENCES

- Achten, H.H. (1997a) Generic Representations – Typical design without the use of types, *CAAD futures 1997. Proceedings of the 7th International Conference on Computer Aided Architectural Design Futures held in Munich, Germany, 4-6 August 1997*, Munich, Germany, August 4-6, 1997, pp. 117-133.
- Achten, H.H. (1997b) Generic Representations – An Approach for Modelling Procedural and Declarative Knowledge of Building Types in Architectural Design, Ph.D. Thesis, Department of Architecture, Building and Planning, Eindhoven.
- Achten, H.H. (1998) Typological knowledge acquisition through a schema of generic representations. *Proceedings of Artificial Intelligence in Design '98*, Lisbon, Portugal, 20-23 July 1998.
- Achten, H.H. and Leeuwen, J.P. van (1998). A feature-based description technique for design processes: A case study. *Proceedings of the 4th Design and Decision Support Systems '98 Conference*, Maastricht, The Netherlands, July 26-29, 1998.
- Achten, H.H. and Leeuwen, J.P. van (1999). Feature-based high level design tools: A classification. *Computers in Building. Proceedings of the 8th International Conference on Computer Aided Architectural Design Futures held in Atlanta, Georgia, USA on June 7-8, 1999*, pp. 275-290.
- Bronsvoort, W.F. and Jansen, F.W. (1993) Feature modelling and conversion, key concepts to concurrent engineering, *Computers in Industry*, 21(1), pp. 61-86.
- Bronsvoort, W.F. and Dohmen, M. and Bidarra, R. et al. (1996) Feature Modelling for Concurrent Engineering, *Proceedings of the International Symposium on Tools and Methods for Concurrent Engineering '96*, Budapest, pp. 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, *Proceedings of the 4th Design and Decision Support Systems in Architecture and Urban Planning Conference*, Maastricht, The Netherlands, July 26-29, 1998.
- 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, 1988.
- Lawson (1990) *How designers think*.
- 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, 1996.
- Leeuwen, J.P. van and H. Wagter (1997) Architectural design-by-features, *CAAD futures 1997. Proceedings of the 7th International Conference on Computer*

- Aided Architectural Design Futures held in Munich, Germany, 4-6 August 1997*, Munich, Germany, August 4-6, 1997, pp. 97-115.
- Leeuwen, J.P. van and H. Wagter (1998) A features framework for architectural information: a case study, *Proceedings of Artificial Intelligence in Design '98*, Lisbon, Portugal, 20-23 July 1998.
- Leeuwen, J.P. van (1999) Modelling Architectural Design Information by Features – An approach to dynamic product modelling for application in architectural design, Ph.D. Thesis, Department of Architecture, Building and Planning, Eindhoven.
- Markus, Th.A. (1969) The role of building performance measurement and appraisal in design method, *Design methods in architecture*, London.
- Maver, Th.W. (1970) Appraisal in the building design process, *Emerging methods in environmental design and planning*, Cambridge, Massachusetts.
- Roozenburg, N.F.M. and Eekels, J. (1995) *Product design: fundamentals and methods*, Chichester.
- Shah, J.J. and Mäntylä, M. and Nau, D.S. editors (1994) *Advances in feature based manufacturing*, Amsterdam.
- Shah, J.J (1991) Assessment of features technology, *Computer-Aided Design*, 23(5), pp. 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, pp. 139-148.
- Vries, B. de and Achten, H.H. (1998) What offers Virtual Reality to the designer? *Proceedings of the int. Conference on Integrated Design & Process Technology*, Berlin, Germany, July 6-9, 1998.