Paper published as: van Leeuwen, J.P., H. Wagter, and R.M. Oxman. Information Modelling for Design Support, a Feature-based approach. In: Proceedings of the 3rd Conference on Design and Decision Support Systems in Architecture and Urban Planning, Spa, Belgium, 18-21 August, 1996. p. 304-325.

Information Modelling for Design Support - a Feature-based approach.

Jos P. van Leeuwen [a], Harry Wagter [a], Robert M. Oxman [a+b]
Eindhoven University of Technology
Faculty of Architecture, Building, and Planning
Building Information Technology
Eindhoven
The Netherlands

ABSTRACT

The research described in this paper addresses the problem of modelling design information with respect to the dynamic nature of design. This requires information models to be highly flexible and extensible with domain-specific information-definitions. The approach followed in this research is inspired by the concepts of Features-technology, mainly applied in Mechanical Engineering, and results in a framework for defining flexible and extensible information models for architectural application. This approach distinguishes Generic Feature Types for the domain of architecture and Specific Feature Types for particular sub-domains. It proposes a classification of Feature Types, as well as an infrastructure that accommodates the definition and particularization of Features Types. The research involves the development of a pilot-system for the computational support of this Dynamic Information-Modelling approach. At the end of this paper we discuss the possibilities of the Feature-based Modelling approach as a basis for design support systems.

1. FLEXIBILITY AND EXTENSIBILITY OF INFORMATION MODELS

An important problem in developing computer support for architectural design tasks is formed by the complex and dynamic nature of design. The complexity of design is due to the many different aspects involved in design (aesthetics, function, costs, manufacturing, planning), often represented by as many different participants with as many different views on the product of design. This results in a diversity of information with complex relationships and dependencies. The complexity of building information is subject of research in the area of Product Modelling (van Leeuwen and van Zutphen 1994).

The dynamic nature of design is related to how information is dealt with and how information evolves during design. This section identifies some aspects of this dynamic nature of design and indicates the importance of respecting these aspects when developing information models and computer-support for design tasks. The next section offers an introduction to the concepts of Feature Based Modelling as used in

---

[a] Eindhoven University of Technology, Faculty of Architecture, Building, and Planning,
Building Information Technology, PO Box 513, postunit 20, NL-5600 MB, Eindhoven, The Netherlands.
e-mail: j.p.v.leeuwen@bwk.tue.nl http://www.calibre.bwk.tue.nl/jos
[b] Technion - Israel Institute of Technology, Faculty of Architecture and Town Planning, Haifa, Israel.

Mechanical Engineering, an approach which we will use in the rest of the paper to address the problem of dynamic information modelling for design support.

1.1 Information modelling for design support

Information models are representations of information by means of unambiguous, formal definitions of data and procedures. Modelling design information does more than merely serving the communication of information at certain stages in the design process. An important function of information models is the representation of information during the tasks of design in order to support the processes of interpretation, generation and evaluation of information. Preserving the dynamic nature of design will form a critical precondition especially for information modelling for the purpose of design representation. Other relevant research on the representation of design information is under development by Achten, who investigates the usage of so-called generic representations in order to model typological knowledge of building types through the use of graphic representations (Achten 1996).

1.2 Approaches of Product Modelling

Product data is defined by (Wilson 1993) as “data that describes the function and physical characteristics of each unit of a product from its requirements at inception to its configuration at time of retirement”. In other words, product data represents a product throughout its life-cycle. In order to solve the problem of complexity as described at the beginning of this section, two paths can be identified which are being developed over the last several years in the field of Product Modelling (PM).

One path focuses on enhancing the communication between participants by developing the exchange of product data. This approach presumes that the data to be communicated can somehow be agreed upon by exchange partners. De Vries (1996, p. 51) argues that ‘to be able to compare [building] objects, aspects [describing these objects] must have a univocal definition. These definitions form a normalised aspect type library and should be part of the building regulations.’

This leads to the other path which focuses on the integration of product data models, arriving at the definition of a so-called core model for the Building and Construction (B&C) industry. A core model would need to be generally acceptable for all disciplines involved in B&C, defining entities of information that match each of the different views of the multiple participants. Also a core model would need to cover all of the common aspects in the B&C domain. Both requirements are unlikely to be met, regarding the dynamic aspects of architectural design and the B&C domain as described above.

A core model which is currently under development is the BCCM (Building and Construction Core Model) within the context of STEP (Standard for the Exchange of Product model data). This core model comprises a rather detailed structure of entities representing product and process data for B&C, mainly focusing on the ‘as-designed’ stage (Tolman and Wix 1995). In this development the problems mentioned above have not yet been solved and issues like extensibility and general acceptability are still under discussion. A compilation of some of these discussions was made by Wittenoom (1995).

Ramscar (1994) believes that the current PM approaches cannot achieve a complete model that is necessary for an integration or exchange standard, because they result in models with a fixed view, which are unable to deal with unforeseen design data and cannot respond to the changing needs of their users. We identify this problem as lacking support of the dynamic nature of design.

1.3 Dynamic nature of design

As a problem-solving process, design involves the structuring of information that often concerns ill-defined problems. During this process a designer searches for solutions, alternatives, and optimisations, and information is constantly generated, interpreted, and restructured. This means that an information model for support of design-tasks requires a flexibility that allows for (re-)definition and (re-)structuring of information: it requires evolution of the model during design. Flexibility is discussed in detail in section 1.4.

Another aspect of the dynamic nature of design is that designers learn. They change their approach to solving design-problems, finding new techniques, new rules, new concepts. Mitchell refers to this as stylistic evolution, and stresses that CAD systems must provide for this essential component of creative design (Mitchell 1990). We find that the information model underlying design support systems must accommodate this stylistic evolution: by adaptation of the model to the changing demands. A similar evolution can evidently by noticed in the Building & Construction industry as a whole, resulting in the constant development of new techniques, methods, products, materials. These new agents in the field also require adaptation of information models for design support.

Evolution and adaptation of design information models are also subject of research on an Engineering Data Model (EDM) by Eastman. Here it is argued that the schema for CAD data cannot be known beforehand, but is ‘defined incrementally as design proceeds’. The structure of design data depends upon decisions regarding the technology and functions associated with the design components. Other observations by Eastman are that CAD data is incomplete for most of the design process, integrity being added incrementally, and that the design process involves iterative refinement, meaning that information is dynamically added in more detail during design (Eastman 1991) (Eastman 1995).

1.4 Flexibility in information modelling

Although in many information models, the characteristics of building parts are modelled as attributes of those parts, this is not necessarily the way these characteristics have been designed. Clearly, design-decisions do not necessarily start with the physical parts of the building, continuing with the specification of their characteristics. More often, design will start with the characteristics that match particular desired functionality, and will conclude with a solution that combines certain of these characteristics into physical building parts. Not necessarily, characteristics are embodied by a single physical building part, but may just as well be the result of a combination of building parts.

A simple example will make the above more clear. Consider the design problem of the division of space. Suppose the designer does not immediately think about this problem in terms of solutions (products), but analyses it in terms of concepts, then this could result in the specification of concepts such as spatial separation, thermal & sound insulation, structural function, etc. At one point in this design-process, a solution may be found in a simple inner wall combining all these concepts into a physical building part. At a later point, however, the design-problem may be extended with another concept such as visual continuity: visual contact between the divided spaces. The total of concepts can no longer be met by the inner wall solution, which may be replaced by an inner wall with glazing. Yet another solution might be to place steel columns which perform the structural function, and glazing frames in between the columns matching the other concepts.

This simple example shows us several important aspects to consider when creating an information model to support this process:

- The solutions generated by the design process (the physical parts) cannot be seen as the ‘information carriers’ in the model, since they may be replaced at any instance in a single decision, while much of the information involved should remain in the model. The concepts that are to be embodied, however, are more likely to remain within the design-case, since they form the core entities of the reasoning by the designer.
- During the design-process concepts will be grouped together in order to find solutions that can match these concepts. However, it is part of creative design to find the right combination of concepts for the right physical solution. This pursuit may include the evaluation of many different combinations of pieces of information. The information will constantly be reinterpreted and restructured in order to find new possible combinations leading to a solution.
- In the end-result, concepts are not necessarily properties of single building parts. Certain concepts may be represented by characteristics of many different parts at the same time, or, more complicated but very likely, by the combination of several building parts (e.g. sound insulation performed by the combination of the steel structure and the glazing frames, thermal insulation performed by the layers in a cavity wall).

It is important to realise that design information will comprise much more than just physical characteristics in design. In fact, any notion or concept that a designer uses in reasoning about a design may be included in the information model. Examples are concepts such as function, performance, planning, accessibility, maintenance, etc.

1.5 Extensibility of information models

Restructuring information is one aspect of dynamic information modelling. Another aspect is extensibility. Inherent to design as a creative process, the concepts and notions used in design cannot be anticipated. Therefore, a generally acceptable core model for architectural information modelling can only be incomplete or of a very high level of abstraction which would not contribute sufficiently to design support.

A design information model needs to be extensible so the designer can add information definitions that reflect specific or new concepts and notions. It is not just newly invented construction methods or products that form new concepts, also style-specific approaches to a design-problem result in concepts that cannot be anticipated, yet need to be reflected in an information-model in order to have an accurate representation of these concepts.

Extensibility of a conceptual information model for design means that information can be defined by the designer in relation to what the conceptual model already contains, and that these new definitions can be added to the conceptual model. When actually modelling information from a particular design-case, these 'designer-defined' extensions to the conceptual model can be used to accurately reflect the concepts of the particular design.

1.6 A Feature-based approach

The approach of Feature modelling as set out in this paper, proposes to model characteristics and concepts as independent units in the information model which are called Features. A network of relationships between Features comprises the coherence of the characteristics and concepts. The fact that Features are defined and modelled as independent units in the information model, not as attributes of predefined entities, gives the model the flexibility to restructure the network of Features and modify their relationships and dependencies during the progress of design. It also means that the composition of a model can be specified and modified at a higher level of detail, more accurately representing the design.

The conceptual model does not have a statically predefined structure, but consists of an open set of Feature definitions with generic relationships. New, or specialised, domain specific Feature definitions can be added to the model, including the definition of more specialised relationships.

We will first introduce the original concepts of Feature-based Modelling from the field of Mechanical Engineering. Then we will set out how these concepts can be used to model architectural information.

2. FEATURE BASED MODELLING

This section will briefly introduce the concepts of Feature-based Modelling which are developed in the area of Mechanical Engineering.

2.1 History of Feature-based Modelling

As in Construction, Mechanical Engineering as well was confronted with the need for modelling tools with a higher level of information than just plain geometry. High-level models were necessary to extract information for generating process plans for manufacturing, evaluation of manufacturability, and for instance Numerical Control programs. This high-level information is not present in solid geometry models which consist only of nominal geometry and low-level product definitions (Shah and Rogers 1988). The need for higher level information models also came from the desire to construct the model using the terms that are derived from design-practice and that have particular functional significance to design, allowing designers to use the model when reasoning about a design.

New modelling entities were defined to represent high-level information. These entities were called Features and were mainly used to describe the physical characteristics of a product-part. Initially being used for manufacturing purposes, they represented shape and technological attributes associated with manufacturing operations and tools. Feature-technology has now developed to application in product design stages as well.

Since geometry is often a base-point for part-modelling in Mechanical Engineering, so-called Form Features have become the most important category of Features. A commonly acceptable definition of Form Features appears to be given by Shah (1991a): ‘[Form] Features are generic shapes with which engineers associate certain properties or attributes and knowledge useful in reasoning about the product’. A more general definition of Features by Shah (1991b) is: ‘Features are elements used in generating, analysing, or evaluating designs’.

2.2 Classifications of Features

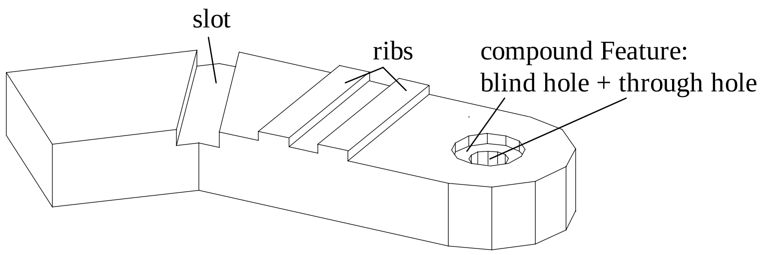
Many different classifications for Features have been made, distinguishing Features by purpose or, in the case of Form Features, by shape. Some of the categories generally recognised are (Shah and Rogers 1988):

- Form Features: elements related to nominal geometry
- Precision Features: acceptable deviations from nominal form of size
- Technological Features: related to performance and operation
- Material Features: material composition, treatment, condition
- Assembly Features: assembly sequence, orientation relative to the part

and (van Emmerik 1990):

- Pattern Features: regular patterns of similar entities

Figure 1: **Some examples of Form Features defining a part**
(from: van Leeuwen, Wagter, Oxman 1995)



- Connection Features: geometric constraints
- Property Features: properties not related to explicit geometry
- Application Features: related to process planning aspects.

Typical Form Feature classifications include Features such as passages, depressions, protrusions, deformations, etc. Figure 1 shows some examples of Form Features defining a part. Although Form Features are dominant in Feature-technology as applied in Mechanical Engineering, the role of other types of Features appearing in the mentioned classifications, is of great importance for product design support and automated reasoning in product design.

2.3 Approaches of Feature-based Modelling

Basically two approaches to Feature-based Modelling have been developed in Mechanical Engineering. Initially, human interpretation of geometry was needed to specify the high-level information as Form Features. This approach is called human-assisted or interactive Feature recognition, and later, in its automated form, automatic Feature recognition: high-level information could be automatically recognised and extracted from geometric models (Henderson and Chang 1988) (Laakko and Mäntylä 1993) (Meeran and Pratt 1993). This approach however, can only result in relatively limited levels of information, because from geometric data only a limited type of information can be extracted. Also, the recognition procedures necessarily have their limitations. Feature recognition can only be done after the modelling of a part in geometry has completed. Changes in geometry must therefore be followed by a repeated procedure of Feature recognition, while valuable information available during design cannot be modelled and is lost.

The approach that logically followed Feature recognition, was to construct the model of the product not on a geometry-basis, but based on Features directly. This approach is called design-by-Feature (Shah and Rogers 1988) (van Emmerik 1990) (De Martino 1994). Definitions of different types of Features that a designer can choose from when modelling a product have been collected in Feature-libraries. However, it is commonly agreed upon that Feature modelling requires an extensible set of Feature definitions, allowing designers to define their own set of Feature types (van Emmerik 1990) (Shah 1991a). The meaning of a Feature is therefore dependent of the context, or view, in which the Feature has been defined. Communicating view-dependent information requires translation of information from one view to another, and this is being provided for by mechanisms of Feature mapping (Shah 1993) (Chamberlain 1993).

Combinations of both approaches of Feature recognition and design-by-Feature are currently subject of research, using the benefits of automatic interpretation of data while allowing early design knowledge to be captured in the model (De Martino 1994).

2.4 Summary of the concept of Features in Mechanical Engineering

After analysis of the developments in Mechanical Engineering on Feature-based modelling, we conclude that Features are collections of information which have a semantic meaning to a particular view on a product-part (van Leeuwen, Wagter, Oxman 1995). The model of the product-part is built up during design by means of an interrelated set of Features that together describe the part. Features may concern the shape of a part (Form Features), but the concept also includes non-geometric information which has significant meaning for a particular engineering point of view.

Computer supported systems for Feature-based modelling allow their users to define types of Features, which can be stored in libraries, and to create instances of Feature types when modelling a part. The structure relating the Features that compose a part is not pre-defined: the relationships between Features are context dependent and defined by the user, partly when defining the Feature, partly when creating the Feature-based model.

In Mechanical Engineering, Features are used only for the modelling of product-parts, not at higher levels in the hierarchy of a product model where aggregations of parts, modules, and information the product as a whole are concerned. This will appear a limitation that we do not wish to include in our approach of Feature-based modelling of architectural information.

3. ARCHITECTURAL INFORMATION AS FEATURES

Currently, Feature Based Modelling techniques are already applied in areas of the B&C industry that resemble Mechanical Engineering. In structural engineering, for example, construction parts are modelled by Features (Watson 1994), representing the construction as designed, mainly by using Form-Features for manufacturing purposes.

Like in Mechanical Engineering, the relevance of Feature modelling has been proved in this field of application. However, the role of Features in the approach to modelling architectural information that we propose is much stronger: Features become the basis of the information model, the entities of information that are used to build up the information structure. In this section we define the term Feature, and make a distinction between generic and specific types of Features. We also discuss the activities that are involved in Feature modelling. In the next section, we will propose a classification of Feature definitions that will form the basis of a framework of Feature-based architectural information modelling.

3.1 Definition of a Feature

The definition of the term Feature in this modelling approach is given below.

> *A Feature is a collection of high-level information 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 several important aspects of Feature modelling which are part of the modelling approach.

Features are high-level information with semantic meaning.
Having semantic meaning, Features can reflect the terminology used by designers and are therefore close to the domain of design. This should make the modelling of the design rationale (i.e. translation into formally defined data-structures) fairly easy. Note that this is not a unique property of a Feature based approach, other Product Modelling approaches also have the intention to model entities with semantic meaning. The difference lies in how these entities are chosen.

Features define sets of characteristics or concepts.
These sets of characteristics or concepts form logical units representing aspects in the particular view for which the Features have been defined. This means that a physical part of a building will probably have different sets of characteristics. For instance a concrete column will be represented by the characteristics of the material concrete, the characteristics of a load bearing column, and maybe the characteristics of a 3D solid geometry with a square section. This physical part may also share these characteristics with other parts: Features may belong to different parts of the building.

This aspect of defining characteristics and concepts as independent units in the information model is the most important in the Feature based approach. It constitutes much of the flexibility and extensibility of the conceptual information model, which we consider critical requirements for architectural information modelling.

Features represent a particular view in the life-cycle of a building.
The definition of information represents a particular view from which this information originates or for which it is to be used. This can be a generic view on design or construction as a whole, but may also be specific to a particular discipline, project, design-style, application, etc. The terms Generic Feature Types and Specific Feature Types are important in the framework that is described further in this paper.

It should also be noted that, in contrast with the original techniques of Feature-based Modelling in Mechanical Engineering, in this approach a Feature may define information at any level of abstraction, not just as describing the properties of a product-part. This means that Features will occur in the building model describing the concepts that relate to the building as a whole, as well as concepts relating to the more detailed views on the project.

3.2 Generic Feature Types and Specific Feature Types

Many definitions of Feature Types will have a very general character. It is likely that a certain basis of Feature Types can be generally accepted by all of the B&C industry. We call these Generic Feature Types. This is not to be confused with the earlier statements in this paper which discussed that a B&C core model is not a plausible option. It is important to keep in mind that Feature Types are independent units of information, they are defined without being fixed within a particular context. Even if certain relationships between Feature Types are of general pertinence, the exact and complete context of Features is defined only during the process of modelling particular design- or building-information. Only at this point the position of a Feature in relation to its context can be fully known. Hence, the set of Generic Feature Types cannot be regarded as a traditionally conceived core model of B&C information. However, our dynamic modelling provisions promote a new concept of core model, which does not predefine a complete structure of information, but rather defines a set of common concepts used in the B&C domain.

It is the intention of the research presented here to investigate the definition of a limited number of Generic Feature Types, in order to demonstrate their meaning and function within the Feature modelling framework. However, the definition of Generic Feature Types should be left to international standardisation institutions, such as the ISO.

Obviously, Feature Types that are not generally acceptable in the B&C domain, must be Specific Feature Types. They are defined for a specific point of view, a specific aspect-model, a specific application, discipline or design-style: in brief, they are defined in the context of a specific domain. Specific Feature Types form the extensions of the information model that consists of the Generic Feature Types.

4. ARCHITECTURAL FEATURE MODELLING

The approach in this research is a ‘design-by-Feature’ approach. Because of the richness of the domain, we believe that Feature recognition will not suffice for architectural design. The design-by-Feature approach, which involves modelling high-level information during (early stages of) the design-process, is expected to be a more useful form of design-support than attempts to recognise information from geometry.

From the previous sections we conclude that four activities are necessary in the Feature-based modelling approach. These activities, shown in figure 2, are (1) the definition of information in the form of Feature Types; (2) the organisation of Feature Types into a classification system; (3) the actual modelling of information by generating Features as instances of Feature Types, and (4) modification of the Feature model during the course of design.

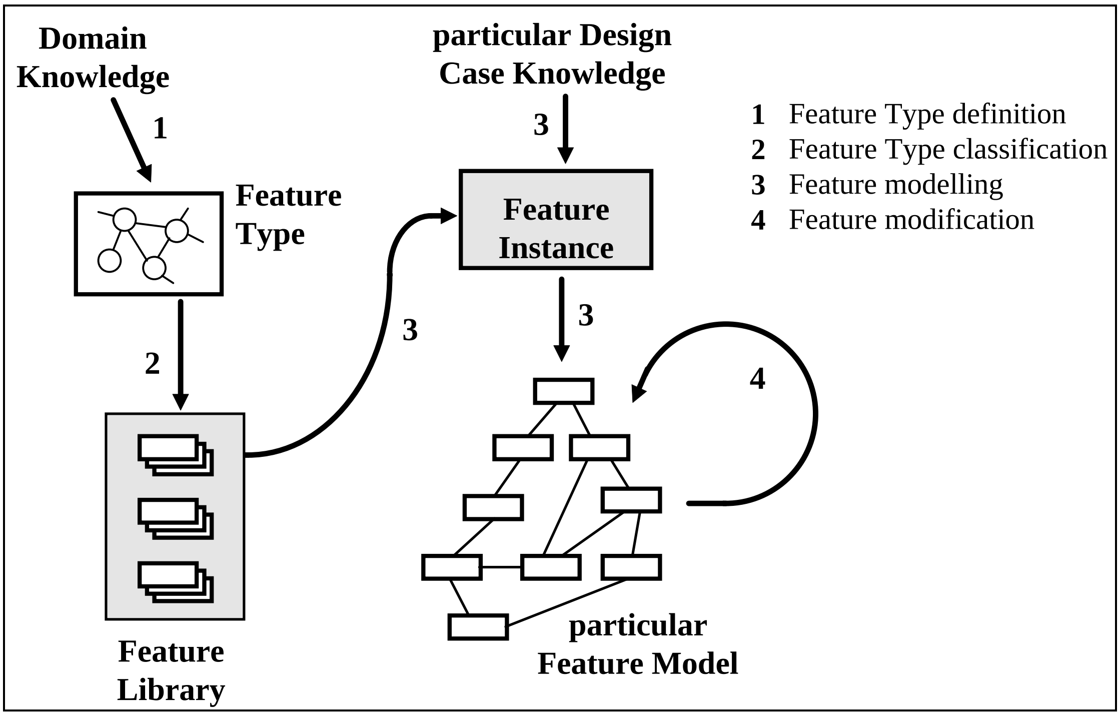
The first activity, definition of Feature Types, involves the translation of particular domain knowledge into formal definitions. It requires that certain domain knowledge is identified as forming a relevant concept, which then needs to be defined using some formal definition method. This formalisation of the concept may include relations to other concepts: the interrelationships between Feature Types.

Once this concept has been formally defined as a new type of Feature, the next activity would be to classify it in relation to previously formalised concepts. For a particular domain, which may represent an aspect of B&C or a specific design-style, in this manner a library of Feature Type definitions can be built up. Such a library would represent (a part of) the knowledge in the domain for which it is defined.

These first two activities are necessary when new design-knowledge needs to be formalised, for instance when a concept has been devised that is not already represented as a Feature Type.

The third activity is the actual modelling of information from a particular design case. This involves the creation of instances of Feature Types selected from a Feature library. Instantiation of Feature Types means providing an instance of this

Figure 2: **Activities in Feature-based Modelling**



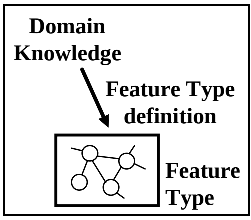
type with the actual information taken from the particular design case and relating it into the structure of Features that make up the model representing the design case.

During the process of design, the design-information will be interpreted and evaluated and will probably result in modification of the information model as well. This activity requires modification of data present in Feature instances and of the relationships between Feature instances. It may also result in changing the type of certain Features in the model.

4.1 Examples

Below we will give some examples of what type of information may be defined as Feature Types. Feature definitions can be devised for any scale of detail, for any level of abstraction, and at any point in the life-cycle of a design or construction-process. They may serve the formalisation of a certain design-style, but also represent certain conventions on details for a specific building-project. They may represent a specific method of construction, or form elements in the planning of a project. They may be defined for early design, as well as for facility management. It is even likely that many of them will serve from early design throughout the design- and construction-process to facility management and even destruction of the building.

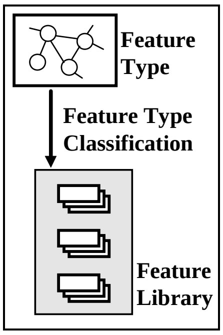
Feature Type definition



Feature Types may be defined to represent knowledge from particular domains. Archetypes, such as a certain type of column, window, or elevation-element, may be represented as types of Features in a library. Also other concepts in a design-style, not being physical building-parts, may be represented as Feature Types. For instance particular detail-solutions such as for the connection of building-elements, concepts like grids and zones, proportions of measures, design-concepts such as transparency, day-lighting, routing, etc.

In later stages of the design- and construction-process, Feature Types may represent concepts that are important for specific work-methods, for planning, particular methods for cost-calculation, structural and climatic design. These concepts may be agreeably defined in a Generic Features Library, or specifically defined for particular requirements.

Feature libraries

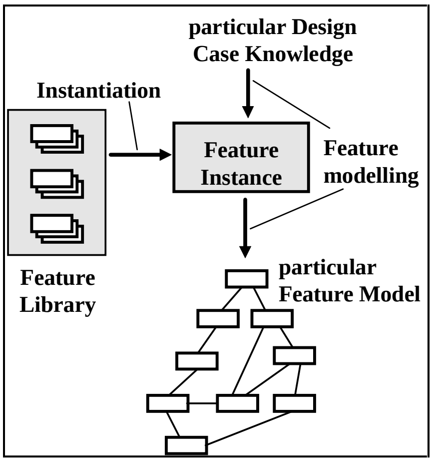


Feature Libraries are classifications of Feature Types for relevant sub-domains in the area of B&C. Generic Feature Libraries, containing Generic Feature Type definitions, may represent generic knowledge and common concepts from sub-domains such as spatial design, structural design, or aspects such as cost-calculation, material-specifications, geometry, which are common to multiple domains. Generic Feature Libraries will need to be standardised by authorised institutions.

Specific Feature Libraries are classifications of Specific Feature Types that will be defined to represent concepts and knowledge that is not common to a domain within B&C, but specific to for instance a design-style, a construction-technique, a product-group, etc. Specific Feature Libraries may also represent typical concepts and knowledge defined for a particular building-project. These Feature Libraries will then serve as a vocabulary in the communication between partners in the project.

Feature modelling

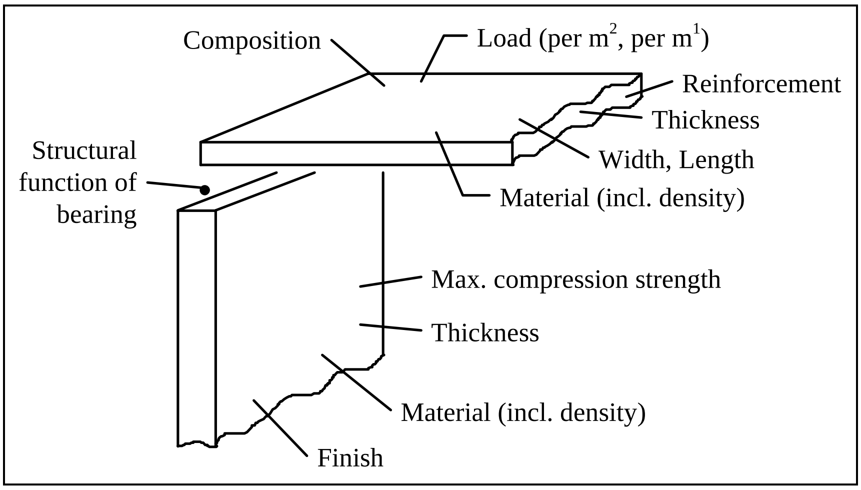
The actual modelling of the information from a particular design case, involves the instantiation of a selected Feature Type, suiting the type of information that needs to be modelled. The flexibility of Feature models can be demonstrated by an example that considers the relation between a wall and a floor which is carried by the wall. The characteristics of both elements are modelled as instances of Feature Types (see figure 3). Reinforcement, thickness, and material, for example, are all separate entities in the model, their relations to each other and to the floor constitute the characteristics of the floor. The model now has the capability of gaining more information as it becomes available during design, since characteristics, new Feature-instances, may be added to the model.



Feature modification

The information in the Feature model may be modified in a very flexible manner. For example the Material Feature of the floor may be shared by many elements in the model, thus allowing to modify this characteristic for all these elements at the same

Figure 3: **A simple example of characteristics and concepts as Features**



time. However, the Material Feature of this particular floor may also be replaced so that it doesn't affect other elements. This flexible behaviour becomes more important when for instance the composition of the floor is changed during design. This will affect many characteristics of the floor, such as the required reinforcement and possibly its thickness, but probably the load on the floor will not be affected, nor its width, length and the relationship to the wall.

The relationship between floor and wall is another example of the requirement that concepts need to be modelled independently of physical building parts. Although it is clear that the floor is structurally dependent of the wall, it is not wise to model this characteristic as a property of the floor. We have seen that replacing the floor does not affect this characteristic, moreover this characteristic may be shared by other elements such as beams and other floor-elements. Nor is it wise to model this load-bearing function as a property of the wall, for replacing the wall with a beam does not change this function. The structural dependency that exists in this example is clearly a concept that requires its own place in the information model, which it can be given with the Feature-based approach.

5. ARCHITECTURAL FEATURE MODELLING FRAMEWORK

Flexibility and *extensibility* are the two keywords in this Feature-based approach to information modelling. The framework described in this section defines the information infrastructure that will accommodate these requirements. At the basis for this infrastructure is a classification of architectural Features which can be used to categorise both generic and specific Feature Types.

5.1 Classification of architectural Feature Types

We have set up a basic classification of architectural Features types, which is mainly based on a survey and analysis of building- and design-related information. This survey included a proposal by Woestenenk (1995) for international conversion tables for parts and functions. A recent study on classification of construction works is found in (Ekholm 1995) and (Ekholm 1996).

The main categories of our Feature Type classification are outlined in table 1. The classification is not complete, and is not intended to be, since we believe it never can be: that is why information models need to be extensible. However, the proposed categories of Feature Types are believed to form a good starting point for the definition of generic Feature Types and to start defining specialised Feature Types for specific cases in design.

The Feature Type classification has been the basis for a framework of Feature modelling, which incorporates a methodology of defining information in a Feature-based manner, and creating information models based on these Feature definitions.

Paper published as: van Leeuwen, J.P., H. Wagter, and R.M. Oxman. Information Modelling for Design Support, a Feature-based approach. In: Proceedings of the 3rd Conference on Design and Decision Support Systems in Architecture and Urban Planning, Spa, Belgium, 18-21 August, 1996. p. 304-325.

Table 1: **Main categories of the Feature Type classification**

FEATURES TYPES	BRIEF DESCRIPTION
Form Features Morphological Features Topological Features Geometrical Features	Form Features describe the form, shape, or topology of other entities in the building model, which can be physical entities, but abstract as well.
Physical Features Compositional Features Material Features Composition performance Features	Physical Features form the group of Features that describe the physical qualities, performances, and requirements of entities in the building model.
Context Features Design conceptual relation Features Interface Features Performance dependency Features	Context Features define characteristics and concepts that form relationships between entities, such as dependencies, adjacencies, and relations of functions.
Procedural Features Planning Features Preparation Features Staging Features Integration Features	Procedural Features include the type of information that somehow describe procedures related to the construction process, from the preparation of the work to the actions on the workfloor.
Life-cycle Features Functionality Features Operation Features Quantitative Features Maintenance Features Re-usability Features Security Features	Life-cycle Features are the ones that describe concepts and characteristics that are especially relevant during the complete life-cycle of the building, particularly when it is being used, maintained, revised, renovated, or is given new functions. Also quantitative information such as costs falls within this category.

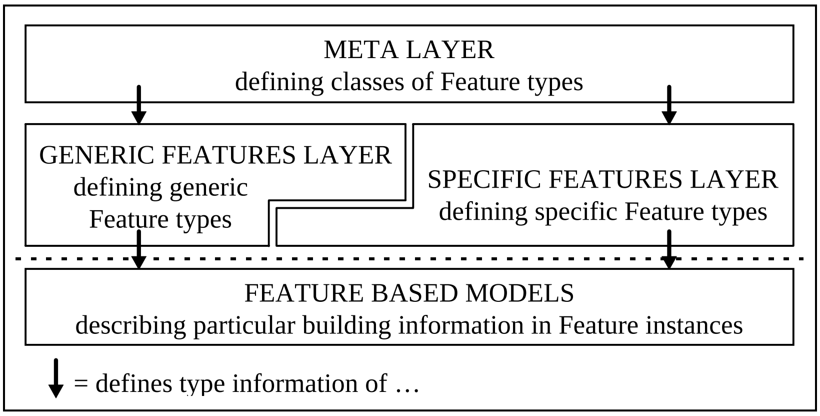
5.2 Infrastructure for Feature modelling

The framework of the Feature-based approach is formed by an infrastructure for modelling design- and building-information with Features. This infrastructure is formed by three layers of information definitions (figure 4).

Starting in the middle, on the left we find the definitions of Generic Feature Types. This layer defines the contents of Feature Types that are generic for B&C. On the right is the Specific Feature layer, defining the contents of Feature Types that are not generic, but defined specifically for a particular domain. Both these layers (generic and specific) define types of Features which may be used to create Feature instances that make up a Feature-based model for a particular design- or building (the lower box in figure 4).

The top layer is the so-called Meta Layer that holds the definition of what information is contained in a Feature Type definition, hence the name Meta layer. The definitions in this layer specify an overall structure in which the Generic and Specific Feature Types must be placed. Using this Meta layer, processes of Feature Type

Figure 4: **Three layers of Feature Type definition**

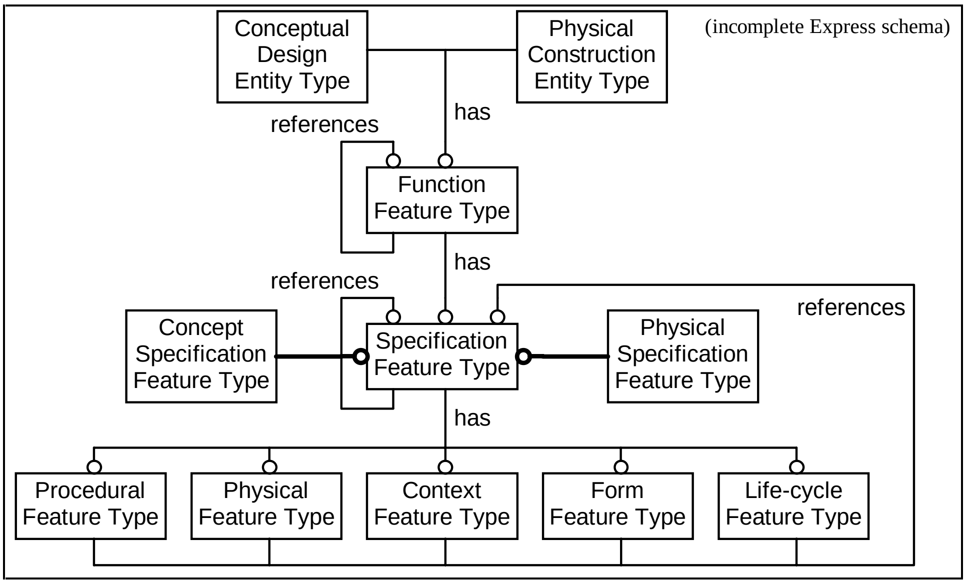


definition and Feature modelling can be generally described and automated in a computational environment.

The Meta Layer is shown in figure 5. Central in the generic information-structure of the Meta layer is the Function Feature Type which defines the type of Features that are used to model the basic functions of entities in the design- or building-model. A Function Feature and its specification (see below) relate entities in the model to each other, and therefore constitute the network of information.

Above the Function Feature Type we see two types of entities that may be described by this Feature: Conceptual Design Entity Type which accounts for design concepts which are not necessarily represented by physical parts in the design, and Physical Construction Entity Type which accounts for elements or parts that will be physically present in the building. These types of entities are described and

Figure 5: **Meta layer: Infrastructure for Feature modelling**



interrelated through the different Features that are specified and attached to them in the actual model, starting with Function Features. Both types of entities may be of different levels of abstraction, meaning that the building itself for instance may be represented by a Conceptual Design Entity, but concepts at a more detailed level will also be represented in the model. Similarly, building parts of higher aggregation levels may be present in the model as Physical Construction Entity (e.g. a south elevation), simultaneous with its decomposed parts (concrete slabs, window-frames, …).

The Function Feature Type is decomposed into Specification Feature Types, Concept Specifications for Conceptual Design Entities and Physical Specifications for Physical Construction Entities. These Specification Features refine the definition of Function Features and have relationships to information stored in the lower line of Feature Types in figure 5. These Feature Types we recognise from the classification of architectural Features that we have presented earlier in this paper.

As stated before, this structure of Feature Type definitions forms a Meta layer in the Feature modelling framework: it describes how Generic and Specific Feature Types can be defined, which in turn describe how particular instances of Features are modelled.

6. A FEATURE-BASED MODELLING SHELL

Currently, work is being done on a pilot-system for computational support of the activities of Feature-based Modelling. The system forms what we called a Feature-Based Modelling Shell (FBM Shell) and implements the Meta layer as shown in figure 5. This will form the basis for implementation of procedures that allow the definition of Feature Types for both the Generic Feature layer and the Specific Feature layer (see figure 4), as well as the instantiation of these Feature Types.

The requirements of the FBM Shell includes many aspects of which we wish to highlight the following:

- *Implementation of non-semantic data-types* to be used in Feature Type definitions. This is the implementation of the lowest level of data-types that will form the basis for Feature Type definitions. It involves simple data-types such as numerical or string-types, but also types which define more complex information such as sets or relationships. Much of the implementation of this level can be compared to implementations of data-types in programming languages.
- *A method or language for formal definition of information in Feature Types*. A uniform method is required to formally define information in Feature Types. This method could involve a descriptive language for data-definition, such as Express as developed in (ISO 1994), but could also be based on a binary format and rely on a dedicated user-interface to specify and read the format. Both approaches require a formal definition of Feature Type contents.

- *Dynamically definable user-interface elements* for specific needs of Feature Type definitions. In a Feature modelling system, two levels of interface are important: the interface needed to allow the user to define and manipulate Feature Type definitions; and the interface needed by the user when modelling a design or building. The first level of interface is defined by the system, it needs to accommodate the procedures necessary for Feature Type definition which are also defined by the system. The second level of interface however, is to be defined by the user when Specific Feature Types are involved, for it is the user who defines the contents and the usage of these Feature Types.
- *Relations of Feature Types to geometric modelling systems*, for instance using parametrics. Many Feature Types will contain or have some kind of relation to geometric data. Therefore, a close integration of a geometric modelling-system would be desirable. Parametric description of geometry is one of the possibilities to incorporate geometry in Feature Type definitions. Parametrics has been subject of research for a long time, an example of successful implementation is presented in (van Emmerik 1990), and although very important for the complete development of a Feature-based Modelling system, it will not be a main focus in our research. It is expected that integration of available techniques of parametric geometry modelling will suffice the needs of such a system.

The FBM Shell accommodates the information structure and methods required for defining Feature Types and creating Feature instances. The proposed classification of architectural Feature Types is incorporated in this environment, so Generic Feature Types, to be provided by a standard, as well as Specific Feature Types, defined for particular domains, can be classified and defined in relation to each other.

The proposed framework for Feature-based Modelling offers a basis for creating dynamic information models. Much of the dynamic behaviour of this model might be formally defined as well. This means that the behaviour of Features may be incorporated in their definition. Doing so would require a very rich definition language with for instance geometric reasoning mechanisms.

7. CONCLUSIONS AND DISCUSSION

The research presented in this paper aims at developing a framework for defining and creating information models for architectural application, which have the important properties of flexibility and extensibility. To achieve this objective we studied the techniques of Feature-based Modelling as applied in Mechanical Engineering.

The integration of the concepts from Feature-based Modelling in approaches of Product Modelling have resulted in a conceptual framework which accommodates:

- extensibility of the conceptual information model;
- dynamic definition of characteristics and concepts as Feature Types;
- Feature definitions at different levels of abstraction;
- flexibility of information models allowing restructuring information during evolution of the design;
- a means for designers to capture specific design-decisions into information models;
- a means for designers to capture specific design knowledge into Feature-definitions.

The development of a pilot-system for computational support is expected to indicate that practical implementation of this framework is possible and may result in a powerful tool for design-support. In addition, the framework is not restricted to information from design-stages only, but allows information from the complete life-cycle of an architectural artefact to be defined and modelled. Although not discussed in any detail in this paper, the framework also contributes to the development of more flexible ways of communicating design- and building-data between parties and across disciplines, offering a common basis for information-structures in combination with flexibility (van Leeuwen, Wagter, Oxman 1995).

7.1 VR-(DIS)^2

The VR-DIS project initiated by the group Building Information Technology of Eindhoven University of Technology and developed in collaboration with several other disciplines within the faculty of Architecture (van Zutphen and Mantelers 1996), forms one of the potential areas of application of the Feature-based Modelling approach. The VR-DIS project, where DIS stands for Distributed Interactive Simulation / Design Information Systems, uses Virtual Reality as a medium for simulating various processes related to buildings or their designs, and as an enhanced interface for design support systems. Because the technology used in the VR-DIS project needs to serve multiple disciplines and various aspects of building information, it requires an approach that allows for easy customisation of data-definitions and software components. Flexibility and Extensibility are also prime issues when VR is to be used directly in early design tasks and early evaluation of design, as described in the research on prototyping of designs in VR by Coomans (1996).

In this paper we have shown that the major benefits of the Feature-based Modelling approach for Architecture are the definition of a generic, yet flexible set of Features for Architecture, and the extensibility of this set with specific Features for specialised domains. Application of this approach in multi-disciplinary support systems such as the VR-DIS project, contributes to the coherence of the overall data-structure of these systems, and thus to the general engineering of the software. On the conceptual level, the Feature-based Modelling approach contributes to a shared basis of information-definitions which advances the possibilities of information-integration and overall system-integration across the multiple disciplines.

8. REFERENCES

Achten, H.H., Bax, M.F.T., and Oxman, R.M. (1996) Generic representations and the generic grid: knowledge interfaces, organisation and support of the (early) design process. *Proceedings of the 3rd Conference on Design and Decision Support Systems in Architecture and Urban Planning*. Spa, Belgium, August 18-21, 1996.

Chamberlain, M.A., Joneja, A., and Chang, T.-C. (1993) Protrusion-features handling in design and manufacturing planning. *Computer-Aided Design* vol. 25(1): pp.19-28.

Coomans, M.K.D., and Oxman, R.M. (1996) Prototyping of design in virtual reality. *Proceedings of the 3rd Conference on Design and Decision Support Systems in Architecture and Urban Planning*. Spa, Belgium, August 18-21, 1996.

de Vries, B. (1996) *Communication in the building industry, a strategy for implementing electronic information exchange*. Ph.D. Thesis. Eindhoven, Eindhoven University of Technology.

DeMartino, 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(8).

Eastman, C.M., Bond, A.H., and Chase, S.C. (1991) A data model for design databases, in J.S. Gero (ed.), *Artificial intelligence in design '91*, pp.339-365. Sydney, Butterworth Heinemann.

Eastman, C.M., Assal, H.H., and Jeng, T.S. (1995) Structure of a product database supporting model evolution. *Modeling of buildings through their life-cycle* (proceedings workshop CIB W78 '95), pp.327-338. Stanford University, CIB W78.

Ekholm, A., and Fridqvist, S. (1995) Object-oriented CAAD, Design object structure, and models for building, user organisation and site. *Modeling of buildings through their life-cycle* (proceedings workshop CIB W78 '95), pp.553-564. Stanford University, CIB W78.

Ekholm, A. (1996) A conceptual framework for classification of construction works. *ITcon*, electronic journal at http://www.fagg.uni-lj.si/~itcon/ vol. 1.

van Emmerik, M.J.G.M. (1990*) Interactive design of parameterized 3D models by direct manipulation*. Ph.D. thesis. Delft, Delft University Press.

Henderson, M.R., and Chang, G.J. (1988) FRAPP: Automated Feature Recognition and Process Planning from solid model data. *Computers in Engineering* vol. 1: pp.529-536. ASME.

ISO TC184/SC4/WG5. (1994) Description methods: the EXPRESS language reference manual, ISO TC184/SC4 10303 part 11. *Industrial automation systems and integration - Product data representation and exchange*. ISO TC184/SC4.

Laakko, T., and Mäntylä, M. (1991) Feature modelling by incremental feature recognition. *Computer-Aided Design* vol. 25(8): pp.479-492.

Meeran, S., and Pratt, M.J. (1993) Automated feature recognition from 2D drawings. *Computer-Aided Design* vol. 25(1): pp.7-17.

Mitchell, W.J. (1990) A new agenda for computer-aided design, in McCullough, Mitchell, and Purcell (ed.), *The electronic design studio*, pp.7. Cambridge MA, The MIT Press.

Ramscar, M. (1994) Static models and dynamic designs, an empirical impasse vs. an inductive solution, in R.J.Scherer (ed.), *Product and process modelling in the building industry* (proceedings ECPPM '94), pp.69-76. Rotterdam, Balkema.

Shah, J.J., and Rogers, M.T. (1988) Functional requirements and conceptual design of the Feature-Based Modelling System. *Computer-Aided Engineering Journal* vol. 5(1): pp.9-15.

Shah, J.J. (1991a) Conceptual development of form features and feature modelers. *Research in Engineering Design* vol. 1991(2): pp.93-108. New York, Springer-Verlag.

Shah, J.J. (1991b) Assessment of features technology. *Computer-Aided Design* vol. 23(5): pp.331-343.

Shah, J.J., Hsiao, D., and Leonard, J. (1993) A systematic approach for design-manufacturing feature mapping, in Wilson et al. (ed.), *Geometric Modeling for Product Realization*, pp.205 - 221. Amsterdam, North-Holland.

Tolman, F.P., and Wix, J. (1995) Building Construction Core Model, ISO/WD 10303-106. *Industrial automation systems and integration - Product data representation and exchange*.

van Leeuwen, J.P., and van Zutphen, R.H.M. (1994) Architectural Product Modelling, A Case Study. *Proceedings CIB W78 Workshop 1994*. Helsinki, CIB W78.

van Leeuwen, J.P., Wagter, H., and Oxman, R.M. (1995) A Feature based approach to modelling Architectural Information. *Modeling of buildings through their life-cycle* (proceedings workshop CIB W78 '95), pp.260-269. Stanford University, CIB W78.

van Zutphen, R.H.M., and Mantelers, J.M.M. (1996) Computational design: simulation in Virtual Environments. *Proceedings of the 3rd Conference on Design and Decision Support Systems in Architecture and Urban Planning*. Spa, Belgium, August 18-21, 1996.

Watson, A., and Crowley, A., et al., eds. (1994) CIMSteel: The Logical Product Model (LPM) version 3.3, ISO TC184/SC4/WG3/N319. *Industrial automation systems and integration - Product data representation and exchange*. ISO TC184/SC4.

Wilson, P.R. (1993) A view of STEP, in Wilson et al. (ed.), *Geometric modeling for product realization*. Amsterdam, North-Holland.

Wittenoom, R., ed. (1995) *Special Requirements of STEP AEC/B&C APs*. http://www.wt.com.au/~ausstep/stepbc/aecforum/aec_spec.html.

Woestenenk, K. (1995) *Proposal for international conversion tables for parts and functions*. The Netherlands, STABU.