Virtual Reality Design Information System : distributed interactive simulation

Virtual Reality
Design Information System /
Distributed Interactive Simulation

Henri Achten, Joran Jessurun,
Jos van Leeuwen, Bauke de Vries

Design Systems Report 1998 / 1

Design Systems Reports

This is a series of reports of the
group Design Systems (Ontwerp Systemen),
Faculty of Architecture, Building, and Planning,
Eindhoven University of Technology.
ISSN: 1388-9591

editors:
dr.ir. Henri H. Achten
ing. Jan Dijkstra
ir. Jos P. van Leeuwen

Copies can be ordered from:
Secretary of the group Design Systems,
Eindhoven University of Technology,
Faculty of Architecture, Building, and Planning,
Main Building 4.05
P.O.Box 513
5600 MB Eindhoven
The Netherlands

CIP. GEGEVENS KONKLIJKE BIBLIOTHEEK DEN HAAG

Virtual Reality
Design Information System /
Distributed Interactive Simulation

Henri Achten, Joran Jessurun,
Jos van Leeuwen, Bauke de Vries
Eindhoven: Technische Universiteit Eindhoven,
Faculteit Bouwkunde, groep Ontwerp Systemen.
ISSN: 1388-9591

ISBN: 90-6814-083-3

Keywords: VR-DIS; CAD; Feature-Based Modelling;
Constraint Solving; Design Studio; DDSS.

Contents

Virtual Reality Design Information System / Distributed Interactive Simulation

0. Introduction

Specific tasks of the design process are effectively supported by computer applications. Nevertheless the design process as such has not changed much under the influence of these applications. Technical issues and reluctance of traditionally trained designers prevent the introduction of new design technologies. As with the introduction of other new technologies (such as electricity) it takes quite some time to find the proper application. The first step in this process usually is to apply a new technology in a conventional way. It seems that the application of computer systems in designing is now at that stage. The 'natural' use of computers and the 'natural' interface with computers has still to be found.

Considering architectural design systems there is a very long tradition in solving design problems. Even before the introduction of computer systems in architectural design much research was spent on systematically describing the design process. Especially the creative part of the design process is very hard to be grasped in a formal way. Though one should not intend to formalize the design process, it seems obvious that specific tasks within the design process could very well be supported by computer applications. The approach towards the design problem using nowadays applications has not changed very much. To make effective use of the computer environment new design technologies must be developed that fit its nature.

A relatively new medium of interacting with computer systems is Virtual Reality (VR). Virtual Reality offers new opportunities for the creation of architectural design systems. Though at the moment VR is mainly used for visualization, it is evident that an immersive three-dimensional workspace allows for new design approaches. If the VR interface is combined with the power of a computer to handle large amounts of data, a new way of designing will emerge. This Design Systems Report provides an outline of the basic work for developing such a VR-Based design system; the so-called VR-DIS research program.

The acronym VR-DIS has two meanings: Virtual Reality Design Information System, and Virtual Reality Distributed Interactive Simulation. With the term Design Information System, we indicate a design system that uses Virtual Reality technology to provide the designer with relevant design information in an interactive and immersive way. Design information ranges from the visual impact of a building (what does it look like) to its behavior (thermal, cost, production, etc.). With the term Distributed Interactive Simulation we indicate the use of such a system by multiple persons (design teams and users), and evaluate how a design will behave when realized.

This report has three sections, each of which deals with a particular subject in the research program.

Section I: An Appraisal of VR-Technology Relative to Current CAD systems

Virtual Reality being a relatively new technology receives much attention in scientific research as well as in public news media. In the first case because of the technical problems that still have to be solved to offer the functionality and performance required by the application developers, and in the second case because of the appealing interface between the application user and the computer system.

Design in general and especially architectural design has a long-standing tradition in solving design problems by mixing artistic and scientific approaches. Research has resulted in design theories and methodologies to better understand the design process and to improve the quality of the designed product.

Before applying Virtual Reality as an enabling technology for design the characteristics of the technology and the application must be specified. This is not an easy job since VR is not yet a very clear and established field, and design is a very complex process. From the knowledge and experience in the Calibre research group and the Design Systems group a first draft of specification is presented so that we can draw some conclusions on the applicability of VR for design. This section provides an assessment of the possible use of VR related to current CAD systems.

Section II: VR-DIS Research Program Issues

At the department of Architecture, Building, and Planning, a research program called VR-DIS (Vries 1997) is established to pursue an innovative multidisciplinary design system. We believe that VR as a User Interface will probably replace many of the existing techniques due to the (extra) possibilities that are available to communicate intuitively between man and machine. Especially in the architectural environment the challenge lies in developing a new work environment in which the design process can take place (Vries and Achten 1998).

A case study of a specific architectural design process has been done to ensure that the developed design system supports design actions at least for that case stage and probably also for other design tasks. First the case is described informally simply registering all design actions and design output. Secondly the case is described formally using a specification language that has especially been developed for these purposes. As a result a formal description of the design model becomes available that can be implemented using modern software engineering techniques.

In this section conclusions will be drawn from the case description about the requirements of a design system for the very early design phase. To conclude, two theoretical concepts are introduced for managing design information and design support. An experimental design system will be constructed that supports the design process in a 3D environment using VR as a technology. The user interface will be described and the underlying design information structure. Concluding the first experiences with the design system will be discussed.

Section III: Application in the Design Studio
New communication media enable new design technologies. To investigate the mutual influence of a new medium like Virtual Reality (VR) and architectural design technologies, a laboratory called the design studio is established. In due time a design system will be developed offering different design technologies in an integrated environment. Interfaces within VR will support various approaches to the design problem. A large research program (Virtual Reality - Distributed Interactive Systems) is carried out to provide state-of-the-art tools for designers to experiment and to give feedback on the results. Section III describes the preliminary setup of the Design Studio.

SECTION I

An Appraisal of VR-Technology Relative to Current CAD systems

1. Characteristics of Architectural Design Problems

The nature of design problems has been subject of prolonged study, starting in the early 1960's (Cross 1984). Design problems are generally characterized as 'ill-defined' (Ibid., p. 145-166), or even 'wicked' (Ibid., p. 135-144) indicating for example a lack of explicit procedures for getting results, a lack of hard criteria for testing solutions, a large number of possible solutions to a problem, etc. Since design problems can not be solved in one run, the design process is lengthy and iterative. The study of design is undertaken from a number of perspectives (Oxman et al. 1995):

- from the design disciplines themselves, in order to achieve understanding of the actual design process,
- from the perspective of computer science, with a focus on design aid systems, artificial intelligence in design, expert systems, etc., and
- from the perspective of design research, with a focus on design as a general cognitive activity.

In the domain of architectural design, characteristics of design are related to the complexity of the design task (encompassing many levels of scale ranging from structural details to urban design), the design context (varying amounts of participants with different mandates and conflicting interests), and the design process (open and flexible). Furthermore, the knowledge available to solve (sub)problems in architectural design varies considerably in quantity and quality (Heath 1984, p. 55). All these characteristics point out that architectural design cannot be undertaken in a general problem-solving manner.

During the long tradition of architectural design, the discipline has generated various approaches to tackle the problematic properties of design. A large number of these concern the use of prestructured solutions to general design tasks. Rowe (1987, p. 80-91) identifies so-called 'heuristics'[1] which aid in constraining the possible solutions to a design problem, and which indicate ways of solving the task. Rowe distinguishes between "anthropometric" and "literal analogies" (transferring properties of an example to the design problem at hand), "environmental relations" (taking constraints from the immediate context of the design problem), "typologies" (generalized complete solutions for particular tasks), and "formal languages" (additional constraints relative to some style).

In this section of the report, we will discusses design from the perspective of computer aided architectural design (CAAD). Research in CAAD has a long history[2], focusing on almost every aspect of design. We will direct our attention to innovative design support that addresses the design approach of architects. Such innovative support must deal with the heuristics mentioned above, since a significant amount of architectural design knowledge is incorporated in them. Design support tools need to resemble features of such heuristics and design techniques, or at least enable the designer to work with them.

However, it is also necessary to note that design tasks are becoming increasingly more complex (number of specialists, requirements, norms, technologies, etc.) and that they have become increasingly more difficult to solve with current heuristics. Therefore, it is necessary to focus on new techniques as well. In general terms, issues that need to be addressed in innovative design aid systems are:

- creation of form,
- design reasoning and inference,
- communication,
- kinds of representations, and
- design techniques and heuristics.

In our opinion, Virtual Reality (VR) technology can play an important role to assist and support the architect in both existing and new ways to deal with design. In order to identify the possible contribution of VR in design aid systems, we will compare it with the current CAAD paradigm. This enables us to highlight the particular properties of VR and to point out its advantages over other approaches. In order to do this, it is necessary to establish a list of aspects on which to compare the CAAD and VR paradigm.

---

[1] Note that the term 'heuristics' here is not equivalent to heuristics in problem-solving such as hill climbing, means-ends analysis, generate and test, and pattern matching (Akin 1986, p. 98).

[2] Advances in CAAD research are documented in conferences such as CAAD futures (see Pipes 1986; Maver & Wagter 1988; Lansdown & Earnshaw 1989; McCullough, Mitchell & Purcell 1990; Schmitt 1991; Beheshti & Zreik 1993; Flemming & van Wyk 1993; Tan & The 1995; and Junge 1997).

2. Aspects of Architectural Design Aid Systems

Aspects for CAAD systems can be derived from general software development. Sommerville (1992, p. 66) proposes five classes of requirements which are termed 'viewpoints':

1. Functional viewpoints: the functionality of the system. Bounding viewpoints result from the actual use of the system and defining viewpoints result from general considerations that can not be directly inferred from the actual use of the system. Examples of functional viewpoints are: client requirements, hardware implementation, kinds of documents used, precision/accuracy of work, kinds of hardcopies, enable conceptual design, support current norms and laws, evaluation functions, speed of feedback/result, etc.
2. Non-functional viewpoints: constraints that do not primarily influence functionality. Examples are cost, security, safety, reliability/robustness, efficiency, etc.
3. User viewpoints: information concerning the user of the system. Examples are skilled or novice designers, particular profession in construction industry, visual representation, ease of learning, etc.
4. Data viewpoints: information concerning the data structures and data flow of the system. Examples are model database, information handling, communication, knowledge-use, exchangeability, etc.
5. Service viewpoints: information concerning use and maintenance. Examples are hardware/software maintenance, (remote) updating/diagnostics, low tech maintenance, etc.

Since the interest is aimed towards support of the architectural design process, viewpoints (2), (4), and (5) will not be considered. The functional viewpoint (1) will be defined in terms of tools, production aim, and representation. The user viewpoint (3) will be more narrowly defined in terms of interface technology. In this manner, the aspect of functional viewpoints deals with *what* a design aid system should be able to do, and the user viewpoint deals with *how* a design aid system should do that.

In order to assess VR relative to current CAD systems, the design function requirements and .the VR interface possibilities will be related and valued.

2.1 Functional Aspects

The present discussion of functional aspects is narrowed down to three categories: tools, production aim, and representation.

Tool aspects deal with various kinds of functionality that design aid tools can have. Schmitt (1990) distinguishes between four classes of computerized tools for architectural design. They can be viewed as stating four different kinds of functionality in terms of design support:

- generic tools: tools that perform a given task regardless of its content (*e.g.* word processors, drawings programs, hypermedia, etc.),
- parametric modeling tools: tools that incorporate design knowledge in structures which allow a range of values for their parts (*e.g.* stairs, classical orders in facades, etc.),
- prototype editors: tools that incorporate design knowledge of complex objects with particular properties (*e.g.* tools for particular kinds of houses, buildings, building parts, etc.), and
- shape grammar generators: tools that generate objects belonging to specific formal languages (*e.g.* Palladian villa's, Victorian houses, etc.).

Production aim concerns the required documents that need to be produced during the design process. These are linked to the stages of the design process. These stages can be viewed as stating seven kinds of functionality in terms of required documents. Schmitt provides the following schematic subdivision of the design process[3]:

- program development,
- schematic design,
- preliminary design,
- design development,
- contract documents,
- shop drawings, and
- construction.

Representation deals with relevant representations that a design aid system must support:

- two-dimensional: (2D) representations such as plan, section, facade,
- three-dimensional: (3D) *e.g.*, perspective, isometric or axonometric projection,

[3] The summary is made in general terms since the design phases, as well as the particularities of each kind of document vary to some degree for each country.

- static: drawing of desired state, and
- dynamic: time, behavior, and interrelationship of elements of the design.

Although the particular kinds of tools, production aim, and representation mentioned here are specific for the discipline of architectural design it seems possible to generalize these requirements to other design disciplines. It can be claimed that other disciplines also make use of pre-structured knowledge in various degrees which offers functionality, that their processes are ordered in design phases which yield specific documents, and that they make use of a limited set of representations to proceed during the design process. To summarize, the functional aspects are:

TOOLS
1. Generic functionality [multi-purpose, wide range of application].
2. Parametric functionality [structures that encode standardized routine knowledge/design].
3. Prototype functionality [structures that bring coherence to large relatively complex wholes].
4. Shape grammar-editing functionality [implementation of formal languages].

PRODUCTION AIM
5. Brief [stated list of requirements of the design in terms of spaces, cost, time, parties, mandates, etc.] .
6. Schematic design [sketch design showing principal solution of the design].
7. Preliminary design [worked-out drawings that show most design decisions].
8. Contract documents [drawings and text for putting out to tender].
9. Shop drawings [detailed drawings of the whole design].
10. Construction drawings [specification of the construction process of the design].

REPRESENTATION
11. 2D [plan, section, facade].
12. 3D [perspective, isometric, axonometric].
13. Static [drawing, desired state].
14. Dynamic [time, behavior, interrelationship].

These aspects of functionality are relevant for design support as discussed in paragraph 2: the *tools* deal with heuristics and design techniques; *production aim* deals with the practice of design and the design process; and *representation* deals with the way design problems are treated throughout the design.

2.2 Interface Technology

The user viewpoint is narrowed down to interface technology. Coomans (1997) establishes the following aspects of user interface: (1) interaction: interactive manipulation of the design in ways that are natural for the architect and which do not draw attention to the handling of the interface; (2) immersion: three-dimensional real-time spatial experience of the design; (3) simulation: visual, acoustic, and haptic simulation of the design for relevant feedback of design decisions; and (4) visualization: visualizing information that is non-visual by nature (such as heat transfer, cost, structural forces, etc.).

To summarize, the interface technology aspects are:

1. Interaction [real-time feedback, natural interface, unobtrusive design actions].
2. Immersion [spatial assessment, three dimensional representation, real-time movement].
3. Simulation [calculation and rendering of (physical) properties of the design].
4. Visualization [representing non-visual information in a visual manner to provide feedback].

These aspects of user interface are relevant for design support as indicated in paragraph 1. *Interaction* refers to the cognitive aspect of the design process, where the architect is engaged in a time-consuming process of iterative cycles of design. In the early stages of design, many ideas are generated and tested in a fast manner. Often these are created by means of sketching, where the rate can exceed 20 drawings per hour (McCall, Johnson and Smith 1997, p. 851). The point here is that design is a process that requires a large degree of interactivity if it is to be well supported in a design aid system. *Immersion* deals with the spatial appreciation of design decisions in the process. *Simulation* deals with the consequences of design decisions, and *visualization* refers to the fact that architects predominantly use visual information and are capable to assess decisions visually more able than for example, numerical information.

3. A Comparison of CAAD and VR Design Systems

Before elaborating on the functional requirements in relation with Virtual Reality, CAAD tools are introduced since they offer a reference enabling us to compare the current status of design systems with a proposed future system.

The functional and interface aspects lead to eighteen aspects on which to compare current CAAD tools and the proposed VR system. Before doing so, we will define the CAAD and VR paradigms.

- **The CAAD Paradigm:** A CAAD system is a computerized system that produces engineering and design drawings. Models are constructed on the basis of primitives using 2D-drawing analogies from the design practice (lines, symbols, hatching, building elements). The interface is controlled by hand-manipulations (mouse, keyboard) on a flat plane (move, drag, pull-down). Domain knowledge is coupled to an entity-level database. The CAAD paradigm is exemplified in systems such as AutoCAD, Arkey, Microstation, etc.
- **The VR Paradigm:** A VR system is a computerized system that produces engineering and design three-dimensional models. Models are constructed using solid modeling techniques and scale-modeling analogies from the design practice. The interface is controlled by the body (glove, head, position, movement) in space (move, drag, scale, mould) featuring immersion and real-time feedback. Domain knowledge is incorporated in both the design process and in a hierarchic multi-level database. The VR paradigm is exemplified in systems using head-mounted displays, data gloves, pointer devices, real-time rendering and textured surface models.

Since many aspects are not stated quantitatively, the comparison of CAAD and VR design systems goes no further than differentiating between '+,' '-,' '0,' and 'n/a':

- '+' Means that the paradigm meets the aspect and that it in fact enhances it.
- '-' Means that the paradigm does not meet the aspect and that it in fact obstructs it.
- '0' Means that the paradigm neither enhances nor obstructs the aspect.
- 'n/a' Means that the paradigm does not feature the technology or aspect, and can therefore not be assessed.

VR technology is relatively young. It has not been applied very much in design aid systems but almost exclusively for presentation when the design is finished. Therefore there is no established standard of VR-based design systems, which makes assessment difficult. Given the appeal of new technologies, it is tempting to state that all aspects will profit from application in VR. This however, is most probably incorrect. Experience from the past has learned that efforts to realize design systems for everything and for everyone have failed. The practice of architectural design necessarily leads to specialized and well-understood design systems, which cover only partial stages of the design process. Furthermore, it is necessary to point out that the list of requirements is largely based on current practice and state-of-the-art of design aid systems. They do not necessarily reflect state-of-the-art in the near future. This issue will be addressed in the next paragraph.
The comparison results in the following matrix of comparison:

Aspects - Interface matrix applying Virtual Reality technology:

		Interaction	Immersion	Simulation	Visualization
	Generic	-	-	-	o
TOOLS	**Parametric**	o	o	-	+
	Prototype	+	+	+	+
	Shape grammar	+	+	o	+
	Program develop.	o	-	o	o
	Schematic design	+	+	+	+
PROD. AIM	**Prelim. design**	+	+	+	+
	Contract docum.	-	-	-	-
	Shop drawings	-	-	-	o
	Construction	-	-	+	+
	2 Dimensional	n/a	n/a	n/a	n/a
REPRESENT	**3 Dimensional**	+	+	+	+
	Static	-	o	o	+
	Dynamic	+	+	+	+

Aspects - Interface matrix applying traditional CAAD technology:

		Interaction	Immersion	Simulation	Visualization
	Generic	-	n/a	n/a	+
TOOLS	**Parametric**	-	n/a	n/a	+
	Prototype	o	n/a	n/a	+
	Shape grammar	o	n/a	n/a	n/a
	Program develop.	-	n/a	n/a	-
	Schematic design	o	n/a	n/a	-
PROD. AIM	**Prelim. design**	o	n/a	n/a	o
	Contract docum.	-	n/a	n/a	-
	Shop drawings	-	n/a	n/a	+
	Construction	-	n/a	n/a	+
	2 Dimensional	o	n/a	n/a	+
REPRESENT	**3 Dimensional**	o	n/a	n/a	o
	Static	-	n/a	n/a	+
	Dynamic	n/a	n/a	n/a	n/a

From the Aspects – Interface matrix applying Virtual Reality technology respectively CAAD technology, the following conclusions can be drawn:

1. 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.
2. CAAD technology shows the best performance in the final design stage, using a 2-dimensional representation.
3. CAAD tools offer good visualization of the design but very poor natural interaction with the user.
4. VR technology is not suitable for production of the traditional documents used for information exchange.
5. VR has the most potential in those areas where CAAD has a poor performance.
6. Generic and parametric design tools are not adequate for Virtual Reality.
7. CAAD does not feature Immersion and Simulation.

A danger of exploiting new technologies on current functions is that only existing functions are implemented. More challenging and probably more promising are the research for new functions that emerge from a new technology such as Virtual Reality.

Apart from prototyping and shape grammars new tools are required to implement the interface between the design system and the designer. These tools are based on the use of three-dimensional metaphors as opposed to the two dimensional metaphors (like Windows) of the user environment of current computer systems. Especially the development of these metaphors in combination with the peripherals to establish the man-machine interface will contribute highly to the adaptation of the Virtual Reality paradigm.

Relating the traditional production aim with a new technology may result in wrong conclusions. It is concluded that VR shows poor performance in producing documents. Instead of trying to solve this shortcoming, research should focus on elimination of the need for these documents. Thus the means should be developed to present information using versatile digital media and to transfer information using electronic networks. Again, the availability of these means will contribute highly to the adaptation of the Virtual Reality paradigm.

We would like to conclude the current section on the appraisal of VR-technology relative to current CAD systems with the following 'predictions' about the future of CAAD systems:

1. VR will replace the 2D user interface.
2. VR is the 'natural' environment for prototyping and for creation of shapes.
3. Designers have to design their own (3D) environment to execute design activities.
4. Experiencing design behavior is only possible using VR.

In the next section, the research program established around the VR-platform in the Design Systems Group is discussed.

SECTION II

VR-DIS Research Program Issues

4. Introduction

At the Eindhoven University of Technology a research program called VR-DIS (Vries 1997) is established to pursue an innovative multidisciplinary design system. As stated above, we believe that VR as a User Interface will probably replace many of the existing techniques due to the (extra) possibilities that are available to communicate intuitively between man and machine. Especially in the architectural environment the challenge lies in developing a new work environment in which the design process can take place (Vries and Achten 1998). The research in the Design Systems group therefore, is aimed at defining the foundations of a VR-based design system. Investigating fundamental issues and implementing a design system seems a good strategy to get an understanding of the complex task.

A case study of a specific architectural design process has been done to ensure that the developed design system supports design actions at least for that case stage and probably also for other design tasks. Conclusions will be drawn from the case description about the requirements of a design system for the very early design phase. Two theoretical concepts are introduced for managing design information and design support. An experimental design system will be constructed that supports the design process in a 3D environment using VR as a technology. The user interface will be described and the underlying design information structure. The first experiences with the design system will be discussed.

5. A Case Study of the Architectural Design Process

The case describes 31 design phases. First the case is described informally simply registering all design actions and design output. Secondly the case is described formally using a specification language that has especially been developed for these purposes. As a result a formal description of the design model becomes available that can be implemented using modern software engineering techniques. For an in depth explanation of the case see (Achten and van Leeuwen 1998). The experimental design system described in this paper focuses on the first part of the case. In this part of the design process the requirements that have been elicited from the principal are 'translated' into a design containing the main building elements and spaces. Primary goals of this design phase are:

- check if all requested rooms and spaces fit on the building site,
- check in communication with the principal if the requirements list is complete,
- get an first impression about possible orderings of rooms and spaces,
- get a first impression about the layout of the plan,
- get a first impression of the facade including windows, doors and material outlook, and
- get a first impression about the possibilities of the roof shapes.

To capture the design process that is taking place, the subsequent phases will be analyzed with respect to the followings aspects: buildings elements, relations between building elements and conditions that must be met by building elements, presentation of the design information and finally the design process moving from one phase to the next.

Excluded from the analysis is all information that is related to the administration about the project such as the name of the customer and the status of the project.

In general, requirements can be in conflict with each other (*e.g.* all rooms on the ground floor whilst there is not enough space on the building site). Therefore some requirements may need relaxation during the design process. Not all requirements are described explicitly in the brief (*e.g.* the living room should be on the ground floor). They are filled in by the designer and checked afterwards with the principle.

5.1 Building Elements

The principal expresses his/her wishes using nouns to specify rooms (*e.g.* kitchen) and verbs to specify activities that will take place (*e.g.* living). Dimensions or floor areas are left to the designer to fill in. For some rooms the interior is described in more detail (*e.g.* the bathroom should contain a shower, a bath, a toilet and two wash basins). The principal has some explicit wishes about the material for particular building elements (*e.g.* brick walls and a wooden construction in the veranda).

5.2 Relations and Conditions

For some spaces and rooms the location is restricted in relation to other spaces or rooms (*e.g.* veranda is an extension of the living room). For some building elements additional conditions are specified (*e.g.* access of a room, material of the floor). The locations of the walls of the rooms are restricted to a grid that is chosen by the designer. For the main building volumes two grids are used, one of them slightly rotated.

5.3 Presentation

Only the first contact with the principle is documented containing the main list of requirements. From that stage on, all design information is documented by using drawing techniques. The first sketches are created to check the total floor area of the plan related to the area of the building site. At the same time some room ordering solutions are investigated to find a cluster of rooms that can be combined into one building volume. Of some building variants the outlook of the facade and the shape roof are presented.

5.4 Process

During the design process, sketches and drawings are produced to get more insight about the composition of rooms and the shape of the main building elements related to the activities that will take place. There is no strict order in which the plans are produced, but rather there is a rapid shift from analyzing one aspect after another.

6. Functional Specification

To describe a design system which supports the depicted design phase of the case, the system functionality is specified looking at the following aspects: building element types, operation types, relation and condition types and presentation.

6.1 Building Element Types

The kind of building elements that are manipulated by the designer can by classified into the following categories:

1. Spaces: Spaces are elements in which people stay or live and that are partly enclosed by physical boundaries or that have no physical boundary at all (*e.g.* hall, dining place).
2. Structures: Structures are all elements that are not spaces and that do have physical boundaries (*e.g.* floor, column).
3. Infill: Infill components are structures that are considered not to be part of the permanent structure of the building (*e.g.* stair, chair).
4. Ground: Elements belonging to the ground category are those elements that are part of the environment of the building (*e.g.* garden, building site).

6.2 Operation Types

Building elements can be created, moved, rotated and scaled. The shape of the rooms is rectangular as a starting point. The shape can be adjusted to fit with other shapes. This means 'wrapping' a shape around another shape or penetrating one shape into another.

Roofs of different shapes can be placed on top of the rooms in case of studying the facades or the building volumes. Floors are generated fitting the room sizes.

A control mechanism is required for locating one building element relative to another one. (*e.g.* one room next to the other). This functionality is especially required for those cases in which the principle expressed this kind of relationship. Implicit relationships between building elements supports the designer in positioning a new building element related to existing ones (*e.g.* surrounding a space by adding walls).

Building elements can be snapped to an (active) grid. The systems 'senses' the neighborhood of a grid line and will position the element according to the grid.

Elements can be grouped together to perform a single operation on a group of elements (*e.g.* changing the shape of all roofs)

Material properties can be attached to selected building elements.

6.3 Relation and Condition Types

Geometrical relationship types that can be recognized between the building elements are:

1. space *adjacent* to ground (*e.g.* veranda adjacent to garden),
2. space *in* space (*e.g.* sitting place in living room),
3. space *adjacent* to space (*e.g.* kitchen adjacent to garage),
4. infill *in* space (*e.g.* toilet in bathroom),
5. space *on* top of space (*e.g.* roof on garage),
6. space *on* top of ground (*e.g.* first floor on building site), and
7. structure *align* space (*e.g.* walls of the living room).

A special kind of geometrical relation type is the restriction of the location of walls on grid lines. The access condition type is translated into geometrical relations during the design (*e.g.* by locating two rooms next to each other and adding a doorway). Conditions that reflect only one building element are described as part of the specification of that building element (*e.g.* outer walls are made of brick).

6.4 Presentation

The kind of presentations that should at least be supported are plan view, perspective view and facade view. Rooms can be identified by displaying a description. Secondly, the use of rooms can be recognized by placing the appropriate components in it (*e.g.* a bed for a bedroom). In general the designer should be able to have a complete overview of the building as well as have some impression about the dimensions of rooms. Already in this stage the infill of the wall openings of the outer walls (door and windows) plays an important role, in particular their size and location. The material requirements as far as they are known should be visible.

The grid can be made visible and active using a toggle. To get a view on the layout of the floor plan, horizontal cross sections of all building elements are generated at a standard height above a floor.

7. Concepts

Two theoretical concepts have been selected to support the management of design data and the manipulation of geometries. Both concepts, Features and Constraints, will be discussed briefly. For an extensive explanation see (van Leeuwen and Wagter 1998) and (Kelleners, Veltkamp, and Blake 1997) respectively.

7.1 Features

Feature Based Modeling finds its roots in Mechanical Engineering. In this discipline Features are almost synonymous with Form Features. Form Feature descriptions are closely related to manufacturing of these forms. In an ongoing Ph.D. study at the Eindhoven University in the Design Systems Group (van Leeuwen and Wagter 1998) the original concept has been applied to architectural design. The main purpose is to investigate if the concept can support flexibility and extensibility.

Flexibility is found to be a key characteristic of the architectural design process. The focus in architectural design shifts between different levels (*e.g.* urban plan, building, room) and different aspects (*e.g.* physical, structural). Design decisions are taken in a cyclic process focusing on a part of the complete design task (*e.g.* the ground floor). Thus, changing a design and its underlying representation is in fact part of the creative process. Moreover design decisions are often dependent on each other. Therefore the design representation must support the linkage of characteristics of design parts (*e.g.* the color). In that case changing a parameter of one building element is propagated to another (*e.g.* all internal doors share the same color). In Feature-Based Modeling any Feature (characteristic) of a building element can be shared by any other building element. For this purpose three relationship types are introduced: Specification, Decomposition and Association. Specification specifies a certain characteristic of a Feature (*e.g.* color). Decomposition declares a building element being a part of another building element (*e.g.* hinge is part of a door). Association is any relation that cannot be expressed by the former two relationship types (*e.g.* a door in wall relation).

The second key characteristic of the architectural design process is the fact that a design evolves in time. Descriptions of building elements are often rather abstract and incomplete in the early design stages. Gradually design parts are refined and design alternatives are accepted or rejected. Sometimes the design process is reversed when reconsidering previously found solutions. All the time new design decisions must be checked for consistency with other design decisions. The representation must support registration of design solutions of the same building element in different stages. So design solutions should never be destroyed unless the designer explicitly wants to do so. In Feature-Based Modeling, Features can be related with each other using the Inheritance relationship. Inheritance specifies a subtype having all the characteristics of the supertype and if necessary one or more extra. Moreover, each Feature is time stamped and carries the name of the creator (an identification of the designer).

A third key characteristic of architectural design is the fact that most designs are unique. Therefore, describing a generic model of a building appears to be an impossible task. To counter this problem, in the proposed Feature-Base-Modeling theory Features can be created on two levels: Feature Type level and Feature level. On Feature level Feature Types are instanced into Features. The designer is offered libraries of predefined Feature Types, but is also allowed to create his/her own Feature Types or Feature Sub Types. In this way standardization can be pursued by the creation of libraries without limiting the designer and refraining him/her of new innovative design solutions.

7.2 Constraints

Reasoning about geometrical relationships between building elements in general requires a common formal geometry description. In VR systems geometry representations for the surfaces of shapes usually consist of lists of triangles (strips). Computer hardware is optimized for displaying these triangles. Ideally constraints also would use the same triangles as primitives. As a start though, we found that Bounding Boxes containing building elements serve well as far as we deal with spatial relationships. Nevertheless there are some limitations; it helps us to focus on the development of algorithms for expressing spatial relationships instead of solving problems that have to do with the complexity of geometry descriptions. The theoretical approach we adopted for managing geometrical constraints is based on Allen's temporal interval algebra (Allen 1983). He recognizes five basic relationships that can exist between a point and an interval: ahead, front-touch, in, back-touch and behind. Because in our design system each of the local axes of the Bounding Boxes representing the building elements are always parallel with one of the global axes, it is appropriate to use interval-interval relationships. This lead to thirteen relationship types. (For a more in-depth explanation about 'Qualitative Spatial Relationships,' see (Gorti and Sriram 1995)). Constraints on a Bounding Box can be rewritten to relations between intervals in three sets, one for each axis.

In the constraint solver (Kelleners, Veltkamp and Blake 1997) five constraint types are recognized:

1. Connection constraints: Touch, AlignSides. These are constraints that show Bounding Boxes are connected with each other.
2. Distance constraints: OppDistance, LatDistance. These constraints specify the distance between two sides of Bounding Boxes.
3. Contain constraint: Contains. This constraint specifies that one Bounding Box should contain another.
4. Non-intersection constraints: NonIntersect. This constraint specifies that two Bounding Boxes should not intersect.
5. Unary constraints: FixPosition, FixSide, FixDimension, FixOrientation, MinDim, MaxDim. These are constraints that put restraints on the position, the dimension and the orientation of the Bounding Box. These constraints are no interval constraints but necessary for the solution process.

For example, a Contains-constraint which specifies that Bounding Box B2 should hold Bounding Box B1 can be rewritten to the following interval constraints:

$$I_b^{B1} < I_e^{B1} \text{ and } I_b^{B2} < I_e^{B2} \text{ and } I_b^{B1} < I_b^{B2} \text{ and } I_e^{B2} < I_e^{B1} \text{ and } I_b^{B1} < I_e^{B2} \text{ and } I_b^{B2} < I_e^{B1}.$$

Here, I_b^B and I_e^B define an interval belonging to Bounding Box B. The subscripted $_b$ en $_e$ mean the beginning and the end of an interval. For the Contains-constraint the above six constraints are created for all three axes.

Since the purpose of the design system is that constraints can be created between building elements rather intuitively there is no control mechanism to check whether the geometrical model is under-constrained or over-constrained. In an under-constrained situation the solving process will produce a large number of solutions that is hardly manageable by the designer. In a over-constrained situation the solving process will find no solution at all. To deal with this problem a solution method is chosen that will always calculate a solution that is 'as close as possible' to the former situation. This method also better relates to design support where an architect implicitly defines constraints between building elements by placing them as he does during the actual use of the system.

8. Implementation

8.1 System Architecture

The model in the system can be viewed in two distinct ways: in terms of Features, and in terms of a three-dimensional rendered scale model view (the typical VR-representation). Any action in either view is reflected in the other view: they refer to the same model. The Feature Model View uses a Relational DataBase Management System for managing the Feature Model data. The Scale Model View directly reads the geometry data from the Feature Model Store and the Geometry Store. The constraint solving process is triggered from the Scale Model View.

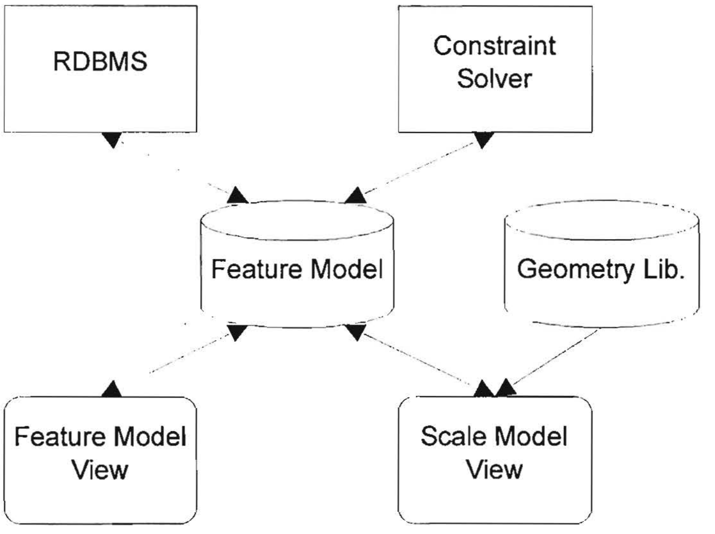

Figure 1: System architecture

8.2 Geometry

The geometry of spaces and structures will be generated from the data in the Feature Model. Infill components reference to a geometry file.

8.3 Constraint Solver

The geometrical relations between building elements will be implemented using constraints. To manage the constraint solver, spaces, structures and infill components are represented by bounding boxes. Five different high level design constraints can be recognized:

1. Building_Element A *adjacent* to Building_Element B: This relation will be implemented with the Touch-constraint. The axis along which the building elements touch must be specified (X or Z)). At first in the building element creation phase, this value will usually be unknown and so the axis will obtain a default value. The constraint will take care of having the building elements touch at one of the sides with a shared plane area > 0.
2. Building_Element A *in* Building_Element B: This relation will be implemented using the Contains-constraint. The constraint will take care of one building element being inside another. It does not express at which location.
3. Building_Element A *on* Building_Element: This relation will be implemented using the Touch-constraint. The axis along which the building elements will touch is always the Y axis.
4. Structure A *aligns* Space_Side B: This relation will be implemented using the AlignSides-constraint and the LatDistance-constraint. The sides (left, right, front, back, top, bottom) of the space and the structure must be specified respectively. The constraint supports the positioning and scaling of walls and floors as space boundaries.

Moreover, to support the design process the following constraints will imposed:

- All Infill components will get a FixDimension constraint for all dimensions. We presume that the components dimensions will not change during this part of the design process.
- All spaces get a MinDim and a MaxDim constraint. These constraints are deduced from the Length, Width and Height of a space by calculating 10 % respectively 200% of each.

8.4 Presentation and Operation

At the moment the presentation of design information consists of two interfaces. One interface for manipulating the Feature Model of the design and one interface for viewing the shapes of the design (the so-called Scale Model) and limited manipulation of the shapes.

8.4.1 Feature Model View

The Feature Model View provides full access to all design information modeled as Features. The Feature Model View consists of two main parts, one for editing Feature Types and one for Feature instances.

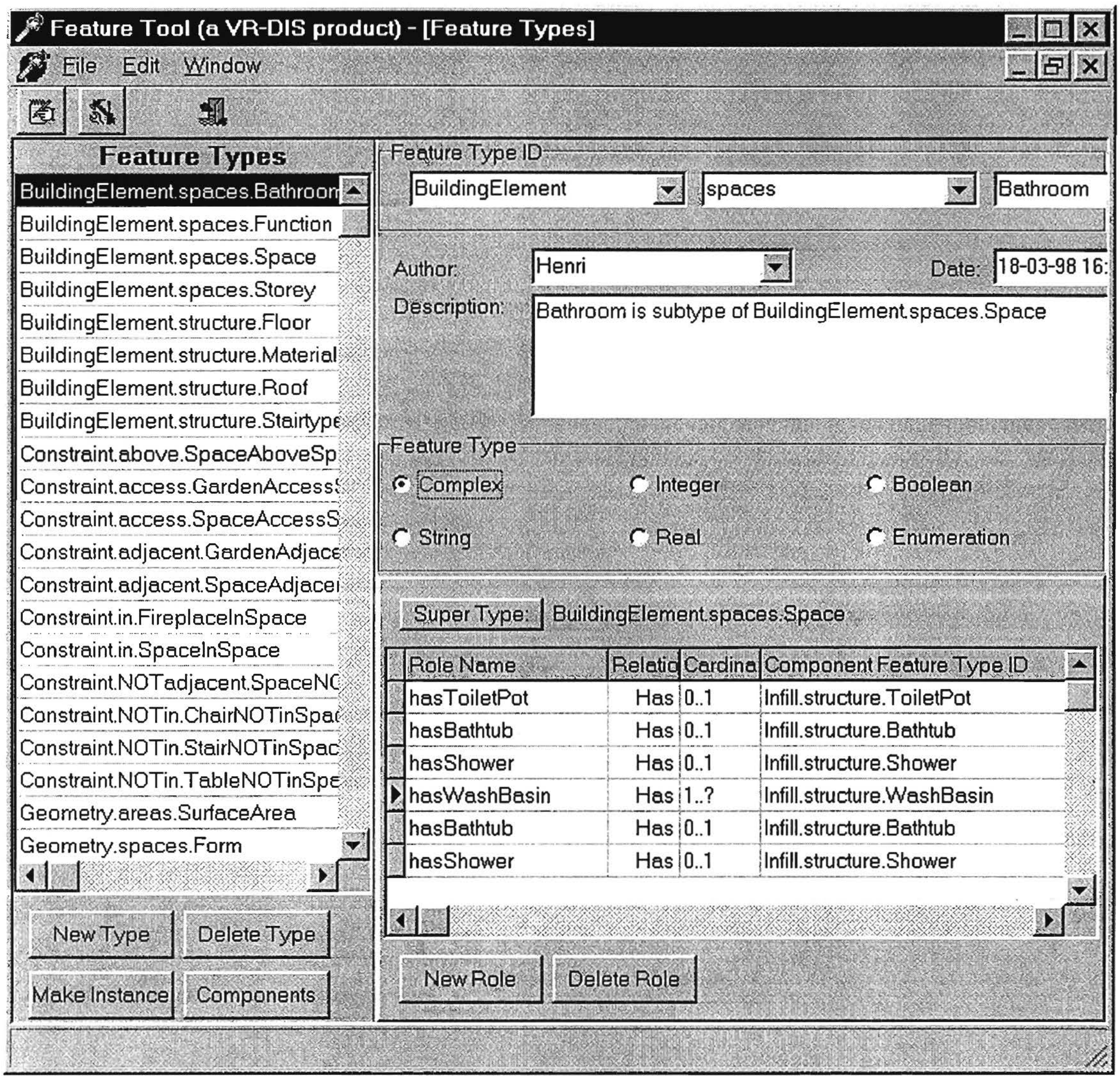

Figure 2: Feature view: Types.

The left part of the screen shows a list of already created Feature Types. In the right part a new Feature Type can be created by choosing one of the basic types (complex, string, integer, real, boolean or enumeration) and entering a description. Moreover the relations are specified by entering a role name, a relation type (has, spec, or assoc), the cardinality and the referenced Feature Component.

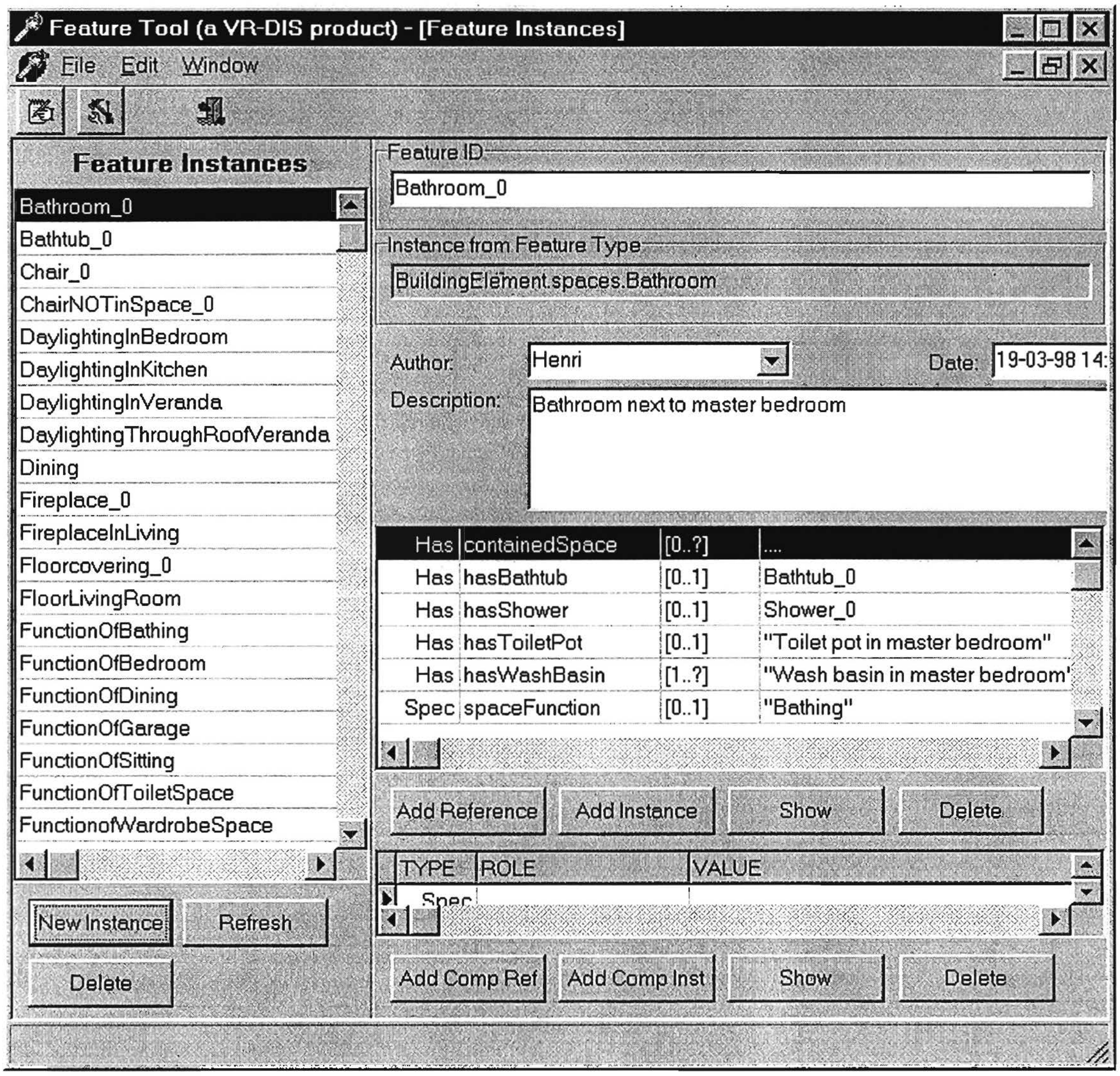

Figure 3: Feature view: Instances.

In the left part of the screen a list of already created Feature Instances is shown. A new instance is created by first selecting a Feature Type from a list. In the right part of the screen a name (Feature ID) and description is entered. At the same time the list is shown of all possible relations as defined for that Feature Type. Now these relations can be filled in by adding references to other (already existing) Feature instances.

8.4.2 Scale Model View

In the Scale Model View the following buttons are available: Perspective/isometric view, View direction (Top, Bottom, Front, Back, Left, Right), Solve constraints, Save model, Toggle object visibility, Object information, Move object, Scale object, Options (Toggle fix infill component dimensions, Toggle limit space dimensions)

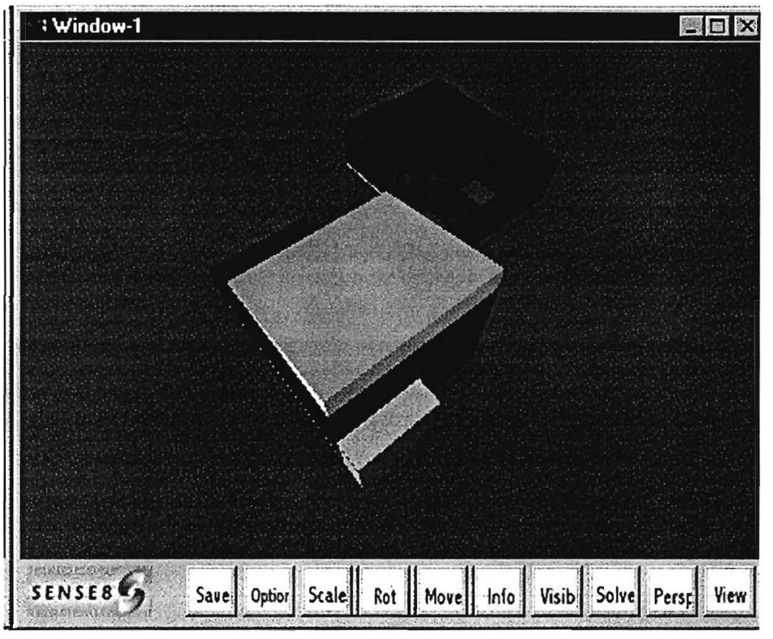

Figure 4: Scale-model view

Spaces are represented by their space boundary, being a box. Structures and infill components are represented by their specific shapes. Spaces, structures and infill components can be moved and scaled. Infill components can also be rotated. Each design phase is stored separately in a database file.

9. Discussion

9.1 Feature Model Characteristics

The characteristics of the Feature Model are most easily explained by comparing the modeling technique with Product Modeling as defined in the STEP standards. The Feature Model appears to be more flexible because of the possibility to create new Feature Types during the design session. Using STEP Application Protocols the product model is restricted to a certain application area and fixed for a period of time. Secondly, Features can be shared by other Features. Though the Express language allows for construction of dependencies, in the Feature model sharing attributes is maintained in a more direct manner, closely related to the way a design expresses these kind of conditions. Thirdly, relations between Features can be added at instance level. Therefore, the user is not limited to the definitions of the existing Feature Types but he/she can extend Features with any kind of relationship with other Features. In this way project-specific information can be handled.

9.2 Constraint Solver Characteristics

The constraint solver acts as a support tool for locating building element geometries at the desired location and keeping topological relationships intact. Lacking facilities like snap in this primitive experimental design system it is the only alternative next to typing co-ordinates. The constraint solver appears to be a good help, but more important is that the way of expressing building element positions by constraints is very close to the designer's way of thinking. Constraints in the described case are used for:

- Generating a first floor layout. Generating plans from requirements usually causes offensive reactions, because designers don't like their primary activity being automated. In this experimental design system on the contrary, the generated layout of the floor plan is not more than a starting point and only maintains those spatial relationship which are explicitly defined by the principle or implicitly by the designer. In practice such relationships are rather limited in number and therefore it is rather easy to understand the system's behavior.
- Placing building elements. Support for placing building elements can be found in more advanced CAD packages. Usually the shape, location and size of a geometrical object in such applications are defined by specifying parameters. Parameter specification is executed in a designerly way using techniques like object snap. In our experimental design system though, any object can be restricted in its dimensions and location. The kind of restriction that the designer wishes to impose depends on many factors. In the

experimental design system these factors are roughly classified by the distinction in four building element categories as mentioned in section 6.1 (Spaces, Structures, Infill, and Ground). This proves to be adequate for the envisaged design goal.

- Checking if imposed geometric relationships are validated. Instead of solving a design problem expressed by constraints, constraint validation will do in many occasions. The system should indicate which objects are part of a constraint that is invalid. Then the user can decide on which constraint to relax. This facility is not (yet) present is our design system.

9.3 Shortcomings

Comparing the initial system requirements with the experimental design system implementation one can deduce the following shortcomings:

- There is no support for the use of grids.
- Material properties cannot be added to building objects.
- Building element geometry cannot be transformed otherwise than by means of scaling.

Evaluating the experimental design system one should bear in mind that we deliberately focused on the development of the core of a design system that allows us to manipulate all information during the design process. In the near future we will extend the system with 'real' design tools and design knowledge (Mallory-Hill 1998). The Windows interface of the Feature View will be replaced by a VR Feature manipulation tool (Coomans 1998).

The next section will deal with the application of the developments of the VR-DIS research program in a laboratory setting for evaluation and testing of the work; the so-called Design Studio.

SECTION III

The Design Studio: Application and Test of Work in the VR-DIS Program

To experiment with VR based design systems, a design studio is furnished with state-of-the-art equipment. The equipment is used for constant development of new additions to the design systems. Experienced and inexperienced designers are invited to use the equipment for their design tasks to get feed back on the tools that are developed in the VR-DIS research program, such as the experimental design system described above. In this section the IT architecture of the design studio will be described, the theoretical background that it is based on, and the current status of the research.

10. Studio Architecture

The design studio can be regarded in two ways:

- The layout and equipment of the room or building where the studio is located. The current situation will be described at the end of this paper. However this will constantly change because of new hardware and software developments. A less implementation dependent way to describe the information architecture therefore is the second approach:
- A group of workstations where specific design tasks will be performed. *Design tasks* will be executed using one or more design techniques. Whether or not different design techniques can be used on the same computer equipment depends on the computer environment. *Design techniques* are closely related to the way the design object is represented. Design techniques can be used in any order by the designer to support a specific *design methodology.*

To explain the information technology (IT) underlying the design studio, the last approach will be followed. First, design techniques in an IT context will be elaborated. These techniques can be used in traditional presentation media and in new ones. Virtual Reality being a new medium will be analyzed with respect to the internal representation of information. To conclude, the interfaces necessary to manipulate entities in VR will be considered.

10.1 Design Technologies

Design techniques and design methodologies have been defined in order to create a clearer view on the design process and in doing so to support the designer while performing his/her design task. In this paper design technologies are introduced as a specific class of design techniques in an IT context. *A design technology is a specific way of creating and manipulating design objects that requires a specific design information representation.* Starting point in the survey for design technologies is the assumption that no traditional means (such as a sketch board and a pencil) will be used. By excluding them, we force ourselves to look for new ways of creating and manipulating design objects.

Design technologies that are distinguished:

- Sketching. *Sketching* is a design technology that produces shapes with no explicit dimensions. Since computers can only handle explicit values, the sketches have somehow to be translated into known geometric entities. The traditional approach to this problem is trying to find a way to make the computer recognize what the designer has sketched. The translation process is very difficult, especially the recognition of the meaning of the shapes (floor, tree, etc.).
- Drawing. *Drawing* is a design technology that produces shapes with explicit dimensions. Almost all CAD systems support this technology. Only few of them are capable of attaching a semantic meaning to drawing entities. As a precondition such drawing systems must be (building) object oriented.
- Simulation. *Simulation* is a design technology that tries to predict the behavior of the design. Behavior can be regarded towards behavior of the building such as structural behavior, thermal behavior, etc. It also can be regarded towards behavior of the occupants such as perception, way finding, etc.
- Managing. *Managing* is a design technology that refers to the manipulation of the design data or the data flow. Design data do not merely consist of geometrical data but also of additional attributes and functions that describe the characteristics and the behavior of a specific building object. Data flow exists within the building project and outside the building project. To keep the design information up-to-date and consistent, control is required for the information exchange process.

10.2 Presentation Media

Presentation media impose their characteristics upon the design technology being used. Designing a spatial object like a building in two dimensions is in fact very unnatural but nevertheless common in architectural practice. One of the main skills an architectural student acquires during his/her training is to create a three dimensional image from a two-dimensional sketch or drawing. Especially for those not familiar with reading building drawings perspective views are created to give an impression of the building. The introduction of CAD systems has not changed much in this practice. Design activity still takes place in two dimensions but now on the monitor screen instead of a piece of paper. Recently released design packages draw 3D (solid) objects, so images of facades etc. can be generated very quickly. As we argued in Section I, to make the next step in the development of design tools of an architect, a new presentation medium is required: Virtual Reality (VR). *Virtual Reality* is characterized by the illusion of participation in a synthetic environment rather than external observation of such an environment. VR relies on three-dimensional (3D), stereoscopic, head-tracked displays, hand/body tracking and binaural sound. VR is an immersive, multisensory experience (Earnshaw and Gigante 1993). In VR the architect (or engineer) designs the building as if he/she is creating a full-scale model. During the creation process the designer can take any position to work from such as in one of the rooms or standing in front of one of the facades. Obviously the designer needs appropriate tools to manipulate the building parts and to easily change view. In fact the design technologies sketching, drawing, simulation, and managing as mentioned in the previous paragraph should be supported.

10.3 Representation in VR

The application of Virtual Reality until now is mainly restricted to visualization of shapes including texture mappings. VR software (such as World Tool Kit and Device) is optimized for generating pictures of a building model consisting of objects with a shape representation and a texture mapping while navigating through the model. Moreover objects can exhibit some behavior like gravity and collision detection. With these capabilities very realistic images can be created of a building which only exists in the mind of an architect. The architect himself/herself or the principal can use VR to judge a design on its esthetical and its functional qualities.

Visualization in VR is dynamic but not interactive at all. To take VR one step further the designer must be able to interact with the design by creating, modifying and deleting design objects and by evaluating the design on certain aspects. Like in Object Oriented modeling, design objects can be identified and they will have a specific behavior (Booch 1994). VR offers new interesting possibilities for evaluation of the building design in terms of:

- Thermal behavior. The air flow and the air temperature can be visualized within the three dimensional model. Moving around the building specific locations can be inspected to see in which direction the air moves, at which speed and its temperature. The exact values can be compared with standards that must be complied.
- Structural behavior. Stress and displacement can be shown for structural parts. By adding a color gradient to a specific stress value, stress can be visualized. Displacement is exaggerated to get some 'feeling' of how the structure behaves. The exact values can be compared with standards that must be complied.
- Human behavior. The way people will walk around the building can be shown. Statistical or experimental collected data are used to depict the paths people will use and how much time it will take to get from one place to another. These data can be compared with for instance safety regulations.

Separate packages for thermal behavior, structural behavior, human behavior, etc. already exist. Sometimes CAD packages are used as a front-end tool. Integration of evaluation tools with design tools has always been very difficult because of the difference of the internal information representation. VR will make no exception to that. However, since the VR environment is still in its premature stage it seems a good time to create a new kind of representation which allows for different views. At the moment we can take advantage of much research that has been done on this issue (Bronsvoort 1996, Luiten 1994).

10.4 Interfaces in VR

The nature of man-machine interfaces in Virtual Reality will depend on the design task. The windows interface, which has become standard for almost all computer applications, is inappropriate for the three dimensional environment of VR. An *interface* is determined by a set of functions that manipulate the design data and by the way these functions are presented to the user. Functions that must be available in an architectural design system are:

- Create, update, delete and select entity. This set of functions is closely related to database management. The interface must be able to directly *manipulate entity attributes and entity relations.* From the available data, selections can be made to focus on a specific part or aspect of the design. Entities and entity relations will develop in time. The development reflects the design process. Cyclic process steps require 're-activation' of older parts of the data structure. Some process steps need to be frozen since they reflect a certain design stage (*e.g.* preliminary design). *Workflow management* within a building project requires an interface that resembles office activities such as planning, archiving, version management, etc.

- Create and modify shape and topology. This set of functions is partly related to tools that can be found in CAD packages. *Shape edit* functions can support the creation and alteration of forms that are visualized instantly. Less common is the support of *topology* constraints between shapes. Adding topology constraints will keep building components (*e.g.* a wall and a floor) connected while moving one of the components. Sketches are used to support the process of the creation of new shapes as a response to former sketches. Shape algorithms *generate* new *shapes* under specific preconditions that are set by the designer.
- Match design patterns. The first application of this function is as an aid in trying to use existing *design knowledge* for specific design problems. Design problem (*e.g.* the layout of a floor plan, thermal requirements that must be met) and design knowledge must be described using the same methodology so that they can be compared. Design knowledge is available from design research. Possible solutions to the design problem can be offered and adapted into the design model. Secondly this function is required to create *multiple views* of the design. Since each discipline (*e.g.* structural engineering, building physics, building contractor) uses its own data structure, matching of data structures is inevitable. Support of the matching process based on a common library of design patterns will improve the quality of information exchange between disciplines.
- Apply engineering constraints. This function is a collection of constraints that the architect or engineer wishes to impose upon the design because he/she wants the design to display specific *behavior*. For instance the designer wants a wall to behave as a bearing wall. Of course proposed behavior can interfere with form conditions. Thus the wall should in fact have sufficient contact area with the floor that it bears.

11. Systems Architecture

In the design studio, design technologies are supported by interface functions that are implemented in a VR environment. The system's architecture of the design studio consists of the interface, storage of information and dedicated computers. In Figure 5: System's architecture, the relationships between design technology, interface function and information storage are outlined. Information storage is divided into five stores, each of them capturing a specific subset of information about the building design. A subset is a defined by meta class. The meta classes are interrelated in Figure 6: Meta class model. For processing the data from the stores triggered by some interface function, computer systems are used that are optimized for the execution of a specific task. Interface functions might access one or more systems. These relations are not shown in the figure for readability reasons.

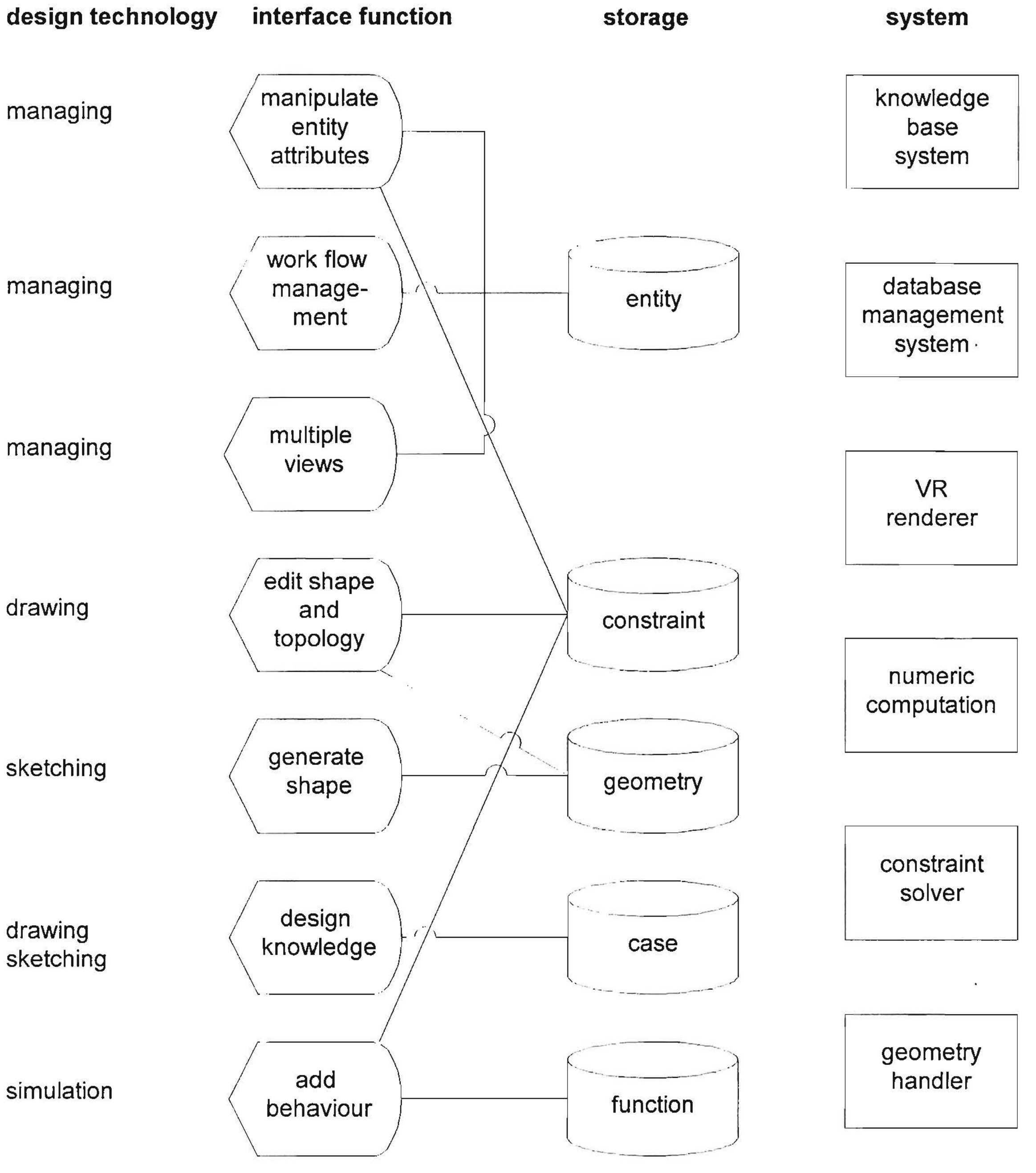

Figure 5: System's architecture.

To show how the sub-stores are related Object Oriented modeling notation is used in Figure 6. An entity (like an object from OO-modeling) is defined by a set of variables and a set of functions. Entities can be related by the following relation types: aggregation (part-of), specification (is-a) and constraints (*e.g.* coplanar, bears). An entity may have one or more geometrical descriptions (CSG, Brep). Functions can be added or overloaded to display specific behavior by entities.

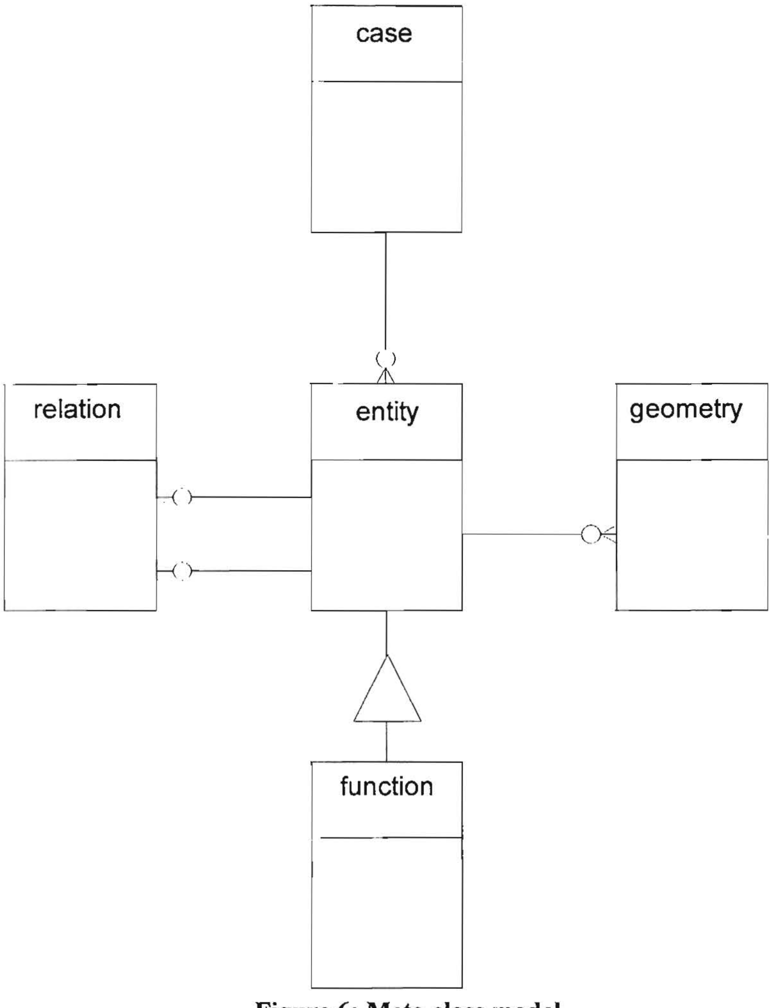

Figure 6: Meta class model.

12. Concepts

The concepts being used to implement data storage and data processing are briefly described below. For a more extended explanation, see the references.

12.1 Feature-Based Modeling

Feature-based modeling is a technology mainly applied in mechanical engineering. As described in the previous Section, 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. Features are proposed as the key concept to support dynamic information modeling. Feature Types represent knowledge from a particular domain (*e.g.* a column as a structural component) and can be defined during the design process. Feature Types are not similar to building components since they can enclose any information that a designer wishes to be captured (*e.g.* thickness of all floors, material of bearing walls, etc.) Feature Types are instantiated to create a Feature model of the design. Relationships known from Object Oriented modeling are supported. Feature models are extended and updated during the design process (van Leeuwen, Wagter, and Oxman 1996).

12.2 Constraint Management

Constraints are necessary to have the design display specific behavior. Two main categories of constraints can be distinguished in a design system: geometric constraints (*e.g.* wall are coplanar) and engineering constraints (*e.g.* wall bears floor). Constraint satisfaction requires a mix of constraint solving strategies (*e.g.* local propagation, relaxation, numeric equations) to provide the designer with a set of zero (over-constrained situation), exactly one or more (under-constrained situation) design solutions that meet the specified constraints. The constraint solving strategies are conducted to trigger the proper solver at the proper time. In an interactive design system constraints must be solved simultaneously (Donikian, S. and Hegron 1995). See also the more extensive discussion in Section II.

12.3 Activity Networks

Designing a building product is a cyclic process. This process can be decomposed into a series of activities. Managing design processes requires well-defined activities. For a specific building project activities are combined in an activity network. Since the network being used is a Petrinet, it is possible to keep the information consistent. Activity networks are useful on the single user level as well on the group level. In the last case the network acts like a kind of planning mechanism for a multidisciplinary design and building team (Vries 1996).

12.4 Generic Representations

To capture knowledge about building types generic representations of a particular building type are proposed. A generic representation (*e.g.* simple contour, schematic axial system, circulation scheme, etc.) is described by: name, source, graphic representation, textual description, graphic units and icon representation. They are established by means of analyzing architectural designs. The design process using a particular building type can be encoded in a sequence of generic representations that develops a particular issue (*e.g.* shape, structure, etc.) (Achten, Bax, Oxman 1996, Achten 1997).

13. Status of the Research

Currently the design studio is equipped with six systems (PC running Windows NT) that are dedicated for drawing 3D models (Autocad, 3DStudio Viz) and fast rendering in a Virtual Reality environment (World Up). The VR user environment consists of a projector (Barco) displaying on a wide screen, a Head Mounted Display and several navigation interfaces like a 6D mouse and a joystick. A prototype of a VR design system was developed using a combination of Autocad en World Tool Kit (Coomans and Oxman 1996). Research to fill in the system's architecture of Figure 1 is going on now for:

- developing object classes for defining Feature models. The object classes define the entity storage data structure using a data base management system,
- developing an interface in VR for manipulating Feature models. The VR interface comprises the function: manipulate entity attributes. Through accessing the entity also the geometry parameters and constraint parameters can be updated,
- developing a tool for the use of design knowledge captured in Feature models. To support drawing as a design technology, design knowledge must be accessible in the case store using a knowledge base system and a database management system,
- developing a system for geometric constraint solving. A constraint solving system is implemented to provide the designer with design solutions that meet the specified constraints in the constraint store,
- developing a system for the simulation of thermal behavior. To simulate the thermal behavior of a building or building part entity functions will be available using the numeric computation system,
- developing a tool for accessing knowledge about building physics. To support the use of knowledge about building physics of a building or building part a case store with requirement indicators will be available using a knowledge base system and a database management system, and
- developing a method for measuring way-finding in VR. To simulate the behavior of human beings in a building, functions will be available to accommodate circulation activity from the function store using the numeric computation system.

Most research programs will result in a Ph.D. thesis within four years.

14. References

Achten, H.H. (1996a). Generic Representations: Intermediate Structures in Computer Aided Architectural Composition. *Approaches to Computer Aided Architectural Composition.* Asanowicz, A. and Jakimowicz, A. (eds.), 1996, TU Bialystok, Bialystok.

Achten, H.H., Bax, M.F.Th., and Oxman, R.M. (1996b). Generic Representations and the Generic grid: Knowledge Interface, Organization and Support of the (early) Design Process. Timmermans, H.J.P. (ed.) *Proceedings of the 3rd Design and Decision Support Systems in Architecture and Urban Planning Conference, August 18-21*, Spa, Belgium.

Achten, H.H. (1997a). Generic representations - Typical Design Without the use of Types. *CAAD futures 1997.* Junge, R. (ed.) Kluwer Academic Publishers, Dordrecht, p. 117-133

Achten, H.H. (1997b). *Generic representations - An Approach for Modeling Procedural and Declarative Knowledge of Building Types in Architectural Design.* Ph.D. thesis. Eindhoven University of Technology, Eindhoven

Achten, H.H., and Leeuwen, J.P. van (1998). A Feature-Based Description Technique for Design Processes: A Case Study, *Proceedings of the 4th Conference on Design and Decision Support Systems in Architecture and Urban Planning,* Timmermans, H.J.P. (ed.) Maastricht, The Netherlands, July 26-29, 1998.

Akin, Ö. (1986). *Psychology of Architectural Design.* Pion Limited, London.

Allen, J.F. (1983). Maintaining knowledge about temporal intervals, Commun. ACM Vol 26 No 11, 1993, pp. 832-843.

Booch, G. (1994). *Object-Oriented Analysis and Design with Applications.* Redwood City, The Benjamins/Cummings Publishing Company.

Bronsvoort, W. F. (1996). *Papers on Feature Modeling from the Computer Graphics and CAD/CAM group.* Delft University of Technology, Delft.

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

Coomans, M.K.D., and Timmermans, H.J.P. (1997). Towards a Taxonomy of Virtual Reality User Interfaces. Paper presented at the *International Conference on Information Visualization-IV,* 27-29 August 1997, London.

Coomans, M.K.D. (1998). A VR User Interface for Design by Features, *Proceedings of the 4th Conference on Design and Decision Support Systems in Architecture and Urban Planning,* Timmermans, H.J.P. (ed.) Maastricht, The Netherlands, July 26-29, 1998

Cross, N. (ed.) (1984). *Developments in Design Methodology.* John Wiley & Son Ltd., Chichester.

Donikian, S., and Hegron, H. (1995). Constraint Management in a Declarative Design Method for 3D Scene Sketch Modeling. *Principles and Practices of Constraint Programming* v. 1, (22). Saraswat, V. and Hentenryck, P. van. The MIT Press, Cambridge, Massachusetts, p. 427-444.

Dijkstra, J., and Roelen, W.A.H. and Timmermans, H.J.P. (1996). Conjoint Measurement in Virtual Environments. Timmermans, H.J.P. (ed.) *Proceedings of the 3rd Design and Decision Support Systems in Architecture and Urban Planning Conference,* Timmermans, H.J.P. (ed.) August 18-21, Spa, Belgium.

Dijkstra, J., and Timmermans, H.J.P. (1996). The Application of Conjoint Measurement as a Dynamic Decision Making Tool in a Virtual Reality Environment. *CAAD futures 1997.* Junge, R. (ed.), Kluwer Academic Publishers, p. 757-770.

Earnshaw, R. A., Gigante, M. A., and Jones, H. (1993). *Virtual Reality Systems.* London, Academic Press.

Flemming, U., and Wyk, S. van (1993). *CAAD futures '93. Proceedings of the Fifth International Conference on Computer-Aided Architectural Design Futures.* North-Holland, Amsterdam.

Gorti, S.R., and R.D. Sriram (1997). From symbol to form: a framework for conceptual design, *Computer-Aided Design* 28, p. 853-870.

Heath, T. (1984). *Method in Architecture.* John Wiley & Sons Ltd., Chichester.

Junge, R. ed. (1997). *CAAD futures 1997.* Kluwer Academic Publishers, Dordrecht.

Kelleners, R.H.M.C., Veltkamp, R.C., and E.H. Blake (1997). Constraints on Objects: a Conceptual Model and an Implementation, *Proceedings of the 6th Eurographics Workshop on Programming Paradigms in Graphics*, Budapest, 1997, pp. 67-78.

Leeuwen, J.P. van, Wagter, H. and Oxman, R.M. (1996). Information Modeling for Design Support. Timmermans, H.J.P. (ed.) *Proceedings of the 3rd Design and Decision Support Systems in Architecture and Urban Planning Conference*, Timmermans, H.J.P. (ed.) August 18-21, Spa, Belgium.

Leeuwen, J.P. van, and Wagter, H. (1997). Architectural Design-by-Features. *CAAD futures 1997*. Junge, R. (ed.). Kluwer Academic Publishers, Dordrecht

Leeuwen, J.P. van, and H. Wagter (1998). A Features Framework for Architectural Information, *Artificial Intelligence in Design '98*, Gero and Sudweeks (eds.), Dordrecht, Kluwer, 1998.

Luiten, G. T. (1994). *Computer Aided Design for Construction in the Building Industry*. Ph.D. thesis Delft University of Technology.

Mallory-Hill, S.M. (1998). Building a Case-Based Design Assistant for Workplace Environment Design, *Proceedings of the 4th International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, Timmermans, H.J.P. (ed.) Maastricht, The Netherlands, July 26-29, 1998.

Maver, T., and Wagter, H. eds. (1988). *CAAD futures '87. Proceedings of the Second International Conference on Computer Aided Architectural Design Futures, Eindhoven, The Netherlands, 20-22 May 1987.* Elsevier, Amsterdam 1988.

McCall, R., Johnson, E. and Smith, M. (1997). Design as Creating a Graphical Hyperdocument. *CAAD futures 1997*. Junge, R. (ed.) Kluwer Academic Publishers, Dordrecht.

McCullough, M., and Mitchell, W.J. and Purcell, P. eds. (1990). *The Electronic Design Studio. Architectural Knowledge and Media in the Computer Era.* The MIT Press, Massachusetts.

Oxman, R.M., Bax, M.F.Th., and Achten, H.H. eds. (1995). *Design Research in the Netherlands.* (Bouwstenen 37) Faculteit Bouwkunde, Technische Universiteit Eindhoven.

Pipes, Alan ed. (1986). *Computer-Aided Architectural Design Futures. International Conference on Computer-Aided Architectural Design*, CAAD Futures 1986.

Rowe, P.G. (1987). *Design Thinking.* The MIT Press, Cambridge, Massachusetts, London, England.

Schmitt, G.N. (1990). Classes of Design - Classes of Tools. *The Electronic Design Studio. Architectural Knowledge and Media in the Computer Era.* McCullough, M. and Mitchell, W.J. and Purcell, P. (eds.) The MIT Press, Massachusetts, p. 77-90.

Schmitt, G.N. (1991). *CAAD futures '91. Proceedings. Education, Research, Applications.* CAAD futures.

Sommerville, I. (1992). *Software Engineering - Fourth Edition.* Addison-Wesley Publishers Ltd.

Tan, M. and The, R. eds. (1995). *The Global Design Studio.* Center for Advanced Studies in Architecture, National University of Singapore, Singapore.

Vries, B. de, Leeuwen, J.P. van, and Achten, H.H. (1997). Design Studio of the Future. *Proceedings of the CIB W96 Conference, IT Support for Construction Process Re-Engineering*, July 9-11, Cairns, Australia, p.139-146.

Vries, B. de, (1996). *Communication in the Building Industry: A Strategy for Implementing Electronic Information Exchange.* Ph.D. thesis Eindhoven University of Technology, Eindhoven.

Vries, B. de (1997). VR-DIS Research program, Internal report Faculty of Architecture and Building, Eindhoven University of Technology, Eindhoven.

Vries, B. de, and H.A. Achten (1998). What offers Virtual Reality to the Designer, Paper presented at the *Third biennial world conference on Integrated Design & Process Technology*, Berlin, Germany, July 6-9.

Vries, B. de, and A.J. Jesserun (1998). An experimental design system, Virtual Environments and Systems, *18th ASME International Computers in Engineering Conference*, Atlanta, Georgia, September 13-16.

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

Uitgave: Technische Universiteit Eindhoven FACULTEIT BOUWKUNDE

Design Systems Reports
is a series of publications by the Design Systems group of the Faculty of Architecture, Building, and Planning of Eindhoven University of Technology.
The Reports present the ongoing research efforts and results of the group, showing the state-of-the-art computation in architectural design.
Research topics include the development and application of Virtual Reality in design, innovative design tools and methods, design information modelling, and interdisciplinary design.

Design Systems Reports can be ordered from the secretary of the Design Systems group:

Eindhoven University of Technology
Faculty of Archtecture, Building and Planning
Design Systems group
PO Box 513
NL-5600 MB Eindhoven
The Netherlands

WWW: http://www.ds.arch.tue.nl
Email: H.L.J. Aretz@tue.nl
Phone: +31 (0)40 - 247 22 62
Fax: +31 (0)40 - 245 03 28