Coomans, M.K.D. and van Leeuwen, J.P. **2001**. ‘Abstract but Tangible, Complex but Manageable’. In: *Proceedings of AVOCAAD 2001*, Brussels, Belgium, April 5-7, 2001.

ABSTRACT BUT TANGIBLE, COMPLEX BUT MANAGEABLE

M.K.D. COOMANS AND J.P. VAN LEEUWEN
Eindhoven University of Technology, Faculty of Architecture, Building and Planning, P.O. Box 513, 5600 MB Eindhoven www.ds.arch.tue.nl

Abstract. In the VR-DIS research programme, an innovative design-information modelling technique has been developed that is based on features. In this modelling technique, the designer is invited not only to model the form and spatial aspects of his or her design, but also to model the structure of the data behind the design. The designer is offered a way to control how design characteristics are structured and stored. With this new information modelling technique, we expect that the designers will be better capable to handle the complexity of diverse kinds of information involved in a design process. This new way of computer aided design offers a unique design freedom: any design concept becomes addressable. On the other hand, this technique also puts the responsibility for the content of the CAD database entirely in the hands of the designer. In order to be able to enjoy the design freedom fully and at the same time handle the responsibility over the design database, a computer tool is needed that shows the precise content of the database, and that is easy and quick to interact with. We have developed a 3D graphical User Interface to realise this requirement. It shows the data objects in a 3D interactive graph. The intuitiveness and user friendliness of the interface was improved by adding features like the visualisation of the browsing history, the visualisation of link-semantics, and animated visual feedback effects. The hardware part of the interface is worked out as a Fish Tank VR set-up. This hardware configuration improves the experienced realism of the displayed 3D objects up to a feeling of physical presence. The interface as a whole therefore provides a highly attractive display of the abstract design data; abstract but tangible. It is a tool in which complex data structures can be explored and controlled: complex but manageable.

1. Introduction

Architectural design, like many other human activities, benefits more and more from the ongoing development of information and communication technologies. Software tools are available for modelling and evaluating most architectural aspects. Unfortunately, the designers from different architectural disciplines each model “their part” of the design information using “their way” of modelling. Despite the fact that all the design information is digital, differences in approaches soon lead to digital confusion of tongues.

Attempts has been undertaken to develop common description languages in the form of product data models. These are aiming at the representation and communication of the information related to a building in all its aspects and during its complete life-cycle. Unfortunately, product data models turn out to be too rigid to be used during the design process itself. Especially during the early stages of design, the concepts used by designers in reasoning about the design develop as part of the design process. A design modelling technique is needed that facilitates this dynamic handling of information.

In the VR-DIS research programme, a Feature-Based Modelling approach is developed to overcome these problems by allowing high levels of extensibility and flexibility of the information structures that describe desings (van Leeuwen, 1999). In this modelling technique, the designer is invited not only to model the form and spatial aspects of his or her design, but also to model the structure of the data behind the design. Relationships can be modelled that are not typologically defined. This provides the designer with a high level of freedom to model the design rationale using both typical design concepts and ad-hoc design concepts (van Leeuwen and Jessurun, 2001). This new way of computer aided design offers a unique design freedom: any design concept becomes addressable. On the other hand, this technique also puts the responsibility for the content of the CAD database entirely in the hands of the designer. In order to be able to enjoy the design freedom fully and at the same time handle the responsibility over the design database, a computer tool is needed that shows the precise content of the database, and that is easy and quick to interact with.

This trend towards the availability of large amounts of digital design information raises the problem of how to use and handle this data efficiently. The problem is actually not specific to CAD, but is becoming a problem for many computer disciplines. According to Card (Card et.al.,1999), the visualisation of large abstract data structures is one of the crucial tasks to bring computers closer to the general public. As an introduction to a panel discussion at Siggraph 1999, Bill Buxton stated: “It borders on banal to state that we live in an ever-more-complex world, and much of that complexity is due to the previous generation of technology. It seems equally obvious that the basic litmus test of future designs should be: does it enhance our ability to cope with that complexity?” (Harris et al, 1999). Applied to our new design information modelling approach, this question turns into: does this Feature-Based Modelling approach indeed enhance our ability to cope with the complexity of architectural design?

The complexity of Feature-Based Modelling is handled in a number of ways. Firstly, the feature modelling approach is enhanced by techniques like feature type recognition. This aims at supporting the designer in the process of concept identification and formalisation, and in ensuring consistency in the usage of design concepts during the task of modelling a complex design (van Leeuwen and de Vries, 2000). Secondly, 3D graphical user interface are developed to facilitate intuitive communication between designer and design model. In the next sections, we present this 3D graphical user interface that was developed to support Feature-Based Modelling.

2. Tangibility

Since it’s emergence in the mid 80’s, Virtual Reality (VR) has often been regarded as a promising new technology for solving the problem of data visualization. VR can potentially communicate large amounts of data in an easily understandable format. Several authors have suggested that VR-based user interfaces would have a big impact on our ability to deal with information (Erickson,1993), (Biocca and Levi, 1995), (Chorafas and Steinmann, 1995).

Although VR looks very promising, its successful application has turned out to be harder then first thought. Several authors ascertain that the success of VR applications stands or falls with the quality of the chosen visualization and interaction method (Bryson, 1995), (Erickson, 1993). However, it is hard to find information on how these paradigms must be designed. It is still a very new technology, and very little scientific knowledge is available on how and when it can and should be applied.

Initial research in his field was mainly focussed at the development of virtual environments that completely replaced our current working place, immersing the user in a new artificial world. VR interfaces that make use of head-mounted displays and gloves belong to this approach. More recently, research has shifted towards finding optimal combinations of our normal workplace with computer-mediated artefacts. Researchers following this approach have termed their work Augmented Reality (AR), Graspable User Interfaces (Fitzmaurice, 1995), and most recently Tangible User Interfaces (Ishii, 1997).

The two keywords to VR, AR, Graspable and Tangible interfaces are: *visualization* and *natural interaction*. The goal of visualization is to present data in ways that make them perceptible, such that it can simply be *experienced* be rather then cumbersomely *interpreted* by our rational. For applications that deal with large databases, VR researchers make use of the visualisation expertise of the Information Visualisation research community.

The second key-characteristic of new user interfaces, natural interaction, allows an easier responds to the computer. Through drawing on our experience with real world interaction, users should no longer have to shift their attention from what they want to do, to how they have to do it. Depending on the approach, the natural interaction is performed on either digital objects (i.e. an object displayed on the screen; this is "classical" VR&AR), or it is performed on physical objects that are tracked by the computer (i.e. the hands of a real clock are used to scroll back in a log file; used in Graspable and Tangible interfaces).

Our approach for the development of a user interface for Feature-Based Modelling followed the following combined strategy. Firstly, we designed a "physique" for features. Their form and behavior on the computer screen was designed such that it reflects the functionality of the feature data model. Secondly, the gap between the virtual world of features (on the computer screen) and the user's physical workplace is bridged by Fish-tank VR functionality. The Fish-tank functionality improves the experienced realism of the displayed 3D objects up to a feeling of physical presence.

3. The physique of a feature

In this architectural form of Feature Based Design, feature types can represent any physical or abstract concept involved in the design process: space, function, costs, safety, comfort, form, … . Designing by features involves the creation of features for any design part or characteristic. The designer can instantiate *simple features* (individual features) or *complex features* (predefined clusters representing complex architectural concepts) that are found in the *type library*. We have chosen to represent features and feature-types as 3D graphical blocks. They have labels attached to identify their type-name, feature name, and possibly their value (e.g. in case of dimension feature).

Feature instances are added to a *feature model* that holds all information of a particular design task. In the feature model, the features can be linked to each other to model their physical and/or conceptual relation. Feature models therefore evolve into large clusters of features. The links are partially type-defined (when a predefined complex feature type was used), and partially designer-defined. Both type-defined and designer-defined relations can be one of 3 kinds: specification (i.e. the height feature telling something about the wall feature), composition (i.e. wall feature is part of the building feature), and association (i.e. the link between the building feature and the site feature). These different kinds of links play a dominant role in Feature Based Modelling.

From a visualization point of view, feature models are large graphs, containing labeled nodes and relations. A number of visualization methods have been proposed and successfully implemented for similar information structures; e.g. on the annual Graph Drawing Symposium (Graph, 1992). Unfortunately, all these visualization methods are developed for data-sets in which relation types are of little or no importance. Because of the importance of relation types in Feature models, a simple color or line-shape polishing of an existing graph drawing method is not sufficient. A designer will recognize relations' semantical differences much better when these are expressed in the spatial layout, and not in color and linetype only.

We developed a new graph visualization method to emphasize relation type differences. It was implemented in a prototype application, called DDDiver[1]. In DDDiver, relations are grouped by type, and each group is given a characteristic direction. Distinct arrow styles and colours further improve the recognizability of that relation's type. The relations themselves are straight by default, but bent into S-like shapes when the connected features are not in line. When bent, the relation-end-arrows preserve the characteristic direction and the rest of the relation forms a fluidly connecting spline. See figure 1.

[1] DDDiver stands for 3D Interactive Visualization of Entity Relationships.

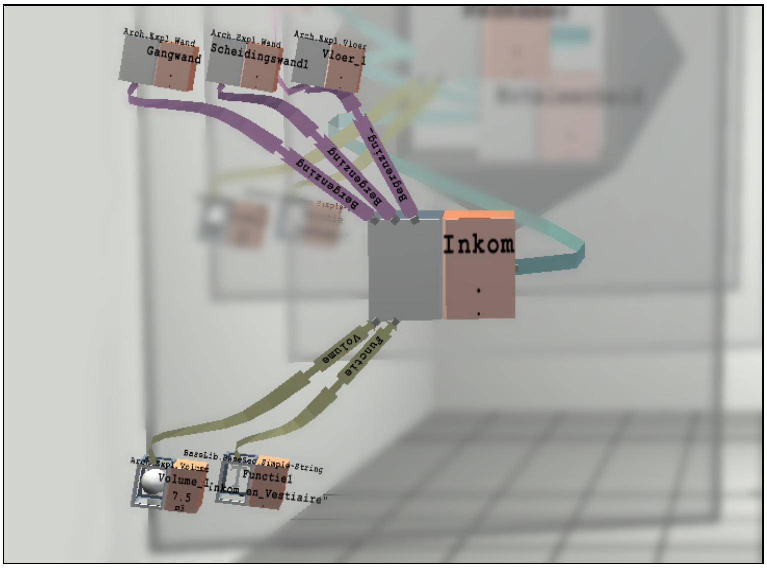

*Figure 1. DDDiver's v*isualisation of objects and relations.

DDDiver visualizes the Cad database as a 3D cubic box. When loading a design model, the box folds open. In front of the opened database, transparent workplanes are placed. Each layer can host one central feature and its directly related features. When the user asks to see additional data of a related object an extra tranparant layer is layed on top (appears dynamically). The selected features moves to the centre of this new layer, and it's own relations are displayed around it. The original feature in focus stays on the layer in the back, as visual background information. After multiple browsing steps, a pile of layers is formed. When a new layer is added, existing layers automatically move one step backwards. See figure 2.

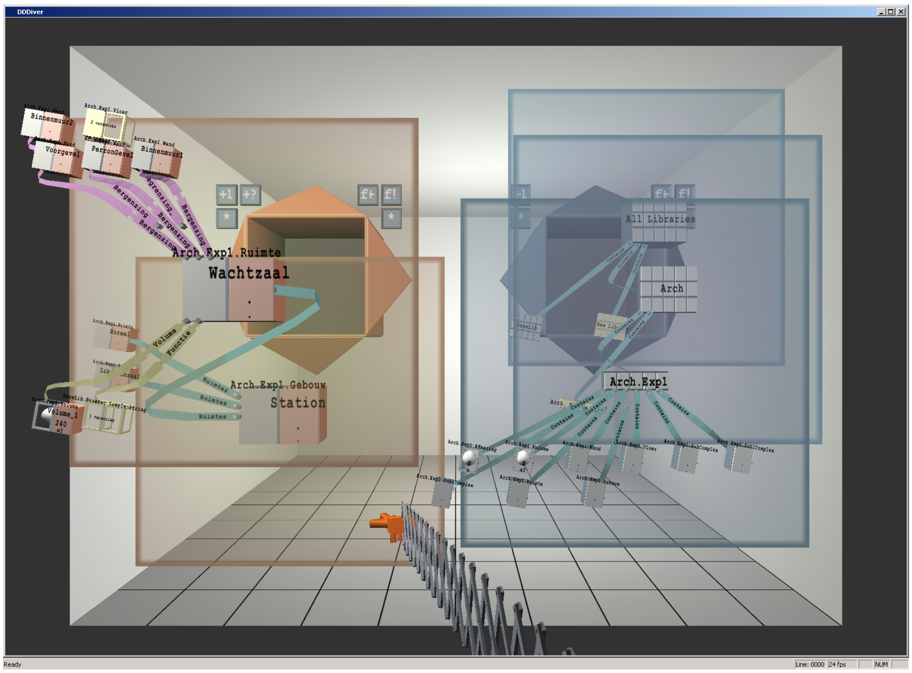

Figure 2. DDDiver's 3D graphical user interface to support feature based design. On the left, a feature model is visualised, on the right the feature-type library.

Feature based design requires repetitive access to both feature model (actual design) and type library. Therefore both are visualized side-by-side in two separated graphs. Both are functionally linked: a feature type can be instantiated by dragging its representation from the type graph to the instance graph. The designer can add a new type to the type library by dragging a prototype of the new architectural concept (typically a cluster of features) from the instance graph to the type graph.

3D animation effects graphically support all feature manipulations. Features never jump from one location to another, but gradually move to the other location. When a feature is deleted, it shrivels up and shrinks until it is disappeared. When a feature is removed from the current view, it floats back into the database box. Research has pointed out that such animation affects do contribute to a better understanding of the user. User errors are reduces because the consequences of each action are illustrated by these animations.

We refer to the combined set of animation effects for specific types of visualisation objects as the *behaviour* of those objects. We argue that a coherent design of a 3D form and behaviours leads to highly attractive user interfaces. A thoughtful design of object behaviour provides the interface objects a certain character and personality. This personalisation improves the user's engagement with the system.

Especially when dealing with abstract and complex systems, it is important to translate the abstract into something physical. Donald Norman (1988) has pointed out two important kinds of cognition: experiential cognition, and reflective cognition. The first we use as a respond on perceived events around us, without apparent effort or delay. The second is that of concepts, planning, reconsideration, comparison and decision-making. In (Coomans, 1998-b), it was argued that the ideal user interface would entirely be based on experiential cognition. Abstract data is only perceived as abstract when there is no perceptible representation that goes along. DDDiver demonstrates that a coherent graphical and behavioral representation can open up such abstract information in an attractive, easily interpretable data environment.

4. Fish Tank VR

To improve the perceived realism of the feature representation, we have experimented with "fish tank VR" functionality. In such a system, the perspective distortion of the 3D image on the screen is in real time updated according to the viewer's head position in front of the screen. As a result, the screen's front is experienced as if it is the glass front of a fish tank that can be looked through. The virtual objects (the "fish") are viewed inside the tank.

When we move our heads while looking at our environment, we perceive movement parallax that our perception system links and compares with the performed head movement. Smets (Smets, 1992) has suggested that movement parallax is sufficient for tele-presence, the feeling of being physically present with the virtual objects. Fish tank VR can present virtual objects on a desktop in such a convincing way that users can work with them as if they are real.

An additional advantage of Fish tank VR is that the perspective–head movement coupling provides depth perception. Ware (Ware, et al., 1993) has shown that movement parallax is at least as important as stereoscopic viewing for depth perception. As discussed in the previous section, DDDiver's data browsing mechanism makes use of multiple layers laid on top of each other. Fish-tank functionality improves the user's capability to distinguish what objects are located on which layer.

5. Integration

In the context of the VR-DIS research, DDDiver is considered to be one of a number of views that is required to support Feature-Based design. The final VR-DIS system will feature several task-specific user interfaces of which each will be characterised by a mixed representation of the task domain. In (Coomans and Achten, 1998), the authors argue that a feature view as presented in this paper, should al least be combined with a scale-model view on the same data. Figure 3 shows two possible implementations.

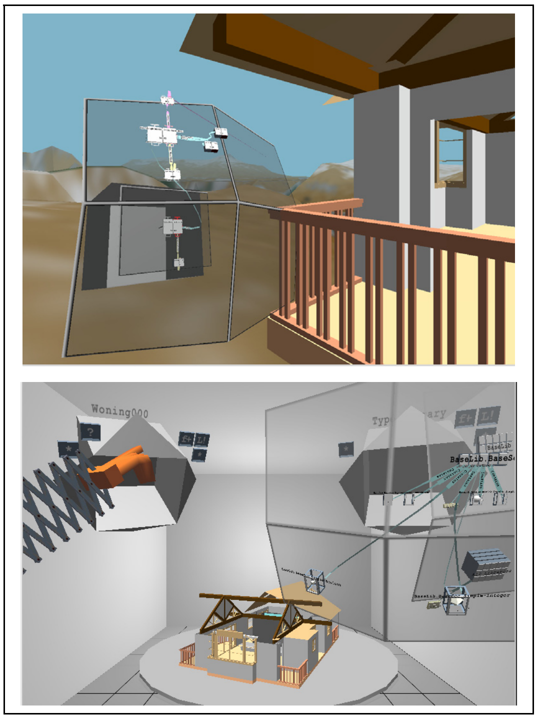

Figure 3. Two possible mixed representation interfaces for Feature-Based design, combining a pictorial "scale model view" and a descriptive "feature view".

6. Conclusion

We have presented a 3D graphical support tool for Feature-Based modelling. Designing by features puts the responsibility for the content of the CAD database entirely in the hands of the designer. In order to be able to enjoy the design freedom fully and at the same time handle the responsibility over the design database, a computer tool is needed that shows the precise content of the database, and that is easy and quick to interact with.

We pointed out VR is a promising technology to develop such user interfaces. We have argued that next to the graphical representation, the behavioural aspects of a user interface are crucial. We argued that complex and abstract CAD information can become easily to interpret by designing an appropriate graphical and behavioral representation. Such representations can be further improved by realizing a Fish-tank VR interface. VR technology based user interfaces can provide a highly attractive display of the abstract design data. What is abstract can be made tangible. Such interfaces are primarily based on experiential cognition, rather than reflective cognition. The user's metal resources are therefore reserved for the actual design and data-modelling task, which stays complex but manageable.

References

Biocca F., Levi M.R.: 1995, Virtual Reality as a Communication Medium, *in* F. Biocca and M.R.Levi, eds. *Communication in the Age of Virtual Reality*, pp. 15-32, Lawrence Erlbaum Ass., Hove, UK

Bryson S.: 1995, Approaches to the successful design and implementation of VR applications, *in* R.A.Earnshaw, J.A.Vince, H.Jones (eds.) *Virtual Reality Applications*, p.328, Academic Press Limited, London,

Card, S.K., Mackinlay, J.D., Scheiderman, B. (eds.): 1999, *Readings in Information Visualization*, Morgan Kaufmann Publishers

Chorafas D.N., Steinmann H.: 1995, *Virtual Reality: Practical applications in business and industry* (sections 1 and 2), Prentice Hall, London

Coomans, M.K.D. and Achten, H.H.: 1998, Mixed Task Domain Representation in VR-DIS, in *Proceedings of the 3rd Asia-Pacific Conference on Human Computer Interaction*, APCHI'98, (Shonan village Center, Japan, July 15-17, 1998)

Coomans, M.K.D.: 1998-b, A Virtual Reality User Interface for a Design Information System, *in* Tom De Paepe (ed.), *Journal for the Integrated Study of Artificial Intelligence, Cognitive Science and Applied Epistemology*, (**15**)4, p321-338

Erickson, T.: 1993, Artificial Realities as data visualization environments: problems and prospects, *in* A. Wexelblat (ed.) Virtual *Reality - Applications and explorations*, pp. 3-22, Academic Press, London

Fitzmaurice, G.W., Ishii, H., Buxton, W. : 1995, Bricks: Laying the foundations for Graspable User Intefaces, in *Proceedings of CHI'95*, p. 442-449

Graph drawing, GD : yearly international symposium 1992 -1999; proceedings as of 1995 by Springer-Verlag, Berlin

Harris, M., Ishii, H., Chung, C., Dodsworth, C., Buxton, B.: 1999, Natural and Invisible Human Interfaces (panel), in *Conference Abstracts and Applications of SIGGRAPH '99* , (Los Angeles, California USA, August 8-13, 1999), ACM Press, pp. 127-129.

Ishii, H., Ullmer B.: 1997: Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, In *Proceedings of CHI'97* , (March 22-27, 1997), ACM

van Leeuwen, J.P.: 1999, *Modelling Architectural Design In-formation by Features*, PhD. Thesis, June 1999, Eindhoven: Eindhoven University of Technology.

van Leeuwen, J.P. and A.J. Jessurun, Added Value of XML for CAAD, in *These proceedings of AVOCAAD 3rd International Conference 2001.*

van Leeuwen, J.P. and B. de Vries: 2000, Modelling with Features and the formalisation of early design knowledge, in Gonçalves et al. (eds.), *Product and Process Modelling in Building and Construction*, Proceedings ECPPM2000 Lisbon, Rotterdam: Balkema.

Lieberman, H.: 1994, Powers of Ten Thousand: Navigating in Large Information Spaces, *Proceedings of the ACM Symposium on User Interface Software and Technology*, p.15-16

Norman, D. A.: 1988, *The psychology of everyday things*, Basic Books, New York, 1988, 257 p.

Smets, G.J.F.: 1992, Designing for telepresence: the interdependence of movement and visual perception implemented, in *IFAC Man-Machine Systems*, (the Hague, The Netherlands, 1992), pp. 169-175

Ware C., Arthur K. and Booth K.S.: 1993, Fish Tank Virtual Reality, in *Proceedings of the international conference on human computer interaction, INTERCHI'93,* (24-29 April 1993), ACM, pp. 37-42

Vries, B. de: 1997, VR-DIS research program, on the web site of the Design Systems group, http://www.ds.arch.tue.nl/