Mixed Task Domain Representation in VR-DIS

Citation for published version (APA):

Coomans, M. K. D., & 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* (pp. 22-27)

Document status and date:

Published: 01/01/1998

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Mixed Task Domain Representation in VR-DIS

M.K.D. Coomans and H.H. Achten

Eindhoven University of Technology, Faculty of Architecture, Building and Planning
PO BOX 513, Mail station 20, 5600MB Eindhoven, The Netherlands

Abstract

Direct manipulation interfaces are characterised according to the representation of the task domain: the domain objects and the effect of user operations on those objects. In the VR-DIS system (Virtual Reality - Design Information System), the task domain is represented by a mix of a verbal (textual) and analogue (pictorial) representation. Objects are represented by a mixture of a descriptive spatial 3D graph of text blocks and a realistically rendered perspective view. It is postulated that user interfaces that present mixed views of the task domain can better support differences between users' working styles and sub-task dependent types of information exchange.

Keywords: visualisation, virtual reality

1 Introduction

The performance of a user interface is largely defined by the way in which it presents task data to the user. In this paper, we will discuss the benefits of integrating multiple data representations in a single user interface. Theoretical foundations are discussed and design demands are formulated. We finish with a brief description of a mixed representation that we are implementing in a VR-DIS system.

2 Task domain and domain representation

Software applications are always built for a specific task. They offer tools and functions that support the execution of that task. The body of knowledge that is related to a particular task is called the "knowledge domain" of that task. The user will execute the task on the basis of his knowledge of the task domain. In order to support this task, a software application has to be designed on the basis of that domain knowledge.

In general, applications do not support one single primitive task, but they support complex tasks that can be subdivided in (primitive) sub-tasks. The user utilises the provided tools for these sub-tasks either simultaneously or consecutively.

Different single tasks involve different data types. For example, writing text involves characters, words, sentences; project planning involves task-descriptions, dates, periods, manpower; the drawing of maps involves lines, colours, areas, symbols, legends, etc. The input and the manipulation of these different types of data often requires accommodated input channels. This has led to complex graphical user interfaces in which the screen image is subdivided in areas with specific input functionality: menu systems and pallets for pointing and clicking, text editing areas, fill in fields, 'drop' boxes, drawing areas, etc.

The data of the job at hand are typically visualised in a particular way throughout the software application. A text editor, for example, represents language in one particular way to the user: as written text. Contemporary text editors do provide a few different views like an edit view, a layout view, and a preview. However, these are all modalities of the same general type: structured written text. Further, these views are not used simultaneously but consecutively. A text editor with an actual mixed representation might be one which provides both a written representation of the text, and a spoken representation. A selected piece of text might simultaneously be visible on the screen and be heard through the computer sound system. An additional feature may be that recorded text such as present in memo-recorders can be used in the text editor.

In reality, applications which support such a true mixture of simultaneous representations are very rare. The multimedia capabilities of present-day computers do provide the means for the development of mixed representation interfaces. However, the current generation of multimedia applications typically assign one particular representation "medium" to a type of data. Applications typically do not provide more than one representation for a particular chunk of information.

In this paper, we discuss the potential benefit of presenting task related information in multiple integrated ways. The potential benefit of mixed representations relates to the fact that representations are not neutral. Representations can only present a subset of reality. They do this by means of abstraction, multiplicity, consistency, functionality, and organisation [1]. Most task domains involve complex data types with multiple aspects. A

particular representation can highlight certain aspects more than others. A combination of representations can provide a more complete view of the information.

Different representations support the mental models of users in different degrees. A mental model is the way a user "sees" or understands a particular domain problem and its solution. These mental models differ between users [2]. A single user can prefer different representations for different sub-tasks. In a multiple representation interface, different users can all individually choose (the part of) the representations that they like most. With a truly mixed representation, a user can very quickly shift his or her attention from one part of the interface representation to another part. This supports best the execution of real tasks that are build of hybrid sub-tasks.

Since a few years, there is a clear trend towards the integration of applications in bundles. These bundles all offer tools needed by people with a specific type of job. The so called "office suites" are probably the best known examples of this trend. They comprise text editors, spreadsheets, planners, and database programs. Another example are the application bundles for the video graphics industry: 3D modelling, animation, and rendering applications are integrated with video editing software and 2D paint tools. These application suites provide new possibilities for combining data chunks of different formats, as well as tools for accessing these data chunks in different ways. We propose not to offer these representations consecutively, but in an integrated way instead. In this way, the objective of simultaneous usage of the different tools is more fully supported. The benefit of mixed representation interfaces is expected to grow with the evolution towards integrated application bundles. The aim of VR-DIS is to develop and provide such an integrated application.

3 VR-DIS

VR-DIS stands for Virtual Reality - Design Information System. VR-DIS is both the name for the research project and the prototype application that will be developed. The goal of VR-DIS is an integrated design system for the building and construction industry. The innovative nature of the project concerns the integration of information from different disciplines within a dynamic design process in combination with the application of the new information presentation capabilities that VR offers to the user.

VR as a User Interface is expected to replace many of the existing techniques due to the (added) possibilities which are available to communicate intuitively between man and machine. In particular in the architectural environment the challenge lies in developing a new work environment in which the design process can take place. [3]

In [4], the authors discuss how VR can fundamentally improve the support of CAD systems for the early,

conceptual stage of the design process. They identify three improvements: 1. VR based interfaces can provide high engagement, and as such, provide a structural performance advantage. 2. VR can present an architectural design with a high level of verisimilitude, and can communicate large amounts of abstract information in an easily understandable format (through advanced visualisation). Both highly verisimilitude views and abstract data views can facilitate early design evaluation. 3. Interactive designing and design reasoning requires the exploitation of the human skill of unconsciously performing motor actions. This is better facilitated by the typical VR-input devices than by the mouse and keyboard interface.

The design of the User Interface for VR-DIS is founded upon the design of the data representations. This high-ranking position of the data representation is inspired by both UI design theory and architectural design theory. Both knowledge bases provide good reasons for the importance of the data representation in VR-DIS. In the next two sections, both will be discussed.

4 UI style for VR-DIS

A diverse set of UIs is used in various computer systems. There is currently no true taxonomy available that can systematically categorise all UIs in a complete set of non-overlapping categories [5]. However, some classification is possible on the basis of "dialogue style" [6][7]. Shneiderman distinguishes 5 basic dialogue styles: menu selection, form fill-in, command language, natural language, and direct manipulation.

Direct manipulation has been selected as the basic UI style for VR-DIS. This choice is based on the knowledge that direct manipulation interfaces perform better than indirect interfaces, in particular for tasks that require complex cognitive operations [8]. Building design involves such complex headwork.

Direct manipulation is characterised by a model world metaphor. The user interacts with an interface that represents the task domain itself, the domain objects and the effect of user operations on those objects [9]. Users work with the interface through pointing out a domain object, modifying it, and immediately perceiving the result of the modification.

The better performance of direct manipulation interfaces is ascribed to three characteristics. Firstly, it supports inter-referential I/O. Secondly, it exploits the human motor capabilities; and thirdly, it provides immediate feedback.

Direct manipulation interfaces visually represent the task domain on a display that is shared between the user and the computer. Both the user and the computer communicate by writing on the shared display, or by referring to something already on the display. This latter capability of referring to previous parts of the conversation is one of the important features of human

conversation and constitutes the difference between monologue and dialogue. Draper has termed this characteristic "inter-referential I/O" [10]. Inter-referential I/O leads to the a typical "grab and manipulate" functionality of direct manipulation user interfaces.

When we come to a new task, such as typing, we first have to think about finding keys, and so on. With practice, the typing skill becomes autonomous. This means that we no longer have to think consciously about it. Clearly, skilled performance, such as typing, requires a lot of mental processing, but that processing occurs independently of conscious thought. This leaves the rest of the cognitive system free for other tasks such as determining what to do. [11]

Direct manipulation interfaces can easily exploit skilled human motor capabilities, especially for what we have called the "grab and manipulate" functionality. Depending on the type of computer system, this grab and manipulate functionality is developed differently. In an ordinary computer system, it is realised as a 'point and click' manipulation, using a cursor, controlled by a mouse. In VR systems, it can be realised analogously to the user's real world way of grabbing and manipulating. The user experiences immersion: he is virtually present between the domain objects. He can grab and manipulate through the natural movements of his hand or head. Actually, such a VR implementation better fulfils the concept of direct manipulation than conventional desktop GUI interfaces. It leads to a much higher level of engagement [12].

The third key characteristic of direct manipulation is the immediate feedback. The object representations interactively show the possibility of user's actions. Consequently, direct manipulation interfaces encourage exploration [7]. Clearly, design systems, such as VR-DIS, benefit from exploration, since exploration is an important strategy for obtaining creative design solutions.

Direct manipulation has been chosen as the main UI style because of the general performance advantage as a result of inter-referential I/O and use of motor activities, and because of its exploration-stimulating characteristic.

5 Design support

The VR-DIS project focuses on the early phases of the architectural design process. It aims at including the design expertise of all parties involved in architectural design; i.e. the architect, structural engineer, HVAC engineer, planner, facility manager, etc. The early design phase is characterised by its ill-structuredness, problematic definition of the design task, insufficient knowledge, and lack of algorithms for attaining solutions [13][14].

In order to tackle these problems, which are not unique for architectural design but hold for design problems in general, the professional community has developed several strategies, which can roughly be divided into design strategies and knowledge structures. Design strategy heuristics provide the designer time-honed procedures for solving design tasks. Examples of such strategies are refinement (working from global to specific), decomposition (breaking the design task into smaller, more manageable tasks), and case-based reasoning (changing existing solutions to satisfy the task). Knowledge structures provide pre-defined sets of solutions or parts of solutions that present a coherent answer to a compositional whole of issues. Examples of knowledge structures are building types, analogies, formal languages, etc.

During the design process, the architect intensively uses graphic representations (sketches, diagrams, drawings) to externalise and fix the current state of the design solution in the design process. Many conventions of depiction are developed such as the plan, section, perspective, and axonometric projection, leading to a large number of drawings which may be instrumental in representing and developing the design [15].

The central role of representations in the design process suggest that this should be a central focus in the VR-DIS project. However, the three-dimensional immersive property of VR-technology is quite outside standard design practice. This means that new representations have to be defined (following design strategies and knowledge structures) that both are intrinsic to the VR-DIS platform and that address the basic and expert cognitive and practical skills that designers in the architectural community have.

6 Shared domain representation

Both the selected UI style and design theory indicate the importance of a domain representation that is shared between the user and the computer. How does this shared representation relate to the other components of the user interface? Figure 1 depicts a general model of human-computer interaction. (See also [5]). It distinguishes between 6 representations of task information.

Information exchange between human and computer necessarily has to go through a "physical medium". In the physical medium, the information is represented as a specific energy state. The computer is "connected" to these physical media through input and output devices. I/O devices form the bridge between the physical media and the digital information carried inside the computer. The physical signals received through the input devices are highly fragmented and contain no explicit semantics.

Figure 1: A model of human-computer interaction

For example, suppose a user wants to commit some numbers and words to a (conventionally equipped) computer. He might choose to do this through hitting the appropriate character-keys on the keyboard. In this way, he communicates the original word through its textual representation, using the physical means of button pushes. The input device (the keyboard) detects these physical signals, and describes them in a digital form, resulting in a list of separate characters. After that, an "abstraction function" is necessary to recognise the larger symbolic forms in this fragmented list: the words and numbers. The result of the abstraction function is still merely a representation and not the concept behind the representation. To retrieve the concept, the computer must use an "interpretation function". The retrieved concepts can then be further processed, or stored in some "internal computer representation". An opposite two-step transformation process is necessary at the output side. For example, when a user wants to retrieve a list of numbers from the computer database, the information must be "represented" in some form. This could be a table with the numbers in textual format, or it could be a bar-chart with the numbers represented as bar lengths. The chosen representation must subsequently be specified or "rendered" into a raw description of the representation that the output devices can handle.

This model of human-computer interaction is valid for all interfaces. Direct manipulation interfaces are a subset of all interfaces. In Figure 1, their specificity is visualised by the dashed rectangle: both input and output use the same single representational modality. The communication from the user to the computer and that of the computer to the user is expressed in this one representation. As described in the previous section, this shared interface representation can excellently fulfil the designer's need for an external representation, given a suitable design.

7 Domain representation in VR-DIS

For direct manipulation, the benefits of the shared task domain representation are at the level of primitive task execution: easy command execution. For design, the benefits of a shared task domain representation are at a higher level of task execution: design reasoning, generation of alternatives, performance evaluation, etc.

To exploit the potential benefit on both levels, the domain representation must be designed appropriate for both levels. At the level of primitive task execution, the representation must support motor actions for selecting, activating, and modification, as well as immediate feedback during (motor) manipulation. The metaphor used for the representation should suggest which primitive actions are possible and how they can be performed. At the higher task level, the domain representation must resemble the user's mental model of the design task. The system must represent the task information to the user in a way that is compatible to his/her mental representation. Thus, the task domain representation of the interface must correspond to the user's mental model of the task domain, both for primitive tasks and higher level tasks.

Van der Veer [2] states that the chosen interface representation influences the further development of the user's mental model of the task domain. More specifically, the interface influences the user's model of how the task can be executed using the system. Parts of the interface's representation can be found again in the mental models of the users. Thus, the design of the interface representation partly determines the solution strategies that the user will develop. This is another reason for thought-out interface representations.

What does all this mean for the design of the VR-DIS User Interface? VR-DIS is set up to compile all disciplines that are involved in an architectural or urban design process. As a result, VR-DIS will handle information of many different kinds: geometry data, lay-out data,

appearance data, stability data, hydro-thermal data, cost information, aging data, organisational data for the building process, and so on. What kind of representation can present all these different kinds of data? UI modality analysis distinguishes between linguistic and non-linguistic interface representations (e.g. [16]). In cognitive psychology, an analogue distinction is made between verbal and pictorial mental representations. For VR-DIS, both types of representations seem equally relevant. Some data can only be communicated verbally (e.g. names, labels), other data can only be communicated in a pictorial way (e.g. appearance characteristics), but most information can be communicated by both types of representations. Further, the type of representation that best serves can be expected to change, depending on the specific task. For example, thermal isolation data is best communicated verbally when specific details are being fine-tuned, but in the early design stages a designer needs a quick rough insight in the thermal performance of the different parts of the building. In this latter case, a pictorial visualisation serves better. Hence, it seems that a software application which deals with such an extensive number of information types cannot be based on a single representation. A combination of multiple representations is required. Especially the combination of verbal and pictorial representations seems generally useful.

It is for this very reason that we have chosen to develop a mixed representation for VR-DIS. The interface representation of VR-DIS will be a combination of two views: the "feature" view and the mock-up view. The feature view is a spatially structured verbal representation of the data structure that underlies the VR-DIS system. This data structure is organised in "features", hence the name. The modelling of architectural design information in features has been developed by van Leeuwen [17].

The mock-up view is a pictorial representation of the design information, based on the physical appearance of the design building/space/urban area. This view will feature a high level of verisimilitude. Figure 2 shows a part of the current design proposal for the visualisation of features and mock-up in the VR-DIS user interface.

Figure 2: Integrated feature and mock-up view

8 Integration

In order to be effective, a mixed interface representation must satisfy four demands:

1. *Supplementary representations.*

The most important demand is that the different representations must supplement each other. A representation is only valuable if it communicates some part of the task data more clearly than the other representations.

2. *Integrated in one view.*

Multiple representations can be offered to the designer in different ways (see figure 3). The first way is to offer them as alternative views. In this case the user has to switch between the different representational modes. A second way is to offer them simultaneously, next to each other (typically in different windows). Now, the user does not need to execute a switch command; the different views are permanently present. The user can also use them together; however, there is no representational hint on how the different representations fit together. The third and final way is to integrate the representations in one view. Now, the different representations can really supplement each other. This is the most effective case.

3. *Mutually consistent.*

The different representations must at all times stay mutually consistent. For instance, when data is updated through the altering of one representation, the system must immediately update the other representation accordingly. If this is not possible (e.g. because it requires some time to compute), the other representations must without delay be altered to visualise its temporarily incorrect (unreliable) state. The representations must also stay mutually consistent with respect to the feedback on the manipulation actions of the user. For instance, when the user selects an object in one representation, the other representations must correspondingly visualise what has been selected.

4. *Mutually referring.*

Representations should mutually refer to each other in order to suggest the use of the other representations where that is beneficial. Take the example of a user manipulating an object. If similar or related actions can be performed using another representation, this must be communicated to the user; if the effect of an action is visible in another representation than this should be clearly indicated.

For the VR-DIS system, we have integrated the feature and the mock-up representations on the basis of the four presented demands. The most important aspects of this integration are the following.

Supplementary representations: the features will display exact numbers (e.g. "width = 5.20m" and "thermal performance has been estimated at 67K"). The mock-up will only visualise a selection of these numbers in qualitative way (e.g. dimensions are presented as such;

thermal performance could be visualised by a colour in the range from red (badly isolated) to blue (well isolated).) On top of that, the mock-up will provide a means to evaluate the spatial experience of the designed spaces.

Integration in one view: while "walking" through the virtual mock-up, the user will be able to point out an object (i.e. a building part) and ask for the features that described that object. The demanded features will appear in a frame near the object. The frame with the feature representations is itself an object that behaves similarly to other objects of the mock-up.

Consistency: the consistency will be maintained between the representations of the task data. For example, the dimensions of an object in the mock-up will always correspond with the numbers that can be read from the dimension features of that object.

Mutually referring: the frames in which features appear will be visually connected to the mock-up object. The objects in the mock-up and the corresponding features in the frames will always be simultaneously highlighted. When a feature is touched (pointed out) or grabbed (selected), both the feature and the mock-up object will be highlighted, and vice versa. Finally, when a property of either an object or a feature is modified by manipulating a particular handle or hot-spot, then all the handles and hot-spots of the other representation with which the same result could be obtained, will also be highlighted during the execution of the operation.

This mixed feature and mock-up view is now being implemented and will be experimentally tested.

9 Conclusion

We have discussed the theoretical foundations of mixed representation interfaces. Both the UI theory and design theory indicate the importance of a shared task domain representation between user and computer. We argued that a single-task representation often only highlights a subset of the task data. This is more becoming an issue as software applications are increasingly dealing with hybrid types of data. Mixed representation interfaces can provide an answer to this problem. Especially the combination of a verbally oriented and a pictorially oriented representation seems generally valid.

References

- [1] O. Akin, *Psychology of Architectural Design*, Pion Limited, London
- [2] G.C. van der Veer, "Mental Models of Computer Systems: Visual Languages in the Mind", in *Cognitive Aspects of Visual Languages and Visual Interfaces*, M.J. Tauber, D.E. Mahling and F. Arefi (eds.), North Holland-Elsevier Science, Amsterdam, 1994
- [3] B. de Vries, "VR-DIS research program", the WWW-site of the Building Information Technology Group, <http://www.calibre.bwk.tue.nl/oz/>, 1997
- [4] M.K.D. Coomans and R.M. Oxman, "Prototyping of Designs in Virtual Reality", in *Proceedings of the 3rd Conference on Design and Decision Support Systems in Architecture and Urban Planning*, Spa, Belgium, August, 1996
- [5] M.K.D. Coomans and H.J.P. Timmermans, "Towards a Taxonomy of Virtual Reality User Interfaces", in *Proceedings of the conference on Information Visualisation*, London, August, 1997
- [6] R.M. Baecker and A.S. Buxton, "Interaction Styles and Techniques", in *Readings in Human-Computer Interaction: A Multidisciplinary Approach*, Morgan Kaufmann, San Mateo, California, 1987, p. 427-435
- [7] B. Shneiderman, "A Taxonomy and Rule Base for the Selection of Interaction Styles", in *Readings in Human Computer Interaction: Toward the Year 2000*, R.M. Baecker, J. Grudin, W.A.S. Buxton and S. Greenberg (eds.), Morgan Kaufmann, San Francisco, USA, 1995, p. 401-410
- [8] J.A. Ballas, C.L. Heitmeyer and A.P. Manuel "Evaluating Two Aspects of Direct Manipulation in Advanced Cockpits", in *Proceedings of the ACM conference on Human Factors in Computing Systems - CHI'92*, May 3-7, 1992, p. 127-134
- [9] E.L. Hutchins, J.D. Hollan and D.A. Norman "Direct manipulation Interfaces" in *User Centered System Design*, D.A. Norman and S.W. Draper (eds.), Lawrence Erlbaum Ass., Hillsdale, 1986, p. 87-124
- [10] S.W. Draper, "Display Managers as the Basis for User-Machine Communication", in *User Centered System Design*, D.A. Norman and S.W. Draper (eds.) Lawrence Erlbaum Ass., Hillsdale, 1986, p. 339-352
- [11] H. Thimbleby, *User Interface Design*, Addison-Wesley, Amsterdam, 1990, p. 470, p. 44-46
- [12] G.J.F. Smets, P.J. Stappers, K. Overbeke, C. vander Mast, "Designing in virtual reality: perception-action coupling and affordances", in *Virtual Reality Software & Technology - Proceedings of the VRST '94 Conference*, World Scientific Publishing, London, 1994, p. 97-110
- [13] H.A. Simon, "The Structure of Ill-Structured Problems", in *Developments in Design Methodology*, N. Cross (ed.), John Wiley & Son Ltd., Chichester, 1984. Originally published in *Artificial Intelligence*, 4, 1973, p. 181-200.
- [14] H.W.J. Rittel and M.M. Webber, "Planning Problems are Wicked Problems", in *Developments in Design Methodology*, N. Cross (ed.), John Wiley & Son Ltd., Chichester, 1984. Originally published as part of "Dilemmas in a general theory of planning", *Policy Sciences*, 4, 1973, p. 155-169.
- [15] H. Achten, *Generic Representations - An approach for modelling procedural and declarative knowledge of building types in architectural design*, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, 1997
- [16] N.O. Bernsen, "A toolbox of output modalities", the Amodeus-II WWW-site at <http://www.mrc-apu.cam.ac.uk/amodeus/>, 1995.
- [17] J.P. van Leeuwen and H. Wagter, "Architectural Design-by-Features", in *CAAD futures 1997. Proceedings of the 7th International Conference on Computer Aided Architectural Design Futures*, Munich, 4-6 August 1997, Junge, Richard (ed.), Kluwer Academic Publishers, Dordrecht, p. 97-115