Designing Virtual Environments for Usability

Kulwinder Kaur

Submitted for Examination of Doctor of Philosophy

Centre for Human-Computer Interface Design
City University
London

June 1998

Table of Contents – Volume 1

Thesis Tables

Thesis Figures

Acknowledgements

I would like to thank my main supervisor, Professor Alistair Sutcliffe, for his dedicated help, advice, inspiration and encouragement throughout my PhD. I would like to thank my second supervisor, Dr Neil Maiden, for his continuous support and encouragement, and active assistance and organisation. Thanks as well to the Engineering and Physical Sciences Research Council and the School of Informatics, City University, for their funding. Thanks to the subjects and designers that took part in the studies, in particular the Royal Navy School of Educational and Training Technology and LTCDR. Dennis Dowsett; The Rural Wales Development Board and VR Solutions, including Mr Phillip Trotter, Professor Robert Stone and Mr Andrew Connell; and Superscape including Mr Andy Wilde, Mr Alex Tearse, Mr Dave Griffith and Mr Garf Bailey. I also thank Mr Simon Attfield, Mr Ivor Benjamin, Dr Fergus Bolger, Mr Simon Bore, Ms Bernice Bright, Ms Elizabeth Bromley, Ms Beverley Copeland, Ms Linda Coughlan, Ms Maia Dimitrova, Ms Anne Doubleday, Dr John Dowell, Dr Geoff Dowling, Mr Mark Ennis, Mr Peter Faraday, Professor Anthony Finkelstein, Mr Oliver Grimwood, Mr Trevor Hackett, Mr Hu Jiawei, Ms Anny Koh, Mr Darrel Manuel, Ms Carol Mason, Dr Shailey Minocha, Mr Nigel Mitchem, Mr Cornelius Ncube, Mr Miguel Ortega, Ms Uma Patel, Dr Gordon Rugg, Ms Michelle Ryan, Ms Yael Shmueli, Dr Walter Smith, Dr David Snowdon, Dr Mark Springett, Ms Jolanda Tromp and Mr Andy Whitcroft for various acts of help and support. Finally, I wish to thank friends and family for their ongoing support.

Declaration

The author grants the power of discretion to the university library to allow the thesis to be copied in whole or in part without further reference to the author.

Abstract

This thesis investigates user interaction in virtual environments and usability requirements to support that interaction. Studies of the design and use of virtual environments are used to demonstrate the need for interface design guidance. A theory of interaction for virtual environments is proposed, which includes predictive models of interactive behaviour and a set of generic design properties for supporting that behaviour. The models elaborate on D.A. Norman's cycle of action to describe the stages involved in three modes of behaviour: task and action based, exploratory and reactive. From the models, generic design properties are defined for various aspects of the virtual environment, such as its objects, actions and user representation. The models of interaction are evaluated through empirical studies of interactive behaviour which compare observed interaction patterns with those predicted. The generic design properties are evaluated through usability studies that investigate the links between missing design properties and usability problems encountered. Results from the evaluation studies provide general support for the theory and indicate specific refinements required. A controlled study is used to test the impact of the theory on interaction success, by comparing performance in virtual environments with and without implementation of the generic design properties. Significant improvements in interaction are found with the use of a virtual environment, after the predicted design properties have been implemented. Design guidelines are then developed from the theory and a hypertext tool designed to present the guidelines. The tool and guidelines are evaluated with industrial virtual environment designers to test the usability and utility of the guidance. Results indicate that the guidance is useful in addressing the practical problem of designing virtual environments for usability. Therefore, this thesis fulfils its objective of developing interface design guidelines for virtual environments, using interaction modelling as a theoretical base. Furthermore, it provides an improved understanding of user interaction in virtual environments and can be used to inform further theories, methods or tools for virtual environments and human-computer interfaces.

Chapter 1

Introduction: Designing Virtual Environments for Usability

This chapter introduces the problem and gives an outline of the thesis.

Chapter 1

Introduction: Designing Virtual Environments for Usability

As new technologies emerge, different types of computer system become possible, with often more graphical and sophisticated user interfaces. A suite of technologies such as head-mounted displays, position-trackers and, most of all, powerful graphics computers have enabled Virtual Environment (VE) interfaces to be realised. However, new technologies must undergo a process of maturity and frequently suffer teething problems. For example, in the early technology-driven phase (Winograd, 1995), a new technology can be difficult to employ and its potential benefits may not be wholly apparent or widely realised. Currently, VEs appear to be in this phase of maturity. Design is practised as a craft using intuition, based on experience (Long and Dowell, 1989). However, others do not learn from this experience and novice designers face a steep learning curve. Therefore, this thesis begins with the assertion that designers should be able to practice at a more engineering level, according to a set of principles and methods.

The design of computer systems is a complex process, involving a number of stakeholders. Major stakeholders are designers, end-users, and the sponsors and organisations involved. Different stakeholders can have different requirements and the resulting system will inevitably be a compromise. Considerations in design range from development cost, reliability, utility and general human factors issues. Human factors include health and safety, ergonomics, motivation, and usability. Usability centres around how comprehensible, easy to use, efficient and pleasant the system is for the end-user. It is important in the overall success of a system in that it affects how well the user can carry out their task or meet goals when using the system (Nielsen, 1993). For voluntary-use systems, such as entertainment or marketing applications, usability is also important in affecting the user's motivation to use the system. There has been only partial application of usability principles in industry, but these have led to some improvements in interface design. For example, graphical user interfaces were introduced as a more natural and easier to use alternative to command-based interfaces (Sutcliffe, 1995). The alternative to considering usability is poor quality and extra cost incurred by making required changes later in development (Gould and Lewis, 1985). Users may have to cope with frustration, fear and failure as a result of being faced with excessive complexity, incomprehensible terminology and chaotic layouts in a system (Shneiderman, 1992). Therefore, a second assertion of this thesis is that usability is a key factor in the design of a system. Guidance on designing for usability exists for conventional interfaces, such as Direct Manipulation (DM) interfaces (e.g. ISO, 1996). However, VEs differ in important ways from conventional interfaces and require specialised guidance. Therefore, the aim of this thesis is to develop usability guidance for VE design.

1.1 Virtual environments: a novel interface style

Virtual Environments are three-dimensional, computer-generated, simulated environments that are rendered in real time according to the behaviour of the user (Loeffler and Anderson, 1994). Virtual environments differ from conventional interface types, bringing new challenges to human-computer interface design. Comparing the interface structure of VEs with a predecessor, direct manipulation, highlights some major differences:

- VEs are structured as 3D graphical models. Therefore, they involve an additional dimension and consist of graphical objects embedded in a world, as opposed to a mixture of graphical icons and text labels in DM interfaces.
- The VE model represents some real-life or artificial structure or place, that has a fairly static spatial organisation. DM systems provide a 2D display area that continually presents objects of interest (Shneiderman, 1982) to the current task.
- Only a sub-section of the model is available through the VE interface at any one time, according to the current user position. The user must navigate around the model to locate objects of interest. There is no concept of user position in DM interfaces (apart from the mouse cursor). The user manipulates the DM interface as required, for example by moving or re-sizing interface objects and opening/closing windows.

The following figures illustrate these differences. Figures 1.1 and 1.2 give screen-shots of example DM interfaces, showing 2D display areas for current interface objects of interest, such as icons and menus. Figures 1.3 and 1.4 give screen-shots of example VE interfaces, showing viewpoints to 3D large-scale structures.

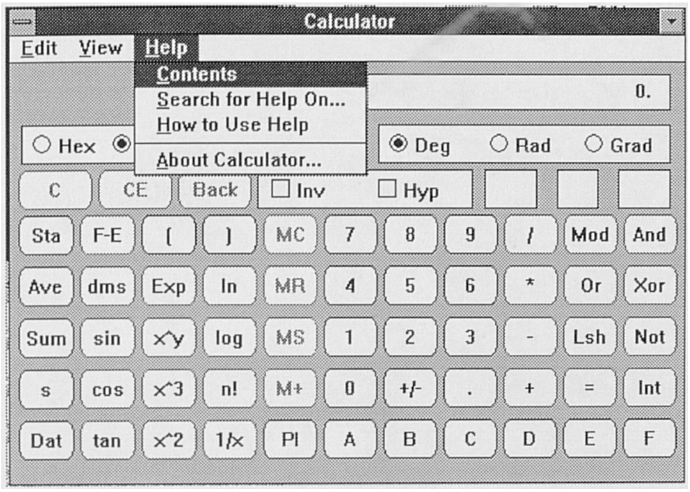


Figure 1.1: Screen-shot of the Windows Calculator direct manipulation interface.

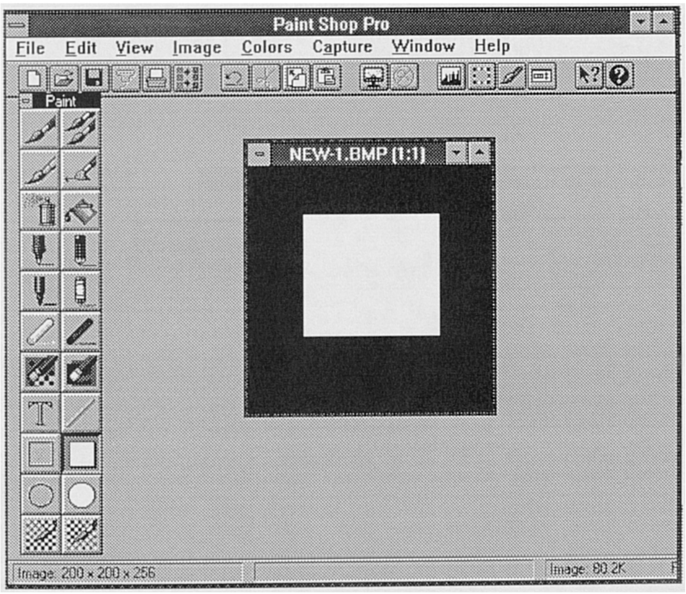


Figure 1.2: Screen-shot of the Paint Shop Pro direct manipulation interface.

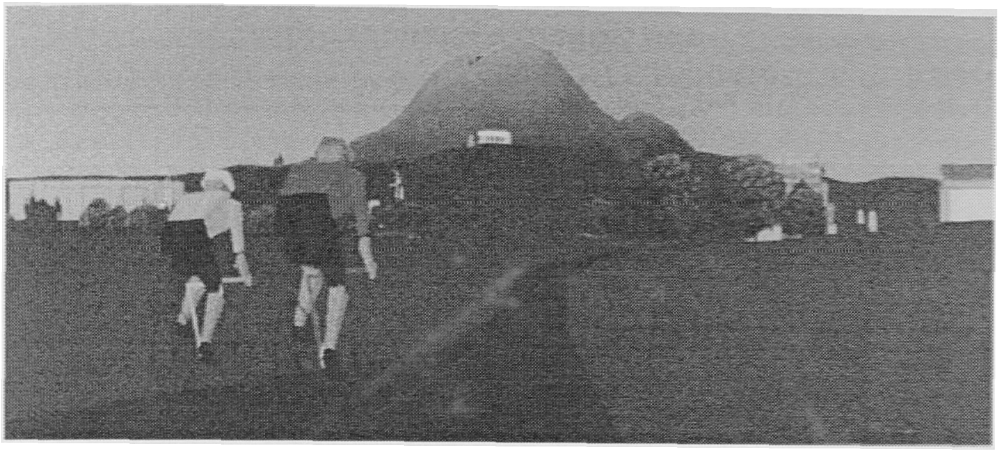


Figure 1.3: Screen-shot of the Diamond Park virtual environment interface *(from Waters et al., 1997)*.

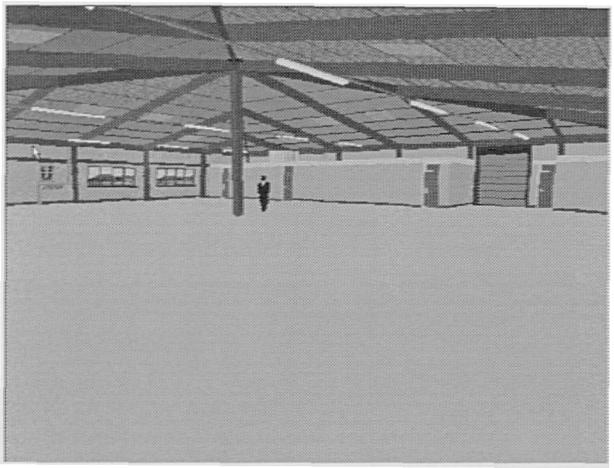


Figure 1.4: Screen-shot of a business park virtual environment interface, developed by VR Solutions for the Rural Wales Development Board.

Other special features particularly include the more exploratory nature of interaction in VEs, due to the typical open-ended task structure. For example, the task of learning about a subject area in an educational VE will involve much exploratory behaviour. Virtual environments are often active, with objects operating independently of the user's actions (Bryson, 1995). Novel input and output devices are used, such as head-mounted immersive displays, data gloves and 3D mice (see figure 1.5). Virtual environments typically model real world domains, and use multi-modal and natural interaction styles, such as human-like navigation and viewpoint control. Graphical representations of the user, for example body shapes, are used to provide information such as position, activity and capabilities (Benford et al., 1995). A general aim of

VEs is to promote user presence, the experience of 'being in' or, with a lesser degree, 'being in contact with' the VE (Loomis, 1993).

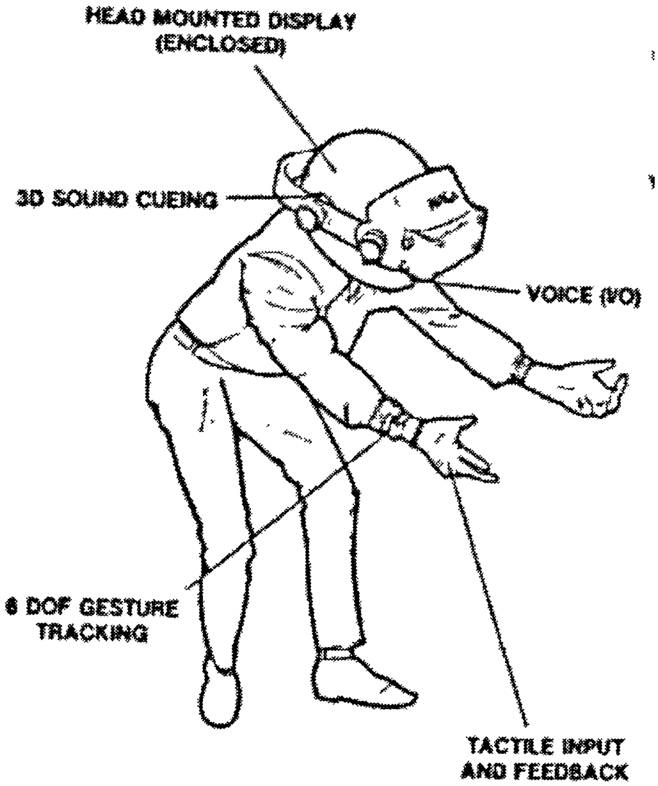


Figure 1.5: Some of the specialist input and output devices used in virtual environments, from Fisher (1990).

1.2 Design and usability issues with virtual environments

Virtual environments offer considerable potential for addressing certain business problems, such as rapid product development and early testing of prototypes (Dai et al., 1996). However, the novel concepts and technology involved with VEs lead them to be significantly more difficult to design, implement and use than conventional interfaces (Herndon et al., 1994). Current understanding about VEs is poor and there is a need for better-designed VE systems (Bolas, 1994). There is little knowledge about how VEs are being designed, what issues need to be addressed, and little guidance about how design should be carried out. Previous work has mainly cited isolated experiences in design and focused on technical issues, such as improving run-time performance. For example, Hubbold et al. (1993) suggest adaptive rendering techniques to maintain performance, such as only showing the parts of the graphical model within the field of view. Human factors are an important area that needs to be addressed in VE design (Macredic, 1995; Rushton and Wann, 1993; Höök and Dahlbäck, 1992), after all, the purpose of the system should be to serve the end-user, not to use a specific technology or be an elegant piece of programming (Norman, 1986). Some progress has been made towards methods specifically for viewpoint control (e.g. Drucker and Zeltzer, 1994) and research has begun into a user's perceptual requirements during interaction (e.g. Rushton and Wann, 1993; Carr and England, 1993). More extensive research has been carried out into motion sickness (e.g. Oman, 1993; Regan, 1995; Kennedy et al., 1997) and the 'presence' factor for immersive VEs (e.g. Held and Durlach, 1993; Welch et al., 1996; Hendrix and Barfield, 1996a & 1996b, Sheridan, 1996; Slater and Wilbur, 1995; Tromp, 1995). However, comprehensive design guidance does not exist and some user issues have received no directed research. In particular, there is a need to consider general usability requirements (Herndon et al., 1994; Boyd and Darken, 1996; Höök and Dahlbäck, 1992) that will cover the range of interaction tasks a user may be engaged in, from goal-formation to perception and action.

Users interacting with VEs have been known to suffer from frequent difficulties, which can result in frustration and an overall low system usability (Miller, 1994). For example, common problems are disorientation, perceptual misjudgements and confusion with unnatural interactions. In part, this is because although, in an ideal VE, interaction would perfectly mimic interaction in the real world, in practice, VEs often differ from their real world counterparts. There may be necessary *limitations* in interactions, such as the absence of tactile feedback, *substitutions*, such as the use of gestures for navigation instead of whole body movement, or *empowerments*, such as the ability to walk through walls. The user needs to be able to adapt to and tolerate limitations, understand and adapt to substitutions and take advantage of empowerments. Instead of perfectly natural interaction styles, VE interactions are more complex and introduce usability issues that require interaction support and careful design.

1.3 Modelling interaction to inform usability

The principle goal of the field of Human-Computer Interaction (HCI) is to provide an understanding of how human motivation, action, and experience place constraints on the usability of computer equipment (Carroll, 1987). To do this requires knowledge of human behaviour. We need to understand what people do when they use the systems and how much the average user knows about the interface and task (Reisner, 1987). Designers have difficulty understanding the average user, since they themselves know too much about the interface and task (Rheingold, 1990). Therefore, to inform design guidance for usability, theoretical models have been used in HCI.

Models can cover three aspects of the interaction of users and computers (Olson, 1987):

- The behaviour of the computer system - task requirements and characteristics of the computer system.
- The person's processing - ongoing mental and motor activity.
- The person's cognitive capacity, strengths and limitations in processing information. For example, limitations include memory and patience, whilst strengths include visual scanning and integration of patterns in time and space.

There are various approaches to modelling in HCI. Models can focus on different aspects of the interaction, such as system models (e.g. PIE, Duke et al., 1994), models of cognition (e.g. Interacting Cognitive Subsystems, Barnard, 1987), models of user knowledge and its use (e.g. PUMS, Blandford and Young, 1995; Cognitive Complexity Theory, Kieras and Polson, 1985), and models of interaction (e.g. GOMS, Card et al., 1983; Norman 1988). The focus of a model will affect the scope and depth of design requirements it can be used to inform. This thesis approaches HCI modelling by focusing on the processes involved in user interaction. Focusing on the system side of interaction can help to design a coherent set of interface modules, but cannot ensure a user's ability to successfully use the interface. Focusing on user cognition can help to check the interface demands on working memory and cognitive complexity are realistic, but this is a narrow focus which has little to say about the user's wider goals of understanding the system and completing their task. Focusing on user knowledge can help to ensure the user's ability to understand, learn about and form a mental model of the system. However, a user's priority, especially in a highly interactive task, will be the interaction process itself and their current task, rather than gaining knowledge about the system. Focusing on the processes involved in interaction can help to ensure the user's ability to successfully complete general interaction tasks. Therefore, this approach is appropriate for informing general usability requirements, as required for this thesis, although it does sacrifice depth for breadth in that highly precise or detailed guidance will not result.

Models can differ in the type of computer system for which they model interaction. For this thesis, modelling of interaction with VE systems is required. Some models, such as Norman's cycle of interaction (Norman, 1988), aim to describe interaction at a general level, applicable to any type of computer system. However, such models necessarily describe interaction at a high, abstract level and, therefore, can only offer generalised guidance. More specialised, subsequent models have been developed for interface styles such as direct manipulation (e.g. Springett, 1996) and information retrieval (e.g. Sutcliffe and Ennis, 1998). Specialised models can describe interaction at a detailed level for a particular style of interface and usability requirements can be related to interface objects specific to that style. Therefore, this thesis involves the development of models of interaction specifically for VE interfaces. A useful approach to modelling interaction for specific interfaces is to elaborate a general model. For example, Springett (1996) described interaction for direct manipulation interfaces by elaborating Norman's (1988) cycle of interaction. This model is particularly suitable for use in developing specialised process models of interaction because it is general, simple and well established. Therefore, in this thesis, the Norman cycle of interaction is used to develop specialised models of interaction for VEs.

Different methods can be used to develop and test HCI models. In this thesis, hypothetical models are first developed and then evaluated through user studies.

Models can be developed from data on interaction behaviour, but a large and diverse data set is needed to ensure the resulting models are not just representative of the particular systems observed, but are more widely applicable to the interface types of interest. Models can be developed, tested or refined using formal logic and computational implementation (e.g. Rauterberg, 1997; Kieras and Polson, 1985). Formal logic or computational implementation can help to validate the logical consistency and correctness of the models and show that paths through the models can be traversed successfully. Alternatively, models can be hypothesised, from knowledge about the interface type, and then tested by experimentation (e.g. Kitajima and Polson, 1996). Experimentation with users can help to evaluate how well the models describe actual observed interaction behaviour, and discrepancies found can be used to refine the models to be more accurate and representative. The models of interaction for VEs, in this thesis, are developed by hypothesising about interaction behaviour and then experimentation is used as the main method for testing and refining the models.

1.4 Presenting usability requirements to designers

Usability requirements are defined by reasoning about what the user requires from a design for successful interaction, as described in relevant interaction models. This thesis uses models of interaction in VEs to define guidelines for designing VEs for usability. Usability requirements can be presented to designers in various forms, from design methods and tools, guidelines and principles, to evaluation checklists. It is preferable to apply usability requirements during the design rather than in the evaluation phase, since there is less leverage in summative product evaluation when it may be too late to change the system (Card et al., 1983). Design methods incorporate guidance into an agenda of issues to be attended to in a structured set of procedures. Formal design methods are verifiable and can produce more reliable results. However, methods, and especially formal methods, can be restrictive, and a more flexible approach can better facilitate creative development. Also, designers can be preoccupied with meeting schedules, with little time to consider usability (Gould and Lewis, 1985), so usability requirements need to be quick to apply. Principles and guidelines do not incorporate any structured procedures so are more flexible and quicker to apply. Principles are high-level usability requirements and too general to be applied directly, so they are better translated into specific design guidelines, which describe heuristics of good practice. Guidelines can be useful memory prompts which show the variety of issues that need to be considered, but they need to be given with scoping rules and caveats to explain when they may apply, to avoid the guidance being too vague or conflicting (Reisner, 1987). To present guidance, computer-based tools have advantages over paper documents in that they can better structure the guidance and allow quick and flexible access. Therefore, this thesis delivers usability requirements for VEs using design guidelines, given with a context-of-use and examples, and presented in a computer-based tool.

1.5 Thesis objective, scope and hypotheses

This thesis addresses the problem of designing usable VE interfaces. The main objective is to develop guidelines for designing VEs from a usability perspective. The scope is the usability requirements to consider when designing basic VEs, since little existing HCI research has been carried out for this interface type. Therefore, the thesis focuses on the more basic and common type of VEs, which are single-user systems, generally modelled on real world phenomena. Abstract and multi-user VEs introduce additional usability considerations, such as metaphor design and communication between users, which are not within the scope of the thesis. The thesis research is more applicable to desktop rather than immersive VEs. Although it aims to model interaction behaviour at a level of abstraction beyond devices used and degree of immersion, evaluation studies to test and refine the models are carried out with desktop applications rather than with the currently less stable immersive applications. Additionally, it is important to consider desktop VEs, rather than only the more 'ideal' type of virtual reality provided by immersive environments, because most VEs created in industry are non-immersive. Within the development cycle, the thesis is concerned with interface design activities, in particular, presentation design (specifying the representation of interface components) and dialogue design (specifying interaction flow, user operations and system feedback). Figure 1.6 shows the scope of the thesis research.

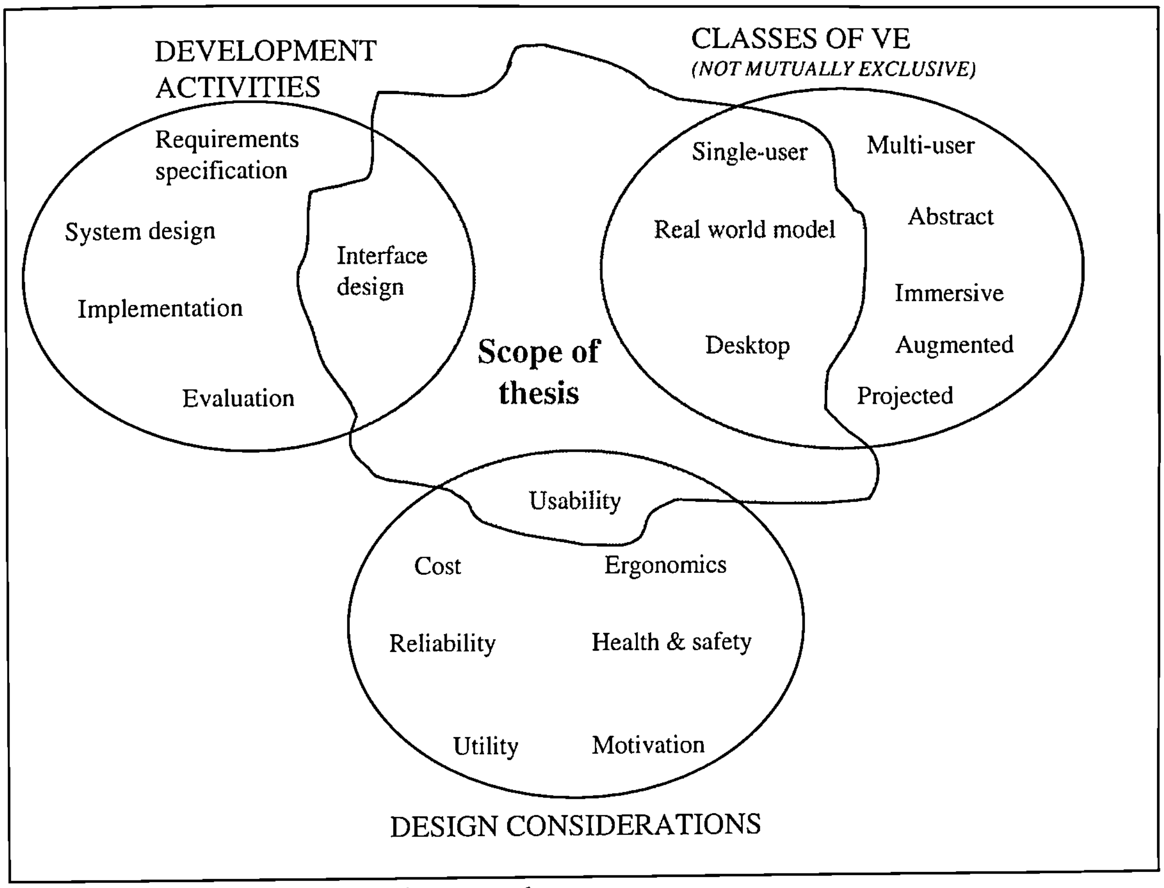


Figure 1.6: Scope of the thesis research

The hypotheses that structure the research are:

H1 There is a need for interface design guidance specifically for VEs.

H2 General patterns of interaction with VEs can be predicted, through theoretical models (i.e. major stages of behaviour and the common links between these).

H3 Design properties required for interaction can be predicted using the general patterns (H2).

H4 Interaction can be improved by implementing the design properties (H3).

H5 The design properties (H3) can be presented in a usable form to support VE interface design.

1.6 Methods used

The method for testing hypothesis one involves two studies into the design and use of VEs to assess the requirement for interface design guidance. The design study involves fact-finding structured interviews of industrial designers. The user study involves a usability evaluation of a VE in practical use, where observation is used to gather data on usability problems encountered. For hypothesis two, interaction modelling is used to predict major stages of interaction and common patterns of behaviour involving these stages. Empirical studies of user interaction behaviour in a VE are used to test the predictions, using 'think-aloud' protocol analysis techniques and observation.

The patterns of interaction are used to inform usability requirements, in hypothesis three, by systematically reasoning about design properties required to support predicted interaction behaviour. The method for testing the required design properties involves using the properties to assess the usability of a VE and predict likely usability problems with it. The problem predictions are then tested in a usability evaluation of the VE, by comparing them against actual usability problems encountered, from observation and 'think-aloud' protocols.

To test hypothesis four, missing design properties are implemented in a second version of a test VE. A controlled user study is carried out, comparing interaction success with and without implementation of the design properties. Observation and 'think-aloud' protocols are used to gather data on usability problems and task performance, and a memory test is used to investigate knowledge gained through interaction. Finally, for hypothesis five, the design properties are translated into design guidelines and a sub-set are presented in a hypertext tool. To test the usability and utility of the tool and guidelines, expert evaluation and critiquing is used within the context of focused design scenarios.

1.7 Thesis outline

This thesis consists of seven chapters. The next chapter, chapter two, describes existing research relevant to the thesis. It includes details about the design of VEs, what is known about how users interact with them and preliminary guidance for VEs. It also includes models and guidance for conventional human-computer interfaces. Chapters three to six describe the research carried out to meet the thesis objectives.

Chapter three describes two studies carried out to investigate hypothesis one, that there is a need for VE interface design guidance. The first study is a usability evaluation of a VE which highlights the major usability problems that exist. The second study investigates the design of VEs and shows that designers lack a coherent approach to design, especially interaction design, and do not consider user issues.

Chapter four describes the theoretical work carried out for hypotheses two and three. A theory of interaction behaviour in VEs is described, consisting of modes of interaction (task-based, exploratory and reactive) and stages of interaction. From this, generic design requirements for usability and relevant elements of user knowledge are derived. Correspondence rules link these components and predict the conditions under which usability problems are likely.

Chapter five describes empirical work carried out to test the theoretical work developed for hypotheses two and three. A study of interaction, using protocol analysis, provides data on observed user behaviour, which is compared with that predicted in the models. Data on usability problems encountered is compared with predictions made using the correspondence rules, and assessments of usability requirements met in a test application. A controlled study is used to test hypothesis four, by comparing usability with and without implementation of the predicted design requirements. Finally, the logic of the correspondence rules is tested through a partial computational implementation. The evaluation work provides general support for the theory components and results are used to refine and improve the theory.

Chapter six describes the development of design guidelines from the theory, and the evaluation of a guidance tool, for hypothesis five. The predicted usability requirements are translated into concrete guidelines and a hypertext tool is designed to present the guidelines. A partial implementation of the tool is evaluated by expert VE designers.

Chapter seven summarises this research and concludes with a discussion of implications and possible future directions. The research provides an improved understanding of interaction in VEs and delivers design guidance. Future directions include further work on VE interaction, completion of the guidance tool for designers, and work on methods for evaluating VEs.

Figure 1.7 shows the structure of the thesis.

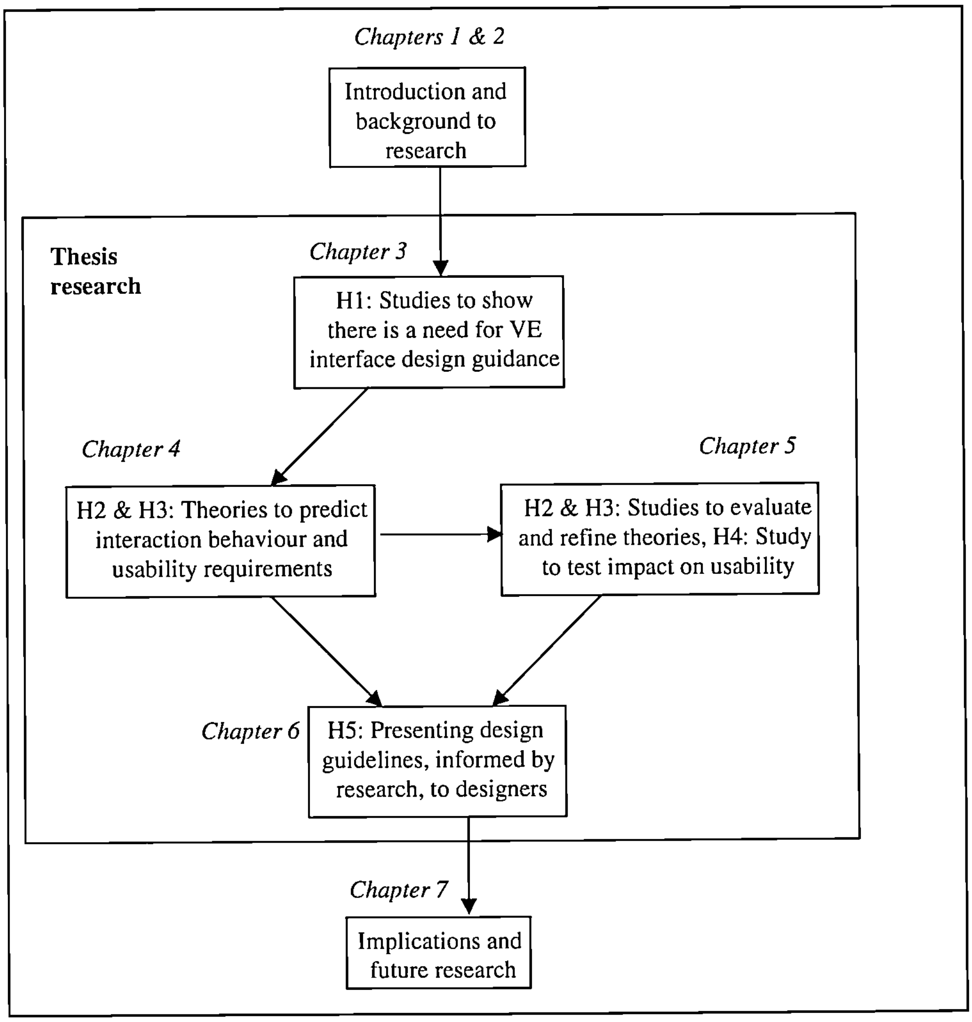


Figure 1.7: Structure of the thesis.

1.8 Contributions

This thesis has implications for research in VEs and HCI.

For VEs:

- A study of designers provides an improved understanding of how VEs are designed and what design issues exist.
- A predictive model of interaction provides an improved understanding of user interaction with VEs.
- Application of the model provides an identification of the design properties that are required to support interaction with VEs, and an identification of the probable usability problems where interaction is not supported.
- A controlled study demonstrates that addressing usability issues, using the theory, can significantly improve interaction with VEs, to a level where users can successfully utilise VEs to meet their goals.
- Extensive user studies provide further knowledge about interaction behaviour and usability issues.
- Concrete design guidelines, based on the theoretical research, provide clear guidance on designing VEs for usability.
- A prototype guidance tool provides the research results in an effective form for helping designers develop more usable VEs.

For HCI:

- The theoretical work provides an improved understanding of human-computer interaction, and helps define the nature of VEs and VE interaction, highlighting ways in which they differ from conventional interfaces.
- The model evaluation studies show that different modes of interaction behaviour, such as task-based and exploratory, are important to recognise and can co-exist in any interaction session.
- The theoretical work provides an identification of some general design properties to support interaction, which can be applied to other interface types.

- The modelling approach demonstrates a successful path to developing models for specific interface types, and developing usability requirements from these models.
- The model evaluation studies demonstrate a useful way of testing complex, informal HCI models, using protocol analysis techniques.
- The guideline evaluation work demonstrates that HCI research can be successfully delivered to industrial designers.

Chapter 2

Virtual Environments as Human-Computer Interfaces

This chapter provides background knowledge for the thesis and details relevant research.

Chapter 2

Virtual Environments as Human-Computer Interfaces

2.1 A definition of virtual environments

Virtual environments are also referred to as *virtual worlds*, and the concept they capture can be referred to as *virtual reality*. There are no generally agreed definitions of these terms. Loeffler and Anderson (1994) define virtual reality as an artefact:

> *"Virtual reality is a three-dimensional, computer-generated, simulated environment that is rendered in real time according to the behaviour of the user."*

Ellis (1993a) provides a more detailed definition of the artefact:

> *"A virtual environment consists of content (objects and actors), geometry and dynamics, with an egocentric frame of reference, including perception of objects in depth, and giving rise to the normal ocular, auditory, vestibular, and other sensory cues and consequences."*

Alternatively, Gigante (1993) defines the experience of interacting with a VE:

> *"Virtual reality is an immersive, multi-sensory experience. It is characterised by the illusion of participation in a synthetic environment rather than external observation of such an environment."*

From these definitions, the key features that characterise VEs appear to include 3-dimensional graphics and an environment model, representing some real-life or artificial structure or place. A user conceptually inhabits the environment, having a current position within it and, therefore, a limited view. The user has the ability to travel through and interact with the environment. Environment perception and interaction can be modelled on reality, for example multi-sensory stimuli and movement on a base plane.

Virtual environments often involve specific, novel hardware, which can provide sophisticated interfaces and new possibilities for interaction:

- Hand-held devices may be used to move around the environment in all dimensions, with varying degrees-of-freedom, such as *joysticks, space balls* or a 3D *mouse. Data gloves* which fit over the hand and can be used to determine hand shape/gesture, for example for interaction with virtual objects. *Position trackers* or *video imaging* can be used to sense head or body movements, for movement through the environment or interaction with objects.
- *Head-mounted displays*, which include independent screens for each eye, may be used to provide an immersive, stereoscopic display. *Video projection* can display the environment onto physical areas, such as walls of a room or a table-top.
- *Tactile/force feedback devices* can be used to simulate sensation applied to the skin, or emulate gravity or resistance to motion. *Three-dimensional audio devices* can provide stereoscopic sounds to indicate the location of sound sources in the environment.

Virtual environments can be differentiated according to their relationship to the real world (Bolzoni, 1994) and the hardware involved:

- *Immersive environments* involve immersive head-mounted displays and do not include data from physical reality (Slater and Usoh, 1995).
- *Desktop environments* involve desktop displays, which allow the user to maintain awareness of physical reality.
- *Projected environments* involve a physical space onto which the virtual environment is projected, such as a room as in Cave projects (Wloka, 1996) and The Virtual Dome (Hirose, 1996) or a workbench (Krüger et al., 1995).
- *Augmented environments* involve virtual objects that are overlaid onto the real world, possibly using see-through head-mounted displays (Adam, 1993).

The different styles of VE interface offer specific benefits. Immersive environments have the potential to provide the user with an absorbing experience of the environment. However, they can be isolating, involve intrusive technology and have been associated with health problems (Travis et al., 1994). Desktop environments avoid these problems, but do not provide a fully immersive experience, instead relying on psychological immersion taking place (Robertson et al., 1993). Projected environments involve a more natural setting and the capability to share the experience with other users (Krüger et al., 1995). Augmented environments enhance the real world with extra virtual content and information (Adam, 1993), but require precise correlation between the real image and superimposed information (Wloka, 1996).

2.2 Applications

Virtual Environments have been used in several specific application areas. For *arts* and *entertainment,* VEs can provide a more exciting and interactive experience for the player, for example virtual reality games or 3D theatre. *Marketing* applications, such as guided travel tours before booking a holiday or a virtual exhibition stand, attract the attention of potential customers and enable them to experience products in new and more realistic ways. In *teleoperation,* VEs provide a realistic interface through which hazardous or remote real world tasks can be carried out indirectly by manipulating robots, for example repairing nuclear reactors or computer network maintenance. *Telepresence* or *collaborative* VEs, such as telemonitoring surgery (Docimo et al., 1997) or virtual cities (Loeffler and Anderson, 1994), enable remotely located people to communicate or work together in a realistic environment. *Design & evaluation* applications enable the visualisation of a design and provide for more effective testing, by allowing the design to be viewed from inside and around, and manipulated. For example, animated mannequins to evaluate clothes or testing fire-safety by navigating through virtual models of buildings. For *education,* VEs enable a user to explore and learn about a subject area by experiencing it more directly, especially where the subject is not accessible in reality (Bullinger, 1996). For example, learning about physics through NewtonWorlds (Dede et al., 1994), or learning about chemistry by interacting with virtual chemical structures. *Training* VEs, such as flight simulators for pilot training or virtual operations to train surgeons, provide realistic environments in which to practice certain skills, especially spatial cognition skills (Regian and Shebilske, 1992). Similarly, *treatment* applications, such as for treating acrophobics (Hodges et al., 1995) or role playing for patients undergoing psychotherapy, provide realistic environments in which to help patients overcome problems through exposure to gradually more difficult situations. Finally, *information visualisation* VEs provide an interactive 3D visual model of a set of information, potentially aiding exploration and interpretation of the data, for example information management with the NASA VIEW system (Fisher, 1990) or virtual libraries.

Therefore, VEs have been usefully applied to a variety of applications. These applications share a common theme, which is the provision of computer-generated worlds for the user to experience, explore and manipulate. The worlds represent some real-life, or possibly artificial, structure or place. Where real-world phenomena is modelled, VEs may be especially useful if the physical counterpart cannot be visited for reasons such as its size, location or danger. Finally, VEs are not useful for applications where it is not appropriate for the user to move around the space and experience it from different positions.

2.3 Designing virtual environments

Creating VEs primarily involves designing the environment model and designing user interactions. Graphical components in the environment model can be categorised as (Ellis, 1993a; Thalmann, 1994):

- the *background space*, or *geometry*;
- the *user*, or *self*, which can carry out actions and controls the viewpoint;
- *agents*, or *virtual actors*, which have intelligence to carry out actions independently of the user, and
- *objects* that populate the background space. Objects vary in their level of interactivity and changeability.

Basic user interactions in a VE are (Bordegoni, 1993; Herndon et al., 1994):

- *navigation* and *viewpoint control*, and
- *object interactions*, such as picking an object; grabbing, rotating and moving objects; manipulating objects to change their state; and, querying to find out the content of an object.

The design of components and interactions in VEs have been investigated individually, and various possibilities, techniques and issues have emerged.

2.3.1 Designing the environment model

The background space of the environment may be based on an existing real world domain, such as a particular building. Alternatively, it can be based on different spatial metaphors, such as a city or countryside metaphor (Benyon and Höök, 1997). It can be represented in graphical views, such as wire-frame (Osborn and Agogino, 1992) to minimise obscured elements (Stytz et al., 1995). For example, visualisation modes, possible in surgical applications, can include transparent views or views with and without blood circulation (Krüger et al., 1995). Lighting of the background space is an important issue, for example good lighting can increase perceptual precision (Stoper and Cohen, 1993). Illumination from above is a natural lighting style (Murta, 1995), but multiple light sources, such as street-lights, may be useful, for example in producing complex shadows (Wann and Mon-Williams, 1996b).

Within the visual field, Slater and Usoh (1995) claim the most significant structure is the representation of the user's own body. Therefore, the user representation is an important part of the environment model. User objects can aid communication in multi-user applications (Capin et al., 1995), for example by providing information about location and identity (Benford et al., 1995). In certain applications, user objects are useful for ergonomic assessment of different body postures, such as a kneeling posture (Wilson et al., 1995). Slater and Usoh (1994) assert a general requirement for the user object to be consistent and predictable, for example by providing a good match between the virtual body and proprioceptive expectations about the position and orientation of the user's limbs.

Various techniques have been used to represent the user object, but few comparative studies have been carried out. Techniques include human representations, such as blockies (Benford et al., 1995), an arm with hand (Slater and Usoh, 1994) or just a hand (Bordegoni, 1993). The view can even be delimited with edges that look like noses, cheeks and eyebrows (Psotka et al., 1993). Alternatively, the user object can assume a new body shape, such as that of a fish (Adam, 1993). The user object can be represented as just tools for different interactions, and this avoids a hand obscuring the view, for example an arrow may be used for navigation and 'cutters' for manipulation (Bordegoni, 1993; Poston and Serra, 1996). Dynamic cursor-like representations can be used, for example, a pointer which changes to a hand for direct manipulation (Osborn and Agogino, 1992) or a jack cursor which appears to indicate the availability of object translation (Venolia, 1993). Special cursors have been proposed for 3D interaction, such as a cone to better indicate position and orientation (Venolia, 1993) and a volumetric 'silk cursor' with levels of transparency to allow objects within the cursor and those occluded by it to be seen (Zhai et al., 1994). The silk cursor was found to better facilitate target acquisition than a simple wireframe cursor. Figure 2.1 shows the 3D cone cursor selecting an object and then changing to a jack cursor to allow object translation.

Agents are a possible component in the environment model. Agents in VEs can serve as instructors or assist users in navigation (Zeltzer and Johnson, 1994; Billinghurst and Savage, 1996; Mason, 1996), directing them towards important parts of the VE (Gallery et al., 1996). For example, in an oil-rig training VE (*demonstrated by Virtuality*), an agent addresses visitors and helps them escape a fire at the oil rig. The behaviour of agents can include specific events for attracting the user's attention or communicating with the user, for example through gestures, mimes and speech. There may also be more complex behaviours, such as following the user around or navigating for the user (McGlashan and Axling, 1996; Gallery et al., 1996; Billinghurst and Savage, 1996; Capin et al., 1995). Agents can be represented in similar ways to the user object. Thalmann (1994) proposes a behavioural animation system to help design and implement agents, which is composed of a *locomotor system*, a *perceptual system* (e.g. synthetic vision), and an *organism system* concerned with rules, skills, motives, drives and memory. Thalmann and Thalmann (1994) state very general qualities required in a virtual human, such as freedom, intelligence, perception, behaviour, memory, emotion and adaptation. However, detailed advice is not given on incorporating these qualities.

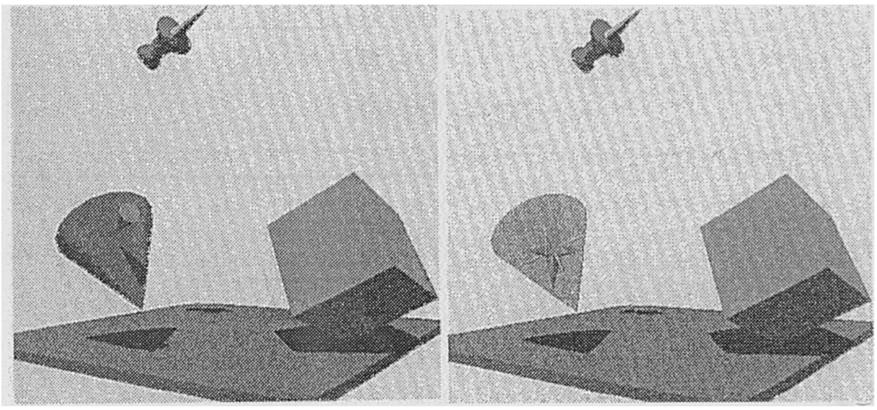


Figure 2.1: Cursors for direct manipulation in 3D, from Venolia, 1993. The left picture shows the 3D cone cursor selecting the cone-shaped object. The right picture shows the cursor changed to a jack to indicate the availability of a translation operation on the object.

The environment model will be populated with objects. Object representations have been generally based on real world phenomena. There has been little differentiation in representation techniques between different object types, such as interactive or non-changeable. However, some techniques have been discussed for representing commonly used objects that have specific functionality through virtual *tools*. Tools are graspable, portable and manipulatable, and when in use the tool can be a sort of extension of the user's hand (Gibson, 1986). Tools may be picked up from locations in the environment, or may be attached to the user (Rygol et al., 1995), for example in a 'tool belt/rack' (Rygol et al., 1995; Poston and Serra, 1996), or in a virtual surgical tray (Delp et al., 1997). Alternatively, tools may be permanently presented on the screen, for example the dashboard containing movement controls in some internet-based VEs (Mohageg et al., 1996).

Major general issues that have emerged in the representation of VE content are realism, depth-cueing and multimodality. Objects in the VE can be represented at differing degrees of realism. They can be exact copies or surrogates of reality, due to limitations such as level of detail, or they can enrich reality with extra information (Bolzoni, 1994). Choosing appropriate levels of realism has been discussed and factors proposed, but there is no concrete guidance available. The representation of objects need not be photorealistic; the level of fidelity required will depend on the task and perhaps also the user's expertise (Herndon et al., 1994). For greater realism, there is a need to model finer details, such as the terrain, lighting, shadows and textures (Gigante, 1993) and imperfections, such as stains, dents and natural variations (Fisher et al., 1996). More unrealistic representations can be effective in certain cases, for example, in medical displays, structures can be delineated with stronger contour lines for better clarity (McConathy and Doyle, 1993). High levels of detail can be reserved for more important objects (Brooks, 1994), nearer objects or objects in the centre of the display (Schraft et al., 1995), for example high-detail insets (Watson et al., 1995; Brelstaff, 1995). Greater detail can also be used in more complex scenes and where complex navigation and interaction is involved (McGovern, 1993). Algorithms and selection criteria have been outlined for determining appropriate levels of detail (Schraft et al., 1995; Reddy, 1995; Sen et al., 1995), which may prove useful in developing automatic level of detail management. Alternatively, information about where the user's eye is fixating may be used (Arthur et al., 1993), the user may select areas for detailed view (Yamaashi et al., 1996) or user trials can be used to determine levels of detail requirements (Pratt et al., 1997).

Communicating depth information is important for representing 3D objects. Various factors have been investigated and some progress has been made in comparing the benefit given by different depth cues. Pictorial depth cues may be employed, such as linear perspective, occlusion, shadows and detail perspective (Pimentel and Teixeira 1993, Wanger et al. 1992). Object motion can be useful for recovering 3D form (Proffitt and Kaiser, 1993), real-time shadows can indicate relative positions of objects (Arthur et al., 1993; Buck et al., 1996) and luminance and opacity can aid the perception of volumetric structure (Russell and Miles, 1993). Studies have shown that stereo viewing (with adjusted displays for each eye) can aid the perception of structure and position (Russell and Miles, 1993; Volbracht et al., 1997), especially with increased scene complexity and decreased object visibility (Kim et al., 1993) and for high precision localisation (Wann and Mon-Williams, 1996a).

As well as the visual channel, other modalities have been used in VEs. The literature discusses experiences and possibilities with the use sound and tactile interaction. Sound can be used to provide auditory background, communication or warning signals (Gilkey and Weisenberger, 1995), or be represented as sound effects causally linked to specific actions and events (Pressing, 1997). Sound can inform about invisible proceedings (Astheimer et al., 1994) and provide a second channel for information (Brown et al., 1989). For example, sound was used to provide extra feedback about energy values involved in a molecule manipulation system, whilst retaining visual attention on the manipulation (Cruz-Neira, 1996). Sound was used to create atmosphere in the Diamond Park VE (Waters et al., 1997), for example sounds of the wind and birds were used for a tranquil outdoor setting. Tactile interaction can be used to communicate properties of objects when manipulating and exploring (Bergamasco, 1994), such as shape, hardness, texture and speed (Johnson and Cutt, 1992), or can be used to provide feedback in dark or visually occluded scenes (Richard et al., 1996). For example, touch feedback is important for surgery applications, to provide the feeling of skin, soft-tissue and bone (Krüger et al., 1995). Sensory substitution may be used, where tactile sensation is not available. For example, when colliding with objects, auditory feedback may be substituted (Fisher, 1990). Multiple sensory channels may be used to provide redundant cues, for example both haptic and auditory feedback can aid performance in simple object manipulation tasks over lesser conditions (Richard et al., 1996).

Therefore, in designing the environment model, different components have been investigated and various techniques have emerged which highlight the range of possibilities available when creating VE interfaces. Some general issues in representation design, such as realism, have been highlighted. However, little evaluation or comparitive work has been carried out on the different techniques and there are few established standards or rules of good practice.

2.3.2 Designing user interactions

Viewpoint movement or navigation is an important basic interaction type in VEs and various techniques have been used to implement it. Navigation can be based on a point-and-fly, eyeball in hand (altering the eye-point within the scene) or scene in hand metaphor (Astheimer et al., 1994). Simpler techniques using a standard mouse are click go/click stop and slide and go (Strommen, 1994), or the use of different mouse buttons for forwards, backwards or stopping movement (Bliss et al., 1997). Alternatively, a world grabbing and pulling gesture (Mapes and Moshell, 1995), a virtual joystick (Mohageg et al., 1996) or physical 'walking in place' can be used. Walking in place can improve sense of presence in the VE, although pointing techniques can result in easier navigation (Slater et al., 1995). The viewpoint can be linked to objects or actions in the environment (Bolter et al., 1995; Stytz et al., 1995) and this helps maintain focus on objects of interest (Eyles, 1993). For example, in a space visualisation system (Eyles, 1993) viewpoints could be locked to the stars, Earth-fixed or move with spacecraft.

Issues that have emerged in the design of navigation are the degrees of freedom and the speed of navigation. The navigation technique can involve up to six degrees of freedom, but controlling all six can be problematic (Drucker and Zeltzer, 1994). Navigation can be restricted to two dimensions, excluding the ability to fly (Bliss et al., 1997), or different movement modes, such as pure positional (orientation preserving), pure orientational and combined (Deering, 1995) can be used. For example, in the Berlin 2010 project (Vorsteher, 1996) modes of navigation include free fly through or travelling as a pedestrian or passenger on a train. The navigation speed needs to be appropriate for the size of the scene (Mohageg et al., 1996). Faster speeds can reduce effort, giving rapid initial target acquisition, but slower speeds make attaining precise target approaches easier (Johnsgard, 1994). Constant velocity can result in problems, such as judging when to stop. Variable velocity is a more natural mode (Rushton and Wann, 1993) and gives users control over the speed of movement (Mohageg et al., 1996).

The other major interaction type in VEs is object interaction. Object interactions can be provided using techniques such as ray intersection for object selection, direct manipulation for object placement, or gestures (Herndon et al., 1994). Gestures are a natural interaction technique (Astheimer et al., 1994), especially if there are few restrictions on allowable gestures (Wexelblat, 1995). However, for target acquisition, a mouse can be faster and more accurate than a glove (Johnsgard, 1994). Two hands can be used for manipulating objects (Mapes and Moshell, 1995), for example two handed shrinking and stretching of objects (Mercurio and Erickson, 1990). Alternatively, speech can be used for interaction such as in the DIVERSE VE system (McGlashan and Axling. 1996). Speech interaction may be directed speech commands to agents or the environment generally, or dialogues with objects of interest (McGlashan and Axling. 1996). Although vocabulary may need to be restricted, speech does allow commands to be issued while keeping hands and eyes free, actions to be combined, and objects not in the current view to be referred to (McGlashan and Axling. 1996). Using speech with gestures is a useful technique, which can improve reliability in identifying gesture commands (Hauptmann and McAvinney, 1993). Techniques used for providing feedback about object interactions include sounds, coloration, and wireframe or translucent graphical rendering (Krüger et al., 1995; Dai et al., 1996; Buck et al., 1996; Venolia, 1993). Close-up view windows can be used for during-action feedback with complex interactions (Carlsson and Jää-Aro, 1995).

Issues that have emerged in the design of user interactions include fidelity, supporting interaction and attracting attention. Object interactions can be programmed at different degrees of realism, from being simplified real world operations, *mundane* when they faithfully reproduce real world interaction, or *magical* when they involve operations not possible in reality (Slater and Usoh, 1994). Required realism can depend on the application involved (Herndon et al., 1994). For example, for ergonomic analysis, object grabbing may need to be modelled close to reality, but ordinarily it may be sufficient to grab an object by just touching it (Dai et al, 1996). Where the link with natural reality is less constrained, metaphors can be used creatively to empower the user but effort is needed to ensure the user can understand the metaphor (Bolzoni, 1994).

Providing support for user interaction can be important and various techniques have been investigated to support different interaction tasks. Support can include guiding the user during exploration (Wilson et al., 1995; Mason, 1996) to ensure they are not lost, bored or overloaded with information (Astheimer et al., 1994). To aid object interaction, visible object-handles can be provided (Venolia, 1993) using 3D geometry to better enable manipulation (Buck et al., 1996). Support can be given for precise object alignment (Buck et al., 1996; Venolia; 1993) such as snap anchors (Mapes and Moshell, 1995). Powerful goal-based manipulation can be offered (Buck et al., 1996), such as 'show me object X' (Drucker and Zeltzer, 1994), and available interactions can be highlighted, such as all links in an internet-based VE scene (Mohageg et al., 1996).

Support to aid perception can include the ability to rotate objects and increase detail levels (McConathy and Doyle, 1993). Increasing the density of surface textures can improve velocity perception (Distler, 1996). To aid orientation, a ground plane and backdrop can be useful, for example as a base to judge object positions and whereabouts (Murta, 1995). Similarly, grids and reference lines can be useful (Ellis, 1993b; Nemire and Ellis, 1993) for simple object pick and place tasks (Kim et al., 1993), while walls, a floor and ceiling can help users stay oriented in a dataspace (Dickinson and Jern, 1995). Furthermore, automatic viewpoint correction (Klaiss, 1996) can be provided, such as preserving an upright view (Murta, 1995; Drucker and Zeltzer, 1994). Support for navigation can include landmarks (Stone 1994; Ranson et al., 1996), automatic navigation to specified areas (Mohageg et al., 1996), pathplanning and maintenance of a collision free path (Drucker and Zeltzer, 1994). Overviews can also be helpful (Mohageg et al., 1996), such as the Orientation Centre in the Diamond Park VE (Waters et al., 1997).

Attention can be an important variable in VE interaction (Pressing, 1997). Events in the user's *personal space* (which extends to auditory and other senses) can be used to attract user attention (Pimentel and Teixeira, 1993). For example, in a virtual wood application a dog barking sound alerted users to the presence of animals (Strommen, 1994) and jazz sounds are used to attract users to the Plaza Cafe in Diamond Park (Waters et al., 1997). Visual effects, such as bleaching, darkening and blurring, can successfully be used to highlight targets (Zhai et al., 1997).

Therefore, again in designing user interactions, a number of techniques have emerged which are interesting in that they demonstrate the various possibilities available in designing navigation and object interactions in VEs. Some general issues have also been highlighted, such as naturalness and realism. Some useful techniques have been suggested for supporting different interaction tasks. However, little evaluation work or guidance is available. Therefore, it is difficult to choose between the techniques since there is no clear evaluation of effectiveness or comparison of benefits and drawbacks. Furthermore, a simple set of possible techniques provides rather limited help to a designer and places early emphasis on implementation details. A deeper, conceptual understanding of user requirements when interacting with VEs is needed, to provide more complete and structured guidance that is generally applicable, clearly motivated and validated. To get this understanding, knowledge of interaction behaviour is essential.

2.4 Interacting with virtual environments

2.4.1 Understanding interaction behaviour

There are no comprehensive models of interaction behaviour in VEs, but there are fragments of knowledge about how users interact with VEs. General movement tasks that users carry out include orientation, navigation, wayfinding and exploration (Drucker and Zeltzer, 1994; Darken, 1995). Three wayfinding tasks are *naive searches*, where there is no knowledge of the target whereabouts, *primed searches* where the target location is known, and *exploration* where there is no relevant target (Darken, 1995). Users may not be systematic in their travel patterns, for example they may not examine all options at a junction before choosing a direction, and may not recognise visited areas of the environment (Strommen, 1994). However, users can employ wayfinding strategies, such as path-following, for example following coastlines or grid lines (Darken and Sibert, 1996). When locating objects, users may move continuously until the target is spotted or may make short movements and then stop to visually scan the environment (Strommen, 1994), especially when at known areas of interest (Watson et al., 1995).

General object tasks include inspecting and interacting with objects (Rushton and Wann, 1993). When the user wants to inspect or interact with an object they must navigate and approach the object, controlling their rotation and velocity so as to end up in the desired orientation and position (Rushton and Wann, 1993). The desired orientation and position is one at which the user can discern relevant visual detail or manipulate the structure (Wann and Mon-Williams, 1996b).

For multi-user VEs, an additional important interaction task is communicating with other users. Benford and Fahlén (1993) propose a spatial model of interaction, which helps understand how spatial properties affect awareness and interaction between users. The model includes the concepts of *aura*, *awareness*, *focus* and *nimbus*. The aura of an object is the sub-space which defines the scope of the object's presence in the environment, according to a specific medium (e.g. audio, visual or text). When the auras of two users collide, interaction between them in the medium becomes a possibility. The awareness between objects is a function of their focus and nimbus, further sub-spaces within which an object directs its attention and presence respectively. Therefore, if another user is within your focus, you can be aware of them, and, if another user is within your nimbus, it can be aware of you. The concepts of awareness and focus can also apply more generally to single-user systems, where the user will have perceptual awareness and focus of different parts of the environment as they change their position.

For the perception of VEs, theories of perceptual psychology can be useful in developing understanding (Carr and England, 1993; Rushton and Wann, 1993). Perception of the visual environment often involves identifying and recognising the objects within it. Rock describes three aspects of the perception of an object (as described in Taylor et al., 1982). First, the physical characteristics of objects are distinguished, such as shape, size and distance. Second, the object may be recognised as familiar and, third, the object may be identified in terms of its function and meaning within the environment. Both direct and constructivist theories can apply to the perception of VEs (Carr and England, 1993). Theories of direct perception (e.g. Gibson, 1986) assume that perception involves directly picking up light information reaching the eye, with little or no information processing. Therefore, VEs may be visually perceived by picking from a range of general or detailed graphical information in the environment model. However, the more generally accepted view of constructivist theories (e.g. that of Bruner and Gregory, as described in Neisser, 1976) is that perception is influenced by the internal hypotheses, expectations and knowledge of the observer. For VE perception, this points to the importance of the user's degree of familiarity with the environment (Carr and England, 1993), from exploration of it or from recognising features from other perceptual experiences.

Cognitive tasks that users carry out include understanding the relationship between motor commands and the virtual self, and building a mental model of the self and world from interaction experiences (Tromp, 1995; Loomis, 1993). There may be systematic perceptual distortions in the VE and perceptual learning will be required to understand and adapt to these. For example, adaptation can cause dramatic reductions in motion sickness symptoms for most individuals (Regan, 1995). Active exploration and familiarity with objects are important factors for perceptual learning (Kohler, 1964; Held and Bossom, 1961). Users learn and utilise spatial knowledge when interacting with VEs (Regian and Shebilske, 1992) and can even use this knowledge when navigating in corresponding real world environments (Witmer et al., 1996). Cognitive map is the name given to these spatial mental models which include landmarks, route schemas and survey representations of the environment (Medyckyj-Scott and Blades, 1991). To construct a good world model, sensory information should not be too complicated, too inconsistent or too limited, and the user has to enter a state of absorbed attention (Tromp, 1995).

The user's sense of presence is important in VE interaction (Slater et al., 1995), for example it can aid learning in novel and unpredictable contexts (Tromp, 1995). However, it may not be necessary for all tasks (Ellis, 1996). Presence is the experience of 'being in' the VE, but the feeling of 'being in contact with' the VE may be more common (Loomis, 1993), or being mentally absorbed on a task (Dede et al., 1994; Tromp, 1995). Presence can be affected by high realism (e.g. multi-sensory interaction, stereoscopic views and wide field of view), a virtual body, interactivity, control, familiarity and understanding, and isolation from the real world (Gilkey and Weisenberger, 1995; Held and Durlach, 1993; Welch et al., 1996; Hendrix and Barfield, 1996a & 1996b, Sheridan, 1996; Slater and Wilbur, 1995; Tromp, 1995). Presence can lead users to behave in a similar manner to everyday reality (Slater and Usoh, 1995). For example, users can tend to navigate around objects instead of through them, even if no wall/object collision is implemented (Hendrix and Barfield, 1996a), and users can expect to be able to navigate into areas that are usually available in reality, such as off a trail in a virtual wood (Strommen, 1994).

Individual differences between user interaction in VEs can be quite pronounced (Dede et al., 1994). Important differences include spatial abilities (Höök and Dahlbäck, 1992), which affect the usability of the system for different users (Benyon and Höök, 1997), concentration or absorption abilities (Psotka and Davison, 1993; Tromp, 1995), susceptibility to claustrophobic feelings (Psotka and Davison, 1993) and seriousness of perceptual problems. Individual differences in spatial ability may be linked to ability in mentally rotating figures, technical aptitude, learning style and experience, although some individual limitations of low spatial abilities may not be overcome by experience (Benyon and Höök, 1997). It may be possible that some people are more susceptible to virtual experiences (Rushton and Wann, 1993). Indeed, VEs may be particularly suited to certain personality types, such as extroverts and activists (Mason, 1996).

To summarise, there is some knowledge of cognitive factors, such as presence, but limited knowledge of general interaction tasks. However, it is during such interaction tasks, for example navigation, where significant usability problems have been found. Therefore, a better understanding of general interaction behaviour is necessary.

2.4.2 Interaction problems

User studies of VEs (Dede et al. 1994, Mercurio and Erickson 1990, Wanger et al. 1992, Psotka et al. 1993) have tended to evaluate the applicability of VEs or the most appropriate device for a particular application, or have evaluated the effectiveness of various depth cues, rather than focus on the overall usability. However, a few general usability problems have been uncovered in user studies.

Maintaining general spatial orientation can be difficult (McGovern, 1993). Disorientation and losing whereabouts commonly occurs when the navigation speed is too fast or the movement direction is not as expected (Miller, 1994). Novices can suffer from steering over-control, which results in oscillation about a desired path (McGovern, 1993). Perceptual problems can be common, such as difficulties distinguishing objects, and difficulties in size and depth perception (Rolland et al., 1995; Miller, 1994), for example users missing objects when reaching for them (Miller, 1994) and not perceiving negative obstacles, such as ditches (McGovern, 1993). There can be incorrect perception of movement with users feeling objects are moving towards them instead of feeling they are flying through the world (Miller, 1994). Problems with unnatural interaction can occur, for example difficulties manipulating grasped objects that are sticking unrealistically out of a virtual hand (Miller, 1994). Specific technical problems, such as slow display update rates, can also result in usability issues, as well as health problems such as motion sickness (Oman, 1993).

Suggestions have been made as to why some of the above problems occur. Perceiving a scene as moving towards oneself can be a result of the powerful dominance of optic flow information over stereo information (Ware, 1995). Misjudgements of depth and distance may be due to the fact that the eyes focus less accurately on virtual, rather than real, images (Carr and England, 1993), or because of the lack of anchor points (e.g. a ground plane) or variations in some virtual images (Rolland et al., 1995; Loomis, 1993). Motion sickness is typically caused by continuous, unexpected or unfamiliar, sensory information concerning the orientation and movement of the body (Oman, 1993). However, few of these suggestions have been translated into specific design guidance for avoiding the problems that exist.

2.5 Guidance for VEs

Creating VEs is difficult (Singh et al., 1994) and various toolkits have been developed to help construct environment models, for example the Bricks toolkit (Singh et al., 1994), the VR-MOG toolkit (Colebourne and Rodden, 1995), and the AVIARY framework (West et al., 1993). However, the design of VE interfaces, involves more than the simple construction of environment models or the recreation of real world models. It is important to consider human cognitive abilities and limitations (Höök and Dahlbäck, 1992), interaction support (Rushton and Wann, 1993) and ease of use and correct functionality (Mason, 1996) during design. However, there are no comprehensive methods or guidelines for considering user issues. Most research so far has been directed to requirements for supporting perception and wayfinding, and the design of viewpoint controls and user representations.

Drucker and Zeltzer (1994) present a framework for designing camera controls in a VE, based on an analysis of required tasks. The framework controls the placement and movement of virtual cameras through a network of camera modules. View constraints are incorporated, such as maintaining the camera's up vector to align with world up, maintaining a height relative to the ground, maintaining the gaze towards a specified object and maintaining the camera's position on a collision free path. A constraint solver combines these constraints to come up with the final parameters for a particular module. The framework has been applied to the design of a virtual museum, for example, pathplanning was used to generate a guided tour through a set of selected paintings. The framework is useful in exploring view controls in a VE. However, the link between view controls and application or user requirements is not detailed.

Benford et al. (1995) propose a set of issues in the design of user embodiments, from experience in the development of collaborative VEs. Therefore, the issues deal with representing a user to other users, as well as to themselves. The issues include presence (conveying a sense of someone's presence), location, identity, activity, viewpoints and actionpoints, availability, gesture and facial expression, efficiency (in terms of computing resources) and truthfulness of representation. Viewpoints represent areas of space that a person is attending to and actionpoints represent areas of space in which the user is able to carry out manipulations. For example, in the DIVE system, simple embodiments, 'blockies', are sufficient to convey presence, location and orientation. The set of design issues form a useful framework for considering and exploring user embodiment, particularly with respect to collaborative VEs. However, again they fall short of providing concrete guidance and, therefore, further work is required to convert the framework into support for selecting and designing appropriate user representations.

Darken and Sibert (Darken, 1995; Darken and Sibert, 1996) present general principles to aid wayfinding in VEs, based on spatial knowledge theory and environmental design methodology. Wayfinding problems, such as difficulty finding places, relocating places recently visited and understanding the overall spatial structure, can result from poor spatial knowledge or poor spatial cues. Wayfinding tasks require survey (map-like) knowledge, a clearly organised space (to eliminate multiple passes or skipping entire areas) and, for target searches, knowledge of the direction and distance to the target. Darken and Sibert propose the following organisational principles:

- Divide the world into distinct small parts, preserving sense of place.
- Organise the small parts under a simple organisational principle.
- Provide frequent directional cues.

Principles for map design are also proposed, such as always showing the observer position and orienting the map with respect to the observer. Darken and Sibert have evaluated the proposed principles by implementing wayfinding augmentations (grids and maps) in test worlds. Subjects given the augmentations were found to execute more effective searches, because the augmentations could guide and optimise searches, and were better able to structure the space. Therefore, the general principles have been shown to be effective in supporting specific wayfinding tasks.

Rushton and Wann (1993) propose that guidelines are required as to what features of the natural optic flow field, from Gibson's work (e.g. Gibson, 1986), must be recreated in a VE for effective control. Each possible type of user movement produces a characteristic flow pattern, such as an expansion or contraction when the user moves forward or backward. Rushton and Wann give some very general considerations about the flow field in VEs, such as:

- Sparse and untextured VEs will have impoverished flow fields.
- Where lags occur, some components of the field, such as rotational components, could be restricted to alleviate this problem.
- The natural mode of locomotion, with variable velocity control, is preferable to constant velocity motion, because it is built on the user's existing knowledge of momentum control.

However, they are carrying out experimental work to determine more detailed optic flow elements required for different tasks. For example, in Rushton et al. (1997) they report that 2D layout appears to play a more important role than 3D layout for the perception and control of heading. Further work is required to translate these perceptual requirements into specific design guidance.

Therefore, some progress has been made on the design of view controls and user representations. Work has also begun on requirements to support wayfinding and specific perceptual tasks. Although, this research may lead to detailed advice for specific parts of the VE design problem, more general advice is also needed which covers the wider range of interaction tasks in VEs, so that all important and common behaviours can be supported. For example, a VE where viewpoint control, wayfinding and perception are well-supported, but task completion, exploration, event interpretation, object inspection and object interaction are unsupported will not have good overall levels of usability and may discourage or hamper some important interaction tasks.

2.6 Summary: virtual environments

Virtual environments are computer-generated worlds, representing some real-life or artificial structure or place, that the user experiences, explores and manipulates. Virtual environments have been usefully applied to various problems, such as training, marketing and evaluation of designs. Components in a VE include a background space, the user, objects and possible agents. User interactions with VEs include navigation and viewpoint control, and object interactions, such as object selection. Various techniques have been used for designing components and interactions, demonstrating the wide range of possibilities when creating VEs. Major issues, such as realism, have emerged but no general standards exist, and little evaluation of the effectiveness of the different techniques has been carried out.

Rather than a simple list of possible implementation techniques, a deeper understanding of user requirements when interacting with VEs is needed. However, whilst much research has been dedicated to implementation techniques, little is known about how users interact with VEs. General interaction tasks include navigation, wayfinding, object approach and object interaction. Users have limited perceptual awareness and focus when interacting, depending on their current position in the environment. Users need to build up an understanding of their embodiment in the VE and its capabilities, as well as an understanding of the world and its spatial structure. In an ideal interaction with a VE, the user would feel present in and directly engaged with the environment. However, users can suffer from common interaction problems, such as disorientation and perceptual misjudgements. Work has begun on requirements to support perception and wayfinding, and the selection of appropriate viewpoint controls and user representations. However, further, more comprehensive work is required on supporting all interactive behaviour and ensuring overall usability.

2.7 Human-computer interface design and usability

Human-computer interface design (HCI) is the study of all aspects of systems involving people and computers. It is, therefore, related to both human psychology and computer science. Figure 2.2 shows how *usability* is a function of the overall *acceptability* of a system. Within the category of practical acceptability, *usefulness* is the issue of whether the system can be used to achieve some desired goal, and this can be divided into *utility* and *usability*. Utility is a measure of how well the system helps the user fulfil one or more real world tasks. Usability is a measure of how well the user can interact with the system and meet their goal. It centres around how comprehensible, easy to use, efficient and pleasant the system is for the end-user.

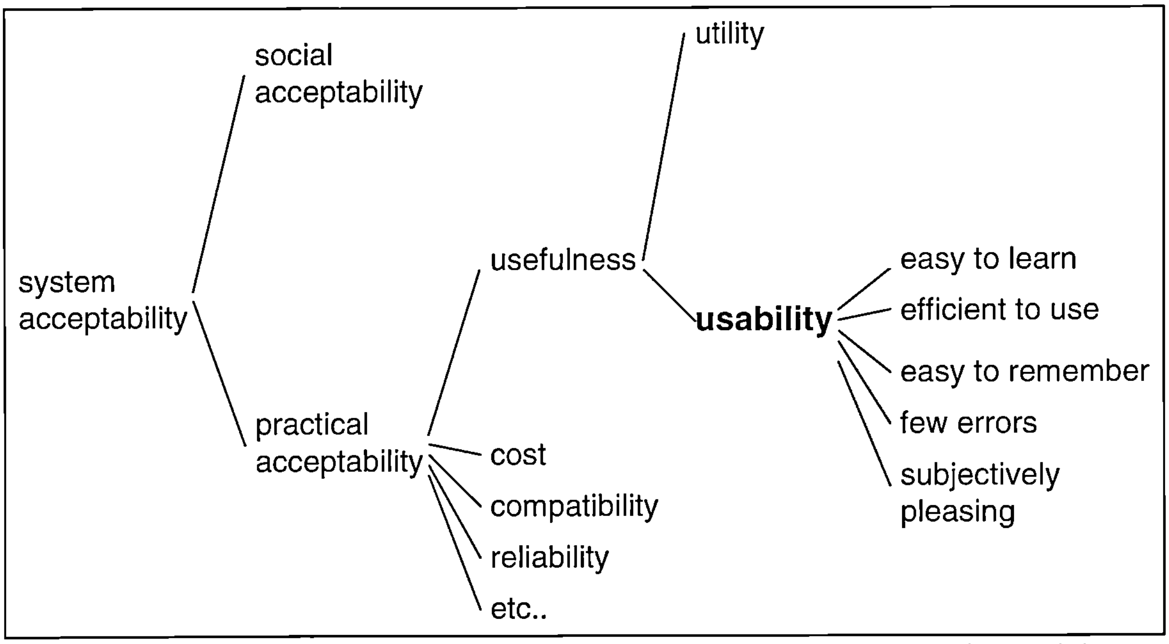


Figure 2.2: The place of usability in the overall acceptability of a system, from Nielsen (1993)

There are general usability requirements, such as learnability, consistency and task conformance (Dix et al., 1993), and specific requirements for different interface styles, which provide more detailed and pertinent guidance for the designer. Therefore, VEs will have some specific requirements for usability depending on their unique characteristics as human-computer interfaces.

2.8 Comparing virtual environments with other human-computer interfaces

The classic early interface style was the command-based interface which promoted a conversational mode of interaction between user and computer (Frohlich, 1993). More recently, direct manipulation (DM) interfaces have promoted interaction based on the user's manipulation of computer based objects, on the grounds that this would place less load on the human cognitive system and is preferred by users (Frohlich, 1993). Direct manipulation is characterised by Shneiderman (1982) by the following properties:

- continuous representation of objects of interest,
- physical actions or labelled button presses instead of complex syntax, and,
- rapid incremental reversible operations whose impact on the object is immediately visible.

The notion of directness is related to the psychological distance between user goals and user actions at the interface, and the psychological engagement of feeling oneself to be controlling the computer directly rather than through some hidden intermediary (Frohlich, 1993).

In chapter one, the differences in the interface structure of DM and VE systems were summarised. Virtual environments are 3D large-scale graphical models representing a coherent structure, whilst DM interfaces are 2D spaces dynamically presenting objects of interest. Direct manipulation interfaces have an established paradigm of interaction (WIMP - Windows, Icons, Mouse, Pop-up menu) but VEs have yet to evolve a dominant interaction paradigm (Bryson, 1995; Colebourne and Rodden, 1995). Wann and Mon-Williams (1996b) suggest that the principles of DM translate easily to VE systems, while Frohlich (1993) discusses VEs as a continuation of the DM trend. Virtual environments reduce psychological distance by representing task domains in a more realistic manner and facilitating more natural, multi-modal (Bryson, 1995) interaction. The computer is removed as an object of perception, allowing the user to interact directly with the generated environment (Hubbold et al., 1993). Sense of presence in VEs goes beyond mere engagement with objects of interest (Frohlich, 1993), encouraging users to become more involved in their task (Pimentel and Teixeira, 1993; Dede et al., 1994).

Object interactions in VEs share some similar features to those in DM systems. Objects are manipulated directly, for example by picking up and moving them, although in VEs the third dimension may also be involved, for example grasping and rotating objects. Furthermore, hybrid systems may be created that involve a VE model and include more symbolic elements (Bolter et al., 1995). For example, the VR-MOG toolkit (Colebourne and Rodden, 1995) includes DM facilities such as buttons and menus, as well as common world components such as walls, lights and rooms. Therefore, VEs continue from and have some similar qualities to DM systems.

Virtual environments also share important features with other interface types. Virtual environments do not continually represent objects of interest. Objects need to be located by navigating and searching a large environment model. In this respect, VEs have similar qualities to hypertext interfaces. Hypertext interfaces involve abstract information spaces which consist of text fragments connected by access paths, such as buttons and links, and a set of standard operations for navigation, such as next page.

Virtual environments are often active, with objects operating independently of the user's actions (Bryson, 1995), for example monitor agents to notify the user of relevant events requiring attention (Yamaashi et al., 1996). Virtual environments also tend to be exploratory in nature, with less structured tasks for achieving goals. Many VEs are rarely if ever, revisited (Darken and Sibert, 1996) so they tend to be used for short periods of time by novice users, because of the nature of tasks associated with VE applications. For example, with marketing applications the user is introduced to a product; with training applications the user practises a skill; and with evaluation applications the user evaluates a design. Such user goals can be fulfilled within the first few interaction sessions and therefore expertise from repeatedly interacting with the VE will often not be built up. In this respect, VEs have similar qualities to walk-up-and-use systems. Walk-up-and-use systems, such as museum information systems, need to have fast learning times, allowing users to be successful from their very first attempt at using them (Neilsen, 1993). Exploratory learning becomes a more important process to support, as opposed to efficient expert use.

Figure 2.3 shows some of the general qualities VEs share with DM, hypertext and walk-up-and-use systems. Previous research in these areas may be useful in defining guidance specifically for VE interface design. However, before existing guidance can be borrowed or new guidance developed, a definition of the human-computer interaction is required (Herndon et al., 1994), as well as, the necessary affordances and tasks for travelling and acting in the VE (Boyd and Darken, 1996). Some general theories of how humans plan and react, such as the work of Suchman (1987), may be applicable to VEs (Höök and Dahlbäck, 1992).

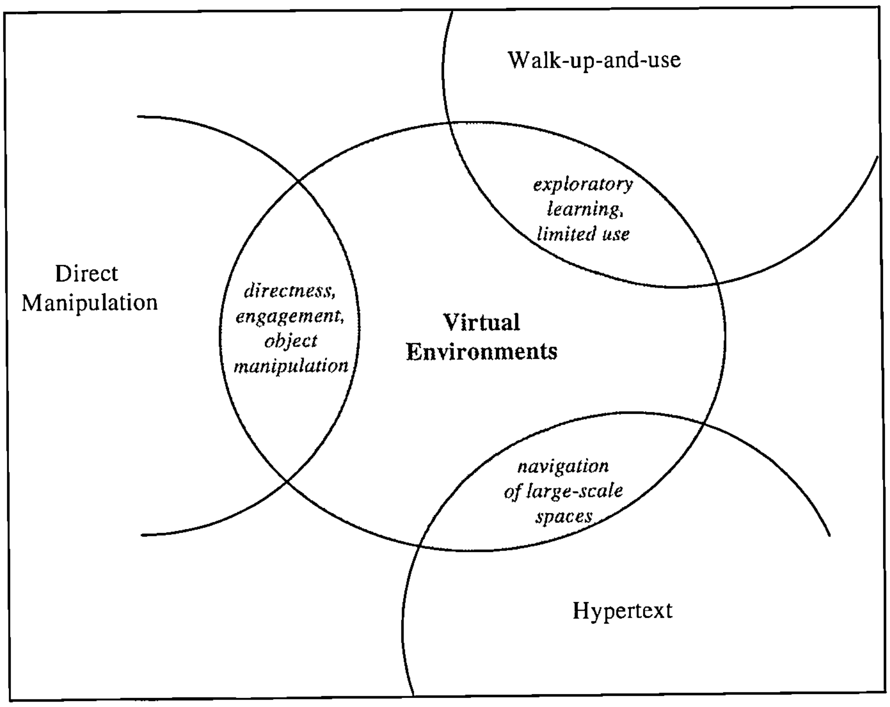


Figure 2.3: General features of virtual environments that are also found in three other interface types.

2.9 Models of interaction

HCI models describe the processes involved in human-computer interaction and help identify the likely root of difficulties (Dix et al., 1993). Modelling frameworks presume a mental architecture and mental procedures, and presume a specific scope and purpose that guides development of the framework (Timmer and Long, 1997). Such frameworks can be inferred from data on user behaviour, such as verbal protocols, or hypothesised and then tested through automated simulation, formal or informal reasoning, as well as experimentation.

Models in HCI include system models, e.g. PIE (Duke et al., 1994), general models of cognition, e.g. The Model Human Processor (Card et al., 1983), models of user knowledge and its use, e.g. Cognitive Complexity Theory (Kieras and Polson, 1985), and models of interaction, e.g. Norman's (1988) cycle of interaction. The PIE model relates sequences of inputs (*programs*) to *effects* through an *interpretation* function. It can be used to describe interactor units that serve as building blocks for specification and development. The Model Human Processor consists of three interacting subsystems; the perceptual, motor and cognitive systems, each with its own memories, processors and principles of operation. The perceptual system senses and stores input, the cognitive system uses knowledge in long-term memory to make decisions about how to respond to the input, and the motor system carries out the response. Cognitive complexity theory represents the user's knowledge of how to use a system, in a procedural notation that permits quantification of the complexity of knowledge required and the cognitive processing load involved in using a system. Components of knowledge involved in operating a device are the user's task representation, device-dependent knowledge and device-independent knowledge; the latter being relevant knowledge not specific to the device in question.

Having a consistent mental model of the device can be important, however for graceful interaction, it is more important that the user can accomplish useful work, interactively with the system in a smooth, elegant and trouble-free way (Frohlich, 1993). Interaction models aim to develop an understanding of what is happening during interaction. For example, Norman's model of interaction (Norman, 1988) shows interaction occurring in cycles of user and system actions, creating a dialogue between the user and computer, see figure 2.4. The seven stages involved are establishing the goal, forming the intention, specifying the action sequence, executing the action, perceiving the system state, interpreting the system state and evaluating the system state with respect to the goals and intentions. Two gulfs exist between the user and computer which have to be bridged by the interface. The *gulf of execution* is when the user has to decide what to do next. There may be a difference between the intentions of the user and allowable actions in the system. The *gulf of evaluation* refers to what happens after the user action, and the effort required to interpret system states. Specific questions guide a user's actions when they are using a novel object, for example:

- Which parts of the object move, which are fixed?
- What kind of movement is possible, e.g.. pushing, rotating?
- Which parts signify the state of the object and which are non-functional?
- Where should you look to detect any changes in the object?

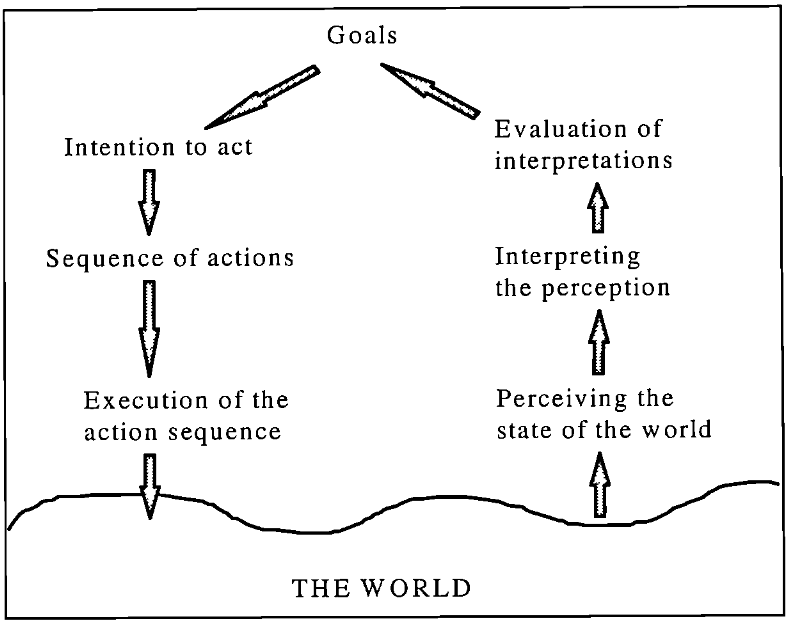


Figure 2.4: Norman's seven-stage cycle of interaction

Norman's model generalises about interaction behaviour, beyond specific interaction styles. Such general interaction models, may be useful in informing the development of more specific models for VE interaction. Indeed, Springett's Interaction Level Models (Springett, 1996) are based on the Norman model of interaction, and describe interactive behaviour with direct manipulation interfaces. Springett describes interaction at different levels for skill-based, rule-based and knowledge-based processing, using Rasmussen's 3 levels of user action (Rasmussen, 1983), for example see figure 2.5. A remedial cycle accounts for the effect of errors on the processing levels, for example problem solving behaviour in handling error situations is included. In Springett's models, the display plays an important role in interaction, for example, in knowledge-based interaction, the display is scanned and features selected for action. For VE interaction, modelling the short-cut processes of skilled users is less important, but display-prompted behaviour is likely to be an important part of the interaction. Norman does not explicitly describe display-prompted behaviour; however, there has been a move away from top-down, plan-based models of interaction in HCI to more situated models.

Suchman (1987) proposed that plans were not necessarily the bases for action. Much of action is situated in that people use their perceived environment to achieve intelligent action. We encounter a succession of situations to which we respond, identifying their features, and matching our actions to them. For effective interaction, the situation must be recognised as an instance of a class of typical situations, and the behaviour of the user must be recognised by the system as an instance of a class of appropriate actions or responses. In situated action, knowledge is seen to be distributed between the user and the world (Bibby, 1992; O'Malley and Draper, 1992). Payne (1991) also argued for the reactive nature of human action and the importance of display-based action in HCI. People do not always work out complete plans in advance and actions may not be remembered if instead they can be recognised on the user interface. In a Display-Based Task Action Grammar (D-TAG; Payne, 1991), display items are scanned and actions can be specified as 'the item on the display which best matches the current relevant task features'. Display-based models, such as D-TAG, may be useful in informing the development of models for VE interaction. For example, scanning for task-relevant, or otherwise interesting, features is likely to be common place in VE interaction.

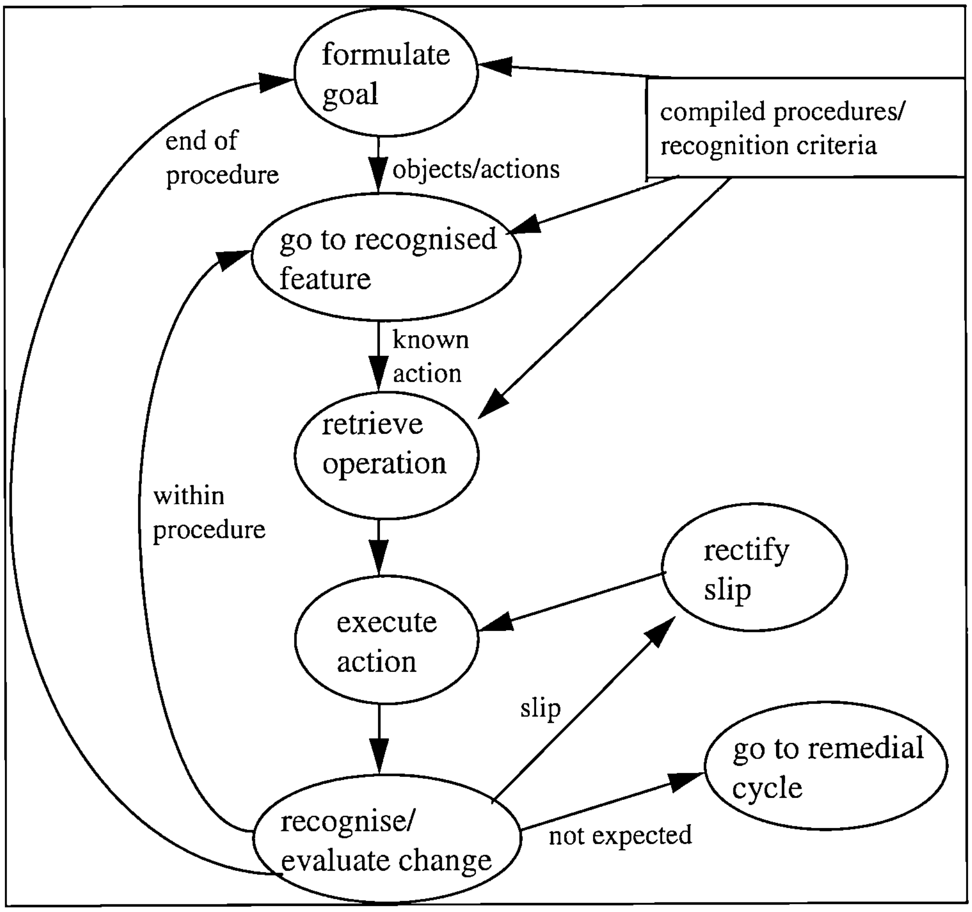


Figure 2.5: Springett's model of skill-based action

As well as generally being display-based, VE interaction is also likely to include more specific behaviour in navigating, exploring and reacting to environment events and agents. Rather than simply planned actions, there are likely to be reactive, exploratory and opportunistic actions, where opportunities for action are stumbled upon in the environment (Hayes-Roth and Hayes-Roth, 1979). Navigation and reactive behaviour have only been previously modelled to a limited extent, for example behaviour in responding to action feedback. However, exploratory behaviour and exploratory learning have been recently studied with respect to walk-up-and-use systems. Users can learn to use systems without manuals, exploiting the knowledge they already have combined with their ability to learn by doing and exploration (de Mul and van Oostendorp, 1995). However, exploratory learning can be chaotic with users failing to use previous successful methods and developing 'false' knowledge, such as 'superstitious' methods (Payne et al., 1992). Therefore, exploratory behaviour needs to be supported to avoid frustration (Howes and Payne, 1990), and speed learning (De Mul and van Oostendorp, 1995).

The CE+ model of exploratory learning (Lewis et al., 1990) aims to inform the design of walk-up-and-use systems. The model contains a problem-solving component, a learning component and an execution component. The problem-solving component predicts a user will use means-ends analysis and choose among alternative actions based on the similarity between the user's expectation of the consequences of an action and the user's current goal. After executing an action, the user evaluates the response and decides whether progress is being made toward the goal. Learning occurs when the evaluation process leads to a positive decision and the action is stored in memory in the form of a rule. CE+ is a learning model and the understanding of exploratory interaction behaviour provided by it is limited. However, it may be useful in understanding basic aspects of exploratory behaviour, for example, the selection of actions based on similarities to user goals.

In general, although there are many previous HCI models, none adequately describe all features of VE interaction. Generalised models can inform basic requirements of a VE interaction model. Display-based and exploratory models can inform additional requirements for describing situated action. However, there is a lack of HCI models describing reactive behaviour to environment events and agents, and behaviour in navigating through large-scale spaces. Therefore, specialised models of interaction are required for VEs, which include all major relevant patterns of behaviour, so that a comprehensive set of requirements for supporting VE interaction can be defined. Where aspects of VE interaction are shared by other HCI systems, existing design guidance may be useful in informing interaction requirements.

2.10 Human-computer interface design guidance

General user interface design guidance exists which applies more to conventional interface styles. For example, usability heuristics given by Nielsen (1993) are *simple and natural dialogue, speak the user's language, minimise the user's memory load,* *consistency, feedback, clearly marked exits, shortcuts, good error messages, prevent errors* and *help and documentation.* Text, shortcuts, errors messages and documentation help are found less in VEs than in conventional command-based and direct manipulation interfaces. Alternatively, Norman (1988) gives principles for general action support, from his work on modelling interaction, which can apply to VEs. For example making important object parts visible; making clear how actions are to be carried out; and providing feedback, clear affordances, natural mappings and clear constraints for object manipulation.

More specific guidance exists, such as the guidance for the design of direct manipulation dialogues. Springett (1996) defines general support roles for the system which are primarily used in evaluating DM interfaces, such as the locator, feature identifier and execution support roles. ISO 9241, part 16 (ISO, 1996) provides more detailed design guidance. For example, it recommends objects that are directly manipulable to be clearly identified from other elements, and the kind of manipulations that can be applied to be clearly indicated. It also recommends guidance for the re-arrangement of the interface, such as moving, removing and re-sizing of elements (e.g. windows etc.).

'Design for successful guessing' is a set of design principles derived from the CE+ theory (Lewis et al., 1990) for walk-up-and-use systems. For example, make the repertory of available actions salient, provide an obvious way to undo actions, offer few alternatives, so the user is less likely to choose an incorrect option, and require as few choices as possible. De Mul and van Oostendorp (1995) propose similar guidelines for supporting explorative behaviour, such as making possible operations distinguishable, making consequences of actions clear and the effects of executed actions visible.

For hypertext interfaces, Nielsen (1990) suggests good design features, such as facilities to utilise the interaction history (e.g. backtracking), landmark nodes (e.g. home), timestamps to record when a node was last visited, levels of overview diagrams with checkmarks to indicate visited nodes, and a recommended reading order of links.

Some of the existing design guidance can be applied to VEs, but some is much less applicable and there are aspects of VE interaction that are not covered by existing HCI guidance. For example, guidance on making direct manipulations clear can also apply to VEs, but guidance on re-arranging interface elements is less applicable because the interface layout is generally static in VEs. From walk-up-and-use systems, a clear repertory of available actions is also important in VEs, but limited choices may be a little restrictive to the open nature of VEs and hamper active exploration. Hypertext guidance such as landmarks and overviews is applicable to VEs, but features like backtracking apply less well. Furthermore, additional guidance is needed on novel VE aspects, such as user representation, viewpoint orientation and reacting to system-led behaviours.

2.11 Summary: virtual environments as human-computer interfaces

Virtual environments continue the direct manipulation trend, extending the concepts of directness and engagement. Navigation through the environment model is an important part of interaction, as it is with hypertext interfaces. Virtual environments are exploratory in nature and many are not continually revisited, similar to walk-up-and-use systems. A definition of the human-computer interaction for VEs is required, so interface design guidelines can be developed. Conventional interface design guidance is only partially applicable to VEs and does not cover the range of issues that arise in VE interaction. Existing interaction models include some features that are likely to be important in VE interaction, such as the general cycle of action and situated actions, but all important behaviours are not covered.

The following chapters describe the thesis research into designing virtual environments for usability. In the next chapter, introductory studies into the design and use of virtual environments are reported.

Chapter 3

Current Design Practice and Usability Problems with Virtual Environments

This chapter describes studies of the design and use of virtual environments, and presents a case for the need for interface design guidance.

Chapter 3

Current Design Practice and Usability Problems with Virtual Environments

Two studies were carried out to investigate hypothesis 1:

H1 There is a need for interface design guidance specifically for VEs.

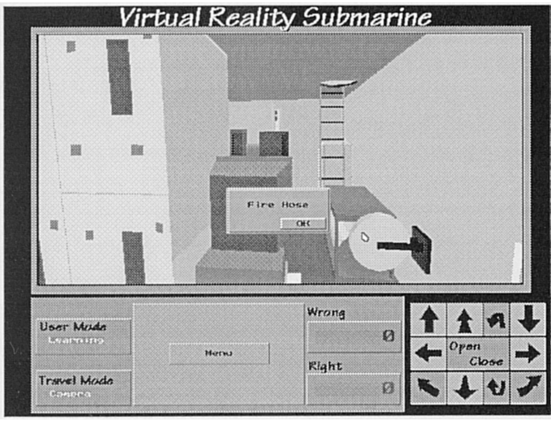
The first was a study of designers, which was carried out because little knowledge existed about how VEs are designed. Previous work had mainly cited isolated experiences in design, such as issues in maintaining run-time performance (e.g. Pimentel and Teixeira, 1993; Hubbold et al., 1993). Therefore, an introductory study of VE design was undertaken to gain more comprehensive information about VE design. The study investigated all stages in development, focusing on designers' perceptions of the development process. Many studies of the design of conventional systems have been carried out, for example Guindon et al. (1987), Bellotti (1988), Rosson et al. (1988), Curtis et al. (1989), Guindon (1990), and Myers and Rosson (1992). These studies have provided useful knowledge for research on methods and techniques for conventional systems, such as knowledge about patterns of design practice, problems faced in design and heuristics and techniques employed.

Additionally, little knowledge existed about the use of VEs and predominant usability problems. Previous user studies tended to evaluate the applicability of VEs, compare different devices or the effectiveness of various depth cues (e.g. Dede et al., 1994; Mercurio and Erickson, 1990; Wanger et al., 1992; Psotka et al., 1993), rather than focus on the overall usability of VEs. Therefore, a second study was carried out which involved investigating a VE in use, to gain knowledge about the usability of current VEs and common usability problems.

3.1 Usability problems

3.1.1 Study method

The study involved an evaluation of the usability of the Royal Navy's Virtual Submarine application. The application was a prototype desktop VE for training submariners in nuclear-powered submarines. The emphasis was on familiarisation with compartment layout and equipment location. The virtual submarine was made up of two rooms - a switchboard room and, above this, a manoeuvring room. A ladder connected the two rooms. An on-screen control panel and 2D mouse was used for interaction. The control panel consisted of buttons for navigation in various directions and a button to open/close a hatch at the top of the ladder. Figure 3.1 shows the switchboard room and the connecting ladder and figure 3.2 shows the manoeuvring room.



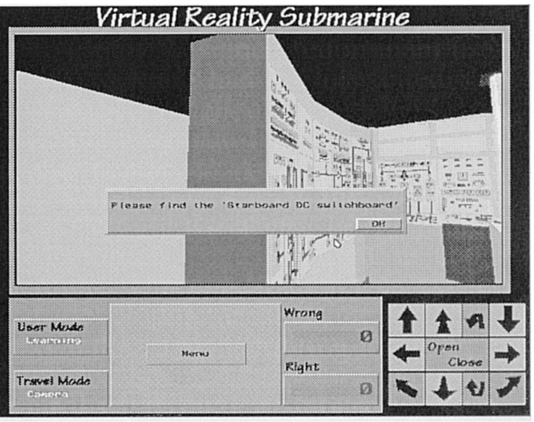


Figure 3.1(left): The switchboard room, including the connecting ladder. The identification of the "fire hose" is shown.

Figure 3.2(right): The manoeuvring room. An instruction to locate the "Starboard DC switchboard" is shown.

Subjects were trainee submariners with the Royal Navy and were involved in a training programme which consisted of traditional classroom-style training, an interaction session with the virtual submarine and a visit to a physical simulator. Most subjects had visited the physical version of the virtual submarine. The interaction sessions involved two parts - exploration and testing. A short time was spent in exploration mode, during which users navigated the submarine and could identify objects by clicking on them, causing a dialogue box to appear naming the object. In testing mode users were required to locate 20 pieces of equipment which they had to click on when found. Figure 3.1 shows the identification of the "fire hose" in exploration mode. Figure 3.2 shows an instruction to locate the "Starboard DC switchboard" in testing mode.

Two groups of four users participated in the study. Each user in the group spent about 10 minutes interacting with the environment while the other users in the group observed and made comments. Users were directly involved in exploration or testing, but not necessarily both. Data on usability problems and user attitudes was gathered from observation of interaction sessions, de-briefing interviews and retrospective questionnaires.

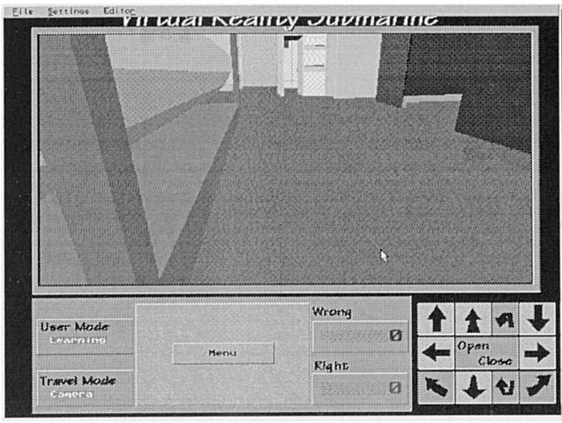
3.1.2 Results

3.1.2.1 Interaction problems

There were clear overlaps in the problems faced by the different users. Table 3.1 summarises the main interaction problems found, which could be categorised as either navigation or object interaction problems. A more thorough analysis of the problems can be found in Malinski (1996).

	Problem
Navigation	Getting the right viewing height and orientation
	Difficulties using the ladder joining the two rooms
	Facing a uniform screen
	Unknown object obstructing movement
	Understanding what the control panel buttons represented
	Movement too fast
Object interaction	Clicking the activating part of an object
	Clicking non interactive objects

Table 3.1: The main interaction problems



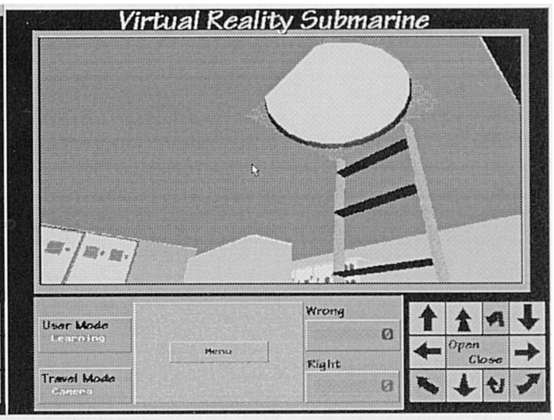


Figure 3.3 (left): Users had problems getting and maintaining a suitable viewing angle, for example, often they navigated at too low a height.

Figure 3.4 (right): Users had problems navigating through tight areas, such as the hatch at the top of the ladder.

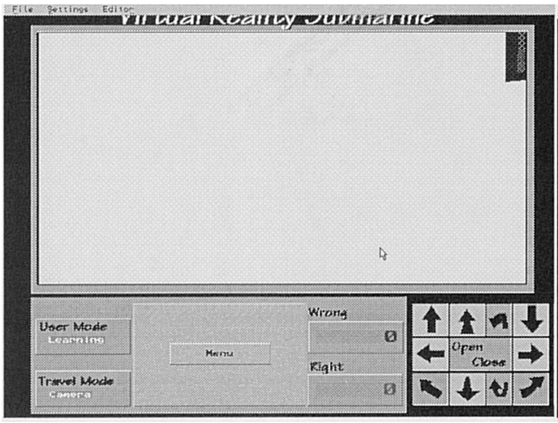
Navigation problems:

- Getting the right viewing height and orientation, see figure 3.3. Users often observed the environment from quite a low position, near the floor of the rooms, which did not provide them with a suitable view. Sometimes the viewing position was too high with users moving above the submarine equipment.
- Difficulties using the ladder joining the two rooms, see figure 3.4. Users had problems getting up and down the ladder. Orientation problems were encountered after going up the ladder. Some users required several attempts to get down the ladder because of difficulties positioning themselves accurately over the top of the hatch.
- Facing a uniform screen, see figure 3.5. Sometimes users walked into walls or got too close to objects. This resulted in the screen being filled with one colour, leaving users unsure of their exact location in the environment.
- Unknown object obstructing movement. Sometimes users found they were unable to move and were not sure what object was in their path.
- Understanding what the control panel buttons represented. Users had difficulties using the navigation control panel because they often failed to guess correctly or remember what movements the arrows represented. For example, users confused the two up arrows.

- Movement too fast. Some users found the speed of navigation too fast.

Object interaction problems:

- Clicking the activating part of an object, see figure 3.6. Users had difficulties interacting with objects because they failed to position the mouse cursor in the right place. Users may not have known where to click on an object or could have missed objects when not close enough to them.
- Clicking non-interactive objects, see figure 3.6. Sometimes users clicked objects expecting information which was not available.



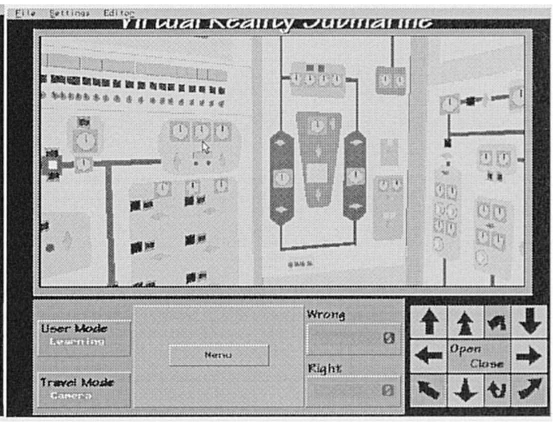


Figure 3.5 (left): Users had problems losing their whereabouts after getting too close to objects (e.g. nose against the wall).

Figure 3.6 (right): Users had problems recognising interactive hot-spots in the environment, for example which buttons on the switchboard could be interacted with.

<u>3.1.2.2 Subjects' comments on usability</u>

Subjects commented on the usability of the virtual submarine during de-brief interviews and in retrospective questionnaires. The main points made were:

- Users felt that a more complete and improved version of the prototype, with the interaction problems tackled, would be helpful for training. The system would be a valuable training aid alongside classroom training, but could not replace a visit to a real submarine.
- Users found the environment was visually inaccurate. The equipment was not shown in realistic enough colours and dimensions. Equipment was not distinctive

enough, for example it could not be recognised from all angles, because insufficient detail was included. There was a lot of equipment, walls and wiring missing and this resulted in there appearing to be too much empty space in the model submarine.

- Users found the environment both difficult to interact with and unpredictable. Controlling movement was difficult and time-consuming. Users felt it would be better to use a joystick or roller ball to control movement. Unrealistic movements were possible, such as being able to walk across the top of the switchboard, and some users found this confusing. It was easy to get lost in the environment. A plan of compartments, which included the current position, was felt to be necessary.
- Users felt some necessary features were not included. A facility to get closer to objects and study any particular piece of equipment in more detail was felt to be important. Users wanted a more realistic virtual submarine, which included noises, such as sounds from the ventilation system, and running machinery. Additionally, help facilities and better graphics (bigger screens, better image quality and 3D viewing) were suggested.

3.1.3 Discussion

The user study provides an indication of the types of usability problems with VEs and users' perceptions about important considerations. Accuracy and realism are deemed to be important qualities. Copies of real world objects need to be represented accurately to facilitate recognition. Environments should be designed to be realistic by including appropriate sounds and dynamic behaviour, and not including inappropriate and unrealistic features. Reliable and effective interaction is essential, especially for navigation. Interaction needs to be quick to learn, predictable and easy to control, and without frequent obstacles or problems. Help facilities are thought to be useful, such as plans of the environment.

In the study, trainee submariners experienced major interaction problems. Some of the problems, such as disorientation and unrealistic representations and interactions, have been found in other evaluation studies of VEs (McGovern, 1993; Miller, 1994). Few motion sickness problems occurred, which have been found in other studies (Rolland et al., 1995; Miller, 1994; McGovern, 1993; Oman, 1993), probably because no head-mounted display was used in this application. Furthermore, the study identified common object interaction problems and further navigation problems, such as realising that collisions have occurred and executing precise navigation. More recent work supports the common problems found, such as the difficulties that can occur with collisions (Witmer et al., 1996; COVEN, 1997b), with knowing what objects can be acted on and how, and with precisely manipulating objects (COVEN, 1997b). Finally, specific problems were found with the navigation technique used in the virtual submarine, i.e. the navigation control-panel. Similar problems in understanding how to use screen controls and with users expecting a more direct form of navigation were found with a virtual joystick used in VEs on the internet (Mohageg et al., 1996).

Comparing the common VE interaction problems with typical problems found with conventional user interfaces, shows some clear areas of difference. For example, Springett (1996) reports some of the common problems found with the MacDraw direct manipulation interface:

- misleading action cues (e.g. menus not making clear the resulting direction of arrow lines)
- expectancy of an action that is not possible (e.g. expecting arrows to be available on curved lines)
- hidden functionality (e.g. the implicit setting of defaults on line styles)
- missing or ambiguous feedback (e.g. inadequate feedback on object selection)

These problems seem to be related to understanding the meaning and functionality of objects, actions and feedback. Such problems may be less predominant in VEs because objects and actions are copied from the real world at a more direct and semantic level, as opposed to a symbolic/metaphorical level in conventional interfaces (e.g. through the use of iconic representations). However, VEs also suffer additional problems, such as maintaining a suitable viewpoint, difficulties in navigation, problems with the basic perception of objects and movement, and specific technical issues (e.g. display update lag). Basic interaction issues appear to be more prevalent, such as not knowing which objects are active, because VEs do not have an established standard that defines active objects (like the menus and icons in DM interfaces). The added spatial dimension in VEs may place more demand on manipulation precision. Locating objects is also likely to be an issue, because much of the interface will be hidden at any one time. In summary, usability problems in VEs appear to be at a more basic level of perception and orientation within the interface, rather than issues in understanding the semantics of the interface. This may indicate that current VEs are being designed to a poorer level of usability, than their conventional counterparts. The next study investigated the design of VEs in practice.

3.2 Design practice

3.2.1 Study method

A range of people from the small population of VE designers took part in a designer study. The subjects were ten designers (nine male, one female) from three organisations in the UK - one VE toolkit vendor/consultancy (5 designers), one VE consultancy (3 designers) and one military organisation (2 designers). The designers had built environments for a range of application areas and a few had built immersive environments as well as the more predominant desktop type. Designers from the VE toolkit vendor/consultancy had built VEs mainly for marketing and product design/evaluation purposes, for example architectural walkthroughs of planned buildings. Designers with the VE consultancy developed VEs for product design/evaluation, training, entertainment and marketing applications, such as building a virtual water processing plant for planning purposes and for handling inquiries. Finally, designers with the military organisation had built only training applications, for example VEs for aircraft maintenance training. Within an organisation, individual designers tended to work on different projects since VE projects tend to be small in size, often completed in weeks, rather than months or years. All but the least experienced had worked on several projects.

Structured interviews were used for knowledge elicitation as they provide a good source of evidence about the nature of a problem domain and the relevant goals and constraints (Evans, 1988), while being economical in terms of time or expense. Previous designer studies, for example Rosson et al. (1988), have used interview techniques for reporting experts' procedural and declarative knowledge. Evans (1988) warns about reliance on interview techniques for accessing procedural knowledge, such as cognitive processes underlying design, because subjects find this difficult to express through retrospective reports. Detailed procedural knowledge was not sought for this study, rather the focus was on gaining a general outlook on VE design. However, techniques such as scenarios or reporting through the use of examples were used where it was felt that the interviewee would have difficulty reporting information retrospectively.

The interviews were structured into the following sections:

- *background* (general experience, VE design experience, work role) and technical working environment (software and hardware used)
- *approach to design* (information sources and established methods used, general design approach, approach to design of three important aspects - interaction, user representation and multi-modality)
- *design problems* - (problems and issues faced by designers)
- *design guidelines* - (guidelines and rules of thumb representing existing design expertise).

One interviewer undertook all the interviews, which were audio-recorded and later transcribed. The interviews took place in the normal work place of the designer, often involving the inspection of past work (i.e. built cnvironments), and typically lasted about forty minutes.

3.2.2 Results

The designers provided a range of qualitative information about the nature of VE design.

3.2.2.1 Background and design environment

Table 3.2 details the varied background of the designers. The majority of designers had studied computing or had some programming experience. However, few designers had HCI experience. Other background areas mentioned more than once were electronics, computer graphics and architecture. Overall, the designers had little experience building VEs. Experience ranged from 1 month to 3 years 8 months, with the median at 8 and a half months.

Designer **Experience**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
VE design (months)	3	3	14	7	44	2	1	42	20	10
computing/ programming	✓		✓	✓	✓	✓	✓	✓	✓	✓
HCI	✓			✓					✓	
electronics				✓	✓	✓				
computer graphics					✓		✓	✓		
architecture/CAD		✓					✓			
photography								✓		
psychology									✓	

Table 3.2: The background of the designers, including experience building VEs and experience in other areas.

Table 3.3 details the software and hardware environments with which the designers worked. Designers relied heavily on VE toolkits and graphics design packages. They also used computer-aided design (CAD) packages to build components and transfer previously built CAD models to a VE. Most designers used a 3D mouse for navigation, but little other VE hardware was used. Two designers had used head-mounted displays, for example to test interaction in immersive environments.

	Environment component	No. of designers
Programming packages	Superscape VRT	9
	Division dvise	2
	MEDIT	2
	programming language	3
Other packages	animation/ graphics package	5
	CAD package	4
	sound package	2
	spread-sheet package	1
VE hardware	space mouse/ space ball	9
	head-mounted display	2

Table 3.3: The reported general use of hardware and software items by the designers.

3.2.2.2 Design Approach

The designers followed common patterns of design practice. Often they observed that clients were not aware of the unique aspects of VEs and had to work with clients to define objectives that would benefit from the use of VE technology. All designers tended to build copies of real world models and gathered suitable reference material before designing. The sources of information included plans and drawings of buildings, photographs of parts of objects for use in texture scanning and shape copying, and survey maps when building external scenes. A few designers devised a graphical structure for the environment, which would be suitable for any planned interactions and would provide good run-time performance. Some of the larger environments were designed collectively. In one example, different designers worked on the external and internal representations of a building.

Designers tended to build the environment using either a mainly top-down or bottom-up approach. Six designers started with a basic structure and gradually added detail. For example, with a room model the floor and walls were built first and then doors, windows and furniture added later. The other four designers built objects one by one and then fitted them together in the VE. All designers enhanced the VE after objects had been built and positioned, with further detail such as texture, lighting, and object interactions. The VE was optimised and checked to make sure all graphic elements would be drawn in the right order of depth.

The designers built and tested iteratively and, as most used visual programming toolkits, changes could be easily and quickly assessed. Designers tended to both test the VEs themselves and use colleagues as testers. Sometimes a short time was spent building demonstration applications which were shown to sponsors, but rarely tested with users.

Table 3.4 shows established techniques and development approaches used by the designers. Overall, design was informal; the reasons given were not having enough time, because of tight deadlines, and the lack of good guidelines for building VEs. Iterative or prototyping development approach was reported by four designers. Two designers, with architectural experience used architectural guidelines, such as working from a basic shell model.

Designer **Established Technique**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
iterative development/ prototyping	✓		✓			✓			✓	
architectural/CAD guidelines		✓					✓			
object-oriented approach	✓									
HCI techniques (e.g. storyboarding)									✓	

Table 3.4: The use of established techniques/approaches for each designer.

Whereas all designers had created the basic features of VEs such as objects and lighting, table 3.5 shows that few designers included features for object manipulation, realistic user representation and multi-modality. Four designers had built only simple walkthrough VEs, where the only interaction type available was navigation. The other six designers built more complex interaction with objects, for example opening of doors and steering a fire engine. Seven designers represented users with cursor arrows only. The other three designers used body figures, right hand or both hands, or hands and feet to indicate user presence. The designers did not use tactile interaction and only three commented that they had used sound, such as ringing phones, fire bells and opening doors.

Designer / **Advanced area**	1	2	3	4	5	6	7	8	9	10
object manipulation	✓				✓	✓		✓	✓	✓
realistic user representation	✓				✓				✓	
use of sound						✓		✓	✓	

Table 3.5: The inclusion of 3 important VE features for each designer.

Those designers that built more complex features did not have a clear or consistent approach. The designers that built interactive objects copied real world object cues and behaviour, but few strategies were used to highlight available actions. Designers tended to let the users explore and find out what actions were available, although one designer used feedback messages for objects with no associated behaviour. Designers limited the input for initiating actions to single button clicks, and therefore complex cues or metaphors were not required to communicate information to the user about how to carry out actions. One designer noticed the problem of users not accurately clicking on cues to activate actions and, therefore, programmed large invisible areas around cues to make interaction easier.

Where more human-like user representations were implemented, designers used very different representation styles. For example, one designer used hands that were always visible to the user, while another used hands and feet that were only visible if the user looked down, thus providing an orientation cue. Few strategies were used to implement sound. In one exception, a designer used walking sounds to give feedback where the VE was large and homogeneous, because the user may have difficulty judging distance and progress in navigation.

3.2.2.3 Problems cited

Individual problems cited by the designers were categorised into nine areas. Table 3.6 lists problem areas faced by designers in three categories. Individual designers cited problems for a range of one to four areas.

	Problem Area	Number of designers
General problems	**P1.** Client perception of VEs	4
	P2. Getting accurate specification	3
	P3. Obtaining design information and guidance	1
Design problems	**P4.** Balancing detail with performance	6
	P5. Distinguishing important features of environment	3
	P6. Appropriate use of lighting	1
	P7. Fitting together model parts	1
Environment problems	**P8.** VE toolkit problems	6
	P9. Hardware problems	2

Table 3.6: Number of designers citing individual problems for each of the 9 problem areas.

General problems:

P1. Problems with clients' perception of VEs. Clients compared VEs with non real-time graphical rendering and animation software and there was a lack of understanding of the unique features of VEs. For example, clients were not convinced of the worth of sound, and expectations of graphics quality were too high.

P2. Difficulty getting a complete and accurate specification. Clients often requested changes and time was wasted initially building an environment the client did not want.

P3. Difficulty getting information about complex areas of VE design. Few people had tackled complex problems and there was no established way of sharing knowledge with the rest of the design community.

Design problems:

P4. Problems balancing level of detail with performance. A common problem was over-detailing which impaired run-time performance and resulted in too much time being spent on building graphical detail.

P5. Difficulty assessing the importance of features in an environment from the point of view of the user. Designers found problems with judging the perceptibility of visual features; often creating over-complex environments because they assumed users would notice every detail.

P6. Problems in suitably positioning lighting and judging its appropriate level.

P7. Mismatch problems when fitting together a VE built by different designers.

Environment problems:

P8. Problems with VE toolkits. For example, difficulties building complex 3D shapes and object behaviour, time consuming sorting of shapes to ensure they are drawn in the right order, transferring models to/from CAD packages, and difficulties in designing collisions between objects.

P9. Hardware problems. For example, low image quality, and difficulties installing and maintaining specialist VE hardware.

3.2.2.4 Guidelines

Individual guidelines offered by the designers were categorised into twelve areas. A summary of the areas is given in table 3.7. Individual designers cited guidelines for a range of one to six areas.

	Guideline Area	Number of Designers
High level guidelines	**G1.** Consider VE goals	4
	G2. Validate design early	2
	G3. Check client's understanding of VEs	2
General design guidelines	**G4.** Control level of detail	6
	G5. Get basic structure, detail gradually	5
	G6. Keep model simple but realistic	4
	G7. Reuse components	3
	G8. Use textures	3
	G9. Get good object structure	2
	G10. Scale model correctly	2
	G11. Get reference material	2
	G12. Use sound	1

Table 3.7: Number of designers citing individual guidelines for each of the 12 areas.

High level guidelines:

G1. Remember the goals of the VE and avoid getting side-tracked. The goals of the environment should be considered during the design process, for example unnecessary features should not be added with respect to the aim of the VE. Also the goals should justify the use of VEs rather than other technologies.

G2. Validate design at an early stage. The designer should ensure implementation is possible before large-scale development begins. For example, making sure complex parts of an environment can be constructed.

G3. Ensure client has satisfactory understanding of VEs. The client should be aware of the difference between VEs and other technologies, or should be educated about it.

General design guidelines:

G4. Control level of detail. The level of detail in the model should be managed to achieve better run-time performance. For example, spreading detail around, detailing the most important parts of the model, and using an animated series of textures to represent motion instead of using moving shapes. A zoning technique may be used where all parts of a model are not shown until the user is sufficiently near to them; for example, objects in a room are only drawn if the user is in the room, otherwise only the doorway is drawn. Another technique is to use simple shapes to replace detailed objects when viewed at a distance.

G5. Design outline first, then add detail. A basic structure for the model should be designed, then detail added gradually. One useful technique involves placing very simple shapes representing the main objects in required positions, then later replacing these shapes with detailed designs. This approach allows designers to check the outline of a VE before detailing it.

G6. Keep model simple, but maintain realism. The user should perceive the VE as looking realistic and natural, but this can be achieved with simpler models. The designer should judge the accuracy needed to ensure that a VE will have adequate realism. For example, if the doors in a building open automatically as the user approaches, a 2D texture map of a door handle is sufficient rather than a 3D shape.

G7. Reuse components. Objects, textures, shapes, sounds and other model components should be reused to save design time. As little as possible should be developed from scratch and a library of well designed components should be set up.

G8. Use textures to enhance realism. Textures can be specifically added to enhance the appearance of a VE.

G9. Plan object structure. Devise a good object structure for the model as this will aid run-time performance and simplify design.

G10. Scale model suitably. A standard measurement scale should be used for environment components, particularly when reusing objects for size consistency.

G11. Gather reference material. Get as much domain-specific reference material as possible for use in model building.

G12. Add sounds. Sound in a VE can be important, particularly in differentiating a VE from CAD models and animations. (This was noted by one designer, although sound was used infrequently by the designers in general.)

Figure 3.7 demonstrates the use of three general design guidelines.

- Design outline first, then add detail (G5) - the left picture shows an outline VE and the right picture shows the addition of detail.
- Keep the model simple, but maintain realism (G6) - simple but recognisable representations are used for the bed, table, clock and wardrobe.
- Use textures to enhance realism (G8) - wood texture on the floor has been used to enhance the environment.

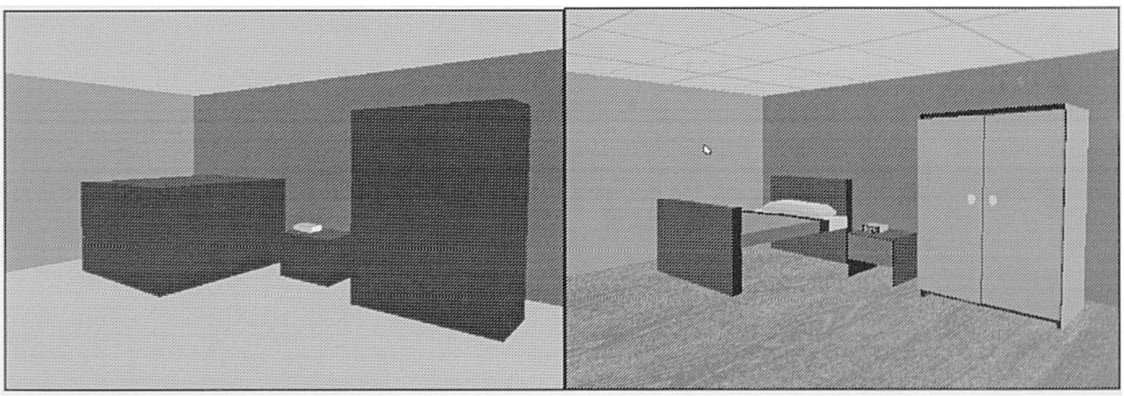


Figure 3.7: Use of design guidelines for a bedroom environment. The left picture shows an outline of the environment, with cubes for the main objects, and the right picture shows the environment after these cubes have been replaced with simple object representations and further detail, such as texturing, added.

3.2.3 Discussion

The study provides new and important knowledge about the practice of VE design. Although individual variations existed, five common, basic processes can be abstracted from current approaches to design (see section 3.2.2.2). These processes were carried out iteratively:

a. requirements specification;

b. gathering of reference material from real world models;

c. structuring of the graphical model and, sometimes, dividing it between designers;

d. building objects and positioning them in the VE;

e. enhancing the environment with texture, lighting, sound and interaction, and optimising the environment.

In more recent work, Bryson (1995) notes the importance of a highly flexible, iterative development process for VEs, while Green and Halliday (1996) report that design activity may be divided into two or three phases: individual object design, scene composition, and narrative design. Narrative design deals with transitions between scenes, for example sequences of scenes that the user explores. Surprisingly, a narrative design activity was not found in this study, although Green and Halliday do note that this is optional. The activities of individual object design and scene composition are present in stage *d* above. Therefore, the design activities identified in this study provide a wider view of VE design.

In a study of the design of general systems, Bellotti (1988) found categories of design activity to be *requirements specification*, *conceptual specification*, *generation of a working prototype*, *testing* and *finalisation*. Current VE design practice differs in that:

- Designers appear to see the VE development problem as primarily involving modelling of the graphics, for example the structuring, positioning and enhancement of graphic elements.
- More emphasis is placed on reference material, since the environments often closely model real world phenomena.
- Testing is informal. In this study, testing could not be identified as a coherent design stage and user testing was very rare. Although Bellotti (1988) also found

testing with users to be rare, in other studies, for example Rosson et al. (1988), some form of user testing was found.

The study results show rclationships between the problems encountered and the guidelines employed by VE designers and, from these, the issues a designer considers can be categorised into three main areas:

1. The most predominant issue is the complex balancing act between three aspects of a VE: performance, graphical detail and realism (see table 3.6, problems P4 and P5). On the one hand, a high level of detail can provide a more realistic environment. However, greater detail can also result in low run-time performance which introduces delays in display update, thereby reducing realism. Moreover, perceptual realism may not necessarily be improved by better detail; there will be a point at which the user will not notice or be affected by increased detail. To tackle this issue, designers used low-level detail management techniques, tried to keep the model as simple as possible to maintain realism and optimally structured the model (table 3.7, guidelines G4, G6, G8 and G9).
2. A second issue concerns the understanding of the concept of VE (see table 3.6, problems P1 and P2). Designers found that clients failed to understand the potential of VEs and this caused problems when establishing requirements. Designers tried to justify the use of VE technologies and suggested educating the client to tackle this issue (table 3.7, guidelines G1 and G3). However, important and distinguishing aspects of VEs, such as interaction, user representation and multimodality, were not frequently implemented by the designers.
3. The third issue is the immaturity of VE technology which makes developmcnt unnecessarily complex (see table 3.6, problems P8 and P9). Designers found problems with toolkits not supporting the creation of complex components. To help tackle this issue designers checked the technical viability of design ideas before beginning development (guideline G2, table 3.7).

In more recent work, some of the same issues and techniques have been highlighted. For example, Grinstein and Southard (1996) suggest reuse, texturing and level-of-detail management. Importantly, although the literature suggests that human factors are important in VE design (Höök and Dahlbäck 1992; Macredie 1995; Wann and Mon-Williams 1996b), they were rarely mentioned in this study, indicating that the designer community is not convinced of their significance.

Comparing the reported problems and guidelines with those found for more conventional systems (e.g. Bellotti 1988; Myers and Rosson 1992; Guindon 1990; Guindon et al. 1987) shows how issues for VE design differ:

- A novel issue is the design aspect of realism and level of graphical detail, which is important in copying real world models for VEs.
- Designers and clients appear to have a generally poor understanding of the nature of the interface type being developed, which exacerbates general requirements specification problems.
- The lack of appreciation for user issues is more apparent than with standard user interfaces. For example, Bellotti (1988) lists exclusion of users from the design process as a design problem *volunteered by designers* of conventional systems and Lauesen (1997) reports that designers are concerned about usability, although there are problems with misunderstanding it and a lack of confidence about solving usability issues.
- Technical issues, such as performance considerations, have greater impact on the design process and product.

3.3 Need for interface design guidance

The designer study results support the need for methods and guidance that are specific to the nature and problems of VEs. The findings show the general lack of a coherent approach to VE design. The complexity of VEs was often limited with designers lacking coherent strategies for the implementation of some important VE features, such as user representation. Design practice and issues differed from those with conventional interface design and designers typically did not find existing methods or techniques useful. Furthermore, with VEs being a relatively new interface type, the designers had little experience compared with designers of conventional systems. The median months experience was 8.5, compared with for example 84 months in the Myers and Rosson (1992) study. According to Long and Dowell's (1989) model of three levels of HCI design practice, VE designers appear to work at the craft level where design is practised by intuition based on experience, but no-one learns from the experience. Ideally, designers should be able to practice at the engineering level, according to a set of principles and methods. The designers faced difficult technological issues, but many of these are likely to become less significant as technology improves. However, problems with interaction design and tackling user issues will persist without appropriate research and guidance.

The user study showed that major interaction problems existed with current VEs, which differed from those typical with conventional interfaces. The frequent difficulties in interaction seriously affected the utility of the application for intended users. The problems generally resulted from a lack of consideration of usability issues, such as highlighting interactive hot-spots, rather than performance compromises. Comparing the two viewpoints, designers appear to be particularly concerned about technology issues although there is also concern about designing visually realistic environments. While users are less concerned about technology, realism and accuracy are deemed important. However, the other main user concern, for reliable and effective interaction, is not shared by designers. The designers showed a lack of guidelines for interaction design and little consideration for supporting the user. For example, none of the main interaction problems found with the virtual submarine, would have been avoided by employment of the designers' guidelines.

To improve VE design, designers need to be encouraged to concentrate more on the interaction and, more generally, to consider user issues. Therefore, guidance is needed to highlight and avoid potential usability problems, during the design process, supporting hypothesis one of the thesis. Furthermore, the guidance needs to be specific to VEs. However, there is little theoretical understanding of VE interaction and usability. Therefore, this thesis aims to fill in gaps in the designers' guidelines by researching into user interaction with VEs, with an improved understanding of current problems and design practice. In the next chapter, a theory of interaction in VEs is described to provide a theoretical basis for design guidance.

Chapter 4

A Theory of Interaction in Virtual Environments

This chapter describes theoretical models of interaction in virtual environments and a set of abstract usability requirements, derived from the models.

Chapter 4

A Theory of Interaction in Virtual Environments

A key component of this thesis is a theory for VEs, which predicts interaction behaviour and requirements for successful interaction. The theory elaborated hypotheses two and three of the thesis. It aimed to model important and common patterns of behaviour, for hypothesis two, and then use this modelling to define generic usability requirements for a self-explanatory VE, to fulfil hypothesis three:

H2 General patterns of interaction with VEs can be predicted, through theoretical models.

H3 Design properties required for interaction can be predicted using these general patterns (H2).

The usability requirements helped in predicting likely usability problems and how to avoid them. The theory did not aim for a complete coverage of all usability issues in VEs. Instead it aimed for the more realistic objective of covering a significant set of issues, relating to generic activities in interaction.

4.1 Overview of theory

The theory can be generally described as a hypothetical set of structured ideas about interaction in VEs. Figure 4.1 shows the main components. Interactive behaviour was predicted in a set of models which divided behaviour into stages of interaction. Using the stages of interaction, desirable properties of a VE design and relevant elements of user knowledge were defined. For each stage of interaction, a set of rules linked the components by predicting usability problems, based on the presence of relevant design properties and user knowledge. The following sections describe each component of the theory, beginning with the models of interaction.

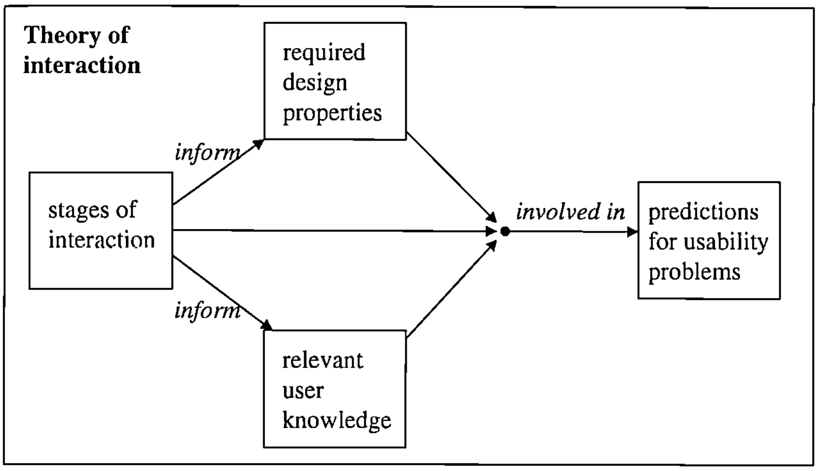


Figure 4.1: Overview of the components of the theory.

4.2 Models of interaction

Norman's (1988) general model of interaction was used as a starting point for describing interaction in VEs. Norman's model is simple, well known and describes interaction at an appropriate level of detail. It has been used in developing interaction models for evaluating direct manipulation interfaces (see Springett 1996). It is a top-down, plan-based model which consists of seven-stage cycles of interaction, as shown in figure 2.4.

The elaboration of Norman's model involved the explicit modelling of exploratory and reactive behaviours, which are important aspects of VE interaction. Tasks in VEs are often loosely structured with more emphasis on exploration and opportunistic action, where opportunities for action are seen in the environment, rather than having been pre-planned (Hayes-Roth and Hayes-Roth, 1979). For example, in many education and marketing applications, the user's task is to investigate the environment so behaviour primarily involves opportunistically following cues. Virtual environments are often active, with objects operating independently of the user's actions (Bryson, 1995). The behaviour of these agent objects can be manifest as events in the environment, which may demand or invite responsive behaviours (Gibson, 1986) from the user. For example, agents can act as monitors, notifying the user of changes in the environment, such as 'someone has entered the room - do you wish to view the doorway?' (Yamaashi et al., 1996). System agents can also initiate complex ongoing behaviours which may affect the user directly, for example by moving them to new locations (Billinghurst and Savage, 1996). To describe task-based, exploratory and reactive modes of interaction in VEs, three models were proposed:

Task action model - describing purposeful behaviour in planning and carrying out specific actions as part of user's task or current goal/intention, and then evaluating the success of actions.

Explore navigate model - describing opportunistic and less goal-directed behaviour when the user explores and navigates through the environment. The user may have a target in mind or observed features may arouse interest.

System initiative model - describing reactive behaviour to system prompts and events, and to the system taking interaction control from the user (for example taking the user on a pre-set tour of the environment).

The models are inter-connected and behaviour switches between them at appropriate points. Behaviour switches to task action mode when the user makes a decision to focus on planned tasks/goals, or to carry out intended actions. Behaviour switches to explore navigate mode when the user decides to explore and learn about the environment, or navigate and locate targets. Finally, behaviour switches to system initiative mode when an event attracts attention, or when the system takes control of interaction from the user.

The task action model was based on Norman's action cycle, with additions for:

- Consideration of objects involved in an action. Since objects in a VE are not continually presented, the user may need to reason about what environment objects are available for carrying out actions.
- Having to search for objects when they are not within the environment section in view. Search tasks are an important part of VE interaction (for example, see Darken 1995).
- Approaching objects and orienting correctly to them. Approaching objects is non-trivial in 3D interaction and appropriate 3D orientations to objects are required.
- Object investigation actions, as opposed to object manipulations. The user may only be interested in examining VE content, rather than manipulating it in some way.

The other two models were developed from the basic processes present in Norman's model. Explore navigate model was developed to focus on movement through the environment and exploratory or opportunistic action. Basic features found in display-driven models were included, such as scanning of the interface and the selection of features (e.g. from Springett's knowledge-based model; Springett, 1996). System initiative model was developed to focus on interpreting and responding to system behaviour. Norman's model was used again to check for completeness in modelling interaction behaviour. Various paths through the models were checked to ensure all relevant basic processes, such as perception and interpretation, had been incorporated.

The following sections describe each model with specific examples. The models consist of stages of interaction (shown as circles in the diagrams), flows of interaction from one stage to another (shown as arrows), and transfers between modes (shown as block arrows). System initiative model also includes environment triggers (shown as curly arrows). Although the environment would be involved throughout, such as in providing feedback in task action mode and a changed viewpoint in explore navigate mode, it was not included in the diagrams for reasons of simplicity. The flows of interaction pass information between stages, but there were no explicit pre- and post-conditions set or rules defined to govern transitions between stages. Such a level of predictivity was not felt to be realistic or necessary for general modelling of interactive behaviour.

4.2.1 Task action model

Walking through task action mode, the user establishes a goal (stage 'establish goals' in figure 4.2), such as to study the electricity supply in a building, and forms an intention to carry out an action to turn on power ('intention task action'). S/he then considers what power objects are available in the environment ('consider objects'), such as mains boxes and switches. If the mains are not within his/her immediate vicinity, a search for them is carried out in explore navigate mode. Once the mains are found, s/he approaches and takes up a suitable orientation to them ('approach/orient'), see figure 4.3. S/he then deduces how to turn on the power at the mains ('deduce sequence') and executes the action ('execute'). S/he interprets feedback in the environment ('feedback') to see whether or not power has been turned on. Alternatively, after approaching the mains, if s/he had an intended action to study the mains, rather than turn on power, s/he would closely inspect and investigate the mains ('inspect'). Finally, s/he evaluates the outcome of this inspection or the action to turn on power, on his/her goal to study the electricity supply ('evaluate').

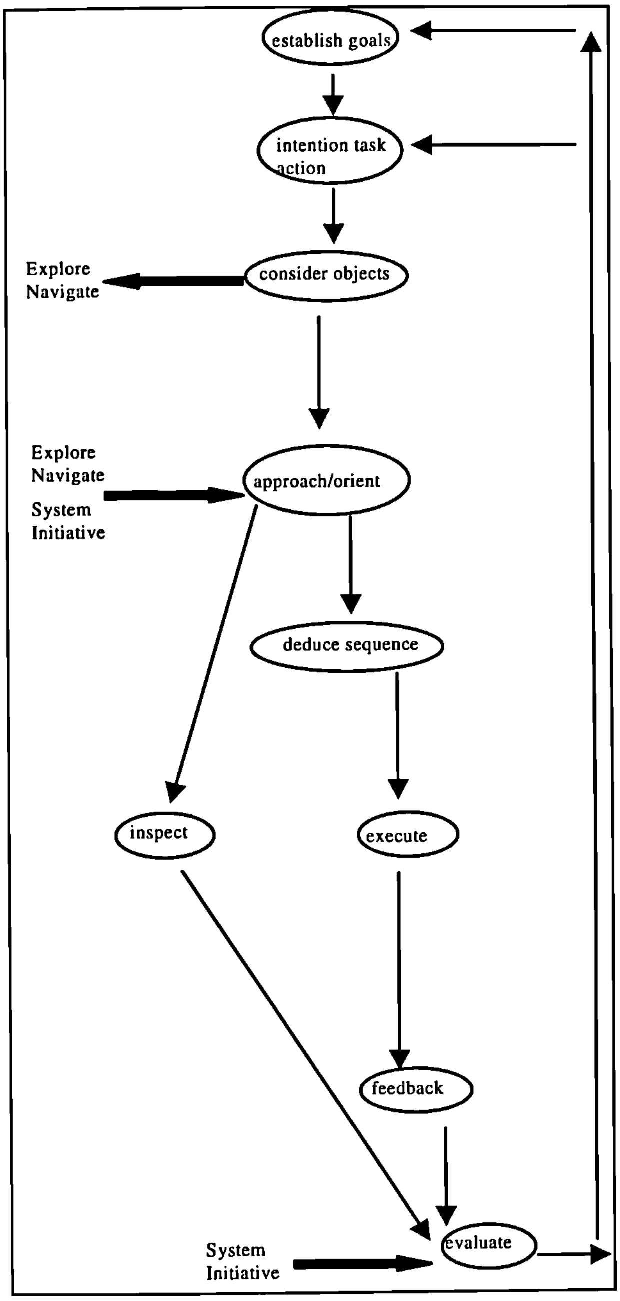


Figure 4.2: Task action model, showing stages (circles) and flows of interaction (arrows).

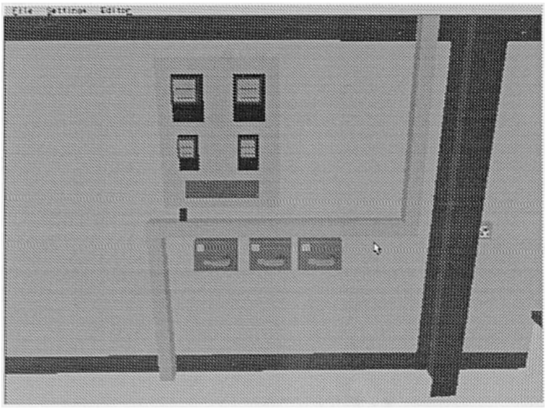


Figure 4.3: The user's view when approaching the mains object.

Table 4.1 gives a brief description of the processing involved in each stage in task action mode.

Stage	Description
establish goals	Formulate high level goal to drive interaction.
intention task action	Formulate next action to carry out.
consider objects	Determine objects or environment section of interest to the current action.
approach /orient	Approach target in vicinity and suitably orient self to target.
deduce sequence	Specify sequence of operations required to carry out object manipulations for the intended action.
execute	Perform the object manipulations using the interaction sequence input from the previous stage.
feedback	Understand the feedback available from the environment with respect to the executed action.
inspect	Obtain required information from approached target by inspecting it.
evaluate	Assess implications of interpreted environment state for the ongoing task.

Table 4.1: Description of processing involved in each stage in task action mode.

4.2.2 Explore navigate model

Walking through explore navigate mode, the user forms an intention to explore the environment (stage 'explore' in figure 4.4), such as a virtual building. S/he scans the observable environment ('scan') and decides to move forward through the building ('plan'). S/he navigates forward ('navigate') and re-scans the environment. If, for instance, s/he sees a cupboard which arouses interest, see figure 4.5, s/he decides to investigate the cupboard ('intention explore action') and this action is now carried out in task action mode. In the case of transfers from task action mode to search for targets, such as the mains boxes, the user scans, plans and navigates and when s/he finds the mains boxes, s/he returns to task action mode to continue with the planned action.

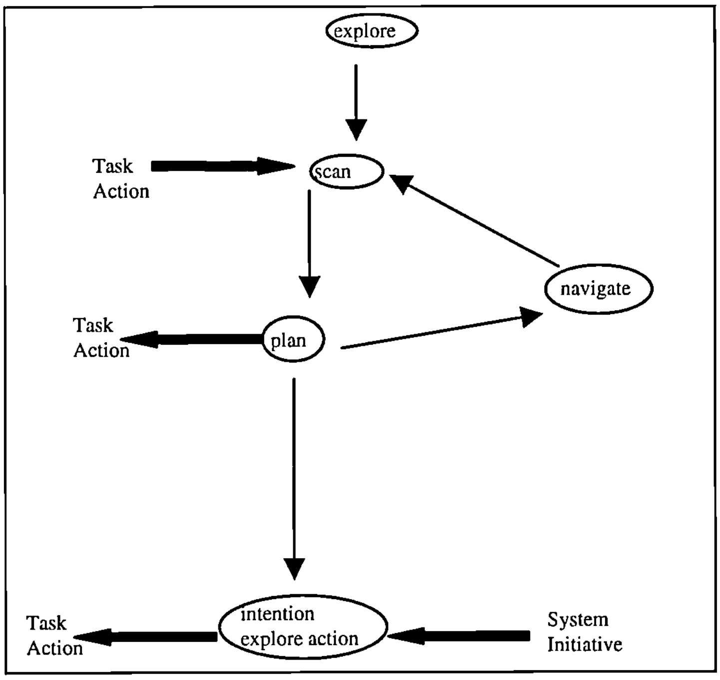


Figure 4.4: Explore navigate model, showing stages and flow of interaction.

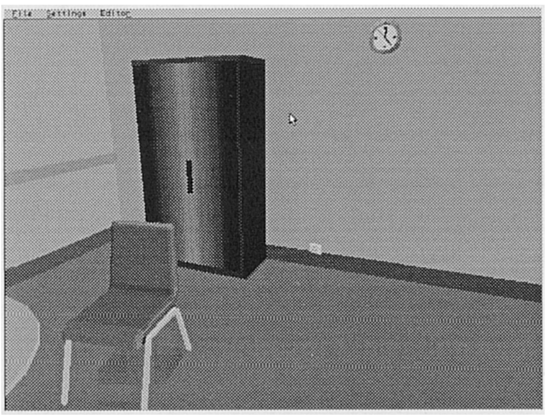


Figure 4.5: The user scans and sees a cupboard object which arouses interest.

Table 4.2 gives a brief description of the processing involved in each stage in explore navigate mode.

Stage	Description
explore	Establish will to explore.
scan	Inspect current state of environment from available output.
plan	Determine appropriate plan of further activity from inspection of environment, with respect to either a target search or an exploration intention.
navigate	Move self to a location elsewhere in the environment, following plans from previous stage.
intention explore action	Formulate action to carry out on a feature of interest.

Table 4.2: Description of processing involved in each stage in explore navigate mode.

4.2.3 System initiative model

System initiative behaviour may either be events or interaction control. In the case of events, the user perceives and interprets an event (stage 'event' in figure 4.6), such as a ringing telephone, see figure 4.7. S/he plans how to respond to it ('plan'). S/he may immediately decide to answer the telephone ('intention reactive action'), and carry out the action in task action mode. Alternatively, s/he may investigate how to use the telephone, in exploratory mode, or evaluate what the telephone ringing event means to his/her ongoing task, in task action mode. In the case of interaction control, the user acknowledges the beginning of system control ('acknowledge control'), such as an automated tour of a building. S/he watches the tour ('monitor') and acknowledges when the tour has ended ('end control'). S/he then plans how to respond to this system behaviour ('plan' again). Whilst watching the tour, s/he may decide s/he has seen enough and would like to quit the tour ('intention control action').

Table 4.3 gives a brief description of the processing involved in each stage in system initiative mode. Appendices 4A to 4C give full descriptions of each stage, for each of the three modes.

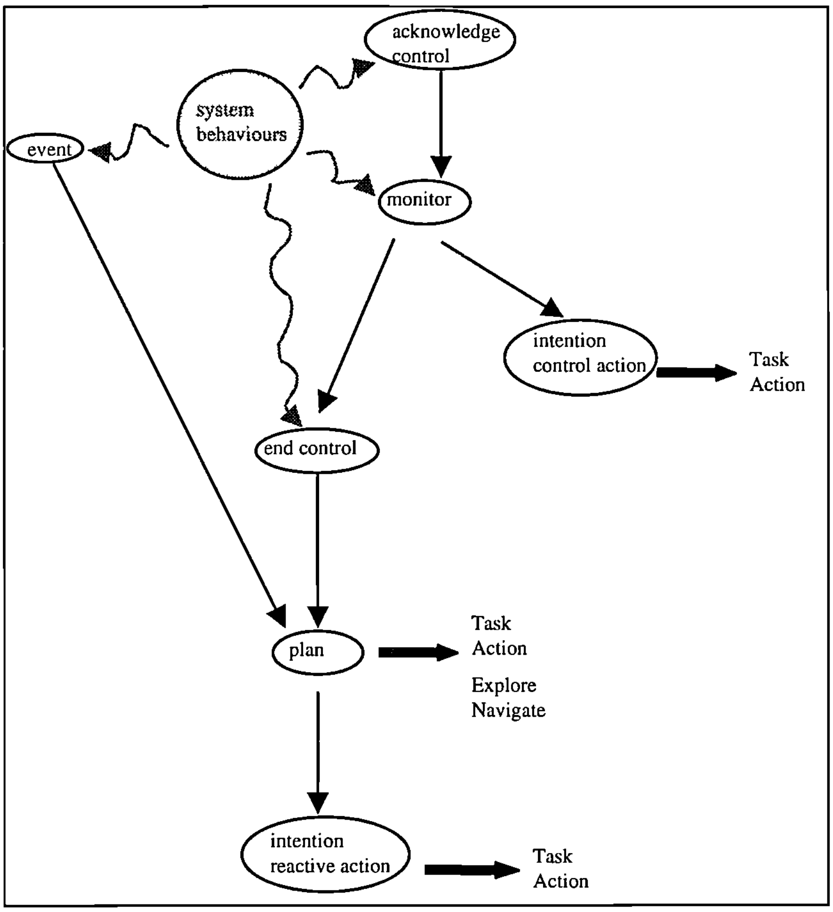


Figure 4.6: System initiative model, showing stages and flow of interaction. Environment triggers (system behaviours) are shown with curly arrows.

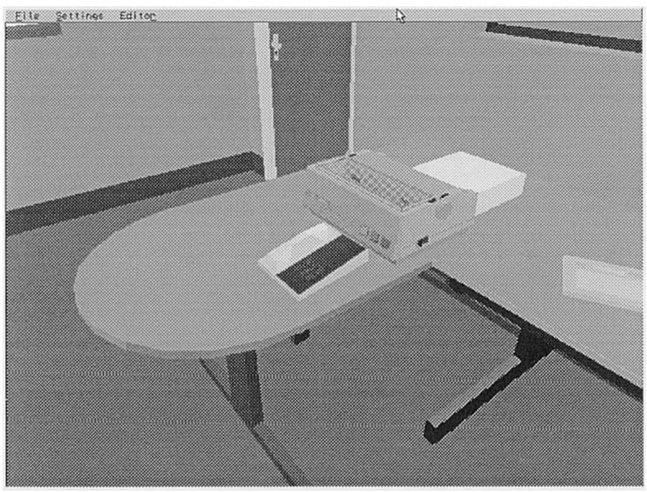


Figure 4.7: The user perceives a telephone-ringing event.

Stage	**Description**
event	Perceive event and establish some understanding of it.
acknowledge control	Realise and accept that system has taken control over interaction.
monitor	While system control continues, monitor system behaviour.
intention control action	Form intention to carry out action to exercise user control.
end control	Realise that system control has terminated and control is now returned.
plan	Determine appropriate plan of how to deal with system behaviour.
intention reactive action	Formulate action to carry out, to react to system behaviour.

Table 4.3: Description of processing involved in each stage in system initiative mode.

4.2.4 Discussion of models

The patterns of behaviour in the models are general and apply across different VEs. For example, for the virtual submarine application, detailed in chapter three, task action mode would capture behaviour in carrying out actions, such as identifying submarine equipment by clicking on objects (see figure 3.1). A user in explore navigate mode could be familiarising themselves with the submarine layout, and a user in system initiative mode may be reacting to system training instructions to locate pieces of equipment (see figure 3.2).

Activity was divided into the different models in order to distinguish different types of interactive behaviour and planning, but also to describe interaction in as complete a way as possible, whilst avoiding redundancy. Therefore, boundaries were set between the models that best separated off modes of interactive behaviour and avoided duplicating activity. For example, since the behaviour involved in carrying out the different action types (task, exploratory and reactive actions) was essentially the same (i.e. approach, deduce sequence, execute, etc.), this behaviour was included once only in task action mode, with transfers to this mode after forming an action intention. As a second example, basic activity in navigating through the environment was included once in explore navigate mode, whether the navigation was for exploration or searching targets, with known or unknown locations.

The models aim to capture the basic flow of interaction, and it should be recognised that behaviour will often deviate from the simple patterns. It is expected that some stages may be skipped, for example in task action mode the 'deduce sequence' stage may not be needed by the skilled user, who has learned the required action sequence. There may be repetitions of stages or parts of models, for example some tasks may involve several object searches or manipulations. Backtracking to previous stages may also occur as remedial activity when the user encounters errors. Finally, the ordering of some stages may differ, for example 'deduce sequence' may be carried out before, instead of after, an object 'approach/orient'. Despite the inevitable complexities of actual interaction behaviour, a general understanding of patterns of behaviour and the basic flow of activity is important in understanding interaction. The models aim to provide such a general understanding of VE interaction.

The models describe interactive behaviour at a relatively high level of granularity. As with the original Norman model, little was included about detailed cognitive behaviour. However, the models were based on some simple and reasonably well-established assumptions about user cognition. A user memory component was assumed, which could be homogeneous and did not have to be divided into short-term and long-term memory, because this level of distinction was not used. The theory focused on the existence of knowledge held by the user and did not elaborate on the retrieval or updating of knowledge, the representation format or the deterioration of knowledge. The models assumed the presence of sensory, perceptual, cognitive and motor activities in the user, and the links between these. Although such activities were not detailed in the models, they have been described extensively in psychology literature (e.g. Glass and Holyoak, 1986) and described specifically for HCI in The Model Human Processor (Card et al., 1983). More detailed psychology research was referred to where necessary, such as to inform requirements to support object understanding, in the next section.

4.3 Generic design properties

The interaction models provided a clear breakdown of interactive behaviour. The models were used to systematically reason about what properties are required in a design to support the user during each identified stage in interaction, so that usability difficulties will be avoided. The models highlighted relevant areas of research for informing required design properties. The main areas being:

- General HCI guidance, such as the need for task support for the task action model, for example the task conformance principle (Dix et al., 1993) to provide actions as required by the user task (see required design property *action support for task*). Also, general HCI guidance for action support (e.g. Norman, 1988) was used, such as knowing how to carry out actions and being able to assess the progress and success of executed actions (see *declared action sequence*, *clear action progress* and *declared action effect/success*).
- Guidance on supporting exploratory behaviour for the explore navigate model, such as making the repertoire of available actions salient (see *discernible repertoire of opportunities for action*), from Lewis et al.'s (1990) design for successful guessing.
- Research on spatial cognition and preliminary work on aiding spatial behaviour in VEs, such as having a clearly organised layout (see *discernible spatial structure*), from the work of Darken (1995).
- Research on basic psychology, such as object perception requiring the observer to distinguish physical characteristics, recognise the object and identify its function and meaning (see *distinguishable object, identifiable object and clear object role*), from Rock's aspects of object perception (in Taylor et al., 1982).

Forty-six individual generic design properties (GDPs) were identified for supporting all behaviour predicted in the models. The GDPs were found to often relate to more than one stage of interaction. For example, requirements for understanding the current view angle (e.g. *clear self position/state*) were relevant in the 'navigate' and 'approach/orient' stages in explore navigate and task action modes. Therefore, rather than organising GDPs by stage of interaction, the set of properties were organised into categories for information about:

- the user's *task*,
- the *overall VE*,
- *spatial knowledge* of the VE,
- the *user's viewpoint and representation*,
- *objects*,
- *system initiative behaviour*,
- available *actions,* and
- *action feedback.*

The task category covered requirements for basic support of the user task and information about task progress. The 'overall VE' category covered properties for providing information about the environment as a whole, rather than the sub-section currently within view. The spatial knowledge category included properties for understanding the spatial layout of the VE and for locating objects. The viewpoint and user representation category covered support for understanding the user object, including the view-angle and any attachments to the user object (e.g. tool belt). The object category included requirements for investigating environment objects, both passive and active. The category for system initiative behaviour covered properties for interpreting system events and ongoing system control. The actions and action feedback categories included requirements for carrying out and assessing the progress and success of actions, and covered both actions on specific objects and actions with the user object (e.g. navigation and the use of attached tools). Table 4.4 lists a few of the design properties and a complete list is given in appendix 4D.

For example, in the virtual submarine, some of the generic design properties listed in table 4.4 were supported. Information was given about the user's task to locate equipment, satisfying the *clear task/ task flow* GDP. Feedback was given when equipment was successfully or wrongly identified, supporting the GDP for a *declared action effect/ success*. However, it was not clear how to open a hatch at the top of a ladder, because the requirement for *declared action sequence* was not satisfied. The hatch was opened through a button on the control panel, at the bottom of the screen, labelled 'open close', see figure 4.8. However, with this design, there was no explicit linking between the hatch and the button on the control panel. A more suitable design might have been to allow the user to click on the hatch itself to open and close it.

Category	**Generic Design Property**	**Description**
User task	*clear task/ task flow*	The user's task in interacting with the environment is clearly defined.
Overall VE	*discernible repertoire of opportunities for action*	The general opportunities for user action within the whole VE can be easily determined.
Spatial knowledge	*locatable object/ areas of interest*	Important objects and areas of interest can be easily located.
Viewpoint & user representation	*detectable self parts*	The parts making up the user representation can be easily located from any orientation, and can be distinguished from the rest of the VE.
Objects	*identifiable object/ objects parts*	The object and its individual parts can be readily and reliably identified. If the object is copied from real world phenomena, then its representation is accurate and matches user expectations.
	accessible object	The object can be easily accessed, i.e. the user can closely approach the object and take up a suitable position/orientation to it.
System initiative behaviour	*declared system control commencement/ termination*	It is clear when the system takes control of the interaction, and later when control is returned to the user.
Actions	*declared action sequence*	The sequence of operations required to carry out the action are clear.
	executable action	The action can be executed efficiently and without frequent obstacles/problems. The demand of manipulation precision and motor co-ordination is within usual human ability.
Action feedback	*declared action effect/ success*	Feedback on the effect of the executed action is given. The success of the action execution can be readily determined and errors easily detected.

Table 4.4: Example generic design properties for each category.

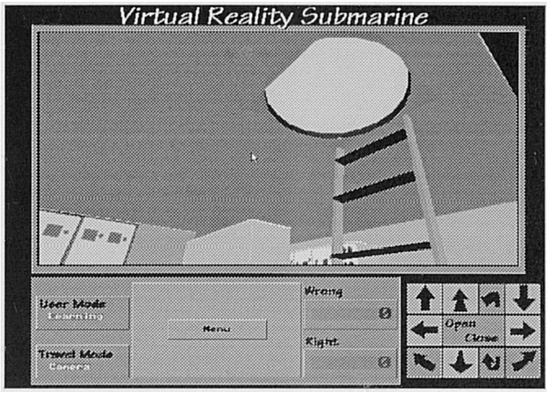


Figure 4.8: The hatch at the top of the ladder is opened by clicking on the 'open close' button, at the bottom right of the screen, not by clicking the hatch itself *(taken from the Royal Navy's Virtual Reality Submarine application).*

The set of generic design properties can be compared with existing HCI guidance, such as Norman's (1988) general design principles and the direct manipulation guidance of ISO DIS 9241-16.1 (ISO, 1996). The main differences are:

- More emphasis has been placed on facilitating perception (e.g. *distinguishable object*), because the added spatial dimension and degrees of freedom in VEs can make perception of VEs more problematic.
- Emphasis has also been placed on encouraging and guiding exploration (e.g. *discernible repertoire of opportunities for action*), because of the more exploratory nature of VE interaction.
- Guidance related to the match between the VE and phenomena copied from the real world has been included throughout (e.g. *discernible spatial structure, identifiable object* and *clear object role*), because VEs commonly model real world phenomena which the user may be familiar with and have certain expectations about.
- Guidance has been included on more novel aspects in VEs such as spatial navigation, viewpoint and user representation, and system behaviour (e.g. *clear navigation pathways*, *clear self position/state* and *declared causality and effects of behaviour*).

The generic design properties were further classified as being either:

- *basic support* (BS) - where the design provides basic requirements for interaction and task completion (e.g. *distinguishable object*), or
- *information provider* (IP) - where the design provides information for interaction that could also be found as knowledge in the user's internal memory (e.g. *identifiable object*).

This classification of the GDPs is important when understanding and predicting usability problems, in section 4.5, since information provider GDPs can be dependent on knowledge the user may have and need to provide information that is consistent and compatible with that knowledge.

4.4 User knowledge sources

User knowledge is important in modelling interaction because it plays a key role in learning and performance (Shneiderman, 1992). Extensive user knowledge, for example through training or interface expertise, can alleviate usability problems caused by poor design properties. The next component of the theory hypothesised relevant sources of knowledge available to the user during interaction. These sources were seen to be distributed between the VE and the user's internal memory. Six knowledge sources in total were identified as potentially important for VE interaction:

Environment Available - an external knowledge source consisting of all that is currently perceivable in the VE, including audible sounds and tactile feedback. This will only be a sub-section of the total environment because the environment will generally be too spatially large to be perceivable at once. The generic design properties, discussed in section 4.3, can be implemented in a VE and therefore presented to the user through this source.

Environment Model - a knowledge source within the user's memory consisting of knowledge about the current VE. It originates from experiences with the VE which result in knowledge, such as environment content and layout, being accumulated during interaction. The spatial aspect of the knowledge may be structured through 'cognitive maps' (Neisser, 1976). Expert users will have a more complete, accurate and well-structured environment model.

Task - user's knowledge about the application task and task progress. It can be used to determine current goals and intentions when interacting with the VE, as well as procedures for carrying out task actions. Generally some knowledge of task goals will exist, although it could be incomplete or ill-defined.

Domain - user's knowledge about the application domain that may be used to make inferences about the content or layout of the VE, depending on the strength of the relationship between the domain and the VE. For example, with the virtual submarine this source would represent knowledge about the physical submarine, modelled in the VE, such as equipment within the submarine. For fictitious VEs, that are not modelled on a real world domain, this knowledge source will not exist.

Real World - user's general knowledge about the real world, that is not specific to the application domain in question. This knowledge may be helpful in making inferences about natural interaction styles and common real world objects represented in the VE. For example, with the virtual submarine this source would include knowledge about the representation and functionality of a ladder.

Other Environments - user's knowledge about other VEs they have experienced. Meta-knowledge of standards and commonalities between environments may be helpful in making inferences for carrying out interaction in the current VE. For example, a common standard for Superscape VEs is set starting positions linked to function keys.

Most of these knowledge sources are found in current HCI theories, for example Cognitive Complexity Theory (Kieras and Polson, 1985). However, domain and real world knowledge were felt to be additionally important for VEs, because applications typically closely model real world domains and use common real world objects (e.g. doors) and natural interaction styles. There can be overlaps between the knowledge sources and redundancies in the overall knowledge. For example, knowledge in the environment model may overlap with domain knowledge, where the VE is closely modelled on a well-known domain. Figure 4.9 gives a conceptual overview of the origins and inter-relationships between the knowledge sources.

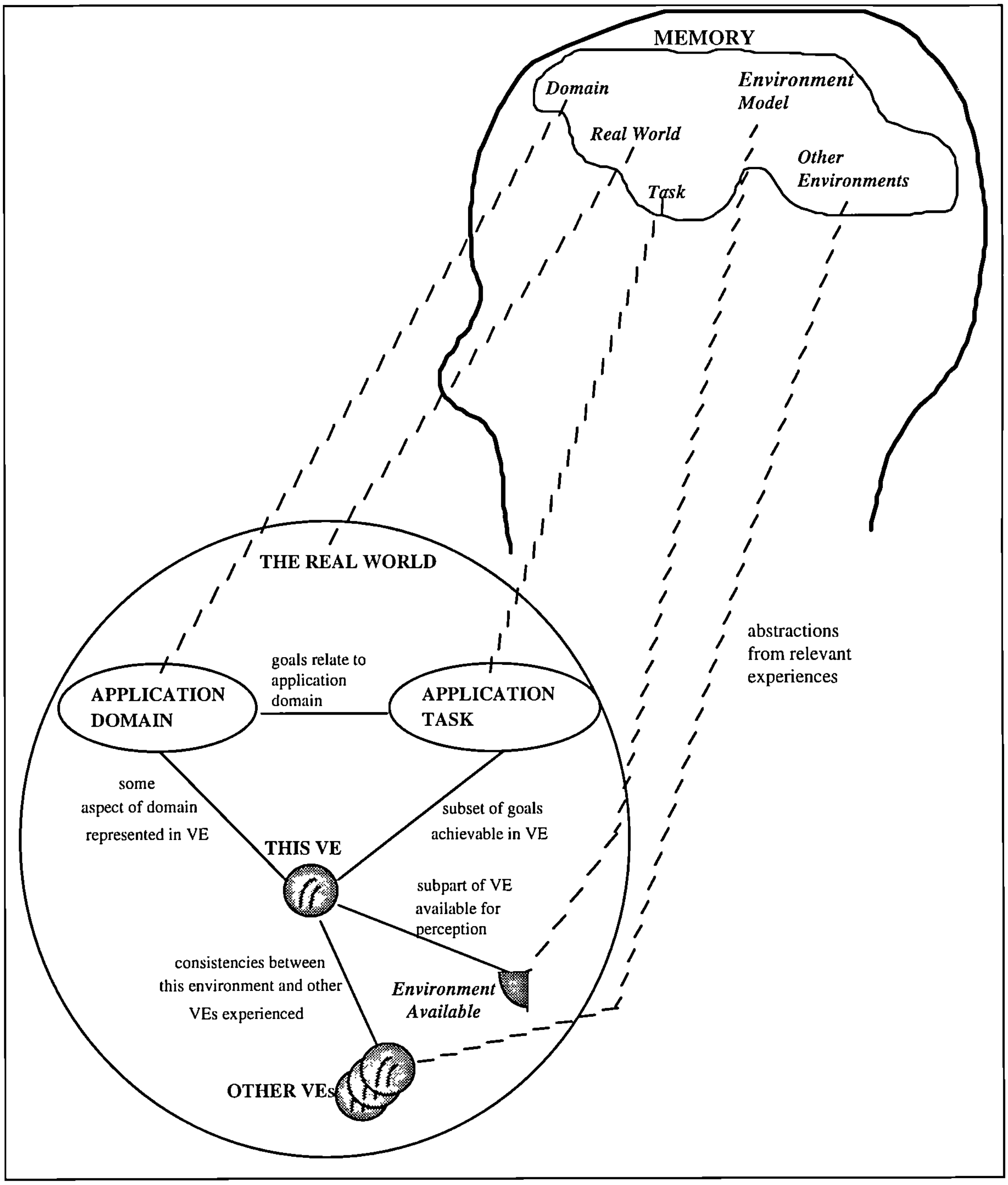


Figure 4.9: A conceptual representation of the knowledge sources, showing their origins and inter-relationships. Experiences with the real world, the application and other VEs are abstracted to makc up the knowledge sources. The VE in question is experienced through the Environment Available source, and abstractions from these experiences make up the Environment Model.

Each of the five internal knowledge sources can provide specific information for the stages of interaction. These items of information correspond with the information provider GDPs. For each IP GDP, there can be an element of user knowledge, which contains corresponding information, and which may be found in one or more of the knowledge sources. For example, during the *scan* environment stage, in explore navigate mode (see figure 4.4), relevant user knowledge includes *identification information for objects* which corresponds with GDP *identifiable objects*. In the virtual submarine, information for identifying equipment would be found in the *Environment Model* source, for equipment seen and identified in previous interactions with the VE; in the *Domain* source, for submarine equipment experienced in the corresponding physical submarine, or in the *Real World* knowledge source for common objects, such as the ladder. During the *navigate* stage, relevant information includes the *location of targets* which corresponds with GDP *locatable objects/areas of interest*. Information about the location of equipment in the virtual submarine would be found in the *Environment Model* source, if the VE had been experienced before, or in the *Domain* source, if the user had knowledge of the layout of the physical submarine. The links between the stages of interaction, GDPs and user knowledge elements were detailed in the final part of the theory, where these components were used to predict usability problems.

4.5 Correspondence rules for problem prediction

The theory predicted that interaction difficulties would be linked to poor environment design properties and/or the lack of user knowledge. In the problem prediction part of the theory, correspondence rules (IF..THEN.. statements) were used to specify the conditions under which potential usability problems would be likely to occur.

Usability problems were defined as critical incidents or breakdowns in interaction, which interfered with the user's ability to efficiently and effectively complete interaction tasks (from definition by Karat et al., 1992). Usability problems were identified through systematic reasoning about 1) required processing during each stage and possible breakdowns during this processing, and 2) required information during each stage and the resulting effects of the information being missing or inadequate. Usability problems could have one or more causes. The possible causes of the usability problems were identified through systematic reasoning about GDPs and elements of user knowledge relevant to each problem situation and required to avoid the problem. Usability problems were predicted for each stage in the interaction models and each GDP was referred to at least once. Seventy-five correspondence rules were defined to predict problems, the basic format being:

```
IF condition1
OR condition2 ...
THEN predicted problem = (...);
```

Each condition involved a possible cause for the predicted problem and referred to one GDP, using one of the following formats:

1. For Basic Support GDPs, the conditions checked on the presence of the GDP using the basic format -

   ```
   ( NOT GDP(...) )
   ```

2. For Information Provider GDPs, the conditions checked on the availability of required information from either the relevant GDP or internal knowledge sources, and also that information provided by the GDP was consistent with user knowledge. This condition type used the function, Available and Matching Information (AMI), and followed the basic format -

   ```
   ( NOT AMI(relevant information(...),knowledge sources(...), GDP(...)) )
   ```

   The format of the Available and Matching Information function was –

   ```
   AMI(relevant_information, internal_knowledge_sources, GDP):
   IF ((relevant_information IN internal_knowledge_sources) OR GDP )
   AND (consistent_information(internal_knowledge_sources, GDP))
   THEN TRUE
   ELSE FALSE;
   ```

The 75 correspondence rules predicted a range of problems related to planning, decision making, locating targets, exploration, perception, understanding VE components and behaviours, identifying interactions, determining action sequences, acting and feedback. Table 4.5 gives six rules for example usability problems that can be encountered in different stages in the three modes. The problems are *difficulty determining whether further interaction possible*, *difficulty determining what objects are in immediate vicinity*, *difficulty locating the destination in the environment, difficulty executing navigation, difficulty assessing progress in navigation* and *difficulty deciding appropriate response to system activity.* A full list of the rules is given in appendix 4E.

Stage: *approach/orient* in task action mode
23 IF (NOT **AMI**(**RI**(*actions available with target object*), **KS**(*EM,D,RW*), **GDP**(*declared available action - actions on target*))) (NOT **AMI**(**RI**(*interactivity of target object*), **KS**(*EM,D,RW*), **GDP**(*clear object type/ significance - target*))) THEN **PP** = (*difficulty determining whether further interaction possible*); If there is not available and matching information about possible actions with available objects Or there is not available and matching information about the interactivity of available objects Then the user is likely to have difficulties determining whether further interaction is possible with target objects

Stage: *scan* in explore navigate mode
42 IF (NOT **GDP**(*distinguishable object - all objects currently perceivable*)) OR (NOT **AMI**(**RI**(*identification information for objects*), **KS**(*EM,D,RW*), **GDP**(*identifiable object - all objects currently perceivable*))) THEN **PP** = (*difficulty determining what objects are in immediate vicinity*); If objects in view are difficult to distinguish Or there is not available and matching information about the identity of objects Then the user is likely to have difficulties determining what objects are in their immediate vicinity

Stage: *navigate* in explore navigate mode
51 IF (NOT **AMI**(**RI**(*spatial structure of environment*), **KS**(*EM,D*), **GDP**(*discernible spatial structure*))) OR (NOT **AMI**(**RI**(*location of areas of interest or target*), **KS**(*EM,D*), **GDP**(*locatable areas of interest/object - target*))) THEN **PP** = (*difficulty locating the destination in the environment*); If there is not available and matching information about the spatial structure of the environment Or there is not available and matching information about the location of target objects Then the user is likely to have difficulties locating their destination in the environment
56 IF (NOT **GDP**(*executable action - navigation*)) THEN **PP** = (*difficulty executing navigation*); If navigation is not easy and efficient to execute Then the user is likely to have difficulties navigating in the environment
57 IF (NOT **GDP**(*clear action progress - navigation*)) OR (NOT **AMI**(**RI**(*semantics behind feedback received while carrying out navigation*), **KS**(*EM,OE,RW*), **GDP**(*clear during action effects - navigation*))) THEN **PP** = (*difficulty assessing progress in navigation*); If there is inadequate feedback about progress whilst navigating Or there is not available and matching information about the effect that navigation actions have on the self object/viewpoint Then the user is likely to have difficulties assessing progress whilst navigating in the environment

Stage: *plan* in system initiative mode
73 IF (NOT **AMI**(**RI**(*significance of system activity to overall goals and interaction*), **KS**(*EM,D,T*), **GDP**(*clear system activity significance*))) OR (NOT **AMI**(**RI**(*appropriate response to system activity*), **KS**(*EM,D,T,RW*), **GDP**(*appropriate response to system activity*))) OR (NOT **AMI**(**RI**(*goals for system control*), **KS**(*EM,T,D*), **GDP**(*clear system control purpose*))) THEN **PP** = (*difficulty deciding appropriate response to system activity*); If there is not available and matching information about the significance of a system activity to the environment and task Or there is not available and matching information about an appropriate response to the system activity Or there is not available and matching information about the purpose of a system control (where the system activity is a system control) Then the user is likely to have difficulties determining how to respond to the system activity
KEY: **AMI** – Available and Matching Information function **RI** – Relevant Information **KS** – Knowledge sources: Environment Model (EM), Domain (D), Task (T), Real World (RW), Other Environments (OE) **GDP** – Generic Design Property of Environment Available **PP** – Predicted Problem

Table 4.5: Correspondence rules for six potential usability problems.

In the evaluation of the virtual submarine (see chapter 3), trainee submariners faced difficulties identifying interactive hot-spots in the environment, such as active buttons on a switchboard (see figure 3.6). As rule 23 indicates, in table 4.5, this was because interactions were not cued and active objects were not distinguished from inactive ones (GDPs *declared available action* and *clear object type/ significance* not supported). Additionally, since users did not have previous experience with the virtual submarine, they had no knowledge about available actions and the interactivity of different objects. Submariners also noted problems in recognising some objects due to inconsistencies between the design and user expectations (see second condition of rule 42). For example, figure 3.1 shows that the representation of the ladder aids recognition but the representation of the fire hose appears too simplistic and does not aid recognition. However, the submarine included a feature, for some objects, which provided an identification label when the object was clicked (supporting GDP *identifiable objects*), and this helped reduce the object recognition problem.

Problem 56, in table 4.5, was found to be a major difficulty with the virtual submarine. Control panel buttons were used for navigation by a set distance in various directions. One common problem with this technique was getting precise positioning which was necessary to go through the hatch at the top of the ladder (see figure 4.8). As the rule indicates, this problem is only dependent on the GDP *executable action*, and there is little in the way of user knowledge that can compensate for deficiencies in the design of this property. To make navigation through the virtual submarine easier to carry out, a less constrained navigation style may be more appropriate, such as the use of a space-mouse. Alternatively, user support in the way of automatic alignment over the hatch could be provided. A further navigation difficulty found with the virtual submarine is described in rule 57 (see first condition). Users often bumped into objects, without knowing it, and were unable to move forward. Again this was due to the design, which gave inadequate feedback during navigation, when colliding with objects. Sounds or visual highlighting effects perhaps should have been used to indicate object collisions.

Problems in locating equipment (rule 51 in table 4.5) were not found to be common because users had good Domain knowledge of the spatial structure of the submarine and the location of equipment, due to classroom training and experiences with the physical submarine. This knowledge compensated for the fact that the corresponding generic design properties were not present, i.e. there was no spatial plan with equipment locations identified. Finally, the virtual submarine included system prompts to locate various pieces of equipment (see figure 3.2). The design made it clear to the user what following action was required (GDP *appropriate response to system activity* supported), so there were no problems knowing how to respond to the system prompts (see rule 73).

In the 75 correspondence rules, there were some areas of similarity between problems predicted for different stages. For example, some of the problems when approaching objects (*approach/orient* stage) overlapped with problems in navigation (*navigate* stage). However, such problems were kept separate so that each stage could be assessed individually. Additionally, no probabilities were incorporated in the rules, as this level of precise prediction was not felt necessary or realistically achievable. However, it would be expected that the more conditions that are satisfied, the more likely it is that the predicted problem will occur. In applying the rules, the rules may be forward-chained to ascertain likely problems given specifications for user knowledge and design properties. Alternatively, the rules could be backward-chained to determine likely reasons for observed problems.

Usability problems linked to interaction stages have been predicted in walkthrough evaluation methods for conventional interfaces (e.g. Polson et al., 1992), although not to the extent of using formal rules with specific problem causes. Differences in the areas covered by the predicted problems tend to follow patterns in the differences in design requirements, as briefly discussed in section 4.3.

4.6 Inter-relationships between theory components

The correspondence rules involve all components of the theory in predicting usability problems. Therefore, the rules show the links between the different components. For each stage of interaction in the models, there are a number of specific usability problems that can occur, each predicted in one correspondence rule. Each usability problem has one or more causes and each cause refers to one GDP (i.e. if the GDP is missing or inadequate then the problem may occur). Some of the GDPs, specifically the information provider GDPs, can be related to prior user knowledge. The information provided by the GDP may be duplicated in user knowledge and may be dependent on expectations from user knowledge. Therefore, for this type of GDP, the conditions in the rules refer to a corresponding element of user knowledge, from a set of possible sources, and check the consistency between the GDP and knowledge element. There are more rules than GDPs because individual GDPs often relate to more than one usability problem and more than one stage of interaction. Figure 4.10 shows the inter-relationships between the theory components. The next section gives a summary of the theory of interaction.

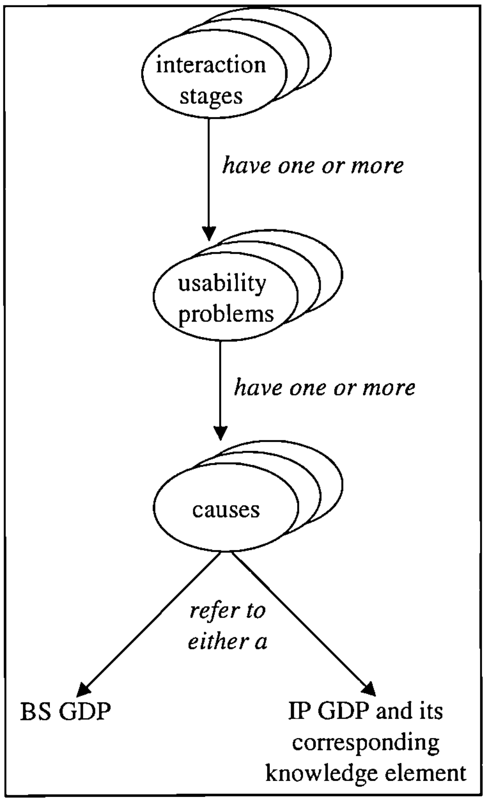


Figure 4.10: The inter-relationships between the theory components. Each interaction stage has a number of associated usability problems. Each usability problem has one or more possible causes. The causes refer to either one basic support GDP or one information provider GDP, with its corresponding user knowledge element.

4.7 Summary: theory of interaction

The theory describes interaction with VEs in three modes of behaviour: task action, explore navigate and system initiative, for task-based, exploratory and reactive behaviours respectively. Within the modes, interactive behaviour is divided into 21 stages of mental and physical action. Forty-six general properties describe requirements of a VE design for supporting all the stages. The properties relate to the user task, overall VE content, spatial layout, viewpoint and user representation, objects, system initiative behaviour, actions and action feedback. Broad areas of knowledge important for VE interaction are the VE interface, the internal model of the VE, application domain and task knowledge, real world knowledge and knowledge about other VEs experienced. From these broad areas, relevant elements of user knowledge are defined for the stages of interaction. In the final part of the theory, potential usability problems are predicted for each stage, depending on the presence of relevant elements of user knowledge and required design properties. The theory includes 75 correspondence rules, which predict usability problems and their causes.

The theory is an important step towards addressing the problem of interface design for VEs. It provides:

- an understanding of interactive behaviour and an identification of stages of interaction that need to be supported by an environment design;
- an identification of usability problems that should be avoided, and those likely given a user's prior knowledge;
- an identification of required design properties for supporting interaction and avoiding usability problems.

Therefore, the theory provides an improved understanding of user interaction with VEs and usability requirements. In this thesis, the theory also provides the basis for interface design guidance for VEs. However, before using the theory, it was evaluated against actual interaction behaviour to assess its accuracy and representativeness. The next chapter details the evaluation of the theory and chapter six describes the use of the theory to inform design guidance.

Chapter 5

Evaluation of the Theory of Interaction

This chapter describes empirical work carried out to evaluate the models of interaction and generic design properties. A controlled study is described to test the impact of the design properties on interaction success. Refinements are made to the theory in light of the results.

Chapter 5

Evaluation of the Theory of Interaction

The theory of interaction was developed to be extensive and complex, so that it would cover a significant set of usability issues for VEs. Therefore, in evaluating it, only major components of the theory could be tested, such as the stages of interaction and the generic design properties. The aim was to gain general support for the theory, rather than validate its predictions completely. Therefore, support was sought for key predictions in the theory, such as the modes of interaction. Detailed theory components, such as individual rules and rule conditions, could not be fully tested, but an overall good correspondence between theory predictions and actual observations was sought. Additionally, it was important to assess the effect that application of the theory would have on interaction with a VE. With general support for the theory and an indication of its positive impact on interaction, it could be used with some confidence to inform design guidance.

The theory of interaction was evaluated by comparing predicted aspects of interaction with observations made in empirical studies of users. Discrepancies found with observed interactions were subsequently used to refine the theory.

5.1 Overview of the empirical studies

The empirical studies provided data on user interaction behaviour, usability problems encountered and the affect of the GDPs on interaction.

5.1.1 Test application

The test application used was a business park simulation, developed by VR Solutions, which was being used by The Rural Wales Development Board for marketing of business units to potential leaseholders. It was a desktop application consisting of two worlds - an external view of the business park and an inside view of a unit in the park. The unit could be viewed as either an empty complex or a factory or office. Hot-keys, 'SHIFT-H' and 'SHIFT-G', were used to move to the cxternal world and to different views of the inside world. Information about features in the unit was available by mouse clicking on related objects, such as windows and lighting. Figures 5.1 to 5.5 show the external world, the different inside views of the unit and an example information box.

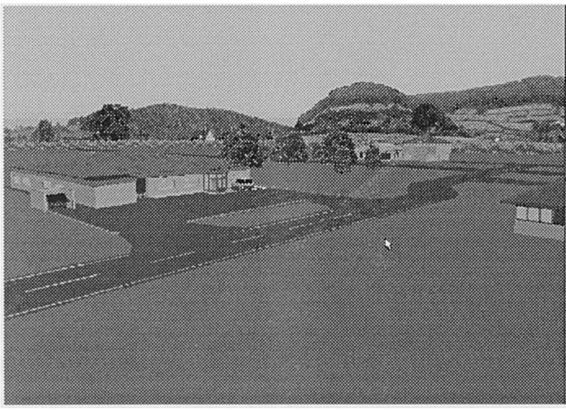


Figure 5.1 (left): The external world showing the outside view of the business park. The business unit represented internally is shown in the left of the picture.

Figure 5.2 (right): The inside world showing of one of the units in the park, as an empty complex.

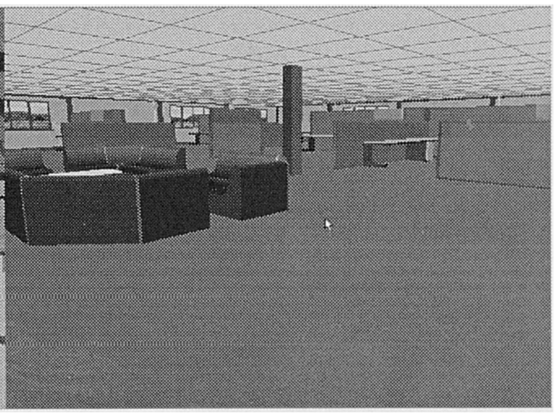


Figure 5.3 (left): The inside world showing of one of the units in the park, as a factory complex.

Figure 5.4 (right): The inside world showing of one of the units in the park, as an office complex.

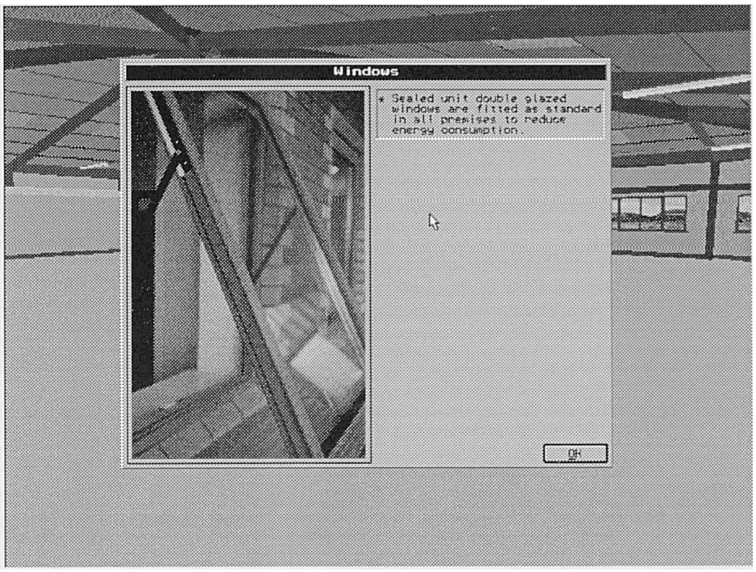


Figure 5.5: The inside world representing an empty complex. From clicking on one of the windows, an information box is displayed describing the double-glazed windows in the unit.

Changes were made to the application to ensure it allowed for the range of behaviours to be evaluated in the theory. The application lacked any aspect of system control. Therefore, an automatic guided tour was introduced to show the user around the external world. There was only one system event in the application, therefore two more were added - a speech text appearing from a man (when the user approached), and a telephone ringing. The application had a very basic user representation (mouse pointer only). However, technical constraints with the application platform prevented a more sophisticated user representation being included. Alterations made to the application followed general styles that had been observed in other VEs, for example the style of speech text used.

The application, with the above changes, will now be referred to as the *original* version of the test application. This version was assessed, using the theory of interaction, and predictions were made for likely usability problems, as described later in section 5.3.1. A second, *amended* version of the application was produced by implementing some of the missing generic design properties, identified in the assessment (described later in section 5.4.1). Both the original and amended versions of the test application were used in the following empirical studies. The versions were run on a PC with a 21inch monitor, a joystick was used for navigation and a standard 2D mouse for interacting with objects.

5.1.2 Subjects and groups

Three groups of subjects were involved in the studies: a Control group, a Domain group and a GDP group. The control group was given the original application to interact with. The domain group was given the original application and additional information about the application domain (see appendix 5D). The GDP group was given the amended version of the application. Pre-study questionnaires (see appendix 5A) were used to select subjects and balance them across groups according to sex, age and experience in direct manipulation interfaces, video games, virtual reality systems, and property evaluation (the experiment task). Twenty-nine subjects, eighteen male and eleven female, with an age range of 21-54 years, took part in the studies (excluding pilot subjects). The subjects were staff and students at the School of Informatics, City University, and were paid £10 for participating in the studies. Three subjects were unable to attempt all tasks and were therefore excluded from the groups. These subjects either lacked the spatial skill required to navigate effectively in the VE or, had to stop the experiment, complaining of nausea. The remaining subjects in each group were as follows: Control group – 8 subjects, Domain group – 8 subjects and GDP group – 10 subjects.

The domain group was set up to test the effect of an additional knowledge source on interaction and usability problems encountered. However, subjects did not use the domain information effectively to help complete tasks, although the information was relevant, and some subjects did not use it at all. Subjects got involved with the VE and forgot about the domain information, or found it difficult to refer to external documents whilst interacting. This was unexpected, since there was nothing in the literature to indicate the possible severity of this problem for desktop VEs. Therefore, there was little difference between the control and domain groups in interaction and performance. The few areas of difference related to the map of the park and unit, which was sometimes used to guide or prompt exploration or exhaustive searching. This appeared to result in the domain group spending longer on some tasks. In general, little insight was gained into the effect of the *domain* knowledge source but, since this was not a major component of the theory, no further attempt was made to test it. However, the interaction sessions of subjects in the domain group provided useful data to evaluate other components of the theory.

To gain data on interaction behaviour, the interaction sessions of 10 randomly selected subjects, from the control and domain groups, were used. For data on usability problems encountered, the interaction sessions of all 16 subjects from the control and domain groups were used. To assess the effect of the GDPs on interaction success, subjects in the control group were compared with those from the GDP group.

5.1.3 Tasks and experiment

The task scenario that subjects were given involved an overall goal to gather information about the architecture and basic services of the site represented in the VE, as if they were sales people who had to explain the site to potential leaseholders. Subjects were told that, following the experiment, they would be questioned on the site. A range of specific tasks tested all aspects of the theory, such as exploration, targeted searches, actions and object investigation (see appendix 5B). Subjects were given 10 minutes to explore and familiarise themselves with the VE. There were then eight set tasks, with no time limits for individual tasks or overall. Two of the tasks involved finding and investigating objects, namely the windows and the water tank. For three tasks, subjects were asked to carry out specific actions - open the loading bay door, switch on power and tilt the drawing board. For the other three tasks, they carried out general analysis and problem solving – finding areas with a special floor covering, investigating provisions for power sockets and comparing the three toilets in the building (disabled, men's and women's).

A pilot study was first undertaken to refine the experimental procedure. In the experiment, subjects were asked to provide a concurrent, 'think-aloud' verbal protocol (as defined by Ericsson and Simon, 1984). They were first given a training task to practice 'thinking aloud', which involved carrying out mathematical calculations using the Calculator package in Windows. For the interaction session, subjects were provided with basic interaction notes, such as how to use the joystick (see appendix 5C) and, if in the domain group, were provided with the application domain notes (see appendix 5D). Subjects began their interaction session by first completing the exploration phase. They then carried out all the eight set tasks. The interaction sessions were video-recorded. When subjects had completed the tasks to their satisfaction, which typically took 40 minutes, they completed a paper memory test on the site (see appendix 5E), then took part in a de-briefing session with the experimenter. The memory test consisted of structured questions to ascertain what relevant information subjects had gained during interaction. The de-briefing session was used to clarify any points about the interaction session, such as reasons for actions carried out which had not been given in the verbal protocol. Finally, subjects completed a retrospective questionnaire eliciting their views about the interaction session (see appendix 5F).

A few minor amendments were made to the experiment material, for the GDP group, to account for general changes in the amended version of the test application. For example, questions were included in the retrospective questionnaire on the helpfulness of the added cues and support information, and a comment was included in the interaction notes to say that the main interactive objects would be highlighted.

The empirical studies were used to evaluate three major components of the theory, namely the models of interaction, the problem prediction rules and the generic design properties. The following sections describe thc cvaluation of each of these.

5.2 Evaluation of the models of interaction

5.2.1 Study method

To evaluate the models of interaction, 10 subjects were selected from the control and domain groups, for data on interaction behaviour. The subjects were randomly selected from those who gave rich enough verbal protocols (i.e. where the protocols were audible and comprehensive). A selection of four tasks was analysed for each subject, because of time constraints for carrying out the data analysis, which meant it was impractical to completely analyse subjects' interaction sessions. The exploration phase was analysed along with one object investigation task, one action task and one analysis task. The number of times each of the different set tasks was analysed was balanced overall and, within this constraint, tasks were randomly selected for each subject. Subjects' interaction sessions selected for analysis ranged from 16 to 35 minutes (mean 23).

Verbal protocols from selected tasks were transcribed, noting concurrent physical and system behaviour. Speech segments were matched to mental behaviours using general verbalisation categories, which were commonly found in the literature (e.g. Ericsson and Simon, 1984) and therefore independent of the theory. The categories were refined in a preliminary analysis of speech segments. The major categories were:

observations - comments describing the observed environment or assertions giving reasoning and interpretations about the observed environment (e.g. from subject A - 'ok windows straight ahead', 'erm no window handles').

observations - changes - comments describing observed changes in the environment resulting from system feedback after an executed action, or system behaviour (e.g. from subject L - 'here I'm I seem to be being taken through it', 'I err yes its a drive-through').

objectives - expression of goals or planning of the experiment task (e.g. from subject I - 'let's find the drawing board', 'oh I'll have to give up on that task I can't do it').

intentions - verbalisation of intent to do some specified, low-level action (e.g. from subject B - 'err open the door', 'go in').

problem reports - comments describing problems faced, such as being unable to understand the current situation, or making slips (e.g. from subject K - 'oops', 'bash into the wall a bit').

problem solving - verbalisation of mental activity in considering a task or interaction problem, such as asking questions, developing and testing hypotheses (e.g. from subject R - 'maybe the switches are in here it's a bit unlikely', 'I don't think you have power switches in the boardroom').

reading documentation - verbalisation arising from subject reading aloud and taking information from the provided documentation, such as task descriptions (e.g. from subject M - 'OK task three analysis and problem solving', 'right so main building find out what areas of the building have a special factory floor covering').

For physical behaviour, four categories were defined to cover the range of relevant physical operations:

movement - navigating around and thereby changing the current position in the environment.

adjusting view angle - altering the angle of view whilst maintaining a stationary position, such as tilting the view angle down or heightening it

interacting with objects - mouse-clicking on objects to interact with them

executing commands - executing commands not directly related to objects in the environment, such as pressing 'SHIFT-H' to change worlds or 'F12' to reset the world

The categorisation of verbal protocols was validated through cross-marking by two independent observers. Each observer allocated a verbalisation category to each utterance in a 10-minute transcript from one of the interaction sessions. An inter-observer agreement of 83% was reached in categorisation for the 117 utterances in the transcript. Resulting differences in categorisation were discussed, and where necessary, changes were agreed by both observers.

Rules, represented in decision trees, were defined for matching the verbalisation and physical behaviour categories to interaction stages in the theory. This was a relatively novel approach in analysing protocols which was employed to make the matching of verbalisations to theory elements as objective as possible. The tree for the 'observations - changes' category is shown in figure 5.6. There are five possible theory stages, from the task action and system initiative modes, which can be matched to this category, according to the type of change commented on. The decision trees were systematically applied to the data by selecting the appropriate tree, for a category, then checking the required conditions, from top to bottom, until either a theory stage was reached or no set of conditions were met, in which case an additional behaviour was noted and described. For example, table 5.1 shows an extract of the data for subject A during the window investigation task. In the second row, the verbalisation is categorised as an observation of changes and matched to the 'feedback' stage of task action mode, using the tree shown in figure 5.6.

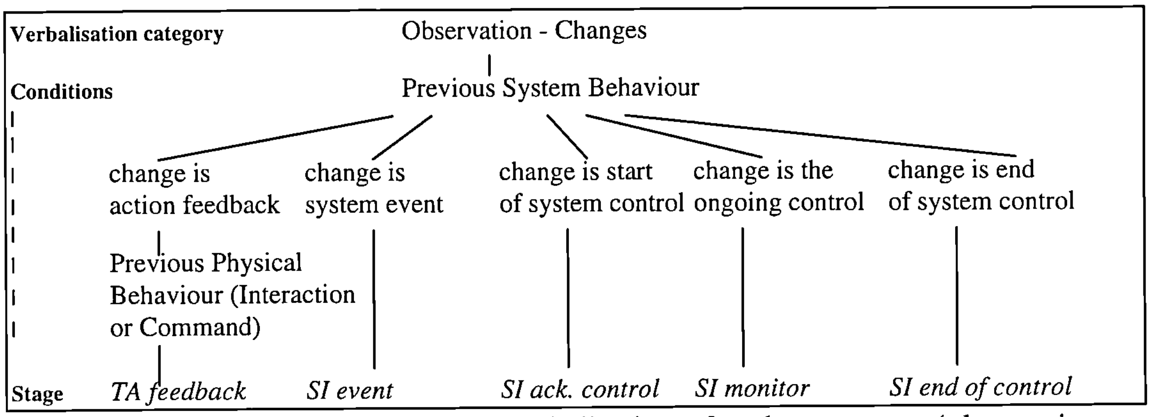


Figure 5.6: Decision tree to match verbalisations for the category 'observations - changes' to a possible five theory stages, from the task action (TA) and system initiative modes (SI).

verbalisation	category	physical behaviour	system behaviour	stage(s) of interaction
well I'll just click on	intention	interacting with object - window	window information	TA intention task action TA execute
ok so its a sealed unit double glazed windows fitted as standard in all units to reduce energy consumption	observation - changes			TA feedback

Table 5.1: An extract of the data for subject A during the windows investigation task. In the first row, the verbalisation instance is categorised as an intention and matched to the 'intention task action' stage of task action mode. The concurrent physical behaviour is categorised as an object interaction and matched to the 'execute' stage.

In the second row, the verbalisation is categorised as an observation of changes and matched to the 'feedback' stage of task action mode.

Later, all noted, additional behaviours which were outside of the theory were refined into a set of additional stages of interaction, by grouping together similar behaviours. For example, there were verbalisations describing the subject's understanding of changes in the current position/orientation of the view as a result of navigation (e.g. 'OK now I'm back at floor level', from subject B), which were grouped as the additional stage 'interpret navigation feedback'.

The matching of interaction stages was also validated through cross-marking by two independent observers. Each observer allocated a stage of interaction to each verbalisation instance and physical behaviour in a 10-minute transcript from one of the interaction sessions. An inter-observer agreement of 81% was reached (agreement in matched stage for 126 of the 156 verbalisation instances or physical behaviours in the transcript). Resulting differences between matched stages were discussed and, where necessary, changes were agreed by both observers.

Therefore, observed instances of interaction behaviour were categorised and either matched to a stage in the theory, when all required conditions were met, or classified as an additional stage of interaction. Data on these observed stages consisted of sequences of stages that occurred for each task analysed for the subjects. The data on observed sequences was entered into a database, which was queried to find the number of times each theory and additional stage occurred, and the number of times transitions between different stages occurred (i.e. the number of times one stage was followed by another). Specific hypotheses were set to test predictions in the interaction models. The interaction models predicted modes of interaction behaviour, stages of interaction and the general flow of interaction between stages. The models aimed to predict the general organisation of stages in each mode, rather than precisely which stage would follow which for any interaction instance. Therefore, the skipping of a stage or returning back to the previous stage was deemed to be acceptable within the models' predictions. To test the existence and cohesiveness of the three modes of behaviour (task action, explore navigate and system initiative), it was hypothesised that in an interaction session:

Hypothesis 1a: There will be significantly more stage-to-stage transitions within mode boundaries, than across different modes.

Hypothesis 1b: Observed sequences of up to five stages long will remain within the boundaries of each mode. *(Five being a reasonable sequence length, beyond which there was not necessarily enough data for reliable significance testing.)*

To test that the 21 stages of interaction in the models together describe important behaviour, it was hypothesised that in an interaction session:

Hypothesis 2: Theory stages will occur significantly more times than any additional stages of interaction identified.

To test that the interaction models represent the generalised pattern of interaction flow, it was hypothesised that in an interaction session:

Hypothesis 3: Observed stage transitions, within each mode, will conform to patterns predicted in the models. More specifically, observed stage transitions will be either exactly as predicted, or jumps forward or backtracks of one stage in the interaction models.

To test for significance, chi-squared and binomial tests were used to compare observed frequencies of events (stage occurrences and stage transitions) against those expected by chance.

5.2.2 Results

5.2.2.1 Modes of interaction

Table 5.2 shows the number and percentage of times each mode of interaction was followed by stages within each of the modes. For task action and explore navigate modes, 79% of stage-to-stage transitions stayed within the mode boundaries and 21% crossed to another mode. System initiative mode had only 51% transitions being within-mode, 22% transfers to task action mode and 27% transfers to explore navigate mode. The number of transitions staying within-mode were found to be significant at $p<0.01$, for task action and explore navigate modes, using a chi-square test with a 50% chance of a within-mode or between-mode transition. Therefore, for these two modes, there were significantly more transitions within the mode boundaries than could have been expected by chance. Therefore, hypothesis 1a was not refuted for task action and explore navigate modes, but rejected for system initiative mode.

To / *From*	Task action		Explore navigate		System initiative		Total
Task action	1594	79%	388	19%	31	2%	2013
Explore navigate	352	19%	1461	79%	41	2%	1854
System initiative	37	22%	44	27%	85	51%	166

Table 5.2: Total number and percentage of stage-to-stage transitions within mode boundaries (shaded cells) and across different modes, for all subjects' analysed tasks. *(Data only includes transitions between the theory stages.)*

Table 5.3 shows the number of within-mode stage transition sequences, by length of the stage chains. The data is reported as a survivorship function, so of the 1594 transitions that were observed between stages in task action mode (in table 5.2), 1162 progressed to three stage chains, 891 progressed to four stages and 712 of the originating sequences progressed to five stages all remaining within task action mode. Other sequences either terminated or crossed a mode boundary, so the percentages express the number of sequences that remained within the mode compared with the total observed at that length. For example, 50% of the five-stage chains, beginning in task action mode, remained within the mode. Expected values for the stage transition sequences were calculated from the total observed frequencies at each length, multiplied by the probability for the stage combinations (i.e. cumulative probability = 0.25 *(0.5*0.5)*, 0.125, 0.06 at lengths three, four and five respectively). The observed values for all lengths, for each of the modes, were significant at $p<0.02$ (chi-square). Therefore, the data showed a pattern of staying within mode boundaries for up to five consecutive stages. Hence, for all three modes, hypothesis 1b was not refuted.

Length of stage chain	Task action		Explore navigate		System initiative		*Minimum N for significance testing*
3	1162	65%	1055	65%	56	34%	*100% = 48*
4	891	56%	811	55%	40	24%	*100% = 83*
5	712	50%	677	47%	34	20%	*100% = 160*

Table 5.3: Number and percentage of stage transition sequences that remained within the boundaries of each mode, by length of stage chain. For each mode, there were a sufficient number of stages for reliable significance testing. The minimum required stages for significance testing were calculated from a formula given by Bakeman and Gottman (1986: p. 137). *(Data only includes transitions between the theory stages.)*

5.2.2.2 Stages of interaction

Table 5.4 shows the number of times each theory stage occurred. Eighty-six percent of observed stages could be attributed to predicted stages in the theory. The remainder were 24 identified additional categories of stages of interaction, listed in appendix 5G. A chi-squared test of the total occurrence of theory versus additional stages, compared to 50% chance levels, was found to be significant at $p<0.01$. Therefore, hypothesis two was not refuted since the theory stages were found to account for significantly more observed stages of interaction, than could have been expected by chance. Indeed, this was a general pattern, since theory stages accounted for the majority of observed stages (between 80 and 90%), for every subject and for each of the different tasks.

The predominant stages, which each accounted for at least 5% of the total, were six stages in task action and explore navigate modes, as highlighted in table 5.4. Together these stages accounted for 71% of the total number of observed stages for all subjects. Uncommon stages, which each accounted for less than 1% of the total, included seven theory stages - 'consider objects' and 'intention explore action' from task action and explore navigate modes, and five stages from system initiative mode. Five of the 24 additional stages identified were reasonably common in that they each accounted for at least 1% of the total. Indeed, the seventh highest occurring stage was the additional stage, 'interpret navigation feedback' (4% of total). The others were - forming an intention to execute a command (such as switching worlds); scanning and inspecting an area of the environment; forming an intention to approach an object; and considering where an object might be located (see appendix 5G).

Theory Stages	**Total occurrence**	**% of all stages**	**Number of subjects**
execute	1001	18.36	10 (all)
feedback	446	8.18	10
approach/orient	345	6.33	10
intention task action	144	2.64	10
establish goals	135	2.48	10
evaluate	115	2.11	10
inspect	68	1.25	10
deduce sequence	56	1.03	9
consider objects	20	0.37	6
TOTAL task action	**2330**	**42.73**	
navigate	1433	26.26	10
plan	340	6.24	10
scan	302	5.54	10
explore	97	1.78	10
intention explore action	43	0.79	8
TOTAL explore navigate	**2215**	**40.62**	
monitor	67	1.23	10
intention control action	54	0.99	9
event	30	0.55	6
acknowledge control	15	0.28	8
end control	4	0.07	4
plan	2	0.04	1
intention reactive action	1	0.02	1
TOTAL system initiative	**173**	**3.17**	
TOTAL theory stages	**4718**	**86.52**	
OVERALL TOTAL	**5453**		

Table 5.4: Occurrence totals for each of the theory stages. The percentage of observed stages falling into each stage is also given. The predominant stages (accounting for at lcast 5% of the total) are highlighted.

5.2.2.3 Flow of interaction

Figures 5.7 to 5.9 show the original models with the observed, common stage transitions for each mode. A cut-off point was used to determine common stage transitions, as suggested by Bakeman and Gottman (1986). Common stage transitions were taken to be those where at least 10% of transitions from the first stage led to a second stage. This resulted in a manageable range of one to four stages that followed any one stage. Flows of interaction in the predicted and observed models were compared. Significance testing was not used for individual flows of interaction because this level of precision was not felt to be appropriate and there was not always enough data for reliable testing. Hypothesis three, on interaction flow, focused on transitions within the models, rather than transfers between modes, so the latter are only briefly discussed. There was very little transition from theory stages to the additional stages (e.g. only one transition to an additional stage above the 10% cut-off point), so the additional identified stages are not included.

Comparing the models in figure 5.7 shows that nine of the 13 within-mode transitions were either as predicted or were jumps forward or backtracks of one stage in task action model (see black arrows). For example, the observed transition from 'establish goals' to 'intention task action' was predicted, but there was a jump from 'intention task action' to 'approach/orient' skipping the 'consider objects' stage, and a backtrack from 'feedback' back to 'execute'. Therefore, hypothesis three was only partially supported for task action mode, since four observed transitions did not follow the general predicted flow of interaction in this model (see grey arrows). Three of these four transitions were jumps or backtracks of more than one stage. The other transition, from 'inspect' to 'execute', crossed paths in the predicted model. Other differences were mainly with the less prevalent stages, which did not have any common incoming transitions. For example, there was no iteration back from the 'evaluate' stage to 'establish goals'. Transfers to explore navigate mode (see block arrows) occurred from five other stages, as well as the predicted 'consider objects' stage. Finally, the three predominant stages in task action mode were linked together, as highlighted in figure 5.7.

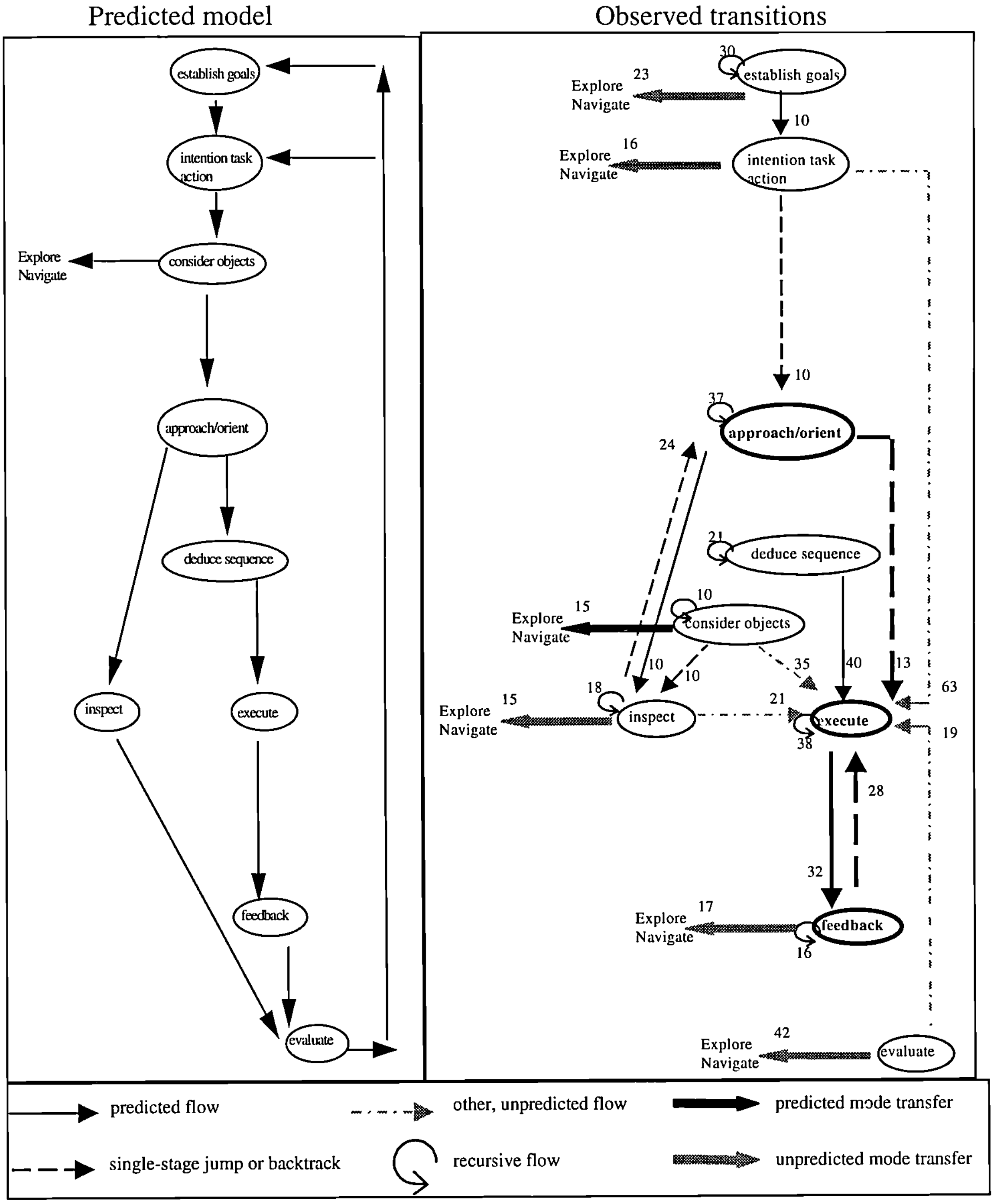


Figure 5.7: Interaction flow diagrams for task action mode. The left diagram shows the predicted model and the right diagram gives the observed common transitions for stages in this mode. Numbers give the percentage of transitions from the starting stage that led to the second stage. Black arrows show predicted flows and single-stage jumps and backtracks. Grey arrows show other, unpredicted flows.

(Numbers may not add to 100% because of uncommon transitions (<10%) scattered between other stages, and recursive flows repeating the same stage. Numbers for the mode transfers give the sum of the common transitions to stages in that mode)

Comparing the models in figure 5.8 shows that four of the five within-mode transitions were either as predicted or were jumps forward of one stage in explore navigate model. For example, the observed transition from 'plan' to 'navigate' was predicted and there was a jump from 'scan' to 'navigate'. Therefore, hypothesis three was partially supported for explore navigate mode, since only one observed transition did not follow the general flow of interaction in this model. This remaining transition was a jump of two stages from 'explore' to 'navigate'. No common transition was found from 'scan' to 'plan', and the less prevalent 'intention explore action' stage did not have any common incoming transitions. Transfers to task action mode did occur from 'intention explore action' but also from 'explore', and no common transfer was found from the 'plan' stage. The three predominant stages in explore navigate mode were closely coupled, as highlighted in figure 5.8.

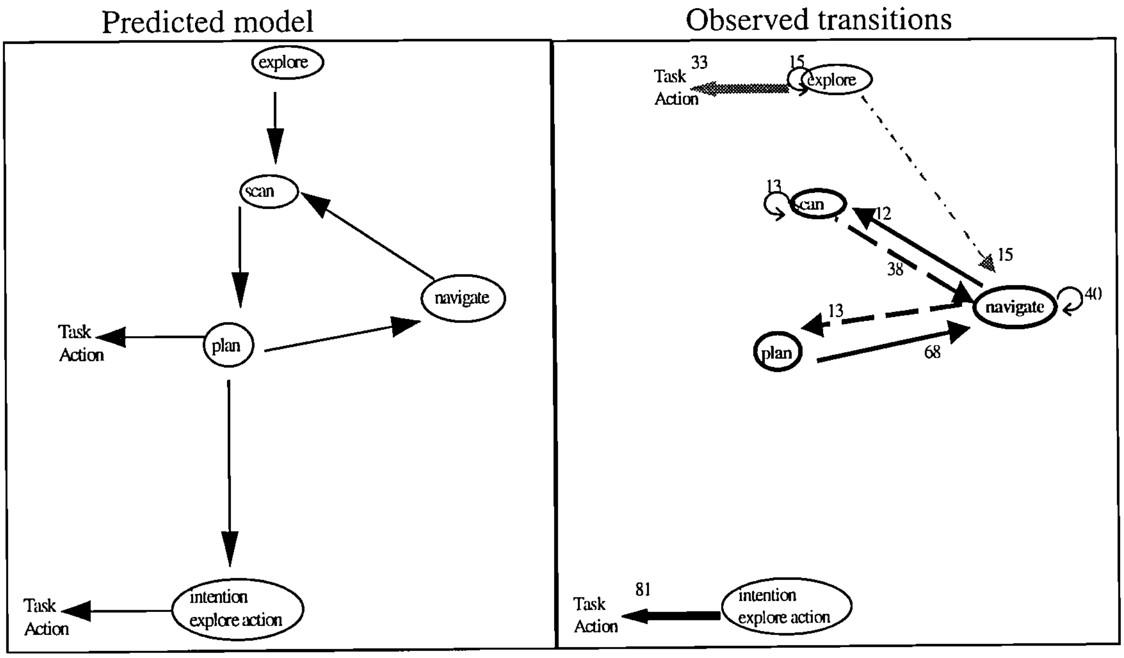


Figure 5.8: Interaction flow diagrams for explore navigate mode.

Comparing the models in figure 5.9 shows that all four within-mode transitions were either as predicted or were jumps forward or backtracks of one stage in system initiative model. For example, the observed transition from 'acknowledge control' to 'monitor' was predicted and there was a jump from 'acknowledge control' to 'intention control action'. Therefore, the results appeared to support hypothesis three for system initiative mode. However, there was a lack of flows linking stages in this mode because, as before, the less prevalent stages, of which there were five in this mode, had no common incoming transitions. Transfers to task action and explore navigate modes occurred from 'intention control action' and 'intention reactive action' stages, but the 'plan' stage was skipped and instead transfers made directly from the 'event' and 'end control' stages.

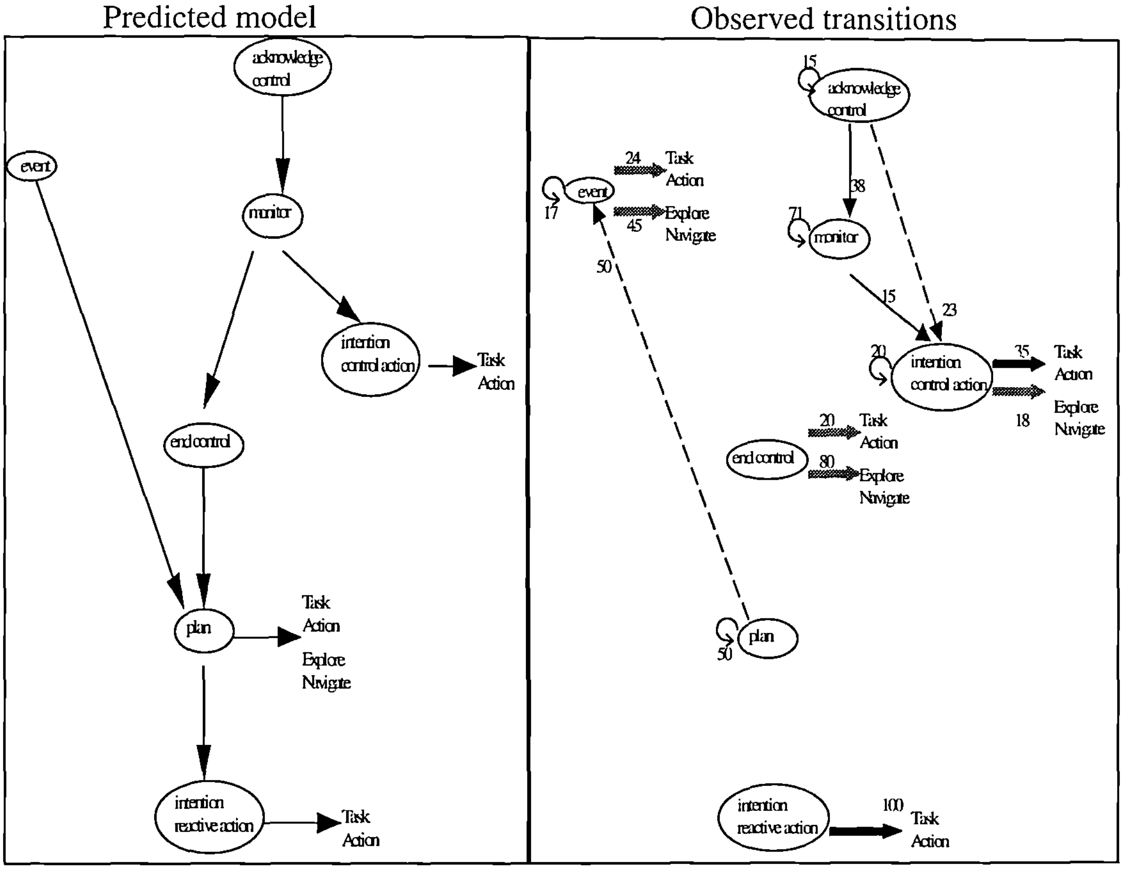


Figure 5.9: Interaction flow diagrams for system initiative mode.

5.2.2.4 Differences by task

Differences in interaction behaviour according to the task being undertaken were investigated for patterns of interest. Results for each task were compared with the average across all tasks (i.e. using a first-order model). Different modes of interaction were found to be more prevalent in different tasks. Task action mode occurred more commonly in the set tasks, especially those involving object manipulation actions and the more complex tasks. For example, for the task of tilting the drawing board, stages in task action mode accounted for significantly more behaviour than would be expected using averages for the tasks overall ($p<0.01$, binomial distribution). Explore navigate mode occurred more commonly in tasks involving a large amount of searching or investigation. For example, in the task of finding and investigating the water tank, stages in explore navigate mode accounted for significantly more behaviour than would be expected using averages for the tasks overall ($p<0.05$, binomial distribution). The first three planning stages of task action mode were not important in exploration tasks, but the others were important for trying out exploratory actions. For example, during the exploration phase, the 'establish goals' and 'intention task action' stages accounted for significantly less behaviour than would be expected using averages for the tasks overall ($p<0.01$, binomial distribution). Similarly, the 'explore' stage, in explore navigate mode, was not important for set tasks.

The additional stages of interaction identified were more prevalent in different tasks. 'Intention to execute a command' was more common where more switching between worlds or set positions was involved, such as in the exploration task, indicating that these were probably the main commands used. 'Interpret navigation feedback' did not follow the patterns for 'navigate' itself, but appeared to be more common at the beginning of an interaction session, when learning navigation, and during the more difficult navigation tasks, such as for the task of comparing the toilets. 'Scanning and inspecting an area' was more common for analysis tasks, where whole areas of the environment, rather than single objects were involved, such as for the toilet task. 'Consider the location of a target' was more common for tasks that involved searching for a target, which may have been difficult to locate, such as the water tank task. 'Intention to approach a target' was generally closely linked to the approach stage.

5.2.3 Discussion

The results provide some support for the models of interaction and indicate required refinements. Support was found for the three modes of interaction, which were found to represent coherent sequences of interaction behaviour. Task action and explore navigate modes differed in importance according to the type of task involved, such as exploratory or object manipulation tasks. However, very little of the interaction was accounted for by stages in system initiative mode, and this mode was found to be less cohesive than the others. This was because the test environment provided limited system behaviour. Therefore, system initiative mode may be more application dependent than the others, and may be unimportant for applications involving very little system behaviour. However, for other applications, such as guided training environments or those involving virtual agents, system initiative mode is likely to be an important part of interaction.

The stages of interaction in the models accounted for the vast majority of behaviour. Six frequent and closely linked stages may be considered as the predominant or core model. These included all the physical behaviours 'navigate', 'execute' and 'approach/orient'. The 'navigate' stage appeared to be almost a default behaviour which subjects would return to, for example after carrying out actions. The extremely high occurrence totals for 'navigate' and 'execute' can be partially explained by the fact that, in the data analysis, physical behaviours were invariably detected from the videos, but not all subjects' thoughts were verbalised because of errors of omission (Russo et al., 1989).

Some theory stages were uncommon and may indicate areas where the models are weak or need to be simplified:

- Three stages within system initiative mode were especially uncommon ('end control', 'plan' and 'intention reactive action'), occurring less than five times each. This seems to be due to subjects preferring only certain responses to the limited system behaviours. For example, the 'end control' stage was uncommon because subjects often exited the guided tour before it reached its end, and subjects often did not answer the ringing telephone for the 'intention reactive action' stage to occur. Subjects often ignored the phone, or could not find it, or, sometimes, attributed the ringing to phones in the real world! Therefore, some types of response to system behaviour may be less important, but further investigation, using a wider range of system behaviours, would be needed to ascertain this.

- In explore navigate mode, the 'intention explore action' stage was expected to occur more often since it was a key part of the model, predicted for exploratory and opportunistic actions. Subjects may have had limited opportunity for exploratory action because, during the exploration phase, they were busy learning how to navigate and familiarising themselves with the environment. Exploratory actions were recorded, however it may be that subjects did not always verbalise specific intentions to carry them out, because there was little conscious planning and they were more inclined to immediately try out the action. Therefore, although the 'intention explore action' stage was uncommon, it seems exploratory actions should have a place in the model.
- Finally, in task action mode the 'consider objects' stage was found to be a less important behaviour because, for most intended actions, little consideration of objects involved was required.

Some common additional stages of interaction were found that perhaps should be incorporated in the theory. The situations in which the additional stages were particularly important were investigated. The most common was 'interpret navigation feedback' which occurred when subjects were learning the navigation technique or had encountered problems in navigation. This behaviour may have been particularly common with the test application because, often during the guided tour subjects tried to navigate for themselves, but found they were not moving in the intended direction. The next most common additional stage was 'intention to execute command' which was similar to the lesser common 'intention to open navigation access (e.g. door)' (see Appendix 5G). These stages involved intentions to carry out actions for moving through the environment, for example, transporting to other worlds or set positions, or opening doors to move into rooms. Although the test application provided commands for moving between worlds/views, which may not be typical, actions for moving to set positions and opening doors are more common in VEs and need to be considered. The other common additional stages indicated three further areas that need to be covered by the models.

Partial support was found for the predicted flow of interaction in the models. The general organisation of stages in the models was found to hold reasonably well, but there were jumps forward, omitting stages, and backtracking to previous stages. Predicted flows to the more uncommon stages were not found, due to the lack of data available. Therefore, interaction flow with the less common stages could not be thoroughly validated. Stage omissions were expected for automated actions in skilled behaviour, and backtracking when re-trying stages after error or repeating stages for multi-operation tasks. However, the models can be refined to be more representative by including the major jumps and backtracks.

Therefore, support has been found for hypothesis 2, stated in the thesis objectives:

> **H2** General patterns of interaction with VEs can be predicted, through theoretical models.

The results show that the models of interaction can predict major aspects of interaction with VEs. Core parts of the models have been identified and section 5.6.1 gives details of the refinement of the models, using the study results.

Having gained support for the models of interaction, the other major components of the theory, derived from the models, were evaluated, beginning with the problem prediction rules.

5.3 Evaluation of the problem prediction rules

To evaluate the problem prediction rules, the interaction sessions of all 16 subjects from the control and domain groups were investigated for data on usability problems encountered. Actual problems encountered were compared with predictions for likely usability problems, made from assessments of the test application.

5.3.1 Assessment of application using the theory

The test application was assessed for the presence of the generic design properties. Assessments were made for general properties, i.e. GDPs in the user task, overall environment, spatial layout, and viewpoint and user representation categories. The main objects, actions and system behaviours were individually assessed. For each element assessed, the test application was inspected and each relevant GDP judged to be supported or not supported, using the GDP definitions (as given in appendix 4D). For example, table 5.5 gives an assessment of which GDPs, in the system behaviour category, were present for the automated drive-through.

Relevant GDP	**Supported**	**Comments**
declared system control commencement/ termination	✗	No information to indicate that automated drive is starting. No information to indicate when automated control is over.
clear system control purpose	✗	No information about purpose of drive and whether it is continuous or will stop and, if so, how long it will take.
declared available actions during control	✗	No information to say cannot navigate during drive. No information to say that function keys are available to stop the drive.
limited system control	✗	Drive continues for two minutes every time start external world. No information given about how to stop the drive if want to.
distinguishable behaviour	✓	
declared causality and effects of behaviour	✓	
clear system activity significance	✗	No information given about purpose and, therefore, importance of drive.
appropriate response to system activity	✓	

Table 5.5: GDP assessments for the automated drive-through system behaviour.

General assumptions were made about what knowledge the average subject would have:

- *Real World* knowledge was assumed to be present.
- *Task* knowledge would be present in the documentation provided to subjects.
- Knowledge of *Other Environments* was generally assumed not to exist, since most subjects (15 of the 16) had little or no experience with VEs in general.
- Knowledge in a current *Environment model* was assumed not to exist because no subjects had previous experience with the test application.
- *Domain* knowledge was assumed not to exist, unless subjects were in the Domain group, which was given information about the business park.

Predicted problems	**Cause**
65 difficulty realising commencement of system control	GDP *declared system control commencement/ termination* not supported
66 difficulty understanding goal of system in taking control of interaction	GDP *clear system control purpose* not supported and no relevant user knowledge
69 difficulty determining whether/ what user control can be exercised	GDP *declared available actions during control* not supported and no relevant user knowledge
70 difficulty deciding whether to investigate exercising user control	Relevant GDPs not supported e.g. *declared available actions during control* and no relevant user knowledge
71 difficulty realising end of system control	GDP *declared system control commencement/ termination* not supported
73 difficulty deciding appropriate response to system activity	Relevant GDPs not supported e.g. *clear system activity significance* and no relevant user knowledge
37 difficulty assessing implications of system behaviour for goals and intentions	Relevant GDPs not supported e.g. *clear system activity significance* and no relevant user knowledge
61 distraction from original task goals and intentions	GDP *limited system control* not supported

Table 5.6: Problems predicted for the automated drive-through system behaviour. *(Problem numbers refer to relevant correspondence rules, listed in appendix 4E.)*

Generalised predictions about likely usability problems were made by inspection of the correspondence rules in the theory (see section 4.5), using the assessments of the application and expectations about likely user knowledge. For each element of the test application that had been assessed, there could be a number of unsupported GDPs. Where these unsupported GDPs were referred to in correspondence rules (i.e. where the GDP was predicted to be linked to a usability problem), the usability problem in question was predicted to be likely, unless corresponding user knowledge was expected to be available to compensate for the unsupported GDP. For example, GDPs in the task category were generally not well supported, but subjects were given clear task instructions, so problems understanding the task were not predicted to be likely. The GDP assessments incorporated a check on consistency with expected user knowledge, i.e. GDPs were judged to be supported if the design provided required information and the information was consistent with expected user knowledge (i.e. general real world knowledge and information in documents provided). This simplified the use of the correspondence rules, which included conditions for GDPs to provide information consistent with prior user knowledge. For example, table 5.6 gives problems predicted for the automated drive-through system behaviour.

5.3.2 Study method

Actual usability problems were noted from observation of video footage and subjects' concurrent verbal protocols, as recommended by Lauesen (1997). Usability problems were defined as critical incidents or breakdowns which interfered with the user's ability to efficiently and effectively interact and complete tasks (from Karat et al., 1992). For example, this would include not being able to understand the current situation, wanting to do something but being unable or not knowing how to, and making slips.

Observed & predicted problems were grouped by the environment element involved (e.g. particular object, action, etc. of interest). Observed problems were found for a total of 62 different elements. Only 22 of these elements had been analysed to predict problems (see appendix 5H). However, this subset covered all the major environment elements, such as navigation and the objects involved in subjects' tasks, and accounted for over 70% of observed problems. For these 22 major elements, observed problems were matched with predicted problems using a scheme based on matching the difficulty, context and cause (following advice of Lavery et al., 1997). These three aspects were captured in the correspondence rules as the predicted problem, stage of interaction and missing GDP or user knowledge, respectively. The matching of problems was validated through cross-matching by two observers. Explanatory discussion was used prior to independent matching, because of the amount and complexity of both data and theory knowledge involved. Each observer allocated a match or non-match judgement for each combination of observed and predicted problems for one of the environment elements. An inter-observer agreement of 97.5% of the combinations was reached (agreement on match or non-match judgement for 195 of the 200 problem combinations for the test element). Resulting differences between matched problems were discussed and reconciled.

Data on observed and predicted problems for each element, and the matches between them, were analysed to test specific hypotheses on the problem predictions:

Hypothesis 1: Significantly more of the observed problems will be predicted than non-predicted (i.e. matched to at least one predicted problem for that element).

Hypothesis 2: Significantly more of the predicted problems will be observed than not observed (i.e. matched with at least one observed problem for that element).

To test for significance, binomial tests were used to compare observed frequencies of problem matches against those expected by chance.

5.3.3 Results

5.3.3.1 Observed problems

There were 351 different observed problems with the elements investigated in the original application, for all subjects. Table 5.7 shows how the observed problems were assessed. Of the 351, 249 were matched to at least one predicted problem (significant at $p<0.01$, binomial distribution and 50% chance of a predicted match). For example, the observed problem 'clicking instead of dragging the drawing board handle' was matched to the theory problem *difficulty determining how to execute action*, predicted for the drawing board element. Therefore, hypothesis one was not refuted for the observed problems, since a significant number were matched to at least one predicted problem.

Assessment	**Count**
Predicted	249
Relevant rule, but not predicted	33
Unpredicted	69
TOTAL	351

Table 5.7: Overall assessment for the observed problems

There were 33 observed problems for which there was a relevant correspondence rule, but the problem had not been predicted to be likely when assessing the application. For example, the observed problem 'looking for the loading bay but not noticing it when in view' could match to the theory problem *difficulty determining whether target is in immediate vicinity*, but this problem had not been predicted to be likely for the loading bay element, when assessing the test application. The remaining 69 observed problems were unpredicted and were categorised according to the general difficulty involved. Thirty categories of unpredicted problem were defined (see appendix 5I). Twenty-two of these only occurred once or twice, many appearing to be unusual or minor incidents (e.g. being unclear about the plan a system control was following). Others included subjects expecting objects, system behaviours etc. to be available in the environment when they were not, or included subjects giving up on their current goal/intention. The more common categories (which accounted for at least 5 observed problem incidents) were:

- difficulties determining where the user could navigate,
- difficulties understanding the spatial structure of the environment,
- expecting actions to be available which were not, and
- difficulties determining what objects were present in the whole environment.

Additionally, one of the more common observed problems (which occurred more than 20 times) was not predicted:

- difficulty in observing the environment from an unsuitable viewing angle.

Number of matching predicted problems	**Frequency**
One	135
Two	89
Three	23
Four	0
Five	2
TOTAL	249

Table 5.8: The number of times the observed problems were matched with one, two, three, four and five predicted problems.

The number of predicted problems matched to any one observed problem ranged from none (unpredicted) to five, see table 5.8. Most observed problems, that were predicted, matched to one or two predicted problems. For example, the observed problem 'trying to flush the lavatory, when this action was not available' was matched to two predicted problems for this element: *difficulty determining whether further interaction possible* and *problems trying to execute action which does not exist.*

5.3.3.2 Predicted problems

Of the 75 possible problems in the correspondence rules, 50 were predicted to occur from the GDP assessments of the original test application. The problems were predicted for different environment elements and there were 164 specific problems predicted by element. Of these 164, 115 were observed for elements predicted, which is significant at $p<0.01$ (binomial distribution, with 50% chance of observing a predicted problem). For example, the theory problem *difficulty interpreting the event*, predicted for the talking man element, was matched with the observed problem 'not sure how and why the speech text appeared'. Therefore, hypothesis two was not refuted for the predicted problems, since a significant number were matched to at least one observed problem.

Each possible problem in the correspondence rules was assessed according to how many times it was observed and successfully predicted, see appendix 5J. Table 5.9 gives a summary of the assessment. Twenty-seven problems were observed for all predicted elements, for example the problem *difficulty assessing progress in action execution* was observed for both the switching worlds command and the drawing board elements as predicted. Twelve problems were observed for at least one predicted element, for example the problem *difficulty deciding appropriate response to system activity* was observed for the predicted drive-through element, but not for the predicted talking man element. Eleven predicted problems, for various stages of interaction, were not observed at all. For example, the problems *difficulty finding feedback not in immediate view* and *difficulty deciding whether to carry out action on target, investigate interesting feature or explore further*, were predicted to be likely, but such difficulties were not found. Five problems in the correspondence rules were not predicted to be likely with the test application, but were unexpectedly observed, at least once. For example, the problems *difficulty determining environment components involved in action* and *difficulty deciding areas of interest to investigate* were not expected but were observed. Twenty problems in the correspondence rules were not tested in this study since they were not predicted and were not observed. These problems were mainly task and user object related, such as *difficulty carrying out task* *as intended* and *difficulty determining self parts involved in posture/orientation change.*

Assessment	**Count**
problems observed for all predicted elements	27
problems observed for some, but not all, of the predicted elements	12
problems predicted to occur but not observed at all	11
problems not predicted to occur but observed	5
problems not predicted to occur and not observed	20
TOTAL	75

Table 5.9: Overall assessments for the 75 predicted problems.

The matching of predicted with observed problems was investigated further. Pairs of predicted problems were sometimes matched to the same observed problem more than once. There were 12 such pairs of predicted problems that appeared to be related, i.e. they appeared together at least five times (see appendix 5K). Some of the pairs of problems were from the same stage of interaction; for example, problems executing navigation and assessing progress in navigation were often matched together (*difficulty executing navigation effectively and efficiently* and *difficulty assessing progress in navigation* respectively). Other problem pairs were from different parts of the models that shared certain interaction issues. For example, there were similar problems determining the current view position/orientation (*difficulty determining current self position and orientation*), during both the object 'approach/orient' and 'navigation' stages. Some predicted problems were found to match with many observed incidents for any one element. There were seven such problems that appeared to be very general, i.e. they covered at least five incidents per environment element (see appendix 5L). These problems were mostly during common, more general activities in the models, for example problems locating a target and problems in navigation (e.g. *difficulty locating the destination in the environment* and *difficulty executing navigation effectively and efficiently*).

5.3.4 Discussion

The results show the correspondence rules to be predictive above chance level. Most observed problems were predicted and most predicted problems were actually found. There were some unpredicted problems which were reasonably common and indicate areas where the rule set may need to be improved. For example, there were unpredicted problems where subjects were not sure about where they could navigate and where not, and there were problems when subjects expected elements in the VE which were not available.

Eleven predicted problems were not found at all and need to be analysed further for possible reasons why they did not occur, such as the lack of relevant situations, and possible simplifications required to the rules. Twenty predicted problems were not tested, i.e. they were not predicted to occur and were not observed. These related to general task problems and problems with the user (self) object, which this evaluation study was unable to adequately investigate. About 10% of observed problems had a matching problem rule in the theory, but the problem had not been predicted to be likely with the test application. This indicates that improvements can also be made with the use of the theory in assessment of an application and determining whether or not a particular GDP is supported.

Although the rules predicted most problems, they did not always appear to describe problems appropriately, at the most suitable level of granularity, or duplicated the actual difficulties that existed. Observed and predicted problems often did not have a one-to-one mapping. Observed problems were more specific but still often matched to two predicted problems. However, such mismatch issues are likely to be inherent when abstracting problems and matching problems from different points in a causal chain of events, due to issues of granularity and causality respectively. Indeed, Hollnagel (1993) discusses distinctions between the manifestations of errors and their underlying causes (phenotypes and genotypes respectively). Others have also noted that problems can often fit more than one category and, therefore, some degree of judgement is required when categorising problems and perfect matches will be rare (Springett, 1996; Lavery et al., 1997). Additionally, some predicted problems appeared to be related in that they were often matched to the same observed problems. Therefore, some of the rules may duplicate possible difficulties, both within a stage and across different stages in the models, and these need to be analysed further for possible simplifications. Some predicted problems, during common activities, appeared to be too general in that they were matched to several observed problems for any one element. These problem rules need to be analysed further to determine whether the general difficulty ought to be split into several, more specific difficulties.

The results provide some support for hypothesis 3 of the thesis:

> **H3** Design properties required for interaction can be predicted using the general patterns.

The results show that the correspondence rules can predict most problems in interacting with VEs. Therefore, the basic assertion of the rules is supported, i.e. usability problems being caused by missing GDPs or user knowledge. Since there was limited relevant user knowledge, the results provide some support for the set of GDPs, which were defined as the requirements for avoiding usability problems. The prevalence of each predicted problem has been investigated and some additional usability problems identified. Sections 5.6.2 and 5.6.3 give details of how the correspondence rules and GDPs were refined from the study results.

Having gained support for the theory's ability to predict usability problems, the next phase of empirical evaluation involved assessing whether the theory could be used to improve interaction, by avoiding the problems.

5.4 Evaluation of the impact of the generic design properties

The generic design properties were further evaluated by assessing the effect that implementation of the GDPs would have on interaction success. The interaction and performance of subjects in the control group was compared with the GDP group, who had been given a second version of the test application with some of the missing GDPs implemented.

5.4.1 Implementation of missing generic design properties in application

Missing GDPs were implemented for a range of environment elements (objects, the self object, actions and system behaviours) that would be most commonly used, i.e. task elements and basic actions, such as navigation. Nineteen different environment elements, that had been assessed for the presence of the GDPs (see section 5.3.1), were then addressed by implementing missing GDPs (see appendix 5H). Additionally, some common, general amendments were made to objects and actions.

For example, for the automated drive-through system behaviour, missing GDPs were implemented by adding a speech track. The speech track included required information about the start of the drive and its termination (for GDP *declared system control commencement/ termination*), the purpose and length of the drive (*clear system control purpose*), and the user actions available during the drive (*declared available actions during control*). An on-screen message was also displayed, throughout the drive, to remind users about available actions. Figure 5.10 shows points in the drive with the current part of the speech track.

General amendments were made to distinguish active from inactive objects (for GDP *clear object type/significance*) and highlight available actions (*declared available action*). Interactivity was indicated by outlining active areas in bright red and white, or, using information signs as cues to actions which provided information about basic facilities (e.g. lighting). Figure 5.11 shows a section of the environment before and after this action highlighting was implemented.

Further examples of the implementation of the GDPs are shown in figures 5.12 to 5.20. In the amended version, objects such as walls sharing an edge were made more distinguishable by using textures to emphasise edges (for GDP *distinguishable object*); objects such as exit doors were made easier to identify by labelling them (*identifiable object*); and areas that the user could not navigate into were marked using 'no-entry' signs, which appeared on approach (*clear navigation pathways*). Navigation was made easier to execute (*executable action*) by adding collision detection on all walls, so the user could not accidentally fall through walls, and limiting the allowable distance from walls, so the user could not stand right up against a wall. The current tilt and height of the viewpoint was indicated through an on-screen figure (*clear self position/orientation*). Previously obscure power switches were highlighted (*declared available action*) and their link to factory machinery made meaningful by starting machinery when the switches were on and indicating this through a low-volume machinery sound (*declared action effect/success* and *clear action effect*). The event involving speech text appearing from a man, was made easier to see and understand by making the speech bubble more obvious and the text more explanatory (*distinguishable behaviour* and *declared causality and effects of behaviour*). The command of switching between worlds was made clearer by adding a speech track to say which world the user was being transported to, and giving the user a view of the main area, which was the room where the internal worlds most differed (*clear action progress* and *clear action effect*). Appendix 5M gives a complete list of amendments that were made to the environment elements addressed.

In implementing the abstract GDPs, a range of specific techniques could have been used. However, there were a number of practical constraints on the amendments possible to implement GDPs. An appropriate representation of the application domain had to be maintained, so amendments could not alter the layout and basic contents of the business park and business unit. Otherwise, departing from a purely natural representation and adding support information (e.g. 'no-entry' signs) was acceptable within the context of the test application. There were experimental constraints, for example, speech output had to be kept to a minimum to avoid disturbing subjects' verbal protocols; furthermore, a plan of the park and unit could not be added without affecting the fair comparison of understanding of the spatial layout, tested in the post-study test. There were technical constraints due to the hardware available and possibilities available with the development toolkit. For example, few changes could be made to the navigation style with the joystick and some desired amendments could not be implemented, such as an easy-to-approach function executed by double-clicking objects. Finally, there were time constraints, which made a complete re-design of the test application impractical. Within the above constraints, the

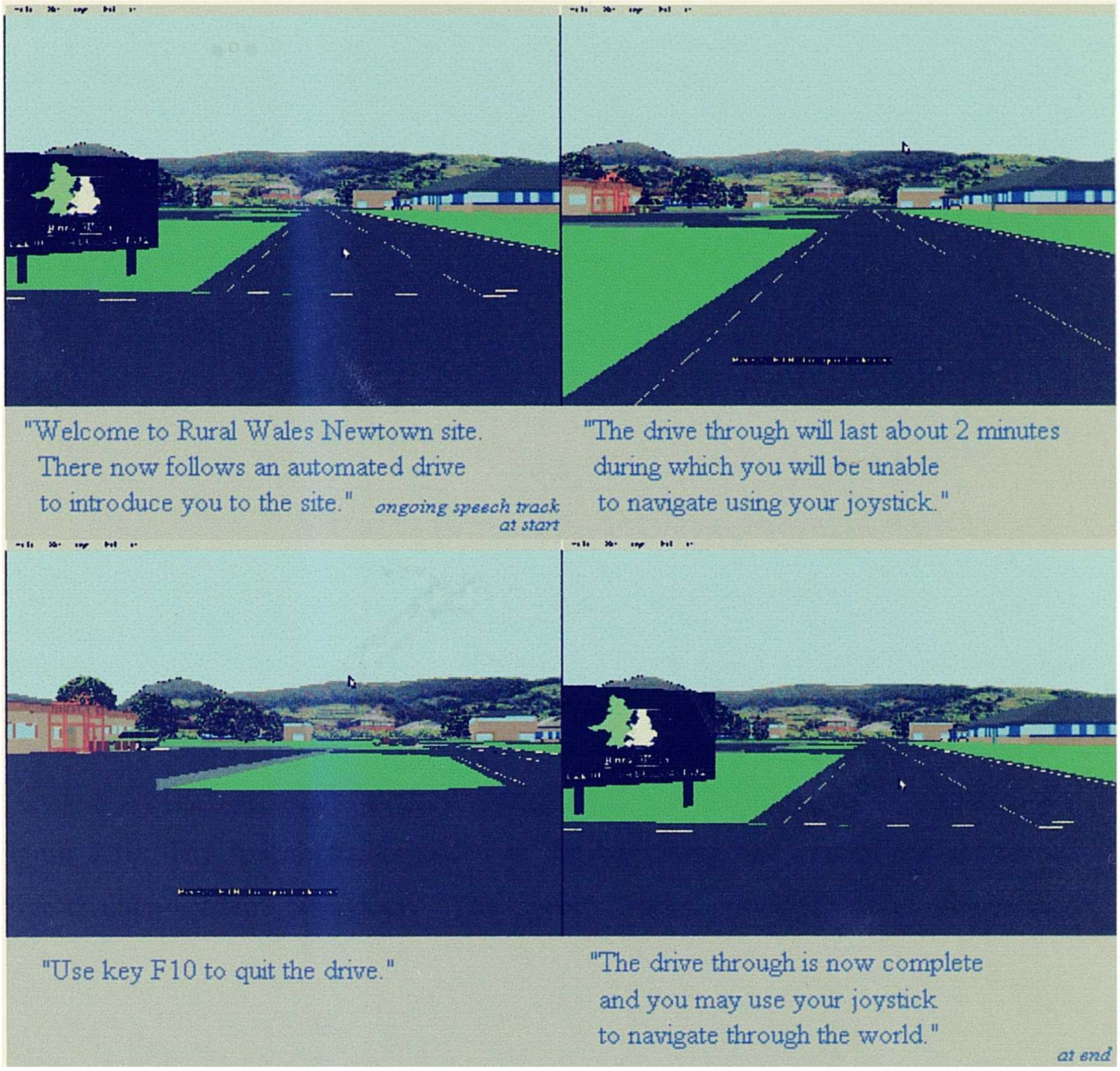


Figure 5.10: The speech tracks added to the beginning and end of the automated drive-through system behaviour. The speech tracks provided useful information about the drive. (*Text of speech track given below screen-shot of VE.*)

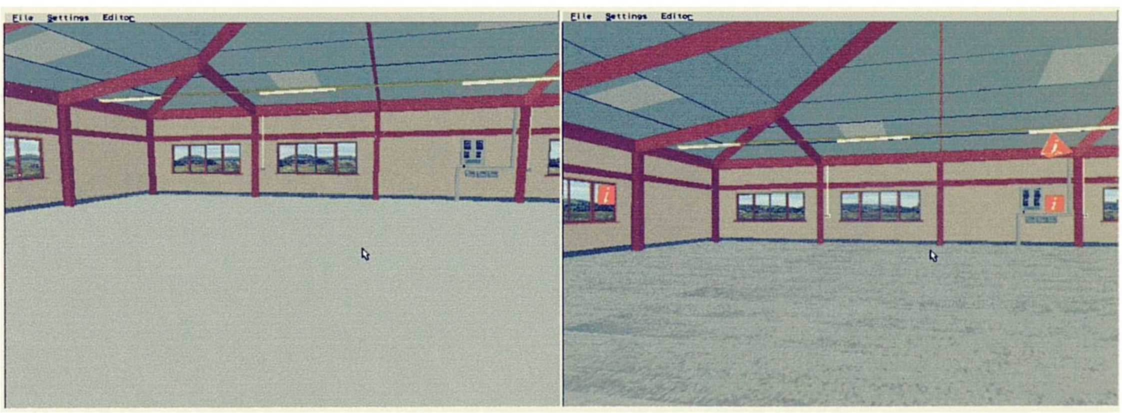


Figure 5.11: Left picture shows the original environment, which included actions for providing information about basic facilities. Right picture shows the amended version where such actions are clearly indicated through the use of information signs.

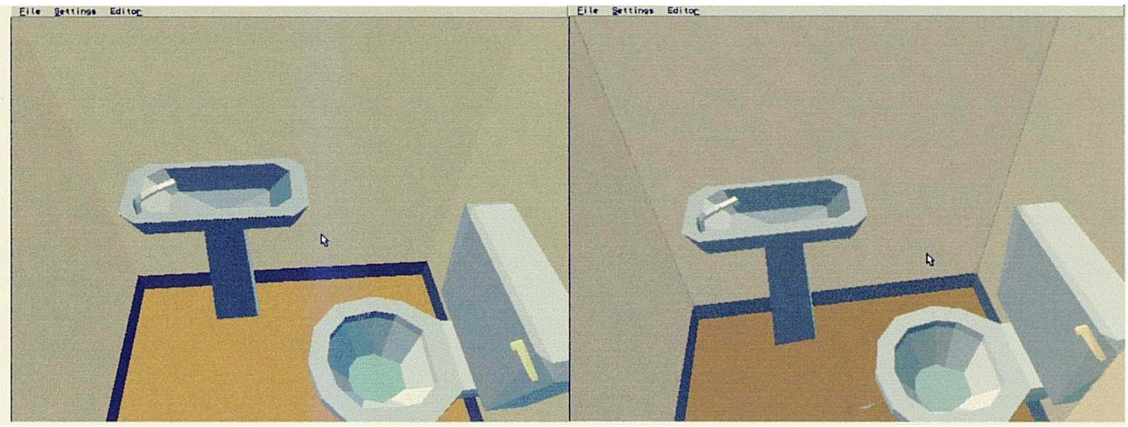


Figure 5.12: Implementation of GDP *distinguishable object* for the toilet walls.

Figure 5.13: Implementation of GDPs *identifiable object* and *clear navigation pathways* for the exit door.

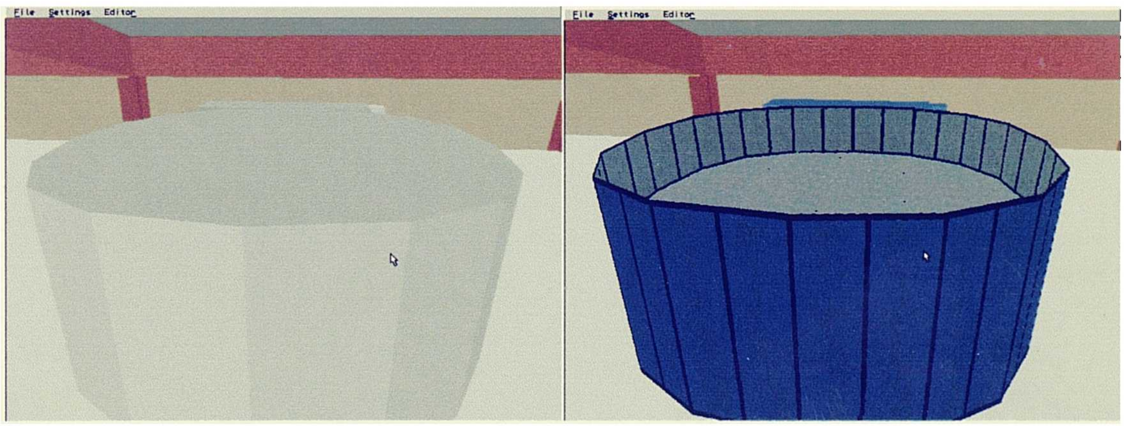


Figure 5.14: Implementation of GDPs *distinguishable object* and *identifiable object* for the water tank.

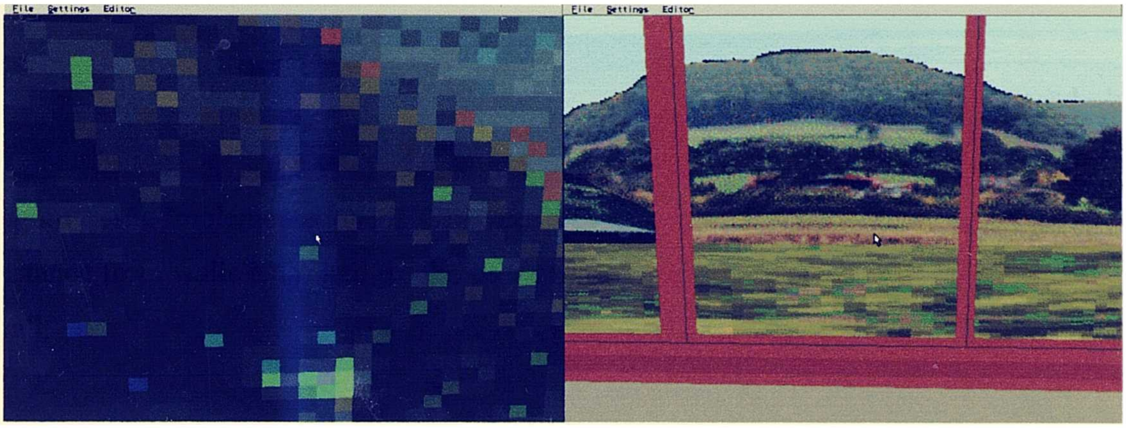


Figure 5.15: Implementation of GDP *executable action* for navigation - allowable distance from walls is limited, so the user cannot accidentally stand right up against a wall.

Figure 5.16: Implementation of GDP *clear self position/orientation* - through a purple figure showing the head tilt and height of the viewpoint, see detailed middle picture. *(Both screen-shots show the amended version.)*

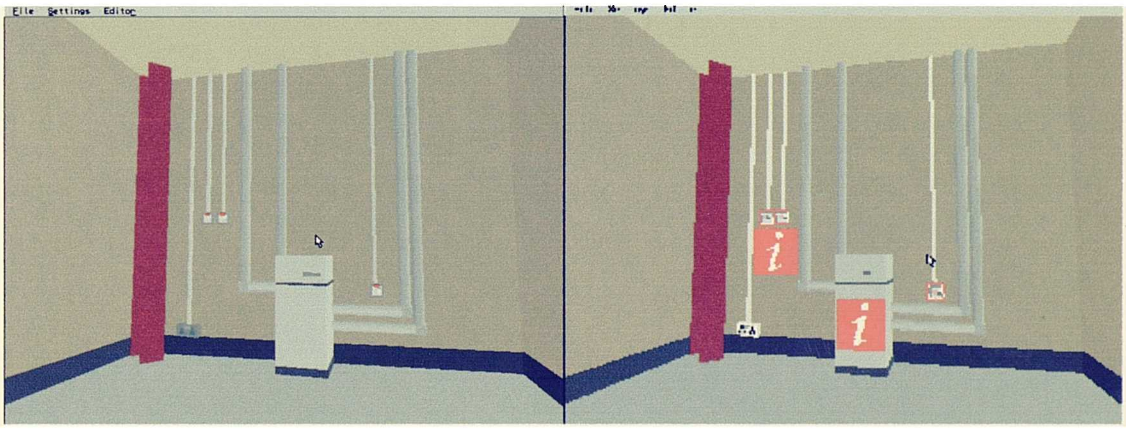


Figure 5.17: Implementation of GDP *declared available action* for the power switches.

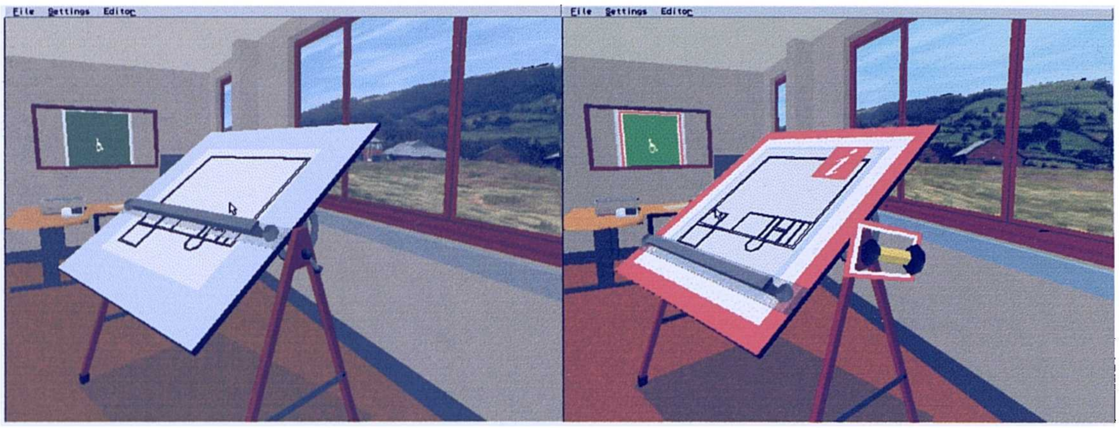


Figure 5.18: Implementation of GDP *declared available action* for actions on the drawing board – loosening the board with the handle and tilting the board itself.

Figure 5.19: Implementation of GDPs *distinguishable behaviour* and *declared causality and effects of behaviour* for the speech text event.

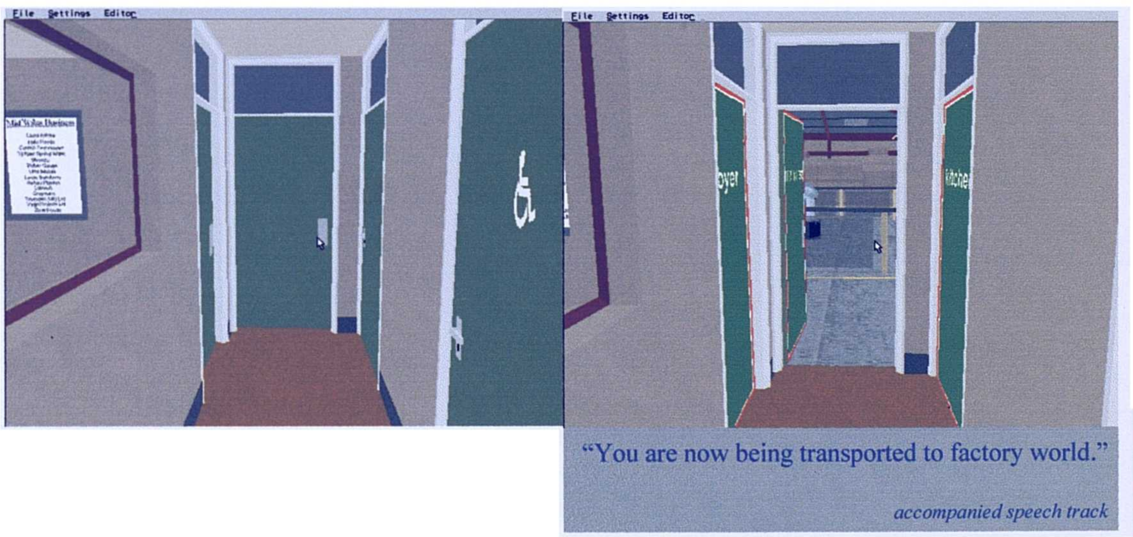


Figure 5.20: Implementation of GDPs *clear action progress* and *clear action effect* for the command of switching worlds.

techniques most appropriately fitting the context of the test application were chosen to implement GDPs. Attempts were made to implement every missing GDP for the elements addressed. In a few cases, GDPs could not be implemented. For example, the missing GDP, executable action, for the door element could not be implemented, because it had failed due to problems with the system not reliably registering mouse clicks, which could not be resolved in the time available.

All amendments were examined by an independent judge, to check they represented only the requirements of the GDPs in question. For each amended element of the test application, the judge was given definitions of the GDPs that had been implemented and details of the specific changes made. The original and amended versions of the application element were shown. In a few cases, the GDP implementations were altered to conform more closely to the requirements of particular GDPs. The original and amended versions of the test application were then used in the following controlled study.

5.4.2 Study method

The interaction and performance of subjects in the control group, who had been given the original application, were compared with the GDP group, who had been given the amended version of the application. Comparisons were made for the following aspects of interaction: number of usability problems, task performance and completion, post-study test scores and subjective responses in the retrospective questionnaire.

Usability problems were noted from observation of video footage and subjects' verbal protocols. Observation was also used to determine tasks completed successfully and the time taken on each task. Criteria were set for successful completion of each task, as summarised in table 5.10. The post-study (recall from memory) test consisted of 31 questions, divided into sections on exploration of the environment, spatial layout, and the object, action and analysis tasks carried out. A common scoring scheme was used which assessed how much information and understanding about the business-park subjects had gained from carrying out the tasks. The scoring scheme involved a maximum score for each question and details of how points should be allocated (see appendix 5N). Subjective responses in the retrospective questionnaire were recorded on a Likert scale and open-ended questions were used to support the quantitative data.

Task type	**Task**	**Completion criteria**
exploration	exploration	Generally explored environment, including internal and external worlds
object	window	Retrieved window information
	water tank	Found water tank
action	open loading bay	Opened loading bay
	switch power on	Clicked all power switches
	tilt drawing board	Tilted drawing board
analysis	floor covering	Retrieved floor covering information
	power sockets	Retrieved socket information OR investigated available sockets in rooms
	compare toilets	Visited all toilets

Table 5.10: Criteria for successful completion of each task

Specific hypotheses were set for testing the impact of implementation of the GDPs on interaction success:

Hypothesis 1: Subjects using the amended version of the test application, with the missing GDPs implemented, will encounter significantly fewer usability problems.

Hypothesis 2: Subjects using the amended version will complete significantly more tasks successfully.

Hypothesis 3: Subjects using the amended version will complete tasks significantly faster.

Hypothesis 4: Subjects using the amended version will gain more useful information from their interaction session, i.e. they will achieve significantly higher post-study test scores.

To test for significance, t-tests were used to compare the performance of subjects in the two groups. *(All the following statistics used an unpaired one-tailed t-test, unless otherwise indicated.)*

5.4.3 Results

5.4.3.1 Usability problems

The GDP group encountered significantly fewer usability problem incidents overall ($p<0.01$; avg. Control=134, GDP=45 problem incidents per subject). Therefore, hypothesis one on the usability problems was not refuted.

addressed element	***P***	**Average no. of problems/subject**	
		Control	**GDP**
window	0.095	3.5	1.9
tank	0.30	4	3.4
open bay area	<0.01	1.6	0
loading bay	<0.01	3	0.6
mains	<0.01	8	1.1
robots	<0.01	1	0
utility switches	0.039	3	1
drawing board	<0.01	9	3.6
floor	<0.01	2.4	0.7
sockets	<0.01	6.1	0.7
toilets	<0.01	8.3	2.2
navigation	<0.01	11	2.2
orientation	0.22	3.9	2.6
doors	0.083	16	9.1
exits	<0.01	4.9	0.2
switch worlds command	<0.01	5.8	1.1
drive through	<0.01	4.8	1.4
talking man	0.043	2	0.7
trunking	0.028	0.9	0

Table 5.11: The average number of problems encountered by subjects in the control and GDP groups, for each environment element for which missing GDPs had been implemented. The statistical comparison of usability problems encountered by subjects between the two groups is given, and highlighted where $p<0.05$.

Table 5.11 gives the average number of problems encountered by subjects, in both groups, for each of the elements addressed in the implementation of GDPs. Looking at environment elements individually, the GDP group encountered significantly fewer problem incidents (see shaded cells) for most of the elements addressed, 15 out of 19. Elements that were originally more problematic tended to show the greatest improvements. For the other four elements (windows, water tank, orientation and doors) there were fewer problems with the amended version, but the difference was not significant. This appeared to be due to common problems, in the original version, still persisting in the amended version, indicating that related GDPs had not been adequately implemented. Table 5.12 lists the persisting problems and gives probable explanations. For example, an on-screen figure was included (see figure 5.16) to indicate the current view angle but, because of severe technical constraints limiting the representativeness of the figure, it was difficult to interpret and did not help achieve appropriate view angles (problem 'difficulty getting the required view angle' still persisted).

element	**persisting problem**	**explanation**
water tank	repeatedly trying to interact with the (passive) water tank	No change was made in the improved version but active objects were highlighted. However, because this was a task object, subjects still expected some interactions with it even though it wasn't highlighted.
	not noticing the tank when it was in view and was the target being searched for	The tank was changed to be more distinguishable and easier to identify. However, it was located in a high area (near roof, above offices) and subjects commonly were not aware that this area existed and did not focus attention to it.
orient-ation	difficulty getting the required view angle	No change was made to the actual joystick commands used, but an on-screen indicator for the current orientation was included and extreme view angles (such as being up-side-down) were not allowed. However, the orientation indicator proved difficult to interpret and did not attract attention when the orientation was being changed.
doors	difficulty navigating through the doorway	Changes were made to prohibit walking through walls/doors. This avoided problems of falling into walls when trying to go through the doorway, but there were additional problems of precisely navigating through the doorway and not bumping into walls.
	having to mouse click several times to get the door to open	No change could be made about the system problem of mouse clicks not being reliably registered.

Table 5.12: Common problems with the original version, which persisted after the amendments were made.

As would be expected, there was no significant difference in problems encountered for most of the elements not addressed, 32 out of 43. Table 5.13 details the 11 elements where there were significant differences. The differences appeared to be mainly due to side effects from related GDP implementations or resulted from general improvements made. For example, three cases were probably due to the use of standard 'I' signs to cue all information actions (see figure 5.11), including those for the heater, boiler and ceiling objects. Another three were probably due to improvements in navigation, which meant that the smaller foyer, kitchen and hall areas could be navigated more successfully.

non-addressed element	*p* =	**Average no. of problems/subject**	
		Control	**GDP**
heater	0.022	2.1	1
boiler	<0.01	1.6	0.2
ceiling	0.012	0.9	0
foyer	<0.01	3.4	0.1
kitchen	<0.01	3.5	0.1
hall	<0.01	3.9	0.4
utility floor	<0.01	0.9	0
phone	0.012	1.5	0.2
computer	0.039	0.6	0
joystick/ mouse	<0.01	2.1	0.7
overall task	<0.01	3.3	0.3

Table 5.13: Environment elements that had not been addressed in the amended version, but which showed significant differences between the groups in the number of usability problems encountered.

5.4.3.2 Task performance

The GDP group successfully completed significantly more tasks ($p<0.01$; avg. Control=7, GDP=8.4 tasks). Therefore, hypothesis two on task completion was not refuted. Table 5.14 shows the percentage of subjects successfully completing each task, for the two groups. The more complex tasks, such as tilting the drawing board, showed the greatest improvements.

The GDP group spent less overall time on the tasks, but this difference was *not* significant ($p=0.13$; avg. Control=42.6, GDP=39 minutes). Therefore, hypothesis three on task time was rejected. Table 5.14 gives the average time spent on each task, for the two groups. The GDP group did spend significantly less time on the "drawing board" ($p<0.03$) and "power sockets" ($p<0.01$) tasks. However, the GDP group also spent significantly *more* time on the "water tank" task ($p<0.01$ - *two-tailed t-test*). This may have been because, with the original version, subjects were less confident of finding the water tank (which was naturally difficult to find), since earlier problems had led them to lose confidence in the application. The GDP group may have expected to be able to find the tank and, therefore, spent longer looking for it.

		Task completion		Average task time	
Task type	**Task**	**Control**	**GDP**	**Control**	**GDP**
exploration	exploration	100%	100%	9.5(mins)	9.7(mins)
object	window	90%	90%	2.4	2.6
	water tank	60%	80%	3.6	7.4
action	open loading bay	100%	100%	2.4	2.4
	switch power on	50%	80%	4.4	3.4
	tilt drawing board	40%	100%	4.8	3.1
analysis	floor covering	80%	100%	3.9	3.3
	power sockets	90%	90%	5.8	1.5
	compare toilets	100%	100%	5.8	5.6
			TOTAL	**42.6**	**39**

Table 5.14: The percentage of subjects successfully completing each task and the average task time, for the two groups. Where the task times showed significant differences, the cells are highlighted.

The GDP group achieved higher total scores for the post-study test and this difference approached significance ($p=0.064$; avg. Control=46, GDP=52%). Therefore, hypothesis four on information gathered from interaction, was not satisfactorily refuted. Table 5.15 gives the scores, by test section, for the two groups. Scores for questions on the action tasks showed the greatest difference. Looking at questions individually, 9 of the 31 individual questions showed significantly improved scores ($p<0.05$). The GDP group was able to recall more accurate information about the speech bubble event in the exploration section (see questions 4-6 in appendix 5E); the power and drawing board tasks in the action section (questions 18, 20 and 21) and the floor and socket tasks in the analysis section (questions 25-27). However, the GDP group also had significantly *lower* scores ($p<0.05$; *two-tailed t-test*) in two of the questions. For question 1, in the exploration section, the GDP group was able to recall fewer environment actions, probably because nine of the original specific object information actions, could be summarised as one general action on an information sign object, in the amended version (see figure 5.11). For question 16, on the power task, the GDP group was able to recall less information about the power facilities, perhaps because the control group, who had difficulties switching on power, studied the power information more often and were therefore better able to remember the details. There were no improvements for the spatial layout section, which was not an area that had been addressed in the GDP implementations.

Section	**Control**	**GDP**
Exploration	27%	26%
Spatial layout	61%	58%
Object tasks	43%	47%
Action tasks	34%	64%
Analysis tasks	57%	61%
TOTAL	**46%**	**52%**

Table 5.15: The average percentage scores for each section of the post-study test, for the two groups. Scores for the action section shows a significant difference, at $p<0.01$.

5.4.3.3 Subjective views

Analysis of subjects' views in the retrospective questionnaire did not show many areas of significant difference between the groups. However, the GDP group showed some more positive views about the VE, as listed in table 5.16. The GDP group perceived the VE to be significantly better at providing information about objects and actions. The GDP group perceived action tasks to be significantly easier and significantly more of the GDP group claimed to have experienced the talking man event. However, the GDP group also indicated *weaker* feelings of 'presence' inside the VE. The reason for this is unclear but may have been due to the GDP group having fewer major problems, which could have the effect of increasing absorption, or finding the VE less realistic because of the added highlighting effects. Appendix 5O gives the average ratings given by the two groups for all the quantitative questions.

Question		*p*	Control	GDP
3	How easy did you find the action tasks?	<0.01	2.1	4.8
6	Did you experience the speech text event?	<0.01	40%	100%
18	How good was the VE in providing information about objects?	<0.01	2.9	4.7
	How good was the VE in providing information about actions?	<0.01	2.1	3.8
24	How strong was your sense of presence in the VE?	0.012 *(two-tailed t-test)*	4.9	3.2

Table 5.16: Questions in the retrospective questionnaire where there were significant differences between the groups. Responses were on a scale of 1[low] to 7[high] and the average response and statistical comparison are given. For question 6, the percentage of subjects claiming to have experienced the event is given.

5.4.4 Discussion

The results are very encouraging and show improvements in interaction across all levels. Since the GDPs were closely linked to predicted usability problems, this is where the greatest improvement was expected and found. There was a 66% reduction in usability problems and, therefore, subjects were able to complete their tasks better, gain more useful information during interaction and had some more favourable opinions about the VE. Notably, the GDP group felt the VE was significantly better at providing information about objects and actions. Overall task time was not improved significantly, but this may have been because users tended to occasionally explore unrelated aspects of the environment, while working on their current task.

There were fewer problems with most of the elements for which the GDPs had been implemented. A few areas did not show expected improvements, such as maintaining orientation, but this appeared to be due to the limited or inadequate implementation of GDPs, rather than the GDPs themselves giving limited or poor advice. This indicates that the GDPs can be implemented to differing degrees, which can result in notable variations in usability. Most elements not addressed did not show differences in problems encountered, unless they were involved in some general improvements or were closely related to areas that had been addressed. This appears to indicate that the improvements in interaction were related to the GDP implementations made. There were particular improvements in the more problematic areas, such as the action tasks with the drawing board and power objects. Enhancements to the speech-bubble event led to more subjects experiencing and understanding this event. In a few specific situations, performance appeared to get worse but there were reasonable explanations why. For example, more time was probably spent on the water tank task because users had more confidence in the amended version, since it included extra cues and support information, and they expected to be able to complete all tasks.

Therefore, support has been found for hypotheses 3 and 4 of the thesis:

H3 Design properties required for interaction can be predicted using the general patterns (in the theoretical models).

H4 Interaction can be improved by implementing the design properties (H3).

The results indicate that the proposed GDPs are important requirements for successful interaction, and a VE interface can be significantly improved by implementing missing GDPs, even with a partial implementation, as in this case. In section 5.6.3, the use of the results to refine the set of GDPs is described. In the next section, a final, non-empirical, evaluation of the theory is described, specifically assessing the logic of the correspondence rules.

5.5 Computational implementation of the problem prediction rules

Before refining the theory, a second level of evaluation was carried out for the problem prediction rules in the theory. It involved assessing the logical consistency and coherency of the rule structure by implementing the rules in a computational model. Data from the empirical studies was used to test the implementation.

5.5.1 Expert system of sub-set of problem prediction rules

The Crystal expert system shell was used to implement the rules. It provided a limited user-interface, but did enable a rule base to be quickly and easily set up. A partial implementation was carried out, since all the rules followed the same logical structure (see section 4.5) and for reasons of practicality. The sub-set of rules selected predicted six problems in the 'acknowledge control', 'monitor', 'intention control action' and 'end control' stages in system initiative mode. Only user knowledge and GDPs relevant to these rules were included. This particular sub-set of rules were selected because they were all related to one element type, a system control, and there were a clear and sufficient set of observed usability problems with system control, from the experimental studies, to test the expert system with.

The expert system included two separate paths for predicting usability problems. Either general information was used about user knowledge and elements supported by a VE, or specific information was used about a particular element in the VE. For the latter option, a menu enabled selection between eight possible element types (task & environment, spatial structure, self & viewpoint, navigation, an object, an action, a system event or a system control), as shown in figure 5.21. The element types were generally based on the GDP categories, with an addition for the common activity of navigation. When general information was used, user knowledge (& GDPs supported) were categorised according to these same element types (e.g. user has knowledge of spatial structure, navigation etc.), instead of using the knowledge sources. This approach simplified the expert system because there was then no requirement to determine when different knowledge sources were relevant to different elements, for example, whether objects could be identified because they were known from previous interactions, were present in the domain or were common real world objects.

Only the system control element type was processed in this implementation. A series of yes/no questions ascertained what, general or specific, elements of user knowledge and GDPs were present. For specific element types, additional information sometimes had to be requested, such as information about related objects for a system control. As in the assessment of the test application (described in section 5.3.1), GDPs had to be consistent with user knowledge to be adequately supported. This simplified the expert system because further information was not required about the level of consistency between user knowledge and the GDPs. In the problem prediction part of the system, each rule was tested to see whether facts gathered about user knowledge and the design would avoid or predict the problem in question. The rules predicted problems if pre-conditions about required user knowledge or GDPs failed. Finally, the expert system provided information about usability problems that had been predicted, from the facts given. Figure 5.21 shows screen shots for each part of thc processing carried out. Table 5.17 gives two example rules for predicting problems, as defined in the expert system. Appendix 5P gives a complete list of the rules in the expert system.

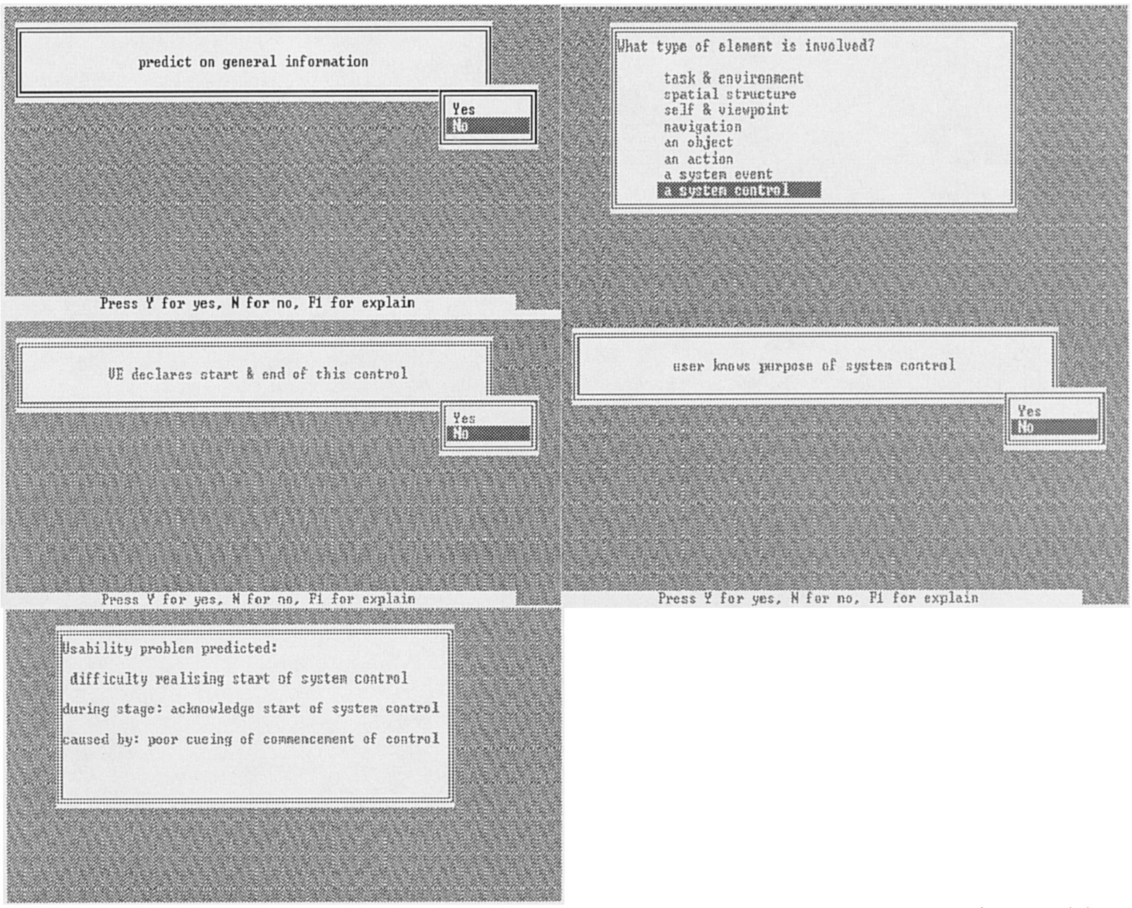


Figure 5.21: Screen-shots of the expert system showing the first screen for problem prediction using general or specific information (top left) and then a menu for choosing a specific element type (top right). A series of yes/no questions follow (e.g. middle left and right) and finally information is given about problems predicted (bottom left).

difficulty realising start of system control IF DO: Test Expression GDPcontrol_start_end<>1
difficulty understanding goal of system control IF DO: Test Expression UKcontrol_purpose<>1 AND DO: Test Expression GDPcontrol_purpose<>1

Table 5.17: Two example rules for predicting problems, as defined in the expert system. In the second rule, the problem *difficulty understanding goal of system control* is set to true if both the conditions *user has knowledge of control purpose* and *GDP clear system control purpose* are false.

5.5.2 Testing of the expert system

The expert system incorporated three variable aspects in the prediction of problems, which were whether problems were predicted using specific or general information, what facts were input about user knowledge and GDPs present, and what predicted problems were actually output. The expert system was systematically tested by covering all major variations in these components, i.e. by checking that all possible paths through the rule set and major combinations of input facts would produce expected outcomes. The scope of possible paths, input facts and outcomes was defined, as listed in table 5.18. Six separate tests were then devised to cover all these variations, as listed in table 5.19.

Further tests were devised using actual data on interaction problems, from the empirical studies (see table 5.19):

- Test *G* checked whether an individual problem was predicted, with known information about missing user knowledge and missing GDPs. The observed problem was *wanting to stop the drive-through but not knowing how to*, missing user knowledge was *information about how to stop or control the drive-through*, and the missing GDP was *declared available actions during control.*
- Test *H* checked whether accurate problem predictions were made with known general information about the average user profile and GDPs supported, in the original test application. The main observed problems with the system control

drive-through were: *not realising that the system had taken control of interaction, not understanding why the system had taken control, being unable to stop or control the drive-through* and *not realising when the system had relinquished control.* The average user profile was that the user had prior knowledge about the task category only and GDPs were judged not to be sufficiently supported in any of the general areas.

- Test *I* checked whether accurate problem predictions were made with known general information about the average user profile and GDPs supported, in the *amended* version of the test application. Few observed problems were found with the system control drive-through, in this version. The most common problem (only 3 occurrences) was *trying to navigate before having heard the full explanatory speech track.* The average user profile was again that the user had prior knowledge about the task category only, but GDPs were now judged to be sufficiently supported in these general areas: *self & viewpoint*, *objects*, *system behaviour*, and *actions & feedback.*

Path through rules	P1. problems are predicted using general information P2. problems are predicted using specific information on individual elements
Input facts	F1. no user knowledge or GDPs present F2. all user knowledge and GDPs present F3. user knowledge present, but no GDPs F4. GDPs present, but no user knowledge F5. a mixture of user knowledge elements and GDPs present
Outcome	O1. no problems predicted O2. six (maximum) problems predicted O3. some but not all problems predicted O4. problems cannot be predicted because the requested element has not been implemented

Table 5.18: Variations possible for three aspects of the tests - path taken through the rules, input facts and outcomes.

	Path	Input facts	Expected outcome	Result
Tests to cover possible variations				
A	P1 - general	F2 - all user knowledge and GDPs	O1 - no problems predicted	← as expected
B	P1 - general	F4 - GDPs, but no user knowledge	O1 - no problems predicted	← as expected
C	P2 - specific - system control	F1 - no user knowledge or GDPs	O2 - six (maximum) problems predicted	← as expected
D	P1 - general	F3 - user knowledge, but no GDPs	O3 - some but not all problems predicted: *difficulty realising start of system control* *difficulty distinguishing system behaviour* *difficulty realising end of system control*	← as expected
E	P2 - specific - system control	F5 - a mixture of user knowledge (UK) and GDPs present: *UK cause-effect-behaviour* *UK identity-object* *GDP control-start-end* *GDP distinguish-behaviour*	O3 - some but not all problems predicted: *difficulty understanding goal of system control* *difficulty distinguishing system behaviour* *difficulty determining whether/what user control can be exercised*	← as expected
F	P2 - specific - actions	-	O4 - element not implemented	← as expected
Tests using empirical data				
G	P2 - specific - system control	F5 - Missing user knowledge: *UK control-actions* Unsupported GDP: *GDP control-actions*	O3 - Expect predicted problem: *difficulty determining whether/what user control can be exercised*	← as expected
H	P1 - general	F5 - User knowledge present: *general task knowledge* GDP areas supported: none	O3 - Expect predicted problems: *difficulty realising start of system control* *difficulty understanding goal of system control* *difficulty determining whether/what user control can be exercised* *difficulty realising end of system control*	all (6) problems predicted, including these 4
I	P1 - general	F5 - User knowledge present: *general task knowledge* GDP areas supported: *self & viewpoint* *objects* *system behaviour* *actions & feedback*	O3 - Expect predicted problems: *none or possibly*: *difficulty realising start of system control* *difficulty determining whether/what user control can be exercised*	no problems predicted

Table 5.19: The nine tests applied to the expert system with details of input, expected output and actual output.

5.5.3 Results and Discussion

Tests *A* to *F* were carried out and found to produce the exact outcomes predicted in table 5.19. Test *G* also produced the expected outcome. Test *H* partly produced the outcome expected. All the problems were predicted, including the four expected and an additional two (*difficulty distinguishing system behaviour* and *difficulty* *interpreting system behaviour).* Test *I* produced one of the possible expected outcomes, no predicted problems.

Therefore, the results in general were positive. The tests using data on actual interaction problems and predicting problems on general information, were not found to be completely accurate due to the generalisations which needed to be made about what was or was not supported. In test *H*, a judgement was made that GDPs on system behaviour were not supported, but some specific areas, such as distinguishing system behaviour, were supported well enough to avoid potential problems. Similarly, in test *I*, a general judgement was made that GDPs on system behaviour were supported, but some specific areas, such as declaring the start of system control, were not supported sufficiently to avoid all potential problems. However, these tests were never-the-less representative of the general trends. The results indicate that when the expert system follows the general path, predicting problems using summarised information about user knowledge and GDPs present, then it cannot give an accurate list of likely usability problems, but will indicate general trends.

Therefore, the results indicate that the problem prediction rules do have a logical structure to them, which enabled the rules to be implemented in the expert system. The implementation of the rules was shown to be effective in making general predictions about usability problems. There were some limitations with this basic partial implementation. In the problem information output, only a general list of possible causes was included, for reasons of simplicity, rather than stating what triggered the problem to be predicted in a particular run. Some element types did require additional information for problem prediction, for example system behaviour required additional information about objects involved in the system behaviour. Therefore, the mapping of relevant problems to element types (object, action, etc.) was not necessarily straight-forward. However, this problem seems inherent since the GDPs do naturally apply to different stages and different usability problems. The Crystal package posed some constraints on the user interface, resulting in the knowledge acquisition dialogue being time consuming. Finally, the judgements about whether GDPs, or user knowledge, were present and adequate were a subjective part of the system that was left to the evaluator. Although this is also the case with other computational models, such as PUMS where user knowledge has to be defined and problem-solving choices input by the evaluator (Blandford and Young, 1995), any future implementation should perhaps include information to help the evaluator make accurate judgements.

The implementation of the rules completed the evaluation of major components of the theory. The next step was to refine and improve the theory of interaction, in light of results from the various evaluation exercises.

5.6 Refinement of the theory

The experiment results and computational implementation generally supported major components of the theory. Before using the theory to inform guidance, it was refined in light of the results, to make it more accurate and representative of observed interaction. A set of conditions were used in assessing how well the study results supported different parts of the theory and how important possible additional parts were. Final judgements were inevitably subjective, but the conditions were important in guiding and helping to record reasoning about the refinements required. The following sections describe the refinement of the models of interaction, problem prediction rules and generic design properties.

5.6.1 Models of interaction

<u>5.6.1.1 Stages of interaction</u>

The 21 stages of interaction were each given one of the following assessments:

Validated – if found to be reasonably common (i.e. accounted for at least 1% of observed stages) and was experienced by at least 50% of subjects.

Retain - if there were good reasons why the stage was not commonly observed in this study, or, it was not expected to be common, but would be never the less important in certain situations.

Remove - if the stage was shown to be a less important behaviour, which did not warrant a separate stage.

Of the twenty-one stages, fourteen were given a validated assessment, six were given a retain assessment and one stage, *consider objects* in task action mode, was given a remove assessment. This stage was found to represent less important and optional behaviour and was therefore incorporated into *intention task action*, to create a stage where an intention to carry out a task action is formed and a possible consideration is made of objects required. Stages retained included 'intention explore action', in explore navigate mode, and five stages from system initiative mode, since it was felt the experiment provided limited opportunity for these behaviours.

The additional observed stages of interaction were deemed to be important to *include* if they were reasonably common (i.e. accounted for at least 1% of observed stages), were experienced by at least 50% of subjects, and if there were no reasons why the behaviour should not be commonly observed in other studies. The additional behaviours were either included as new stages or incorporated into existing, related or closely coupled, stages. One new stage, *intention move action*, was included for actions to move through the environment, and four additional behaviours were incorporated into existing stages. For example, the interpretation of navigation feedback was incorporated into the existing *navigate* stage. Appendix 5Q gives the assessments and reasoning behind decisions made for each existing stage of interaction and the additional behaviours included.

5.6.1.2 Flow of interaction

The predicted flows of interaction in the models (stageA➔stageB) were each given one of the following assessments:

Validated - if at least 10% of the total transitions from stage A went to stage B.

Retain - if there were good reasons why the flow was not commonly observed in this study, or, the flow maintained a basic path joining stages, where jumps and backtracks were observed.

Remove - if the flow was expected to be uncommon generally, the model could be simplified by merging it with other flows, or, one of the stages involved had been removed.

Of the twenty-nine flows of interaction, nine were given validated assessments, thirteen given retain assessments and seven were removed. Flows were removed from the models mainly due to one of the stages of interaction involved being removed, or, inaccuracies about stages involved, resulting in a more common observed flow of interaction replacing the predicted one.

The additional observed flows of interaction were deemed to be important to *include* if they were reasonably common (i.e. at least 10% of the transitions from stage A went to stage B) and there were no reasons why they should not be commonly observed in other studies. The new *intention move action* stage was included in explore navigate mode, because it was related to navigation through the environment and was closely linked to stages in this mode. Overall, there were 18 additional flows of interaction included, which were mainly jumps or backtracks through the models (for skilled behaviour and error-recovery) or additional transfers between modes. Appendix 5R gives the assessments and reasoning behind decisions made for the predicted and additional flows of interaction. Figures 5.22 to 5.24 illustrate the refined models. For example, figure 5.22 shows the refined task action model. The *consider objects* stage has been merged into *intention task action.* Only one of the flows from *evaluate*, returning to the beginning of the model, has been retained, thereby maintaining the original Norman cyclic path. Major unpredicted flows have been added. For example, additions include short-cut paths to ‘execute’ for skilled behaviour (e.g. from ‘approach/orient’), a backtrack from ‘feedback’ to ‘execute’ for re-trying actions after unsatisfactory feedback, a cross from ‘inspect’ to ‘execute’ to link object manipulations to prior inspections, and a transfer from 'evaluate' to explore navigate mode for returning to navigation after evaluating an action.

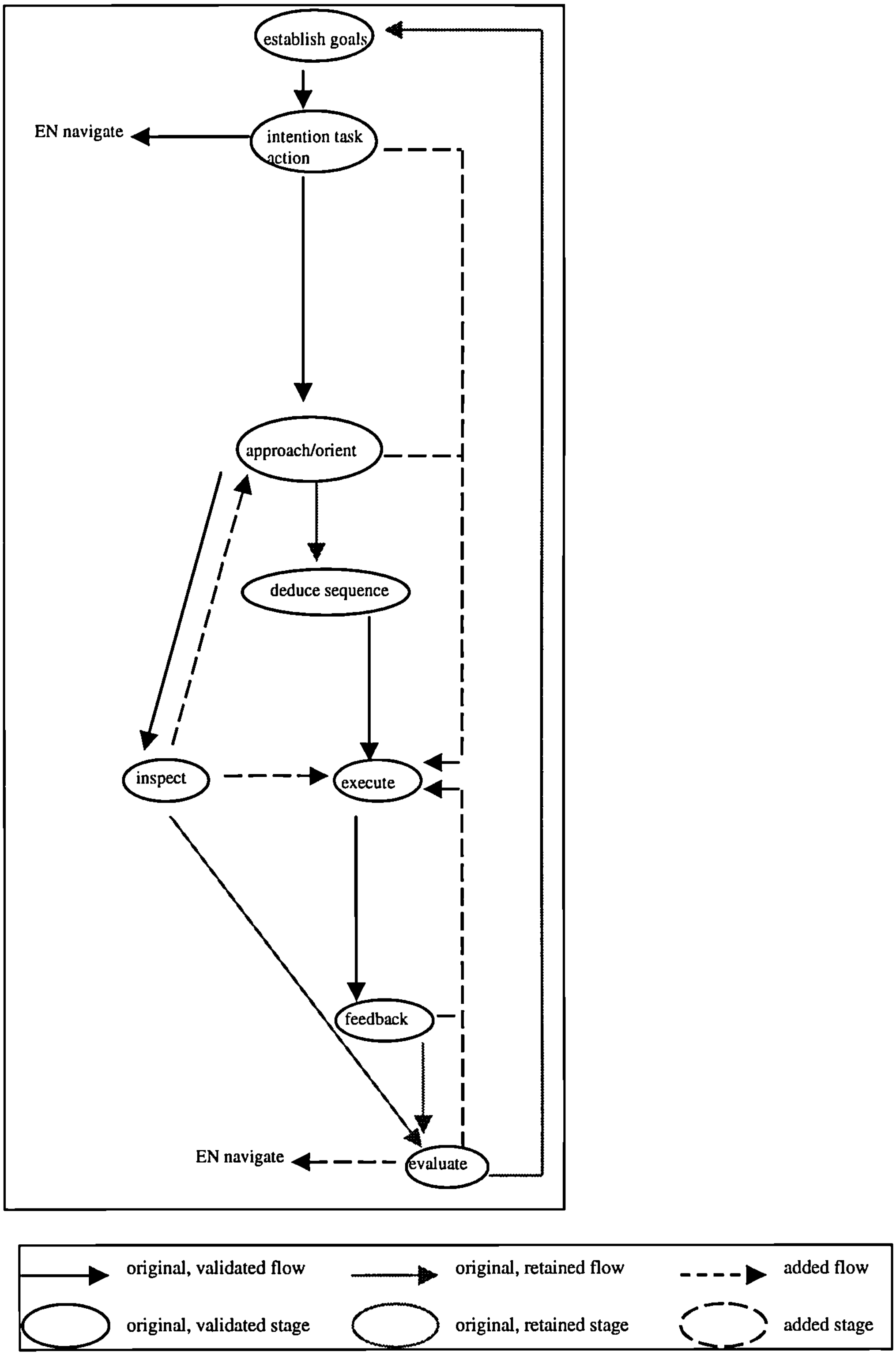


Figure 5.22: Task action model with the refinements. The 'consider objects' stage has been removed. Jumps forward and backtracks to the 'execute' stage have been added.

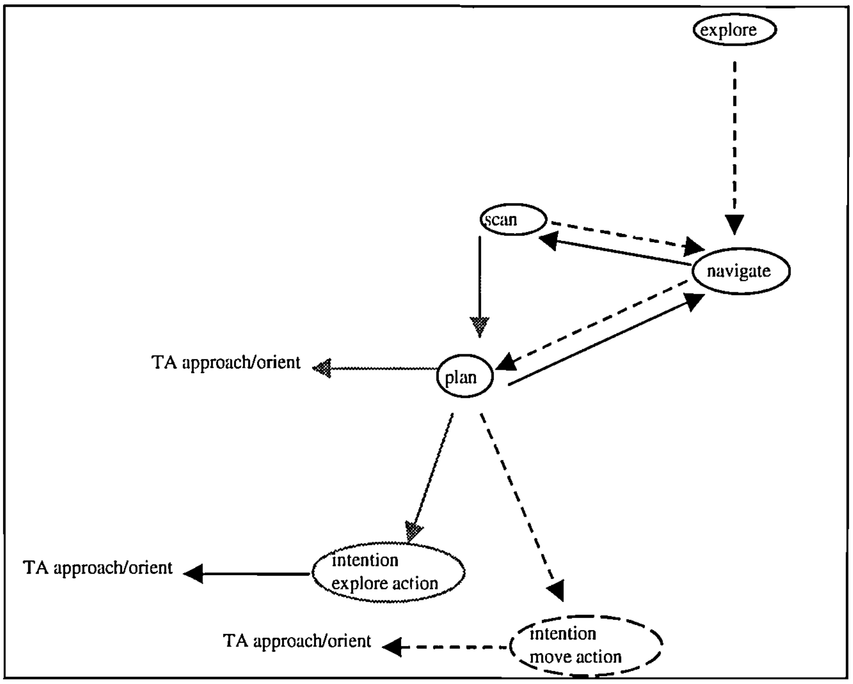


Figure 5.23: Explore navigate model with the refinements. The new *intention move action* stage has been incorporated, the major jumps forward added and the *explore* stage linked to *navigate*, as was observed.

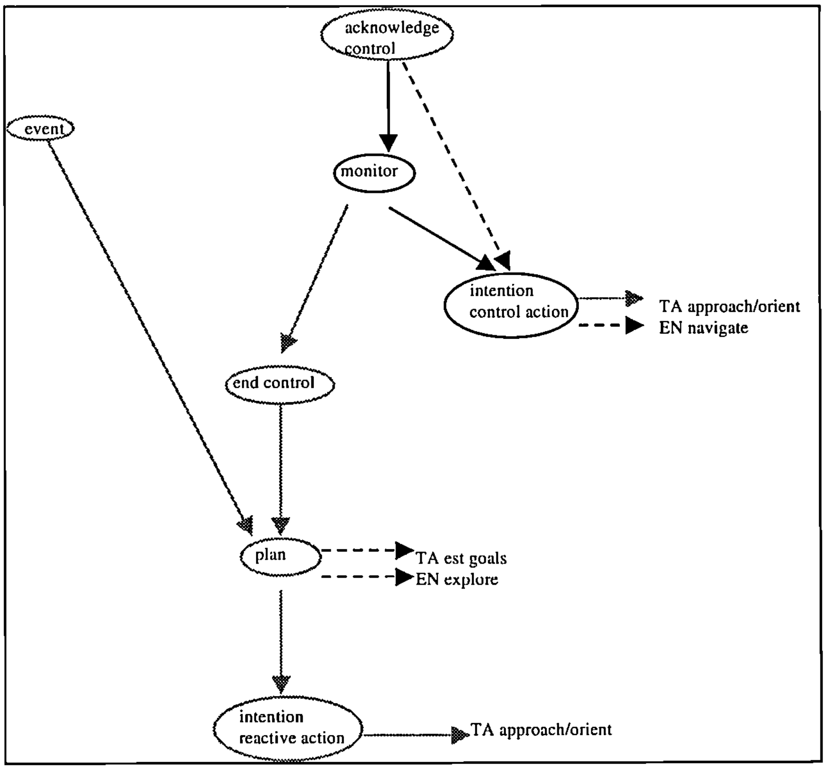


Figure 5.24: System initiative model with the refinements. Most stages were not validated, but were retained because of the lack of interaction data about responses to system behaviour. A jump forward to *intention control action* has been added for

regaining control from the start. Some of the stages transferred to in the other modes, have been changed in light of the study findings.

5.6.2 Problem prediction rules

Each of the 75 problem prediction rules were given one of the following assessments:

Validated - if it was matched with at least two observed problems and did not occur commonly with another rule.

Retain - if there were good reasons why the predicted problem was not matched with observed problems in this study, or, if it occurred commonly with another rule but there was a useful distinction to be maintained, for example if the rule related to a different interaction stage/task.

Not tested - if the area related to this rule was not one of areas for which problems had been matched, and, it did not mimic the difficulty in other removed rules.

Remove - if the rule was shown to be unimportant, duplicated another rule, or was a continuation of the difficulty in a previously removed rule.

Overall, there were 15 validated assessments, 23 retain assessments, 19 not tested and 18 remove assessments. Rules not tested were mainly related to general task problems or detailed problems with the user representation. Rules were removed mainly due to duplications identified in the empirical studies, which indicated areas where the set of problem prediction rules could be sensibly simplified. For example, rule 24, *difficulty determining parts involved*, was removed because it was found to be covered by rule 26, *difficulty determining how to execute action*, since both rules were matched to the same observed problems.

Additional rules for uncovered areas were deemed to be important to *include* if they covered outstanding issues for new or extended stages of interaction, covered more specific difficulties where overly-general problems had been identified, or, if they covered unpredicted difficulties, that had been matched to at least two observed problems and were generally applicable to VEs. Fifteen additional rules were introduced. For example, rules were included for the new 'intention move action' stage, such as *difficulty detecting what movement actions are available.* A new rule was included for the unpredicted problem *difficulties determining available navigation pathways,* and a specific problem, *difficulty moving at suitable speeds*, was included for when executing navigation. Unpredicted problems that related to users expecting elements in the VE which did not exist, were not included because these were seen to be less serious problems that related to difficulties already covered in the rule set. Appendix 5S gives the assessments made for the existing rules and additional rules included, and also gives descriptions for new and changed problem prediction rules.

5.6.3 Generic design properties

Each of the 46 GDPs were given one of the following assessments:

Validated - if implementation of the GDP led to a significant reduction in occurrences of a specific usability problem, or, lack of the GDP caused to a usability problem, for more than one subject.

Retain - if the implementation of the GDP was at fault (e.g. limited or poor), or, there were good reasons why implementation of the GDP did not work or lack of the GDP did not lead to usability problems in this study.

Not tested - if the GDP was not implemented, but was originally adequately supported, for the major elements in the VE (e.g. task and basic elements, such as navigation).

Remove - if the GDP was shown to be unimportant, or less important and better incorporated into another GDP, or, if the GDP only occurred in problem rules that had been removed.

Overall, there were 25 validated assessments (17 of these were where implementation of the GDP had reduced problems), 7 retain assessments, 11 not tested and 3 remove assessments. GDPs not tested were again mainly in the task or user object categories. GDPs (*clear self parts role*, *declared action components* and *declared feedback components*) were removed because all problem prediction rules including them had been removed. For example, GDP *declared action components* was removed because related rules, such as rule 24, *difficulty determining parts involved*, had been removed.

Additional GDPs for uncovered areas were deemed to be important to *include* if they were required for new or changed problem prediction rules. There were two such cases. A GDP for *unobstructed navigation pathways* was required for the new rule, *difficulties finding unobstructed pathways for navigation*, and a GDP for *appropriate navigation speeds* was required for the new rule, *difficulty moving at suitable speeds.* Appendix 5T gives the assessments made for the existing GDPs and the additional GDPs included.

5.7 Summary: evaluation of the theory

In summary, this chapter has discussed the evaluation of major components of the theory through experimentation and computational implementation. First, the models of interaction were evaluated using observed user behaviour and verbal protocol data. The models were found to predict major aspects of interaction with VEs. Core parts of the models and important additional behaviours were identified. The problem prediction rules were evaluated using observation of usability problems encountered by users. The rules were found to predict the majority of observed usability problems, although there were duplications in the rules. Important additional problems were also identified. The rules were further evaluated using computational implementation of a sub-set of the rules. The partial implementation was shown to be effective in making general predictions about usability problems, thus providing support for the logical structure of the rules. The usefulness of the theory in improving interaction was evaluated through a controlled study, which compared interaction in a VE with and without implementation of the predicted required design properties. Implementation of the GDPs was found to significantly reduce usability problems, leading to better task performance.

Therefore, the results provided general support for the theory predictions, which were then systematically refined in light of detailed findings. The controlled study showed that application of the theory was effective in improving interaction. Therefore, the next step was to provide the theory advice in an appropriate form for designers. The next chapter discusses the use of the refined theory to inform design guidance, to fulfil the objective of this thesis. Other implications of the findings of the studies are discussed in chapter seven.

Chapter 6

Design Guidance

This chapter describes design guidelines developed from the theory of interaction. The guidelines are presented in a hypertext tool and details given of an evaluation of the tool.

Chapter 6

Design Guidance

The theory of interaction provided an improved understanding of interactive behaviour in VEs and an identification of abstract design requirements for supporting that interaction. To effectively exploit the theory knowledge for improving the design of VEs it needed to be presented in a form geared towards the target audience, i.e. VE designers. The generic design properties in the theory encapsulated the actual design advice and the other components provided a context and motivation for the advice. Therefore, the GDPs were translated into concrete design guidelines to fulfil hypothesis 5 of the thesis:

> **H5** The design properties (GDPs) can be presented in a usable form to support VE interface design.

6.1 Translating generic design properties into guidelines

Guidelines can be useful prompts for designers, by highlighting the variety of issues that need to be considered. Guidelines need to be given with extra information, such as scoping rules and caveats, to avoid the guidance being too vague or conflicting (Reisner, 1987). Therefore, for each GDP in the theory, a concrete design guideline was written which included extra supporting information, from corresponding components of the theory. There were four parts to each design guideline:

- *Design advice* – which covered the need to incorporate the attributes of the GDP in a VE design.
- *Motivation* – which gave reasons for the importance of the advice, based on supporting interactive behaviour (as detailed in the interaction models) and/or avoiding usability problems (as detailed in the problem prediction rules).
- *Context-of-use* – which gave information about the relative importance and applicability of the advice, based on the users' profile (as detailed in the user knowledge sources) and also the type of application.

- Two practical *examples* of the implementation of the guideline.

For example, the design guidelines for the GDPs *declared available action, accessible object* and *clear system control purpose* were as follows:

GDP: *declared available action*

Design guideline:	The availability for action should be made clear to the user.
Motivation:	Necessary to aid the user in finding available actions during exploration.
Context of use:	Particularly important for exploratory applications. Less applicable where the user has information about actions available in the VE, for example as in the case of VEs accurately modelling activities in a domain well known to the user.
Examples:	In a marketing application for a business park, available actions are highlighted by outlining active objects in red and white. In a virtual supermarket application, the usual flat hand cursor changes to a 'grab' hand cursor when over available actions, such as when over the supermarket trolley handle, for pushing the trolley, and when over purchasable items.

GDP: *accessible object*

Design guideline:	Objects should be easy to access, that is, it should be easy for the user to approach objects and take up a suitable position close to objects.
Motivation:	Necessary when the user is approaching objects and orienting to objects for investigation or for carrying out actions.
Context of use:	More consideration should be given in cluttered environments, where the user is navigating in very restricted areas, or where the user has limited navigation pathways, such as when objects cannot be passed through. More consideration should also be given to the more important objects, such as those that can be interacted with, to the smaller objects and to objects whose position or orientation in the environment can change.
Examples:	In a virtual jewellery exhibition, an access function is provided where the user can double click an item of jewellery to be automatically transported to a close up view of the jewellery at a standard orientation. In a marketing application for a business park, in one of the units a water tank is positioned in a loft area. To aid the user approach the tank, a ladder is included leading up to the tank.

GDP: *clear system control purpose*

Design guideline:	The goal of the system in controlling the interaction should be made clear to the user. There should be a clear indication to the user when control is likely to be returned to her.
Motivation:	Necessary for the user to understand why the control is taking place so that she can benefit from it in any intended ways. Important for the user to be aware how long the control will last, so she can plan future interactions and plan whether she should attempt to regain control.
Context of use:	More important where significant parts of the user's interaction are affected, such as navigation.
Examples:	In a marketing application for a business park, a speech track informs the user that an introductory tour follows which will help familiarise her with the park. The speech track also states that the tour will last for 2 minutes. In a virtual surgery application for training, a trainer agent demonstrates surgical procedures to the user. The agent discusses important points of each surgical procedure before demonstrating it.

Appendix 6A gives a complete list of the guidelines written from the 45 GDPs in the refined theory. Having written all the guidelines, the next step was to present them in a suitable form for designers.

6.2 Hypertext tool to present guidelines

For presenting the guidelines, a form was required which would fit in with current design practice. Knowledge about design practice, gained from the designer study in chapter three, was used to help target the presentation of the guidelines to VE designers. Designers were shown to generally have a poor understanding of usability, and had informal design approaches and short development times (i.e. a number of weeks). Therefore, a presentation style was required which would be simple to understand and allow quick, easy and flexible access to the guidance. As a result, a hypertext tool was designed to present the guidelines in a structured and accessible form. The set of correspondence rules could have been included to advise on the importance of guidelines, given likely user knowledge and interactive behaviour, but this would have added another level of input and processing complexity to the tool, which was not felt to be justified.

The hypertext tool structured guidelines according to the environment elements involved (e.g. events, objects and actions) and the general development stages when the guidelines would apply. Development stages were defined using knowledge of the VE design process, from the designer study, and the common stages of activity found in system development methods, such as task analysis and presentation design (e.g. KAT/TKS, Johnson 1992, and MUSE, Lim and Long 1994). The seven stages outlined were: *define requirements*, *specify components in VE*, *specify interactions*, *design components*, *design interactions*, *build environment* and *evaluate environment*. Separate sections, *explanations* and *documentation*, were outlined for relevant declarative knowledge on VE design and advice on documenting designs.

A demonstration version of the tool was implemented in HyperText Modelling Language (HTML), for evaluation as a format for presenting the guidelines. This demonstration version included a sub-set of the guidelines (12 in total), applying to the two design stages and covering these three VE elements:

- Objects – four guidelines were included for GDPs *distinguishable object*, *identifiable object*, *clear object type/significance* and *accessible object*.
- Actions – five guidelines were included for GDPs *declared available action*, *clear action purpose*, *declared action sequence*, *executable action* and *declared action effect/success*.
- System control – three guidelines were included for GDPs *declared system control commencement/termination*, *clear system control purpose* and *declared available actions during control*.

These particular guidelines were chosen to cover a range of major issues for objects, actions and system control. The guidance for objects was given in the *design components* stage, and guidance for actions and system control were given in the *design interactions* stage. For each element type, a bullet point list of guidelines was given, which was linked to full descriptive texts (taken from the definition of the guidelines in appendix 6A), and screen-shots were included to illustrate examples. A navigation panel enabled rapid access to guidance for different stages, although only the introduction and two design stages were available in the demonstration version. Figure 6.1 shows the starting screen of the demonstration tool. Figure 6.2 shows the bullet point list for the object guidelines in the design components stage, and figure 6.3 shows the full description and example for one of the object guidelines, taken from the GDP *identifiable objects.* Appendix 6B gives the full hypertext in the demonstration version of the guideline tool.

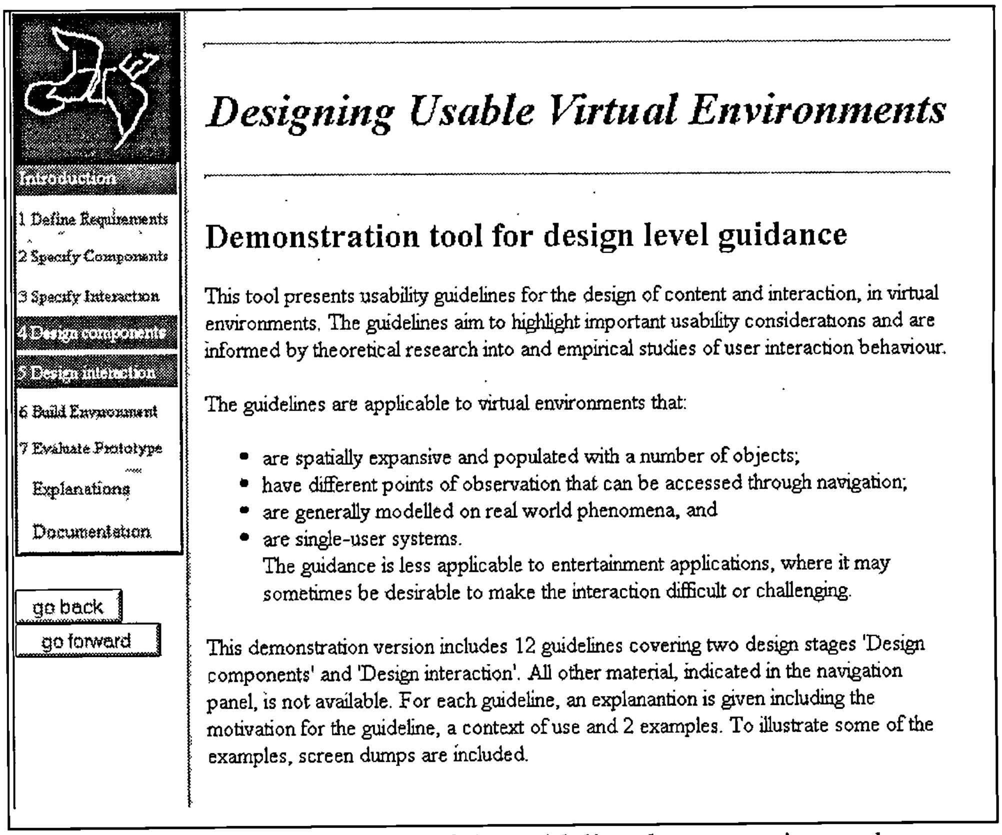


Figure 6.1: The starting screen of the guideline demonstration tool.

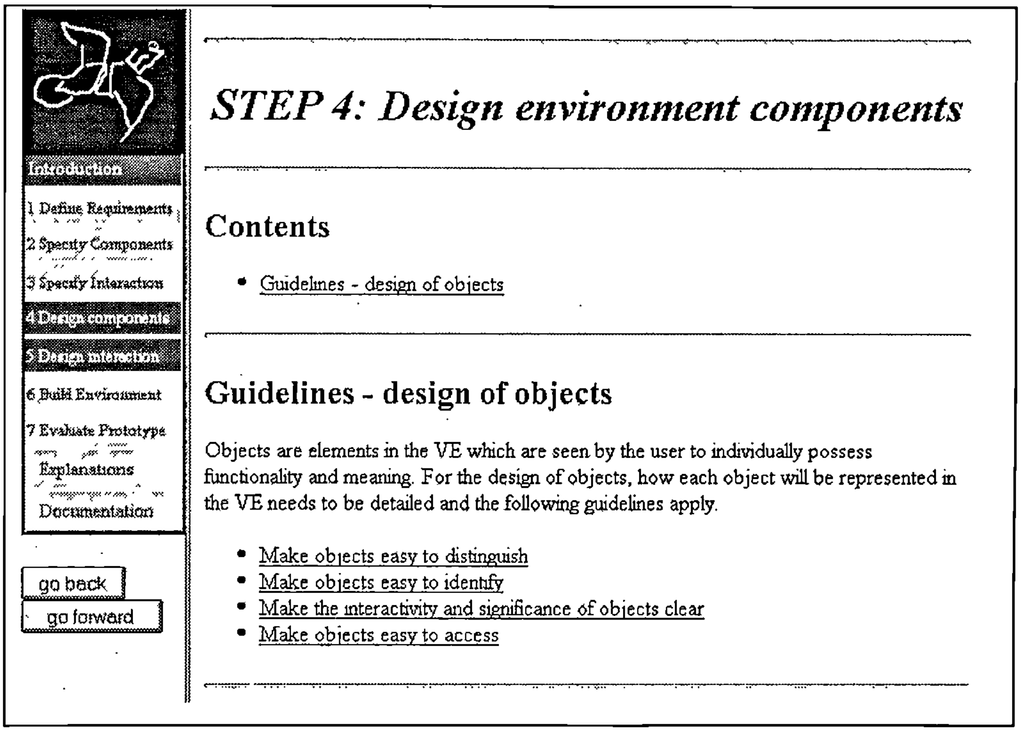


Figure 6.2: The bullet point list for the object guidelines in the design components stage.

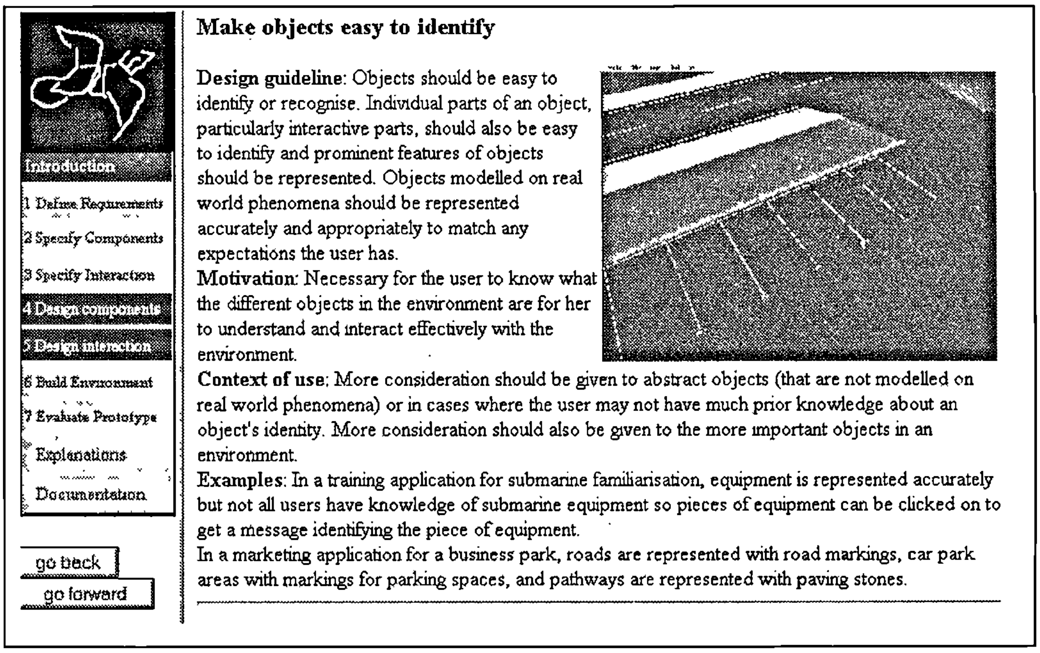


Figure 6.3: One of the guidelines in the tool for designing objects (taken from the GDP *identifiable objects*).

6.3 Evaluation of the tool

The tool and design guidelines were tested using expert evaluation and critiquing to gain direct qualitative feedback about their usability and utility.

6.3.1 Study method

A range of nine, focused design scenarios were defined for evaluating the guidance. The scenarios provided concrete descriptions of design elements, with sufficient detail to allow design implications to be inferred and reasoned about. Table 6.1 lists the nine scenarios that were used. Five of the scenarios involved designing an original element (object, action or system control) from a given set of requirements. Another two scenarios involved re-designing an element chosen by the designers from their previous experiences, so that the guidance in the tool could be tested with actual design problems. There were a further two scenarios for re-designing elements from given descriptions of a current design, including a list of usability problems with the design. Figures 6.4 and 6.5 show screen-shots of the elements described for re-design. The elements in the nine scenarios were chosen so that all 12 guidelines, in the demonstration version of the tool, would represent an important issue in at least one of the scenarios. Full descriptions of the scenarios are given in appendix 6C.

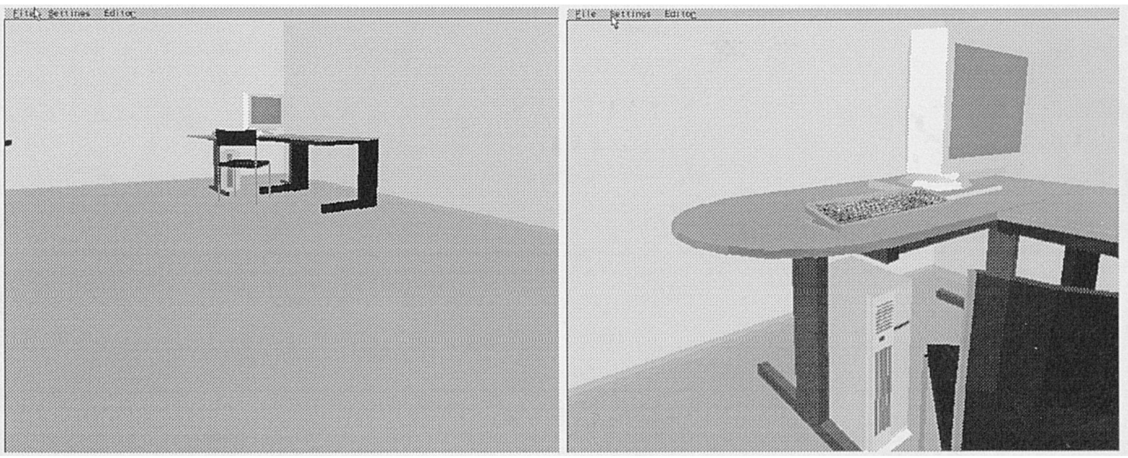


Figure 6.4: The non-interactable computer described for the re-design object scenario.

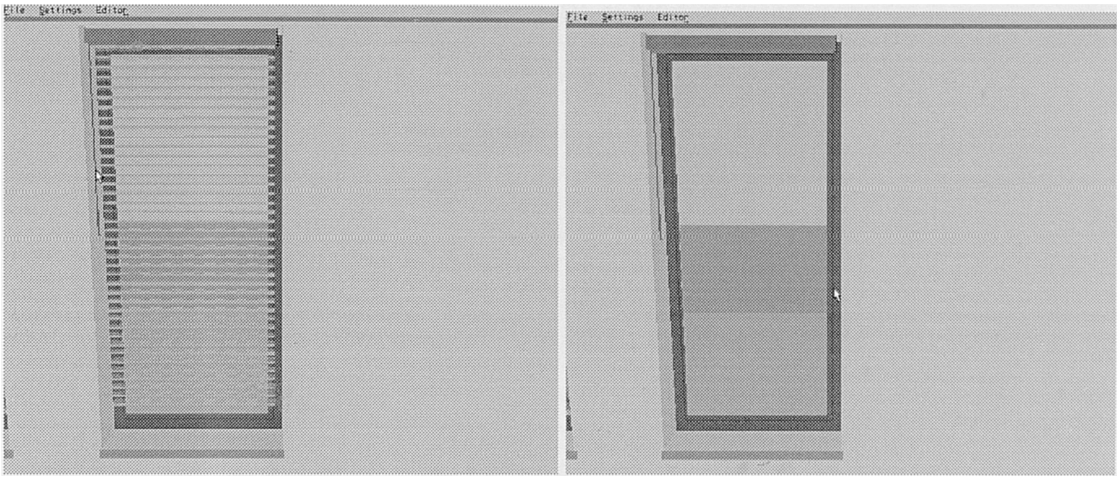


Figure 6.5: The active blind described for the re-design action scenario.

Scenario type	VE element	Description
Design a new element, from given requirements	Object	A corridor with passive and active parts - accessible and inaccessible doors.
		A hatch connecting rooms in a submarine, with two representations for open and closed.
	Action	A simple action for opening and closing the hatch (*see above*).
		A two stage action for producing a standard printout from a printer and then reading the printout.
	System control	An automatic tour of places of interest at a tourist town, including suspending and stopping the tour.
Re-design an element, from the designers' previous experiences	*Either* object, action or system control	*Specific element chosen by the designers.*
	Either object, action or system control	*Another element chosen by the designers.*
Re-design an element, from a given description, including current usability problems	Object	A non-interactable computer, which users currently expect to be able to interact with. (see figure 6.4)
	Action	Drawing a blind up and down and rotating its slats, which users currently have difficulty working out how to do. (see figure 6.5)

Table 6.1: The nine design scenarios used in the evaluation.

Four VE designers from the same organisation took part in a one-day evaluation study. The designers filled in questionnaires (see appendix 6D) about their background, which is summarised in table 6.2. The designers had an average of 2.5 years experience building VEs. They carried out various roles, including working with internet applications, and had designed VEs for a variety of application areas. The designers commonly had computing and graphics backgrounds, and mainly used the Superscape toolkit for development.

Designer	**A**	**B**	**C**	**D**
Role	project manager	senior designer	internet manager	design & marketing
Experience				
VE (years)	3	3	3	1
computing/ programming	✓	✓	✓	✓
electronics		✓	✓	
computer graphics	✓	✓	✓	
art/ photography		✓		
sound engineering				✓
Applications developed				
product design/ evaluation				
marketing	✓	✓		✓
training	✓	✓		
education				
entertainment		✓	✓	✓
information visualisation	✓	✓		
groupware				
tele-operation				
other - internet			✓	
Tools used				
superscape	✓	✓	✓	✓
others – e.g. virtus			✓	✓

Table 6.2: The background of the designers.

The designers were given a 90-minute presentation about the research, including descriptions of how the guidelines had been derived, the tool presenting the guidelines and the evaluation session they were taking part in. The designers were then given the design scenarios, one by one, and asked to produce paper storyboards (sequences of sketches) of design solutions using the guidance in the tool. The designers were given advice on producing storyboards (see appendix 6E), and 20 minutes to complete each scenario. They had access to the demonstration version of the tool throughout. While the designers worked, their conversations were recorded for reference. After all scenarios had been completed, designers filled in retrospective questionnaires eliciting their views about the guidance and tool, see appendix 6F. Finally, the designers took part in a discussion session, with the two study organisers, to explore the issues that had arisen during the use of the tool.

The designers worked in groups of two or three. This was because designer A joined the study late and was only able to take part in the last four design scenarios. The first five scenarios were carried out by one group of three designers (B, C and D). The other four, re-design scenarios, were carried out by two groups, with designers A and D working in one group and designers B and C working in another. The specific elements chosen by designers from their previous experiences, for scenario type 2, were two objects and two actions:

- a board object for a labyrinth game;
- a watchman object (portable, mini television/video);
- standard actions in a virtual world wide web application, and
- the action of adding a new card to a personal computer.

Storyboarding was chosen to record design solutions, because of time constraints which made constructing designs impractical, and to avoid designs being constrained by the hardware and software being used. Also, storyboarding is a common technique used by professional designers during the early stages of user-interface design (Landay and Myers, 1995).

The storyboards of design solutions were collected and scored for usability, using the GDP definitions. The scoring scheme involved consideration of how well each relevant guideline (or GDP) in the tool, had been incorporated in the design solution. For example, for object design scenarios, support for the four relevant GDPs (*distinguishable object*, *identifiable object*, *clear object type/significance* and *accessible object*) was assessed. Individual scores for GDPs were either 0 – for no consideration of the associated usability issues, 1 - for a partial or inadequate consideration, or 2 – for a full consideration of all associated usability issues. For example, for the GDP *identifiable object*:

- a score of 2 was given for the design hatch object scenario (group B C D) because the design solution included a message identifying the hatch, as the mouse cursor moved over it;
- a score of 1 was given for the design corridor object scenario (group B C D) because the design solution included detail about realistic door representations,

but no detail about indicating different door/room types (e.g. exit door or toilet door);

- a score of 0 was given for the re-design gameboard object scenario (group A D) because the design solution did not include any detail about providing identification information for the board or its individual parts (e.g. ball and holes).

A simple scoring scheme was used to give a general indication of the usability of the design solutions, which were inevitably limited because of the time available and recording method used (i.e. storyboarding).

The retrospective questionnaire covered important aspects of the usability and utility of HCI design guidance, described in detail by Lim and Long (1994). Usability was investigated through questions about the understandability, accessibility, applicability, specificity (pitched at appropriate level of description), and acceptability (compatibility with established design practices) of the guidance in the tool. Learnability, another aspect of the usability of guidance, could not be covered because longitudinal data was not available in this one-day study. Utility was investigated through questions about the potential impact of the guidance in the tool, either in highlighting or uncovering usability issues, helping to address usability issues, helping to validate designs from a usability perspective, or, generally helping to develop more usable VE elements.

6.3.2 Results

6.3.2.1 Usability of design solutions

Tables 6.3 to 6.5 give the usability scores for each design solution produced by the designers, for object, action and system control elements. Usability scores for the 13 different design solutions ranged from 50% to 100%, median 75%. Therefore, on average good usability scores were attained. Scores by element type ranged from 63 to 100% for objects, 60 to 80% for actions and 50% for the single system control. Therefore, scores for actions and the system control were slightly lower than scores for objects.

Subjects in group	GDP object	distinguish-able	identifi -able	clear type/ significance	accessible	total /8	%
B C D	corridor	2	1	2	0	5	63
	hatch	2	2	2	1	7	88
A D	game board	2	0	2	2	6	75
	computer	2	2	1	2	7	88
B C	watchman	2	1	2	1	6	75
	computer	2	2	2	2	8	100
	Usage (no. designers)	4	4	4	4		

Table 6.3: Usability scores for each solution produced for the design object scenarios.

Subjects in group	GDP action	declared available	clear purpose	declared sequence	execut-able	declared effect	total /10	%
B C D	hatch	2	1	2	1	1	7	70
	printout	2	1	2	1	2	8	80
A D	world-wide-web	2	1	2	1	0	6	60
	blinds	2	1	1	1	1	6	60
B C	PC card	2	2	2	1	1	8	80
	blinds	2	1	2	1	0	6	60
	Usage (no. designers)	4	2	1	3	2		

Table 6.4: Usability scores for each solution produced for the design action scenarios.

Subjects in group	GDP system control	declared commence/ terminate	clear purpose	declared actions	total /6	%
B C D	guided tour	1	1	1	3	50
	Usage (no. designers)	2	2	2		

Table 6.5: Usability scores for each solution produced for the design system control scenarios.

Most individual GDP scores were either 1 or 2 (53 of the 57), indicating that usually the guidelines were addressed to some extent. Points were generally lost because of missing information in the storyboards, rather than bad designs. For example, if no mention was made of how the design was addressing an issue then a score of 0 was given. Scores for *distinguishable object* and *declared available action* GDPs were always the maximum 2, indicating that these were perceived to be important by the designers or were reasonably easy to implement. No GDP was associated with all zero scores. Scores for the system control GDPs, and the GDPs *executable action* and *declared action effect* were the lowest. The system control scenario appears to have been complex and it may have been difficult to cover all issues arising with the GDPs. Designers tended to pay little attention to making actions easy to execute, perhaps because a standard 2D mouse was used for interaction with a simple clicking interaction style. Designers also tended to pay little attention to action feedback, as their storyboards usually did not describe this area well. This may have been because it was at the end of the action cycle and received less attention, since designers felt they had already addressed the main issues.

In the retrospective questionnaires, designers were asked which guidelines they had used (see bottom row of tables). Object GDPs were most often mentioned. *Declared action sequence* was mentioned by only 1 of the 4 designers, but this may have been because of the simple interaction style involved.

6.3.2.2 Reactions of designers

Table 6.6 gives the responses of the designers to questions in the retrospective questionnaire. Responses were given on a scale of 1 to 7, where 7 represented the most positive response. Responses were generally positive, with averages for individual questions ranging from 4.5 to 6.0 (mean).

The average response for the usability questions was 5.18, and for the questions on utility the average was 5.25. The most positive response was for question 6, about how acceptable the tool would be in practice. Objects were felt to be slightly easier to apply the guidance to than actions or system control. Designer *B* felt the tool did not improve the overall usability of design solutions very much (question 11; response of 2), but gave no reasons for this. The retrospective questionnaire also included an open question on possible conflicting advice in the tool; however, the general response was that there was little conflicting advice.

Q. No.		Question on	A	B	C	D	avg. (1-7)
1	**Usability**	accessibility	6	6	5	6	**5.75**
2		understandability	5	4	6	5	**5.00**
3a		applicability for objects	6	5	6	4	**5.25**
3b		applicability for actions	5	5	5	3	**4.50**
3c		applicability for system control	6	3	4	6	**4.75**
4		specificity - pitched at appropriate level	6	3	5	6	**5.00**
6		acceptability - compatible with design practice	6	6	5	7	**6.00**
8	**Utility**	uncovered usability issues	7	5	4	7	**5.75**
9		helped address issues	6	4	5	6	**5.25**
10		helped validate usability of designs	4	5	6	6	**5.25**
11		improved usability of design	6	2	6	5	**4.75**
12	**Other**	confidence in guidelines validity	5	4	5	5	**4.75**
13		relevance of scenarios chosen	5	5	7	4	**5.25**
14		relevance of guideline examples	6	5	4	4	**4.75**

Table 6.6: Responses of the designers to questions in the retrospective questionnaire.

Designers made comments in the retrospective questionnaires and during the discussion session. Comments related to these main areas:

How the tool was used by designers -

- Designers felt the tool was useful as a reminder of all the usability issues that needed to be considered. The bullet point guideline descriptions were especially useful.
- The guidelines were used more for assessing, rather than brainstorming designs, probably because they could be more naturally applied as a usability checklist to a current design idea. For example, the guidelines were used to identify problems in the design elements chosen from the designers' experiences.
- There was often direct (copycat) reuse from the examples given with the guidelines, such as re-using the technique of outlining active objects.

Issues with the structure and content of the tool –

- Some possible improvements to the design of the hypertext in the tool were suggested. For example, one designer suggested that the guideline sub-categories (guideline, motivation, context of use etc) could be listed and made expandable.

- There were some problems or disagreements with the detailed content of the tool. For example, one designer felt that the tool stressed 'ease' and perhaps underestimated the importance of complex actions. Similarly, one designer believed mouse-dragging was an important interaction method and felt the tool discouraged this.

Useful extensions to the tool -

- The designers wanted more examples and techniques for incorporating the guidelines, to enable re-use by copying. A broader selection of examples in a separate section was felt to be useful.
- More guidance was needed for difficult areas, such as system control, and important areas, such as navigation.
- The guidance perhaps could be extended to help address the issue of how faithfully to model reality, and the possible trade-offs between usability and realism. For example, it was noted that some real world objects may be naturally difficult to interact with, such as the blind. Similarly, some objects may naturally encourage interaction, such as the computer, but all actions may not be available in the VE.

6.3.3 Discussion

The results are encouraging and provide useful qualitative feedback about the guidance and tool. The design solutions produced generally had good usability scores, indicating that the designers were able to incorporate interaction support in their designs, through the use of the tool. Objects appeared to be easier to design for usability than actions or system control. This may be because VE designers focus more on representation design, many having graphical backgrounds, than interaction design which they implement less well. Alternatively, designing actions may be a naturally more complex and demanding interface design problem, than designing objects, because dynamic exchanges or dialogues between the user and VE are involved.

All of the twelve guidelines were used individually, although some were used more and implemented better than others. Object guidelines and some of the action guidelines were used most often and better implemented. Sometimes guidelines were not used well because interactions involved were felt to be too simple to warrant usability support/consideration. However, it appears that further support on implementing guidelines is needed for complex areas, such as system control, and perhaps also for easily forgotten areas, such as action feedback.

Designers' perceptions about the guidance tool were very positive. They felt it was usable and that it helped to analyse and address the usability of VEs. They also felt that it could have an important impact on the design process and could fit in with existing design practices. Designers felt it was most useful as a checklist of all the usability issues that needed to be considered for an element. Designers also tended to directly re-use techniques given in the examples. Such copycat re-use strategies are well known and may be due to designers taking the mentally easy approach of copying techniques rather than reasoning and adapting or defining new techniques (Sutcliffe and Maiden, 1990). Useful extensions to the tool included further examples of techniques for achieving each guideline's effect which designers could directly re-use, and help in addressing the issue of when it would be important to model reality faithfully. Indeed, the issue of judging appropriate levels of realism has been highlighted previously (e.g. Herndon et al., 1994; Boyd and Darken, 1996) and recently COVEN (1997b) also highlighted the difficult trade-off between adding support information and maintaining the realism and naturalness of the VE.

Therefore, the study has provided support for hypothesis 5 of the thesis:

> **H5** The design properties (GDPs) can be presented in a usable form to support VE interface design.

The guidance in the tool, developed from a subset of the GDPs, was found to be usable and appeared to help in designing VE elements for usability. The evaluation was carried out with experienced designers, who produced solutions for a varied set of design scenarios, some of which were chosen by designers from their past experience. Therefore, the tool appears to represent an effective way of delivering the theory knowledge to industrial designers. The study has also highlighted some improvements and extensions that can be made to the tool. However, further evaluation work on the tool would be desirable, because of the limited number of designers that were available for the study. The final chapter of the thesis, discusses further work on the tool and other implications and future directions of the research.

Chapter 7

Discussion and Conclusions

This chapter summarises the thesis research and concludes with a discussion of implications and possible future directions.

Chapter 7

Discussion and Conclusions

7.1 Summary

In chapter one, the aims and approaches of this thesis were outlined. The objective was to develop guidelines to help address the problem of designing usable VE interfaces, using interaction modelling as a theoretical base. Chapter two summarised previous research, which fell into two main areas. Research on VEs focused on technology and techniques, although some research had begun on the psychology of VE interaction, in particular perception and spatial cognition. Research on human-computer interaction included general approaches to modelling and design guidance, and specific theories for conventional interfaces, such as direct manipulation.

Chapters three to six described the thesis research, which was structured according to five general hypotheses. Hypothesis one stated that '*there is a need for interface design guidance specifically for VEs*'. This was investigated in chapter three through studies of the design and use of VEs. These studies highlighted the existence of major usability problems and showed that designers lacked a coherent approach to interaction design, thereby supporting the need for design guidance for VE interfaces.

Chapter four described a theory of interaction for VEs, which included predictive models of interactive behaviour and a set of generic design properties for supporting that behaviour. The interaction models elaborated on hypothesis two, that '*general patterns of interaction with VEs can be predicted*' and the generic design properties elaborated hypothesis three, that '*design properties required for interaction can be predicted using the general patterns of interaction*'. Both hypotheses two and three were then tested in chapter five, in an evaluation of the theory of interaction. Empirical studies of interaction behaviour were used to test predictions in the models of interaction. Usability studies were used to test whether missing generic design properties were linked with usability problems. Results provided general support for the theory and indicated specific refinements that were required. A controlled study was then carried out to test hypothesis four, that '*interaction can be improved by implementing the design properties*'. Significant improvements in interaction were found with the use of a VE, after the generic design properties had been implemented.

Chapter six described the development of design guidelines from the theory of interaction. A hypertext tool was designed to present the guidelines, which was evaluated with industrial designers to test hypothesis five, that '*the design properties can be presented in a usable form to support VE interface design*'. Results were encouraging and indicated that the tool and guidelines were useful in helping to design VEs for usability.

Therefore, this thesis has met its objective of developing usability guidelines for VEs. The approach taken involved interaction modelling to help understand and break down interaction behaviour. Abstract properties were then defined for supporting this behaviour which, following evaluation, were translated into design guidelines. The thesis concluded by showing that the guidelines presented in a structured tool could help to address the practical problem of designing usable VE interfaces. The remainder of this chapter discusses implications and contributions of the thesis research. Limitations of the research are detailed, followed by possible future directions, either through elaborating the research or using it to inform further theories, methods or tools.

7.2 Implications

The thesis research has important implications for the VE and HCI fields. The research has helped define the problem space for VE interaction and usability, provided predictive theories, and practical guidance and tools.

7.2.1 Contributions to research on virtual environments

7.2.1.1 Designing virtual environments

The thesis has provided an improved understanding of how VEs are designed and what design issues exist, which is important in informing the development of methods and tools for VE design. Previously, isolated experiences and techniques had been cited, but the study of designers has provided broad knowledge of the design process, including problems in design, guidelines applied, and the priorities that designers work towards. This knowledge can indicate required method stages, issues to be addressed and guidelines to be incorporated into any VE methods developed. The study has also indicated how VE design differs from the design of conventional systems, such as the close modelling of a real world domain.

The study highlighted other areas for further research:

- An outstanding issue for designers was the general poor understanding of VE concepts and potential applications for VE technology, which needs to be addressed. For example, environments tended to be copies of real world models and this may be because clients and designers did not realise the potential for abstract environments.
- Improvements in VE technology are needed, to avoid designers being forced to limit the functionality of VEs. For example, few designers built immersive VEs and no designers mentioned using gloves or haptic interaction devices. Designers also wanted improvements in current VE toolkits, such as better facilities for handling complex objects and behaviour.

7.2.1.2 Interacting with virtual environments

The thesis has provided an improved understanding of how users interact with VEs, which is important to inform the development of guidelines and evaluation methods for interface design (Reisner, 1987). The models of interaction give a breakdown of the major activities and common patterns of behaviour with VEs. Previously, there were fragments of knowledge about interaction, for example, the importance of the virtual embodiment, sense of presence, and the task of navigation and spatial understanding had been noted (Benford, 1995; Slater et al., 1995; Rushton and Wann, 1993; Darken, 1995). However, no comprehensive models had been developed to explain or predict these aspects, within the context of user interaction, and link them to specific usability/design requirements. Few user studies had been carried out to investigate interaction in VEs and evaluate claims. The interaction models and evaluation studies represent important work in addressing this shortfall.

More recent work has been carried out on understanding interaction behaviour in VEs, focusing particularly on navigation. Benyon and Höök (1997) discuss navigation through general information spaces (e.g. hypertext and visualisations). They suggest that during exploration the user browses, scans and wanders through the information space. The user may want to identify objects, categorise and find information about the objects. These ideas about the general navigation of spaces correspond with descriptions of exploration and object investigation in the models of interaction. For example, task action mode includes a stage, added to the original Norman model (Norman, 1988), for inspecting objects.

Jul and Furnas (1997) go further and offer a detailed framework for the navigation process (see figure 7.1), again relating to general information spaces rather than specifically VEs. Their framework involves the steps: *form goal*, *decide strategy*, *acquire data* (e.g. whether progress is being made), *scan*, *assess*, *form conceptual model* (e.g. a cognitive map) and *act* (e.g. move current position). Navigation subtasks are locomotion, steering, traversal of larger distances from sequences of steering steps, decision-making about where to go, either following a route, finding a route or responding to the environment, and map building. Jul and Furnas's navigation framework provides a more elaborate description of navigation than that given in the explore navigate model. However, the explore navigate model captures the key activities involved in navigation, which the evaluation studies showed to be 'scan', 'plan' and 'navigate' (see figure 5.23). These activities are generally represented in the framework as the inter-linked 'scan', 'assess' and 'act' stages. It is likely that, in situated and skilled navigation, the earlier stages of the framework (e.g.

form goal, decide strategy) will be returned to less frequently or involve little conscious processing.

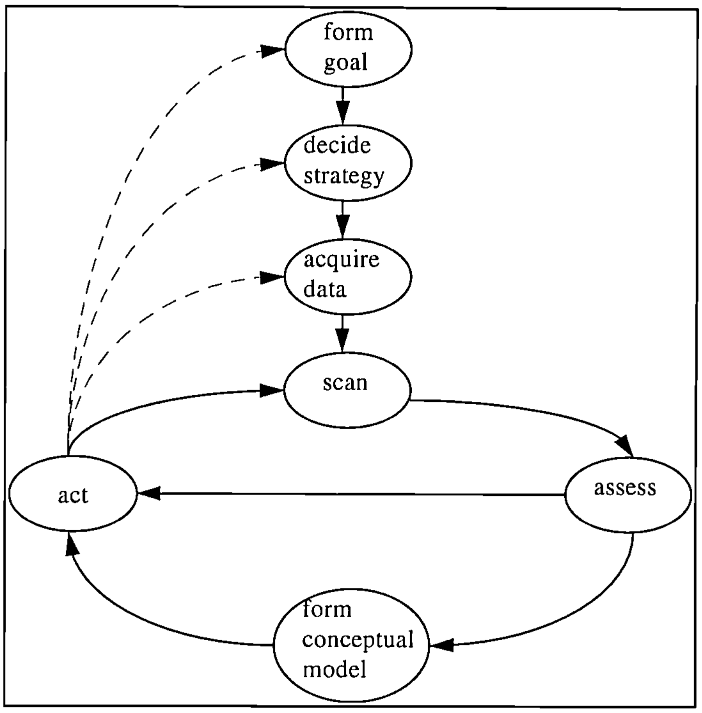


Figure 7.1: A general framework for navigation, from Jul and Furnas (1997)

The interaction models in this thesis provide a broader view of interaction in VEs than this more recent research. Navigation and wayfinding are important and common activities, so a more thorough understanding is useful for generating detailed advice on supporting them. Therefore, the two approaches complement each other. The interaction models provide an overview of interaction behaviour, including all the general activities that need to be supported, and show how different activities are related. The more recent research of Jul and Furnas and Benyon and Höök helps provide a more detailed understanding of some important activities involved. The research also confirms the importance of certain behaviour defined in the models, including exploration and identifying and inspecting objects.

Furthermore, the evaluation studies in this thesis provided not only support for the models, but more extensive and important knowledge about user interaction, such as:

- The studies showed that users can find it difficult to use external sources of information, such as documents, whilst interacting with a VE. This may be a side effect of the 'presence/immersion' factor, in that users become absorbed in the VE and either forget about their surroundings, and any external artefacts available, or cannot selectively attend between external sources and the VE. Therefore, it may be preferable to provide required information within the VE interface itself. For example, in a sports application, statistics about athletes and events can be overlaid on a virtual scene, or information kiosks provided (Bolter et al., 1995).

- The studies indicated that for most users navigation and orientation is a skill acquired reasonably quickly. A few users appeared to suffer severe problems, in particular nausea and continuous navigation difficulties. Both these problems are subject to individual differences and have been reported previously (Kennedy et al, 1997; Höök and Dahlbäck, 1992; Benyon and Höök, 1997). The evaluation studies indicated that nausea can occasionally occur with desktop VEs and supported the link between nausea and unusual orientations (Oman, 1993). The studies also found problems in spatial steering to be a major component of severe navigation difficulties, and demonstrated the extent to which such difficulties can disrupt interaction for the user concerned. Therefore, special consideration seems necessary for users who suffer severe nausea, orientation or navigation difficulties.

7.2.1.3 Supporting virtual environment interaction

The thesis has provided an improved understanding of how to support user interaction in VEs, in particular what properties are required of a design and what are the potential usability problems. Such an understanding is key to informing research on VE guidelines and evaluation methods. Previously, a few usability issues had been identified, such as disorientation (McGovern, 1993), and isolated techniques had been proposed to support interactions, such as functions facilitating precise object alignments (Buck et al., 1996; Venolia; 1993). Research had begun on requirements to support perception (Rushton and Wann, 1993) and general principles had been defined for supporting wayfinding (Darken, 1995). However, the theory of interaction provides a more comprehensive and broad set of design properties and likely usability problems. Furthermore, both are based on a model of interaction for VEs and have been evaluated. Evaluation studies have indicated the more critical issues and shown that the proposed design properties lead to significant improvements in usability. The theory can be used to reason about how supporting interaction in VEs differs from that for conventional systems. For example, it indicates the novel issues to be addressed with VEs, such as supporting the interpretation of system behaviour. Conversely, the theory can be useful in determining what existing HCI principles and research can be borrowed for VE design, for example basic support of task planning and action execution.

More recent work has been carried out on requirements to support interaction in VEs, following from research on understanding navigation. Furnas (1997) discusses basic requirements for navigation in abstract information spaces. For example, the navigator should be able to find the shortest paths to targets from the current view, which means that available routes need to indicate not only the next node but also all other nodes available in that direction. Jul and Furnas (1997) propose specific techniques for supporting navigation, such as providing information about proximity to target and cues to lead to distant targets (e.g. signs). Their advice overlaps with generic design properties in the spatial layout category. For example, requirements to provide information to help find targets is covered by the generic design property *locatable objects/areas of interest* and requirements for information about the shortest paths is covered by the GDP *identifiable optimal routes*. While Furnas offers more detailed and formal requirements that each node needs to satisfy for effective navigation (e.g. target scents), the GDPs cover wider navigation issues such as exploration of areas and overall spatial understanding (see *clear visited areas* and *discernible spatial structure*).

Charitos and Rutherford (Charitos, 1996; Charitos and Rutherford, 1996; Charitos, 1997) present requirements for the spatial design of VEs, using architectural theory. They define a taxonomy of spatial elements for VEs, such as landmark, place, path and domain, and give detailed requirements for each, such as:

- The environment should be structured into *domains*, by means of places and paths.
- *Paths* should suggest a movement direction, have a clear structure and clear start/end points.

- *Landmarks* should be singular, easily identified/recognised and have a prominent spatial location.

Again, the requirements correspond with the design properties in the spatial layout category, specifically the need for a clear spatial structure. However, Charitos and Rutherford offer more detailed advice organised around common architectural elements (domains, places, paths etc.). Furthermore, since the requirements advise on detailed spatial structure, they are more applicable to VEs that do not have to follow a pre-defined spatial layout. This is a narrower focus looking more at abstract environments, because VEs modelled on a real world domain tend to have a defined spatial layout.

Ingram et al. (1996) also discuss the use of architectural features to support the navigation of 3D visualisations. They offer less detailed advice than Charitos and Rutherford, but do empirically show that basic architectural enhancements can aid search tasks. They also compare virtual city forms (e.g. tree and labyrinth) and simple wayfinding strategies. For example, in a computer simulation, they show that a walking strategy, incorporating the use of long lines of sight, leads to faster target acquisition in certain city layouts than a random walking strategy. Consideration of different wayfinding strategies and city forms could prove useful for more intelligent, system-initiated support in facilitating and encouraging optimum navigation strategies for different VEs. On a similar theme, Benyon and Höök (1997) suggest active ways of supporting navigation, such as selecting the next most appropriate destination, indicating relevance or using an agent to help the user find their way around. Conversely, the generic design properties focus on providing more passive basic support and the necessary information during interaction, and do not cover differences in navigation strategy or the layout of the VE. However, such an approach may be a useful complement to the design properties by adding intelligent help to more fundamental support.

To summarise, there has been a variety of recent research on navigation support, both specifically for VEs and for information spaces in general. The research has indicated detailed requirements for navigation and wayfinding. The design properties in this thesis provide a more comprehensive and broad set of requirements for supporting interaction, covering navigation as well as other activities, such as task planning, exploration, object interaction and interpretation of system behaviour. However, detailed more low-level principles for navigation and wayfinding are useful, and complementary to the generic design properties, because these are key activities that need to be supported effectively. Therefore, this recent research can be combined with the GDPs to provide a comprehensive set of usability requirements for VEs which include detailed principles for critical issues.

Recent research has also been carried out on design guidance for collaborative VEs. COVEN (1997a) presents some preliminary, general guidelines based on experiences in the development of collaborative VEs, such as MASSIVE. For example:

- Provide the user with a virtual embodiment he can identify with.
- Enable users to switch smoothly between individual and collaborative tasks.
- Employ realism to aid recognition and understanding of objects and surroundings.

Some general HCI guidelines are also included, such as the requirement to make clear what actions are available. The guidelines are general and will therefore require further detail, so they can be applied in practice. Some of the areas overlap with the more specific generic design properties, such as the requirement to aid recognition and understanding which is covered by several properties, including for example *clear object role*. This is interesting to note in that it provides support for the suggestion that the GDPs are important for collaborative VEs, although they may not cover all the relevant issues. Also, since the COVEN guidelines were derived from practical experiences, rather than from a theoretical base, similarities with the GDPs provide additional practical support for the theory.

7.2.1.4 Design methods and tools

The thesis has provided a set of concrete design guidelines and a tool to present the guidelines in a structured and accessible form. Previous research had focused on tools and guidance for constructing the graphics for VEs (e.g. Singh et al., 1994). However, no comprehensive methods or tools had been reported for usability guidance for designing VE interfaces, although a limited method had been proposed specifically for the design of viewpoint controls (Drucker and Zeltzer, 1994). Recent work has begun on methods for general development. D'Cruz et al. (1996) outline the development of an industry focused framework for identifying appropriate applications, translating the applications into the most appropriate VE and evaluating usability and utility. They plan to use practical experiences in VE development to detail, test and then refine the framework. Their framework has a wider scope than the guidelines and tool in this thesis, covering all development activities as opposed to focusing solely on usability. The framework is useful in showing how different activities fit together, although there is little discussion of how the framework differs from those for conventional systems, and it lacks detailed guidance for individual activities. On the other hand, this thesis provides detailed, comprehensive and tested guidance for the key issue of usability in development. The thesis also advances the general aim of providing designers with a systematic alternative to pure craft practice, by allowing them to work according to a set of tested guidelines. This can deliver a more engineering level of practice (cf. Long and Dowell, 1989) that has the advantage of being more systematic and makes knowledge accessible to experts, for reference, and novices, for training. Ultimately, the guidelines and tool provide a route to improving the usability of VE interfaces by supporting the design process.

7.2.2 Contributions to research on human-computer interaction

7.2.2.1 Defining interaction

For the field of HCI, the thesis has provided theoretical work helping to define the nature of VEs as an interface type and the nature of interaction with VEs. The models of interaction that were developed clearly augmented the initial model of action (Norman, 1988). The main difference being the addition of exploratory and reactive behaviours to the pre-planning model. The importance of exploratory behaviour and, more generally, display-based and situated behaviour has been widely recognised, for example by Suchman (1987) and Payne (1991). The models reflect the importance of these approaches and, therefore, help relate VE interaction to general approaches for understanding human-computer interaction.

For conventional interfaces, recent work on exploratory behaviour includes the IDXL model of exploration (Rieman and Young, 1996). IDXL has some similarities to the explore navigate model in that it describes interactive, unplanned and weakly goal-driven behaviour. The default action in IDXL is to scan the interface for useful features, using a label-following strategy, whereby controls are identified whose labels match key words in the task description. Scanning for features of interest is also present in the explore navigate model, but labels play a much less prominent role in VE interfaces and, therefore, the label-following strategy is less appropriate. Instead, graphical objects are present in the immediate vicinity which are perceived and identified, and may subsequently be acted upon. Whereas IDXL can computationally model label following, by matching key words for example, the image recognition of objects is a more difficult process to model in detail. In general, computationally modelling the perceptual processes involved in VE interaction is a more complex research challenge, than for text based interfaces.

The LICAI model of exploration (Kitajima and Polson, 1996) also focuses on user interfaces consisting of menus and text cues, and employs text-comprehension strategies. Screen objects are selected as candidates for action (3 in total) based on user goals, using strategies including label following. An action is then selected to be performed on one of the candidate objects. LICAI describes a more systematic form of exploratory behaviour which is likely to be less applicable to VEs, since they tend to have less structured interfaces. This again points to the more complex nature of VE perception and exploration, and hence the suitability of more general models of interaction aiming to predict common patterns of behaviour, as opposed to formal models (e.g. IDXL and LICAI), which aim to make narrow and precise predictions about interaction.

Understanding different interface types is important for HCI research because it helps gain insight into the unique qualities of specific interfaces and the generalisable qualities of human-computer interaction. Generalisable interaction behaviour, such as task planning, action execution and perception, is captured in the Norman model of interaction (Norman, 1988), but this thesis indicates that exploration and scanning are also generalisable activities. For example, scanning for features of interest is important in VEs, but is also found in IDXL, LICAI and Springett's interaction level models for direct manipulation interfaces (Springett, 1996). Activities more specific to particular interface types appear to include the following of text labels for direct manipulation interfaces. For VEs, specific activities include some of the key stages identified, such as navigation in 3D space and approaching and orienting to objects, since these activities do not have an equivalent in models for direct manipulation interfaces. Such a comparison also demonstrates that, although there are some similarities, VE interaction differs in significant ways from its predecessor and also has different modelling constraints (such as the level of precision possible).

7.2.2.2 Modes of interaction behaviour

This thesis has shown that it is important to recognise different modes of interactive behaviour, such as task-based, exploratory and reactive. The model evaluation studies demonstrated that different modes of interaction are inter-linked and co-exist in individual interaction sessions. Traditionally, HCI models have focused on only one mode of interaction, particularly plan-based. This thesis indicates that interactive behaviour is more complex and HCI models need to incorporate the different ways in which a user's interactions may be driven. Recently, Fields et al. (1997) also put forward this view. They argue that relying on a single cognitive perspective for interaction modelling provides too narrow a basis for interface design. The user can have different interaction strategies, based on their objectives. The interaction strategies proposed by Fields et al. follow those outlined in the literature - plan following (e.g. GOMS), planning, semantic matching (e.g. display-based interaction), goal-directed exploration and learning by exploration. The different strategies have different support requirements. For example, the planning strategy requires information about a goal and an action-effect map, while learning by exploration additionally requires a history of actions that have already been performed.

The interaction strategies of Fields et al. are more specific than the three modes of interaction proposed in this thesis. However, they do not propose an interaction strategy for interpreting and responding to system behaviour, as addressed by the system initiative model. Although system initiative mode was found to be less common than task action or explore navigate modes, there was no other part of the models that could have described the patterns of interactive behaviour when system initiated events and controls occurred in the environment. For example, task action and explore navigate modes could not have described the interpretation of events, or the monitoring and regaining of control that was found. Therefore, system initiative mode is important and reactive behaviour is likely to be important for other interface types where the system plays a more active role in interaction. For example, system initiative in other interfaces can include events, such as appointment reminding in a desktop interface, or interaction control, such as animated demonstrations (Payne et al., 1992) for learners. Reactive behaviour can also result from interaction with other users in a system, for example responding to messages received or co-operating on a task.

7.2.2.3 Supporting interaction

This thesis has provided an identification of some general design properties required to support interaction, which can be applied to other human-computer interfaces. For example, figure 2.3 showed how VEs share similar features with other interface types, in particular direct manipulation, hypertext and walk-up-and-use interfaces. Although, specific guidance exists for these interfaces, the GDPs may highlight additional relevant properties or provide an alternative view on usability requirements. For example, to support exploration, as well as making the repertory of available actions salient (Lewis et al., 1990), the GDPs also advise on a *discernible environment content set* and for *declared areas of interest*; requirements which can be applied to walk-up-and-use systems and interfaces in general. Furthermore, whereas existing guidance highlights the need to support the understanding and executing of actions (De Mul and van Oostendorp, 1995; Lewis et al., 1990), this thesis provides an alternative view by considering the understanding of objects and the interpretation of system behaviour, as well as user actions. The GDPs also focus on requirements for more direct support of user interaction through affordances and cues, whereas some conventional guidance emphasises dialogue requirements, error messages, help and document (e.g. Nielsen, 1993). For hypertext interfaces, the thesis provides some novel and applicable guidance on the navigation of large-scale spaces, such as the need for *environment enclosure* so there is a clear boundary to the space, and *clear navigation pathways* through the space. Therefore, the GDPs can contribute to research for other interface types by complementing and extending upon existing HCI guidance.

7.2.2.4 Theoretical modelling and evaluation

The modelling approach used in this thesis demonstrates a successful path to developing models for specific interface types, and developing usability requirements from these models. The approach built upon a general theory (specifically the Norman model of interaction) and this placed the resulting models on a firmer theoretical base. The approach described the process of user interaction and the resources available. Other approaches have focused mainly on process (e.g. the Norman model of interaction) or resource (e.g. Cognitive Complexity Theory, Kieras and Polson, 1985), but both are important in situating guidance and indicating usability requirements. The resources for interaction were seen to be distributed between the system and the user (through the design properties and user knowledge elements), as also advocated by Bibby (1992) and as modelled in IDXL (Rieman and Young, 1996). However, emphasis was placed on the abstract design properties which were used to link the models of interaction to concrete design guidance. The resulting theory was a 'bridging model', as advocated by Barnard (1991), in that it fitted between cognitive theory and guidelines, and aimed to provide a design-oriented theory of user interaction. The theory covered a relatively broad set of interaction issues, for a range of interaction stages and interface elements. Narrower focused theories predict more detailed features of an interface design to ensure usability, but they sacrifice breadth for depth. For example, LICAI (Kitajima and Polson, 1996) can give detailed advice for the design of command labels, but it has little to offer on the design of icons, objects or interactive metaphors. Finally, the approach did not involve formalisation and full computational implementation, as used by other approaches such as PUMS (see Blandford and Young, 1995). With the focus on a less structured interface type, formalisation was not practical (see section 7.2.2.1). However, the approach did benefit from being systematic with a clear rationale, i.e. clear links between behaviour and usability requirements, and a clear motivation for the requirements.

The model evaluation studies demonstrated a successful and practical approach to testing complex, informal process models. The aim was to gain general support for the theory using empirical studies of actual interaction behaviour. Focused set tasks were used to encourage the range of behaviours predicted in the theory, and exploration or more general tasks allowed for free and unprompted behaviour. Protocol analysis techniques were used to analyse (mental) behaviour for comparison with the models. Rigorous analytic methods were employed, such as defining objective coding schemes and matching rules, to counteract the subjective interpretation of observed mental behaviour and the potential for verbalisation errors. Detailed results were analysed to identify validated, weak and untested parts of the theory, and issues not covered by it. These analyses were then used to inform systematic theory refinements.

Therefore, this thesis has demonstrated a useful approach to theoretical modelling and evaluation, where the aim is to address specific problems in interaction design and usability. Since the technology is constantly changing, up-to-date research is required on a short time-scale. In this context, formal theories, complete in breadth and depth, and rigorously validated are less practical, because they generate knowledge too slowly. Better focused research, which provides timely insight into a significant set of issues for designers, and for which general support has been obtained through evaluation may be more appropriate. With this approach, HCI theories can be applied early and then matured through revisions based on lessons learned in practice, rather than attempting to validate them completely before application.

7.2.2.5 Delivering interaction guidance

The approach to delivering interaction guidance, used in this thesis, demonstrates a successful path from theory to application in design. The level of complexity of the theory allowed for a substantial set of guidelines (45) to be developed. The evaluation demonstrated that industrial designers could understand and utilise the guidelines. Previously, HCI theories and techniques have been less successful in delivering design guidance, and encouraging designers to apply it (Bellotti, 1988; Lauesen, 1997). Reasons for the apparent success of this approach are believed to be, most importantly that, rather than directly delivering HCI guidance and techniques, the theoretical knowledge was transformed into a presentation style geared towards designers and design practices. Important aspects of the usability and utility of guidance, such as understandability and compatibility (Lim and Long, 1994), were considered individually when designing the guidance tool. For example, the tool was designed to be simple and flexible. The guidance was structured according to aspects of the design problem (e.g. the user' task, environment objects, actions etc.) and a manageable amount of guidance (3-5 guidelines) given for each aspect. The guidance was also situated in interaction behaviour, thus giving a context for application, which is important and is missing from many standard usability guidelines (Long and Dowell, 1989). Furthermore, designers were given an explanation of where the guidance came from and were shown the benefits of applying the guidance, i.e. in improved usability. These results address some of the problems reported in Bellotti's (1988) study of designers which indicated that designers need confidence in HCI as a discipline, as well as requiring HCI techniques to be both quick to use and uncomplicated. This thesis demonstrates that HCI research can be successfully delivered to designers when guidance is suitably presented and motivation for it is given, which is useful to inform the practical application of other HCI theories.

7.3 Limitations of thesis research

Although the thesis has contributed useful research to the VE and HCI fields, there are important limitations in the research which need to be considered. Limitations with the theory of interaction relate to its level of completeness, precision and complexity:

- *The theory of interaction cannot claim completeness in modelling interaction behaviour, design properties, user knowledge and usability problems.* However, it can claim to model the common and important processes and patterns of activity. Furthermore, it is difficult for theories to claim completeness, unless they are based on a verified and complete model of action, which does not exist for human psychology and behaviour (Hollnagel, 1993). Norman's model of interaction (Norman, 1988), which the theory is based on, is however well established and has been used in research for other interface types.
- *The theory cannot claim to make precise predictions about interaction.* For example, the correspondence rules can only predict that a usability problem may be likely, and cannot than say whether or not it will occur in any situation. Again, it is difficult for a theory to make precise predictions unless all independent variables are known, which is not possible for every potential user and interaction situation. The theory does, however, incorporate major factors, such as the stage of interaction, user knowledge and the design. Other possibilities are to include motivation, spatial skill etc.
- *There are gaps of interpretation between elements in the theory and the real world problem of VE interactions.* For example, the generic design properties are abstract concepts and implementing them in a design involves choices about specific techniques to use and the extent to which properties are incorporated. This was evident in the fact that some GDPs were not implemented effectively in the controlled study and this affected the degree of interaction support they provided. Similarly, whether or not a design property is present is also subject to some interpretation. Such gaps are likely to be unavoidable when using abstractions, which need to be matched with or translated into specific instances. Interpretation may be made more reliable through the use of examples, limiting cases, or perhaps more application-specific properties.

- *The level of complexity of the theory can render the descriptions of interaction and inter-dependencies between components difficult to understand and verify.* The correspondence rules help manage the complexity to some extent by describing the links between various elements of the theory. However, a certain level of complexity is inevitable if the theory is to address a significant set of interaction issues without having to over-simplify the problem space.

Although the theory cannot claim completeness and precision, it does model common and important interaction elements, such as activities and user knowledge, and its complexity enables it to cover a significant set of usability requirements.

The limitations in the evaluation of the theory involve issues of completeness and accuracy:

- *The theory was not completely evaluated, because not all parts of the theory were tested or tested in all relevant situations.* For example, interaction within system initiative mode was not fully tested since there was limited system behaviour. There are also unanswered questions, such as whether it is important to model users anticipating system behaviour, rather than just reacting to it, for example expecting doors to automatically open for them. Furthermore, although the test application provided for a generally representative sample of tasks and activities, it covered only a marketing application type with a desktop VE, mouse style object interaction and novice users.
- *The accuracy or validity of the theory evaluation was limited due to subjective elements in the data analysis.* Some subjectivity is unavoidable since there is no standardised and reliable way of categorising human behaviour. The data analyses were made more systematic by using explicit categories and rules, which were independent of the theory, and by using cross-matching with independent observers. However, judgements were still required and there were occasional uncertainties about which code or match, from a limited choice, was most appropriate.

Ultimately, questions of completeness and accuracy about the theory evaluation imply that the validity of the theory cannot be claimed to have been proved completely. What can be claimed is that the evaluation has shown good levels of correspondence between observations about interaction and theory predictions, and no serious contradictions have been found from a series of tests. Therefore, the evaluation has provided general support for components of the theory and further evaluation studies can be used to target weak or untested areas.

Finally, for the guidance tool:

- *Limitations in the tool include the use of only a sub-set of the guidance.* The demonstration tool included 12 of the 45 guidelines and three of the eight possible design elements (objects, actions and system control). Not all guidelines were included for these three elements. Therefore, a more complete version of the tool would be more complex and may, as a result, be more difficult to use. For example, guidelines for the user task element of a VE can apply during the 'define interaction' but also the 'design interaction' stages of development.
- *The evaluation of the tool was limited because of the small sample of designers and the use of the tool in a non-normal work context.* For example, the tool was not used for an extended period of time in a complete VE design project, where tight deadlines and technical limitations were present. However, the initial, focused evaluation has provided support for the proof of concept of the guidelines and presentation style. Also, the industrial designers did indicate the tool to be workable within current design practices.

Although the above limitations exist, this thesis is one of the first research projects directed at VE usability and, therefore, it has made an initial attempt at the problem. Many of the limitations reflect the fact that the research was itself constrained by what it was realistically possible and practical to model and evaluate. However, further work can be carried out to improve and extend on the research.

7.4 Future directions

There are several possible future directions for the research, both practical and theoretical. Work has already begun on extending the research to address the evaluation of VEs (see Sutcliffe and Kaur, 1998) and further work is planned on the guidance tool for designers. A possible theoretical direction is refining the modelling and evaluation approach used and proposing it as a general HCI approach. Also, the theory of interaction can be extended to a cover a wider range of VEs and behaviours. To begin with, however, there are important issues to address with the theory, and these are discussed in the next section.

7.4.1 The theory of interaction

The thesis has made an important start at modelling interaction for VEs and future work includes:

- *A better management of the theory complexity.* A full computational implementation may be useful in recording and structuring the theory, and providing an executable version of it, thereby enabling the efficient and reliable following of paths through the models and testing of theory predictions (Kieras and Polson, 1985). The correspondence rules can be simplified by moving the need for consistency between user knowledge and GDPs from specific rule conditions into the requirements of the GDPs themselves. This was the strategy used to simplify application of the rules when assessing the test application and developing the expert system (see sections 5.3.1 and 5.5.1).
- *Further evaluations of the theory.* Focused studies are needed to evaluate poorly tested parts of the theory. The theory would also benefit from evaluation in a wider range of contexts, particularly with different application types and interaction styles (e.g. gestural), and with immersive VEs and expert users. However, carrying out detailed experimental studies, like those described in chapter five, will be taxing in terms of time and effort. Therefore, it may be most practical to improve the theory through revisions based on lessons learned in future applications of it.

7.4.2 Extending the theory

The theory of interaction modelled error-free behaviour with single-user VEs, representing real-world phenomena. Possible future directions in extending the theory to cover more complex situations include:

- *Looking more closely at common remedial paths to recover from errors.* Although the refined models did include backtracks for error recovery, remedial behaviours can be described more explicitly (see Springett's, 1996, interaction models for direct manipulation). Specific design properties can then be defined to support error recovery. For example, understanding navigation problems and how best they can be recovered from, will be important in enabling low spatial ability users to interact successfully with VEs. For instance, reset functions may help when the user has navigated themselves into a corner and alignment guides may help users navigate in tight areas, such as corridors. Alternatively, the system could adapt itself to the individual skills of users (Benyon, 1993), for example by adjusting navigation speeds according to the spatial ability of the user.
- *Widening the scope of the theory to cover multi-user, abstract and immersive environments.* For example, communication between different users in a VE could be modelled, such as the sending and receiving of messages and collaborations to complete tasks synchronously. Design properties and guidelines could then be defined to support this communication; a requirement highlighted by Steed and Tromp (1998). Similarly, metaphor understanding and transfer could be modelled to define properties required for abstract VEs, such as consistency between metaphors and user knowledge. Furthermore, special requirements for immersive VEs may be usefully investigated, such as properties required to support the use of peripheral vision and levels of awareness, and simultaneous as well as sequential activity.

7.4.3 An approach to HCI theory

The approach taken to modelling interaction and informing usability requirements has been successful and useful future research includes further development so that it can be proposed as a general approach to HCI theory.

The basic approach involves:

1. Process models of interaction, elaborated from established general psychological models, which include the major stages and patterns of behaviour.
2. Contributions to successful interaction, from the interface design and the user, for all behaviour captured in the models. Design contributions can either provide basic support for interaction or provide items of information useful during interaction.
3. Correspondence rules to show the links between different theoretical components.
4. User studies and protocols to gain evidence of interactive behaviour for evaluating and systematically refining components of the theory, by comparing observed interaction with predictions made.
5. The transformation of interface requirements into design guidelines, tools and methods.

There are some issues in this approach that need addressing, in particular:

- Managing the level of complexity can be difficult and computational implementation of the theory may be made more effective by employing a computational architecture for modelling. For example, the COGENT environment (Cooper, 1995) may be used. COGENT has been designed for implementing cognitive models by incorporating generic cognitive elements, such as processes and memory buffers, with formal rules.
- The verbal protocol analysis is very time-consuming and can be difficult to carry out reliably, although it does provide rich data to evaluate process models. Therefore, the approach needs to incorporate a systematic method for analysis of protocols, including general coding categories and guidance on matching back to model components. Toolkits may help to speed up analysis and reduce effort required (e.g. SAPA; Ericsson and Simon, 1984). For example, databases can be provided that include outline tables and standard queries for interaction data.

Finally, the situations in which this approach is particularly suitable need to be elaborated, such as the scope and level of guidance that the approach can deliver.

7.4.4 Methods and tools for designers

The natural direction for the thesis research to take is the development of complete methods and tools for designers. Support was gained for the demonstration tool, in the evaluation study, and useful additions to the tool were identified; in particular, a choice of techniques for implementing guidelines and guidance on balancing issues of usability and realism. Therefore, future work includes re-working of the guidance tool in light of designers' suggestions and including all 45 guidelines. Further evaluation with the complete version of the tool, and with more designers and natural work situations is also important.

Further research may be needed for some of the designers' suggestions. Existing VEs can be investigated for 'best practice' techniques, which incorporate one or more of the guidelines, to include in the tool as reusable examples. Such an approach has similarities with the claims analysis approach of Carroll and Kellogg (1989). For example, in chapter two, various techniques were discussed for providing an appropriate user representation, such as human bodies, hands and task-related tools (e.g. 'cutters') (Benford et al., 1995; Bordegoni, 1993; Poston and Serra, 1996). In this way, the generic design properties can act as a framework for organising available techniques, helping to define what each technique offers the user. The importance of realism in implementing the guidelines is likely to be dependent on the application involved. Some applications will require high levels of fidelity so artificial usability effects will not be acceptable. Other application domains may lack natural affordances or be naturally difficult to interact with. However, different techniques can be used to provide the same usability information but at different levels of realism. For example, instead of the information icons, used in the test application, more realistic information leaflets could have provided the same usability benefits. Therefore, a library of examples in the tool could include a number of techniques for implementing each guideline, at varying levels of realism, for different application requirements.

Furthermore, as well as different degrees of realism, there can be different styles of guideline implementation that can be offered in the tool. For example, guidelines can be implemented by using a different modality (e.g. speech or text) to explain features, highlighting techniques for attentional design, additional tools or interface components (e.g. maps and orientation indicators), or more active techniques like snap-to-object and go-to-location. Different styles are likely to be appropriate for implementing basic support GDPs, as opposed to information provider GDPs. Research into the trade-offs between these styles may be necessary. For example, active techniques may be intrusive and it may be preferable to aid the user in carrying out interactions themselves in the most natural manner, as far as this is possible within technological constraints. One further issue is that often one technique can be used to implement several guidelines. For example, the speech track technique was used to implement three GDPs for the system control drive through in chapter 5. The tool could account for this possibility by including support for a step to consolidate the requirements of relevant guidelines into actual design effects to be implemented.

More ambitious directions for the guidance tool include the following possibilities:

- *Extending the tool to critique designs.* For example, a critic could be embedded in the design environment that extracts components from the design and investigates them for adherence to relevant guidelines. For instance, how well objects can be distinguished may be automated to some extent and combined with specific questions to the designer. Alternatively, design solutions can be assessed against likely user responses in each stage of interaction, by walking through the models, thereby helping designers to understand how users will interact with different design ideas. Potential problems can be highlighted and solutions suggested, by referring to a library of examples.
- *Including a structured method for VE development in the tool.* The demonstration tool included an outline of such a method, which could be elaborated by adding procedural advice with the usability guidelines. The method could also organise design processes around common interaction tasks and behaviours, using the models of interaction. A structured method could be an important tool, especially for novice designers, by helping to explain VE concepts (e.g. user representation and system control), organise development activities, introduce design issues at

appropriate points, and document the design. However, the method would need to be informal and flexible to suit the VE design process.

7.4.5 Evaluating virtual environments

A further, natural direction for the thesis research is the development of methods or heuristics for evaluating VEs. Special methods for VE evaluation are needed (Tromp, 1997) since current methods are insufficient (Höök and Dahlbäck, 1992). In recent work, Whelan (1996) proposed a Virtual Environment Cognitive Ergonomics Evaluation Tool (VECEET). The tool is based on knowledge of cognitive functioning, visual functioning and immersiveness. It consists of a questionnaire composed of 30 items, in three sections: *Cognitive Compatibility Constructs*, such as simplicity, integration and familiarity, *Topological Processing Pathways in the Visual System*, such as change and movement, orientation and positioning, and, an *Immersiveness Model* involving Zeltzer's (1992) three categories of autonomy, interaction and presence. VECEET is an initial attempt at an evaluation tool for VEs and was found to correlate well with ratings given by SART, a situational awareness ratings scale (Taylor, 1989). The VECEET questions cover some important aspects of VEs, such as presence and visual perception, however there is no coverage of critical interaction and usability issues, for example for navigation and action.

Alternatively, existing usability methods, such as cognitive walkthroughs (Polson et al. 1992), may be adapted for VEs. For example, COVEN (1997b) found that the cognitive walkthrough method was applicable to 3D interaction, although there was a need to define specific 3D user problems. The theory of interaction lends itself well to cognitive walkthrough methods, since it includes models of interaction that can be 'walked-though' in an evaluation. Sutcliffe and Kaur (1998) have developed a cognitive walkthrough method for VEs, based on the theory of interaction. A walkthrough analysis is applied for each interaction stage by expanding the correspondence rules into question checklists for the design properties and user knowledge sources required for successful interaction. For example, at the 'scan' stage, in the explore navigate model, the question checklist includes:

When scanning the VE, can the users distinguish and recognise many/few/none of the objects? Can users interpret the identity and role of objects? Does the object appearance match the users' expectations?

The method remains to be fully evaluated, but is a more promising approach to evaluating VEs. It is based on the cognitive walkthrough method that has proved successful with conventional user interfaces (Wharton et al., 1994) and the theory of interaction, which covers important user behaviour for VEs.

A more simple and quick evaluation method, such as heuristic evaluation (Nielsen and Molich, 1990), may be a useful complement to cognitive walkthroughs. Sutcliffe and Kaur (1998) include heuristics in their work on VE evaluation, developed from the generic design properties. For example, the heuristic *clear turn taking* states that where system initiative is used in the VE, it should be clearly signalled and conventions established for turn taking.

The partial rule implementation has also indicated promising future directions with regard to VE evaluation. The expert system may be extended as a tool to predict or explain usability problems. Usability problems could be predicted for an application from inputs about the design and user knowledge. Alternatively, causes of problems, in terms of missing user knowledge or design properties, can be predicted from inputs of observed usability problems and possible solutions offered. The tool would require a complete set of the correspondence rules and an improved user interface and tool structure. The tool would also need to include heuristics to guide judgements on whether design properties or user knowledge were present and adequate. Ultimately, such a tool could support the cognitive walkthrough method for evaluating VEs.

In conclusion, this thesis has provided important early contributions to research on VEs and usability. The thesis has taken steps towards practical application of the research and can be a useful basis for future work in this area, which it is hoped the research will encourage.

References

ADAM J.A. (1993). Virtual reality is for real. IEEE Spectrum 30(10): 30-33.

ARTHUR K.W., BOOTH K.S. and WARE C. (1993). Evaluating 3D task performance for fish tank virtual worlds. ACM Transactions on Information Systems 11(3): 239-265.

ASTHEIMER P., DAI F., GÖBEL M., KRUSE R., MÜLLER S. and ZACHMANN G. (1994). Realism in Virtual Reality. In: Artificial Life and Virtual Reality, ed. by N.M. Thalmann and D. Thalmann. (Chichester, UK: John Wiley and Sons).

BAKEMAN R. and GOTTMAN J.M. (1986). Observing interaction: an introduction to sequential analysis. (Cambridge, UK: Cambridge University Press).

BARNARD P.J. (1987). Cognitive resources and the learning of human-computer dialogs. In: Interfacing thought: cognitive aspects of human-computer interaction, ed. by J.M. Carroll. (Cambridge, Massachusetts: Massachusetts Institute of Technology Press).

BARNARD P. (1991). Bridging between basic theories and the artefacts of human-computer interaction. In: Designing interaction: psychology at the human-computer interface, ed. by J.M. Carroll. (Cambridge, UK: Cambridge University Press).

BELLOTTI V. (1988). Implications of current design practice for the use of HCI techniques. In: People and Computers IV: Proceedings of HCI '88, ed. by D.M. Jones and R. Winder. (Cambridge, UK: Cambridge University Press).

BENFORD S., GREENHALGH C., BOWERS J., SNOWDON D. and FAHLÉN L.E. (1995). User embodiment in collaborative virtual environments. In: Human Factors in Computing Systems: CHI '95 conference, Denver 1995. Proceedings. New York: Association for Computing Machinery, pp. 242-249.

BENFORD S. and FAHLÉN L. (1993). A spatial model of interaction in large virtual environments. In: Third European conference on computer-supported co-operative work, ECSCW '93, Milan, Italy, September 1993. Proceedings. Norwell, Massachusetts: Kluwer Academic, pp.109-124.

BENYON D. and HÖÖK K. (1997). Navigation in information spaces: supporting the individual. In: Human-Computer Interaction: Proceedings of Interact '97, ed. by S. Howard, J. Hammond and G. Lindagaard. (London: Chapman & Hall).

BENYON D. (1993). Accommodating individual differences through an adaptive user interface. In: Adaptive user interfaces: results and prospects, ed. by M. Schneider-Hufschmidt, T Kuhme and U Malinowski. (North Holland: Elsevier Science).

BERGAMASCO M. (1994). Manipulation and exploration of virtual objects. In: Artificial Life and Virtual Reality, ed. by N.M. Thalmann and D. Thalmann. (Chichester, UK: John Wiley and Sons).

BIBBY P.A. (1992). Distributed knowledge: in the head, in the world or in the interaction? In: Models in the mind: theory, perspective and applications, ed. by Y. Rogers, A. Rutherford and P.A. Bibby. (London: Academic Press).

BILLINGHURST M. and SAVAGE J. (1996). Adding intelligence to the interface. In: Virtual Reality Annual International Symposium (VRAIS '96), California 1996. Proceedings. New York: Institute of Electrical and Electronics Engineers.

BLANDFORD A.E. and YOUNG R.M. (1995). Separating user and device descriptions for modelling interactive problem solving. In: Human-Computer Interaction: Proceedings of Interact '95, ed. by K. Nordby, P.H. Helmersen, D.J. Gilmore and S.A. Arnesen. (London: Chapman and Hall).

BLISS J.P., TIDWELL P.D. and GUEST M.A. (1997). The effectiveness of virtual reality for administering spatial navigation training to fire-fighters. Presence: teleoperators and virtual environments 6(1): 73-86.

BOLAS M. (1994). Designing virtual environments. In: The Virtual Reality Casebook, ed. by C.G. Loeffler and T. Anderson. (New York: Van Nostrand Reinhold).

BOLTER J., HODGES L.F., MEYER T. and NICHOLS A. (1995). Integrating perceptual and symbolic information in VR. IEEE Computer Graphics & Applications 15(4): 8-11.

BOLZONI M.L.G. (1994). Eliciting a context for rules of interaction: a taxonomy of metaphors for human-objects communication in virtual and synthetic environments. In: 2nd UK Virtual Reality-Special Interest Group Conference, Theale, Reading, 1994. Proceedings. UK Virtual Reality-Special Interest Group and Contributors, pp. 78-87.

BORDEGONI M. (1993). Gesture interaction in a 3D user interface. European Research Consortium for Informatics and Mathematics. Technical report, number ERCIM-93-RO19.

BOYD C. and DARKEN R. (1996). Psychological issues of virtual environment interfaces: a CHI '96 workshop. Special Interest Group on Computer-Human Interaction (SIGCHI) Bulletin 28(4): 49-53.

BRELSTAFF G. (1995). Visual displays for virtual environments - a review. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 27-37.

BROOKS F. (1994). Is there any real virtue in virtual reality? Lecture given at The Institution of Electrical Engineers, London, November 1994.

BROWN M.L., NEWSOME S.L. and GLINERT E.P. (1989). An experiment into the use of auditory cues to reduce visual workload. In: Human Factors in Computing Systems: CHI '89 conference, Austin, Texas 1989. Proceedings. New York: Association for Computing Machinery, pp. 339-346.

BRYSON S. (1995). Approaches to the successful design and implementation of VR applications. In: Virtual Reality Applications, ed. by R.A. Earnshaw, J.A. Vince and H. Jones. (London: Academic Press).

BULLINGER H.J., MÜLLER-SPAHN F. and RÖßLER A. (1996). Encouraging creativity - support for mental processes by virtual experiences. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

BUCK M., GREBNER K. and KATTERFELDT H. (1996). Modelling and interaction tools for virtual environments. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

CAPIN T.K., PANDZIC I.S., THALMANN N.M. and THALMANN D. (1995). Virtual humans for representing participants in immersive virtual environments. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 135-149.

CARD S.K., MORAN T.P. and NEWELL A. (1983). The psychology of human-computer interaction. (Hillsdale, New Jersey: Lawrence Erlbaum Associates).

CARLSSON C. and JÄÄ-ARO K.M. (1995). The DIVE Laboratory. Presence: teleoperators and virtual environments 4(4): 431-440.

CARR K.T. and ENGLAND R.D. (1993). The role of realism in virtual reality. In: Virtual Reality International '93: third annual conference, London 1993. Proceedings. London: Meckler, pp. 24-33.

CARROLL J.M. (1987). Interfacing thought: cognitive aspects of human-computer interaction. (Cambridge, Massachusetts: Massachusetts Institute of Technology Press).

CARROLL J.M. and KELLOGG W.A. (1989). Artefact as theory-nexus: hermeneutics meets theory-based design. In: Human Factors in Computing Systems: CHI '89 conference, Austin, Texas 1989. Proceedings. New York: Association for Computing Machinery, pp. 7-14.

CHARITOS D. (1996). Defining existential space in virtual environments. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

CHARITOS D and RUTHERFORD P. (1996). Guidelines for the design of virtual environments. In: 3rd Virtual Reality-Special Interest Group Conference, Leicester, 1996. Proceedings. The UK Virtual Reality-Special Interest Group and Contributors, pp. 93-111.

CHARITOS D. (1997). Designing space in virtual environments for aiding wayfinding behaviour. In: 4th UK Virtual Reality-Special Interest Group Conference, Uxbridge, 1997. Proceedings. The UK Virtual Reality-Special Interest Group and Contributors.

COLEBOURNE A. and RODDEN T. (1995). VR-MOG: A toolkit for building shared virtual worlds. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 109-122.

COOPER R. (1995). Towards an object-oriented language for cognitive modelling. In: Seventeenth Annual Conference of the Cognitive Science Society, Pittsburgh 1995. Proceedings. Hillsdale, New Jersey: Lawrence Erlbaum Associates, pp. 556-561.

COVEN (1997a). D2.6: Guidelines for building CVE applications. Public Deliverable report of COVEN ACTS Project N.AC040.

COVEN (1997b). D3.3: Usage evaluation of the initial applications. Public Deliverable report of COVEN ACTS Project N.AC040.

CRUZ-NEIRA C. (1996). Integrating high performance computing and communications with virtual reality for interactive molecular modelling: the VIBE system. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

CURTIS B., WALZ D. and ELAM J. (1989). Studying the process of software design teams. In: 5th International Software Process Workshop (experiences with software process models), Kennebunkport ME, 1989, Proceedings. (IEEE Computer Society Press), pp. 52-53.

DAI F., FELGER W., FRÜHAUF T., GÖBEL M., REINERS D. and ZACHMANN G. (1996). Virtual prototyping examples for automotive industries. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

DARKEN R.P. (1995). Wayfinding in large-scale virtual worlds. Doctoral Consortium paper. In: Human Factors in Computing Systems: CHI '95 conference, Denver 1995. Proceedings. New York: Association for Computing Machinery, pp. 45-46.

DARKEN R.P. and SIBERT J.L. (1996). Wayfinding strategies and behaviours in large virtual worlds. In: Human Factors in Computing Systems: CHI '96 conference, Vancouver 1996. Proceedings. New York: Association for Computing Machinery, pp. 142-149.

D'CRUZ M.D., EASTGATE R.M. and WILSON J.R. (1996). Towards a structured methodology for the industrial application of virtual reality. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

DE MUL S. and VAN OOSTENDORP H. (1995). Encouraging exploratory learning of a user interface. In: Human-Computer Interaction: Proceedings of Interact '95, ed. by K. Nordby, P. Helmersen, D.J. Gilmore and S.A. Arnesen. (London: Chapman and Hall).

DEDE C., SALZMAN M., CALHOUN C., LOFTIN R.B., HOBLIT J. and REGIAN J.W. (1994). The design of artificial realities to improve learning Newtonian mechanics. In: East-West International Conference on Multimedia, Hypermedia and Virtual Reality, Moscow, 1994. Proceedings. Moscow: International Centre for Scientific and Technical Information (ICSTI).

DEERING M. (1995). Holosketch: a virtual reality sketching/animation tool. ACM Transactions on Computer-Human Interaction 2(3): 220-238.

DELP S. L., LOAN P., BASDOGAN C. and ROSEN J.M. (1997). Surgical simulation: an emerging technology for training in emergency medicine. Presence: teleoperators and virtual environments 6(2): 147-159.

DICKINSON R.R. and JERN M. (1995). A unified approach to interface design for data visualisation using desktop and immersion virtual environments. In: Virtual Reality Applications, ed. by R.A. Earnshaw, J.A. Vince and H. Jones. (London: Academic Press).

DISTLER H. (1996). Psychophysical experiments in virtual environments. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

DIX A., FINLAY J., ABOWD G. and BEALE R (1993). Human-Computer Interaction. (Hemel Hempstead, UK: Prentice Hall).

DOCIMO S.G., MOORE R.G. and KAVOUSSI L.R. (1997). Telerobotic surgery is clinical reality: current experience with telemonitoring in adults and children. Presence: teleoperators and virtual environments 6(2): 173-178.

DRUCKER S.M. and ZELTZER D. (1994). Intelligent camera control in a virtual environment. In: Graphics Interface '94 conference, Banff, Alberta, 1994. Proceedings. Toronto: Canadian Information Processing Society, pp. 190-199.

DUKE D.J., FACONTI G., HARRISON M.D. and PATERNO F. (1994). Unifying views of interactors. In: Proceedings of the workshop on advanced visual interfaces '94, Bari, 1994. New York: Association for Computing Machinery, pp.143-152.

ELLIS S.R. (1993a). Prologue. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

ELLIS S.R. (1993b). Pictorial communication: pictures and the synthetic universe. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

ELLIS S.R. (1996). Presence of mind... a reaction to Thomas Sheridan's 'Musings on Telepresence'. Presence: teleoperators and virtual environments 5(2): 247-259.

ERICSSON K.A. and SIMON H.A. (1984). Protocol Analysis. (Cambridge, Massachusetts: Massachusetts Institute of Technology Press).

EVANS J.S.B.T. (1988). The knowledge elicitation problem: a psychological perspective. Behaviour and Information Technology 7(2): 111-130.

EYLES D.E. (1993). A computer graphics system for visualising spacecraft in orbit. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

FIELDS B, WRIGHT P. and HARRISON M. (1997). Objectives, strategies and resources as design drivers. In: Human-Computer Interaction: Proceedings of Interact '97, ed. by S. Howard, J. Hammond and G. Lindagaard. (London: Chapman & Hall).

FISHER S.S. (1990). Virtual interface environments. In: The Art of Human-Computer Interface Design, ed. by B. Laurel. (Massachusetts: Addison-Wesley).

FISHER R.B., FITZGIBBON A., GIONIS A., WRIGHT M. and EGGERT D. (1996). A hand-held optical surface scanner for environmental modelling and virtual reality. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

FROHLICH D.M. (1993). The history and future of direct manipulation. Behaviour and Information Technology 12(6): 315-329.

FURNAS G.W. (1997). Effective view navigation. In: Human Factors in Computing Systems: CHI '97 conference, Atlanta 1996. Proceedings. New York: Association for Computing Machinery, pp. 367-374.

GALLERY R.D., JETHA Z., EVES D.A. and MAH S. (1996). Follow-Me! Paper presented at The Institution of Electrical Engineers Virtual Reality - User Issues Colloquium, Cheltenham, March 1996.

GIBSON J.J. (1986). The ecological approach to visual perception. (Hillsdale, New Jersey: Lawrence Erlbaum Associates).

GIGANTE M.A. (1993). Virtual Reality: definitions, history and applications. In: Virtual Reality Systems, ed. by R.A. Earnshaw, M.A. Gigante, H. Jones. (London: Academic Press).

GILKEY R.H. and WEISENBERGER J.M. (1995). The sense of presence for the suddenly deafened adult: implications for virtual environments. Presence: teleoperators and virtual environments 4(4): 357-363.

GLASS A.L. and HOLYOAK K.J. (1986). Cognition. (New York: Newbery Award Records).

GOULD J.D. and LEWIS C. (1985). Designing for usability: key principles and what designers think. Communications of the ACM 28(3): 300-311.

GREEN M. and HALLIDAY S. (1996). A geometric modelling and animation system for virtual reality. Communications of the ACM 39(5): 46-53.

GRINSTEIN G.G. and SOUTHARD D.A. (1996). Rapid modelling and design in virtual environments. Lab review. Presence: teleoperators and virtual environments 5(1): 146-158.

GUINDON R. (1990). Knowledge exploited by experts during software system design. International journal of Man-Machine Studies 33(3): 279-304.

GUINDON R., KRASNER H. and CURTIS B. (1987). Breakdowns and processes during the early activities of software design by professionals. In: Empirical studies of programmers: second workshop, ed. by G. Olson, S. Sheppard and E. Soloway. (Norwood, New Jersey: Ablex Publishing).

HAUPTMANN A.G. and MACAVINNEY P. (1993). Gestures with speech for graphic manipulation. International journal of man-machine studies 38: 231-249.

HAYES-ROTH B. and HAYES-ROTH F. (1979). A cognitive model of planning. Cognitive Science 3: 275-310.

HELD R. and BOSSOM J. (1961). Neonatal deprivation and adult rearrangement: complementary techniques for analysing plastic sensorimotor co-ordinations. In: Perceptual Learning and Adaptation, ed. by P.C. Dodwell. (Middlesex: Penguin Books).

HELD R. and DURLACH N. (1993). Telepresence, time delay and adaptation. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

HENDRIX C. and BARFIELD W. (1996a). Presence within virtual environments as a function of visual display parameters. Presence: teleoperators and virtual environments 5(3): 274-289.

HENDRIX C. and BARFIELD W. (1996b). The sense of presence within auditory virtual environments. Presence: teleoperators and virtual environments 5(3): 290-301.

HERNDON K.P., VAN DAM A. and GLEICHER M. (1994). The challenges of 3D interaction: a CHI'94 workshop. Special Interest Group on Computer-Human Interaction (SIGCHI) Bulletin 26(4): 36-43.

HIROSE M. (1996). The synthesis of realistic sensations. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

HODGES L.F., KOOPER R., MEYER C., ROTHBAUM B.O., OPDYKE D., DE GRAAFF J.J., WILLIFORD J.S. and NORTH M.M. (1995). Virtual environments for treating the fear of heights. IEEE Computer 28(7): 27-33.

HOLLNAGEL E. (1993). The phenotype of erroneous actions. International Journal of Man-Machine Studies 39: 1-32.

HÖÖK K. and DAHLBÄCK N. (1992). Can cognitive science contribute to the design of VR applications? Paper presented at 5th MultiG Workshop, Stockholm, December 1992.

HOWES A. and PAYNE S.J. (1990). Supporting exploratory learning. In: Human-Computer Interaction: Proceedings of Interact '90, ed. by D. Diaper, D. Gilmore, G. Cockton and B. Shackel.. (North-Holland: Elsevier Science).

HUBBOLD R., MURTA A., WEST A. and HOWARD T. (1993). Design issues for virtual reality systems. Paper presented at First Eurographics Workshop on Virtual Environments, Barcelona, September 1993.

INGRAM R., BENFORD S. and BOWERS J. (1996). Building virtual cities: applying urban planning principles to the design of virtual environments. In: Virtual Reality Software and Technology (VRST '96) conference, Hong Kong, 1996. Proceedings. New York: Association for Computing Machinery.

ISO (1996). International Standards Organisation, draft international standard 9241-part 16: Ergonomic requirements for office work with visual display terminals - direct manipulation dialogues.

JOHNSGARD T. (1994). Fitt's law with a virtual reality glove and a mouse: effects of gain. In: Graphics Interface '94 conference, Banff, Alberta, 1994. Proceedings. Toronto: Canadian Information Processing Society, pp. 8-15.

JOHNSON P. (1992). Human Computer Interaction: Psychology, Task Analysis and Software Engineering (Maidenhead, UK: McGraw-Hill).

JOHNSON A.D. and CUTT P.S. (1992). The tactile sense: an untapped channel in man-machine interfaces. In: Virtual worlds: real challenges, papers from SRI's 1991 conference on virtual reality, ed. by T. Middleton. (Westport, Connecticut: Meckler).

JUL S. and FURNAS G.W. (1997). Navigation in electronic worlds: a CHI'97 workshop. Special Interest Group on Computer-Human Interaction (SIGCHI) Bulletin 29(4): 44-49.

KARAT C., CAMPBELL R. and FIEGEL T. (1992). Comparison of empirical testing and walkthrough methods in user interface evaluation. In: Human Factors in Computing Systems: CHI '92 conference, Monterey, California 1992. Proceedings. New York: Association for Computing Machinery, pp. 397-404.

KENNEDY R.S., LANHAM D.S., DREXLER J.M., MASSEY C.J. and LILIENTHAL M.G. (1997). A comparison of cybersickness incidences, symptom profiles, measurement techniques and suggestions for further research. Presence: teleoperators and virtual environments 6(6): 638-644.

KIERAS D. and POLSON P.G. (1985). An approach to the formal analysis of user complexity. International Journal of Man-Machine Studies 22: 365-394.

KIM W.S., TENDICK F. and STARK L. (1993). Visual enhancements in pick-and-place tasks: human operators controlling a simulated cylindrical manipulator. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

KITAJIMA M. and POLSON P.G. (1996). A comprehension-based model of exploration. In: Human Factors in Computing Systems: CHI '96 conference, Vancouver 1996. Proceedings. New York: Association for Computing Machinery, pp. 324-331.

KLAISS S. (1996). Integrated VR applications: Systecs Visplan. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

KOHLER I. (1964). The formation and transformation of the visual world. In: Perceptual Learning and Adaptation, ed. by P.C. Dodwell. (Middlesex: Penguin Books).

KRÜGER W., BOHN C.A, FRÖHLICH B., SCHÜTH H., STRAUSS W. and WESCHE G. (1995). The responsive workbench: a virtual work environment. IEEE Computer 28(7): 42-48.

LANDAY J.A. and MYERS B.A. (1995). Interactive sketching for the early stages of user interface design. In: Human Factors in Computing Systems: CHI '95 conference, Denver 1995. Proceedings. New York: Association for Computing Machinery, pp. 43-50.

LAUESEN S. (1997). Usability engineering in industrial practice. In: Human-Computer Interaction: Proceedings of Interact '97, ed. by S. Howard, J. Hammond and G. Lindagaard. (London: Chapman & Hall).

LAVERY D., COCKTON G. and ATKINSON M.P. (1997). Comparison of evaluation methods using structured usability problem reports. Behaviour and Information Technology 16(4/5): 246-266.

LEWIS C., POLSON P., WHARTON C. and RIEMAN J. (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In: Human Factors in Computing Systems: CHI '90 conference, Seattle 1990. Proceedings. New York: Association for Computing Machinery, pp. 235-242.

LIM K.Y. and LONG J. (1994). The MUSE method for usability engineering. (Cambridge, UK: Cambridge University Press).

LOEFFLER C.E. and ANDERSON T. (1994). The Virtual Reality Casebook. (New York: Van Nostrand Reinhold).

LONG J.B. and DOWELL J. (1989). Conceptions of the discipline of HCI: Craft, Applied Science and Engineering. In: People and Computers V: Proceedings of HCI '89, ed. by A. Sutcliffe and L. Macaulay. (Cambridge, UK: Cambridge University Press).

LOOMIS J.M. (1993). Understanding synthetic experience must begin with the analysis of ordinary perceptual experience: position statement. In: Symposium on research frontiers in virtual reality, San Jose, California 1993. Proceedings. Los Alamitos: IEEE Computer Society Press, pp. 54-57.

MACREDIE R. (1995). Human factors issues in virtual reality. Paper presented at Virtuality - Human Factors Issues Conference, Cheltenham, March 1995.

MALINSKI J. (1996). Evaluating the usability of a virtual reality prototype system. Masters thesis: Department of Business Computing, City University, London.

MAPES D.P. and MOSHELL J.M. (1995). A two-handed interface for object manipulation in virtual environments. Presence: teleoperators and virtual environments 4(4): 403-416.

MASON C.A. (1996). Desktop virtual reality in the RAF. Paper presented at The Institution of Electrical Engineers Virtual Reality - User Issues Colloquium, Cheltenham, March 1996.

MCCONATHY D.A. and DOYLE M. (1993). Interactive displays in medical art. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

MCGLASHAN S. and AXLING T. (1996). Talking agents in virtual worlds. In: 3rd UK Virtual Reality-Special Interest Group Conference, Leicester, 1996. Proceedings. The Virtual Reality-Special Interest Group and Contributors, pp. 145-162.

MCGOVERN D.E. (1993). Experience and results in teleoperation of land vehicles. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

MEDYCKYJ-SCOTT D.J. and BLADES M. (1991). Cognitive representations of space in the design and use of geographical information systems. In: People and Computers VI: Proceedings of HCI '91, ed. by D. Diaper and N. Hammond (Cambridge, UK: Cambridge University Press).

MERCURIO P.J. and ERICKSON T.D. (1990). Interactive scientific visualisation: an assessment of a virtual reality system. In: Human-Computer Interaction: Proceedings of Interact '90, ed. by D. Diaper, D. Gilmore, G. Cockton and B. Shackel. (North Holland: Elsevier Science).

MILLER L.D. (1994). In the context of the European project 'Metrics for Usability Standards in Computing' (MUSiC): A usability evaluation of the Rolls-Royce virtual reality for aero engine maintenance system. Masters thesis: University College London.

MOHAGEG M., MYERS R., MARRIN C., KENT J., MOTT D. and ISAACS P. (1996). A user interface for accessing 3D content on the world wide web. In: Human Factors in Computing Systems: CHI '96 conference, Vancouver 1996. Proceedings. New York: Association for Computing Machinery, pp. 466-472.

MURTA A. (1995). Vertical axis awareness in 3D environments. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 169-176.

MYERS B.A. and ROSSON M.B. (1992). Survey on user interface programming. In: Human Factors in Computing Systems: CHI '92 conference, Monterey, California 1992. Proceedings. New York: Association for Computing Machinery, pp. 195-202.

NEMIRE K. and ELLIS S.R. (1993). Calibration and evaluation of virtual environment displays. In: Symposium on research frontiers in virtual reality, San Jose, California, 1993. Proceedings. Los Alamitos: IEEE Computer Society Press, pp. 33-40.

NIELSEN J. (1990). Navigation through hypertext. Communications of the ACM. 33(3): 297-310.

NIELSEN J. (1993). Usability Engineering. (San Diego: Academic Press).

NIELSEN J. and MOLICH R. (1990). Heuristic evaluation of user interfaces. In: Human Factors in Computing Systems: CHI '90 conference, Seattle 1990. Proceedings. New York: Association for Computing Machinery, pp. 249-256.

NEISSER U. (1976). Cognition and reality: principles and implications of cognitive psychology. (San Francisco, California: WH Freeman and Company).

NORMAN D.A. (1986). Cognitive engineering. In: User centred system design: new perspectives on human-computer interaction, ed. by D.A. Norman and S.W. Draper. (Hillsdale, New Jersey: Lawrence Erlbaum Associates).

NORMAN D.A. (1988). The psychology of everyday things. (New York: Basic Books).

O'MALLEY C. and DRAPER S. (1992). Representation and interaction: are mental models all in the mind? In: Models in the mind: theory, perspective and applications, ed. by Rogers Y., Rutherford A. and Bibby P.A. (London: Academic Press).

OLSON J.R. (1987). Cognitive analysis of people's use of software. In: Interfacing thought: cognitive aspects of human-computer interaction, ed. by J.M. Carroll. (Cambridge, Massachusetts: Massachusetts Institute of Technology Press).

OMAN C.M. (1993). Sensory conflict in motion sickness: an Observer Theory approach. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

OSBORN J.R. and AGOGINO A.M. (1992). An interface for interactive spatial reasoning and visualisation. In: Human Factors in Computing Systems: CHI '92 conference, Monterey, California 1992. Proceedings. New York: Association for Computing Machinery, pp. 75-82.

PAYNE S.J. (1991). Display-based action at the user interface. International journal of Man-Machine Studies 35: 275-289.

PAYNE S.J., CHESWORTH L. and HILL E. (1992). Animated demonstrations for exploratory learners. Interacting with computers 4(1): 3-22.

PIMENTEL K. and TEIXEIRA K. (1993). Virtual Reality: Through the new looking glass. (USA: McGraw-Hill).

POLSON P.G., LEWIS C., RIEMAN J. and WHARTON C. (1992). Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. International journal of man-machine studies 36: 741-773.

POSTON T. and SERRA L. (1996). Dextrous virtual work. Communications of the ACM 39(5): 37-45.

PRATT D.R., PRATT S.M., BARHAM P.T., BARKER R.E., WALDROP M.S., EHLERT J.F. and CHRISLIP C.A. (1997). Humans in large-scale, networked virtual environments. Presence: teleoperators and virtual environments 6(5): 547-564.

PRESSING J. (1997). Some perspectives on performed sound and music in virtual environments. Presence: teleoperators and virtual environments 6(4): 482-503.

PROFFITT D.R. and KAISER M.K. (1993). Perceiving environmental properties from motion information: minimal conditions. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

PSOTKA J. and DAVISON S. (1993). Cognitive factors associated with immersion in virtual environments. Research report. US Army Research Institute and Catholic University.

PSOTKA J., DAVISON S.A. and LEWIS S.A. (1993). Exploring immersion in virtual space. VR Systems Fall 1(2): 70-82.

RANSON D.S., PATTERSON E.S., KIDWELL D.L., RENNER G.A., MATTHEWS M.L., CORBAN J.M., SECULOV E. and SOUHLERIS C.S. (1996). Rapid scout: bridging the gap between physical and virtual environments. In: Human Factors in Computing Systems: CHI '96 conference, Vancouver 1996. Proceedings. New York: Association for Computing Machinery, pp. 442-449.

RASMUSSEN J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions; Human performance models. Transactions on system, man and cybernetics 13(3): 257-266.

RAUTERBERG M, SCHLUEP S. and FJELD M. (1997). How to model behavioural and cognitive complexity in human-computer interaction with Petri nets. In: 6th IEEE International workshop on Robot and Human Communication - RO-MAN'97, Sendai, Japan 1997. Proceedings. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, pp. 320-325.

REDDY M. (1995). A perceptual framework for optimising visual detail in virtual environments. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 189-201.

REGAN C. (1995). An investigation into nausea and other side-effects of head-coupled immersive virtual reality. Virtual Reality: research, development and applications 1(1): 17-32.

REGIAN J.W. and SHEBILSKE W.L. (1992). Virtual reality: an instructional medium for visual-spatial tasks. Journal of Communication 42(4): 136-149.

REISNER P. (1987). Discussion: HCI, what is it and what research is needed? In: Interfacing thought: cognitive aspects of human-computer interaction, ed. by J.M. Carroll. (Cambridge, Massachusetts: Massachusetts Institute of Technology Press).

RHEINGOLD H. (1990). An interview with Don Norman. In: The Art of Human-Computer Interface Design, ed. by B. Laurel. (Massachusetts: Addison-Wesley).

RICHARD P., BIREBENT G., COIFFET P., BURDEA G., GOMEZ D. and LANGRANA N. (1996). Effect of frame rate and force feedback on virtual object manipulation. Presence: teleoperators and virtual environments 5(1): 95-108.

RIEMAN J. and YOUNG R.M. (1996). A dual-space model for iteratively deepening exploratory learning. International Journal of Human-Computer Studies 44: 743-775.

ROBERTSON G.G., CARD S.K. and MACKINLAY J.D. (1993). Three views of virtual reality: Non-immersive virtual reality. Computer 26(2): 81-83.

ROLLAND J.P., GIBSON W. and ARIELY D. (1995). Towards quantifying depth and size perception in virtual environments. Presence: teleoperators and virtual environments 4(1): 24-49.

ROSSON M.B., MAASS S. and KELLOGG W.A. (1988). The designer as user: building requirements for design tools from design practice. Communications of the ACM 31(11): 1288-1297.

RUSHTON S. and WANN J. (1993). Problems in perception and action in virtual worlds. In: Virtual Reality International '93: third annual conference, London 1993. Proceedings. London: Meckler, pp. 43-55.

RUSHTON S., HARRIS J.M. and WANN J. (1997). Active control of heading and the importance of 3D structure, 2D structure, depth and rotation rates. Investigative opthalmology and visual science 38(4): 379.

RUSSELL G. and MILES R. (1993). Volumetric visualisation of 3D data. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

RUSSO J.E., JOHNSON E.J. and STEPHENS D.L. (1989). The validity of verbal protocols. Memory and Cognition (17:6) 759-769.

RYGOL M., DORBIE A., WORDEN A , GHEE S., GRIMSDALE C. and HARVEY J. (1995). Tools and metaphors for user interaction in virtual environments. In: Virtual Reality Applications, ed. by R.A. Earnshaw, J.A. Vince and H. Jones. (London: Academic Press).

SCHRAFT R.D., NEUGEBAUER J., FLAIG T. and DAINGHAUS R. (1995). A fuzzy controlled rendering system for virtual reality systems optimised by genetic algorithms. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 179-187.

SEN R., YATES R.B. and THACKER N.A. (1995). Virtual reality based on cost-benefit analysis. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 213-221.

SHERIDAN T.B. (1996). Further musings on the psychophysics of presence. Presence: teleoperators and virtual environments 5(2): 241-246.

SHNEIDERMAN B. (1982). The future of interactive systems and the emergence of direct manipulation. Behaviour and Information Technology 1: 237-256.

SHNEIDERMAN B. (1992). Designing the user interface: strategies for effective human-computer interaction. Second edition. (Massachusetts: Addison Wesley).

SINGH G., SERRA L., FAIRCHILD K. and POSTON T. (1994). Visual creation of virtual design environments and virtual worlds research at ISS. Presence: teleoperators and virtual environments 3(1): 94-107.

SLATER M. and USOH M. (1994). Body centred interaction in immersive virtual environments. In: Artificial Life and Virtual Reality, ed. by N.M. Thalmann and D. Thalmann. (Chichester, UK: John Wiley and Sons).

SLATER M. and USOH M. (1995). Modelling in immersive virtual environments: a case for the science of VR. In: Virtual Reality Applications, ed. by R.A. Earnshaw, J.A. Vince and H. Jones. (London: Academic Press).

SLATER M., USOH M. and STEED A. (1995). Taking steps: the influence of a walking technique on presence in virtual reality. ACM Transactions on Computer-Human Interaction 2(3): 201-219.

SLATER M. and WILBUR S. (1995). Through the looking glass world of presence: FIVE: a Framework for Immersive Virtual Environments. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 11-26.

SPRINGETT M. (1996). User modelling for evaluation of direct manipulation interfaces. PhD. thesis: Centre for Human-Computer Interface Design, City University, London.

STEED A. and TROMP J. (1998). Experiences with the evaluation of CVE applications. Paper presented at The 2nd Collaborative Virtual Environments Conference (CVE '98), Manchester, June 1998.

STONE R. (1994). Cognitive aspects of virtual reality. European Association of Cognitive Ergonomics Quarterly 1(1): 13-22.

STOPER A.E. and COHEN M.M. (1993). Optical, gravitational and kinaesthetic determinants of judged eye level. In: Pictorial Communication in virtual and real environments, ed. by S.R. Ellis, M. Kaiser and A.J. Grunwald. (London: Taylor and Francis).

STROMMEN E. (1994). Children's use of mouse-based interfaces to control virtual travel. In: Human Factors in Computing Systems: CHI '94 conference, Boston 1994. Proceedings. New York: Association for Computing Machinery, pp. 405-410.

STYTZ M.R., AMBURN P., LAWLIS P.K. and SHOMPER K. (1995). Virtual Environment research in the Air Force Institute of Technology Virtual Environments, 3-D Medical Imaging, and Computer Graphics Laboratory. Presence: teleoperators and virtual environments 4(4): 417-430.

SUCHMAN L.A. (1987). Plans and situated actions: the problem of human-machine communication. (Cambridge, UK: Cambridge University Press).

SUTCLIFFE A.G. (1995). Human-computer interface design. Second edition. (Basingstoke: MacMillan Press).

SUTCLIFFE A.G. and ENNIS M. (1998). Towards a cognitive theory of information retrieval. Interacting with Computers. *To appear in special edition on human-computer interaction and information retrieval.*

SUTCLIFFE A.G. and KAUR. K. (1998). Evaluating Virtual Reality User Interfaces. *Submitted for publication.*

SUTCLIFFE A. and MAIDEN N. (1990). Software reusability: delivering productivity gains or short cuts. In: Human-Computer Interaction: Proceedings of Interact '90, ed. by D. Diaper, D. Gilmore, G. Cockton and B. Shackel. (North Holland: Elsevier Science).

TAYLOR A., SLUCKIN W., DAVIES D.R., REASON J.T., THOMSON R and COLMAN A.M. (1982). Introducing Psychology. (Middlesex, UK: Penguin Books).

TAYLOR R. (1989). SART: The development of a tool for Aircrew Systems Design. Paper presented at The NATO AGARD AMP Symposium, Copenhagen 1989.

THALMANN D. (1994). Automatic control and behaviour of virtual actors. In: Interacting with virtual environments, ed. by L. MacDonald and J. Vince. (Chichester, UK: John Wiley and Sons).

THALMANN N.M. and THALMANN D. (1994). Introduction: creating artificial life in virtual reality. In: Artificial Life and Virtual Reality, ed. by N.M. Thalmann and D. Thalmann. (Chichester, UK: John Wiley and Sons).

TIMMER P. and LONG J. (1997). Separating user knowledge of domain and device: a framework. In: People and Computers XII: Proceedings of HCI '97, ed. by H. Thimbleby, B. O'Connaill and P.J. Thomas. (London: Springer-Verlag).

TRAVIS D., WATSON T. and ATYEO M. (1994). Human psychology in virtual environments. In: Interacting with virtual environments, ed. by L. MacDonald and J. Vince. (Chichester, UK: John Wiley and Sons).

TROMP J. (1995). Presence, telepresence and immersion: the cognitive factors of embodiment and interaction in virtual environments. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 39-51.

TROMP J. (1997). Methodology for distributed usability evaluation in collaborative virtual environments. In: 4th UK Virtual Reality-Special Interest Group Conference, Uxbridge, 1997. Proceedings. The UK Virtual Reality-Special Interest Group and contributors.

VENOLIA D. (1993). Facile 3D direct manipulation. In: International conference on Human-Computer Interaction (INTERCHI '93), Amsterdam 1993. Proceedings. New York: Association for Computing Machinery, pp. 31-36.

VOLBRACHT S., DOMIK G., SHAHRBABAKI K. and FELS G. (1997). How effective are 3D display modes? In: Human Factors in Computing Systems: CHI '97 conference, Atlanta 1997. Proceedings. New York: Association for Computing Machinery, pp. 540-541.

VORSTEHER A. (1996). Berlin 2010: creating the center - a virtual window into the future of Berlin. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

WANGER L.R., FERWERDA J.A. and GREENBERG D.P. (1992). Perceiving spatial relationships in computer-generated images. IEEE Computer Graphics and Applications 12(3): 44-58.

WANN J. and MON-WILLIAMS M. (1996a). Design issues in virtual environment displays: a formal appraisal from the perspective of human perception. Paper presented at The Institution of Electrical Engineers Virtual Reality - User Issues Colloquium, Cheltenham, March 1996.

WANN J. and MON-WILLIAMS M. (1996b). What does virtual reality NEED?: human factors issues in the design of three-dimensional computer environments. International Journal of Human-Computer Studies 44: 829-847.

WARE C. (1995). Dynamic stereo displays. In: Human Factors in Computing Systems: CHI '95 conference, Denver 1995. Proceedings. New York: Association for Computing Machinery, pp. 310-316.

WATERS R.C., ANDERSON D.B., BARRUS J.W., BROGAN D.C., CASEY M.A., MCKEOWN S.G., NITTA T., STERNS I.B. and YERAZUNIS W.S. (1997). Diamond Park and Spline: social virtual reality with 3D animation, spoken interaction, and runtime extendibility. Presence: teleoperators and virtual environments 6(4): 461-481.

WATSON B., WALKER N. and HODGES L. (1995). A user study evaluating level of detail degradation in the periphery of head-mounted displays. In: FIVE '95: Framework for Immersive Virtual Environments, London 1995. Proceedings. London: University of London, pp. 203-212.

WELCH R.B., BLACKMON T.T., LIU A., MELLERS B.A. and STARK L.W. (1996). The effects of pictorial realism, delay of visual feedback and observer interactivity on the subjective sense of presence. Presence: teleoperators and virtual environments 5(3): 263-273.

WEST A.J., HOWARD T.L.J., HUBBOLD R.J., MURTA A.D., SNOWDON D.N. and BUTLER D.A. (1993). AVIARY - A generic virtual reality interface for real applications. In: Virtual Reality Systems, ed. by R.A. Earnshaw, M.A. Gigante and H. Jones. (London: Academic Press).

WEXELBLAT A. (1995). An approach to natural gesture in virtual environments. ACM Transactions on Computer-Human Interaction 2(3): 179-200.

WHARTON C., RIEMAN J., LEWIS C. and POLSON P. (1994). The cognitive walkthrough method: a practitioner's guide. In: Usability inspection methods, ed. by J. Nielsen and R.L. Mack. (New York: John Wiley).

WHELAN R.J.C.R. (1996). Pilot study for the validation of a virtual environment cognitive ergonomics evaluation tool. Paper presented at The Institution of Electrical Engineers Virtual Reality - User Issues Colloquium, Cheltenham, March 1996.

WILSON J.R., BROWN D.J., COBB S.V., D'CRUZ M.M. and EASTGATE R.M. (1995). Manufacturing Operations in Virtual Environments (MOVE). Presence: teleoperators and virtual environments 4(3): 306-317.

WINOGRAD T. (1995). From programming environments to environments for designing. Communications of the ACM 38(6): 65-74.

WITMER B.G., BAILEY J.H., KNERR B.W. and PARSONS K.C. (1996). Virtual spaces and real world places: transfer of route knowledge. International journal of Human-Computer Studies 45: 413-428.

WLOKA D.W. (1996). CAVE: personal or small group non-HMD-based head tracked wrap-around virtual environment - the system of the future for virtual reality applications? In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

YAMAASHI K., COOPERSTOCK J.R., NARINE T. and BUXTON W. (1996). Beating the limitations of camera-monitor mediated telepresence with extra eyes. In: Human Factors in Computing Systems: CHI '96 conference, Vancouver 1996. Proceedings. New York: Association for Computing Machinery, pp. 50-57.

ZELTZER D. (1992). Autonomy, interaction, presence. Presence: teleoperators and virtual environments 1(1): pp.127-132.

ZELTZER D. and JOHNSON M.B. (1994). Virtual actors and virtual environments. In: Interacting with virtual environments, ed. by L. MacDonald and J. Vince. (Chichester, UK: John Wiley and Sons).

ZHAI S., BUXTON W. and MILGRAM P. (1994). The 'silk cursor': investigating transparency for 3D target acquisition. In: Human Factors in Computing Systems: CHI '94 conference, Boston 1994. Proceedings. New York: Association for Computing Machinery, pp. 459 - 464.

ZHAI S., WRIGHT J., SELKER T. and KELIN S.A. (1997). Graphical means of directing users' attention in the visual interface. In: Human-Computer Interaction: Proceedings of Interact '97, ed. by S. Howard, J. Hammond and G. Lindagaard. (London: Chapman & Hall).

Bibliography

KAUR K., MAIDEN N. and SUTCLIFFE A. (1996). Design practice and usability problems with virtual environments. In: Virtual Reality World '96 conference, Stuttgart, Germany, 1996. Proceedings. Munich, Germany: IDG Conferences & Seminars.

KAUR K., SUTCLIFFE A. and MAIDEN N. (1996). A theory of interaction in virtual environments. Short paper. In: 3rd Virtual Reality-Special Interest Group Conference, Leicester, 1996. Proceedings. The UK Virtual Reality-Special Interest Group and Contributors, pp. 39-52.

KAUR K. (1997). Designing virtual environments for usability. Doctoral Consortium Paper. In: Human-Computer Interaction: Proceedings of Interact '97, ed. by S. Howard, J. Hammond and G. Lindagaard. (London: Chapman & Hall).

KAUR K. and SUTCLIFFE A.(1997). Modelling interaction for virtual reality systems. Poster Paper. In: Human-Computer Interaction: Proceedings of Interact '97, ed. by S. Howard, J. Hammond and G. Lindagaard. (London: Chapman & Hall).

KAUR K., SUTCLIFFE A. and MAIDEN N. (1998). Improving interaction with virtual environments. Paper presented at The Institution of Electrical Engineers Colloquium on 'The 3D interface for the information worker', London, May 1998.

KAUR K., MAIDEN N. and SUTCLIFFE A. (1998). Interacting with virtual environments: an evaluation of a model of interaction. Interacting with Computers. *To appear in special edition on human-computer interaction and virtual reality.*

KAUR K., SUTCLIFFE A. and MAIDEN N. (1998). Usability requirements for virtual environments. Short Paper. *To be presented at The Human-Computer Interaction '98 conference, Sheffield, September 1998.*

SUTCLIFFE A.G. and KAUR. K. (1998). Evaluating Virtual Reality User Interfaces. *Submitted for publication.*

TROMP J., KAUR K., HAND C., ISTANCE H. and STEED A. (1998). Usability for VR interfaces. Paper for discussion at The 2nd Collaborative Virtual Environments Conference (CVE '98), Manchester, June 1998.

Glossary

NOTE: Words in *italics* are also defined in the glossary.

Term	Definition
Additional stage of interaction	Observed stage of *interaction*, from the *user* studies, which was not predicted in the original *theory of interaction*.
Agent	Type of component in a *virtual environment* that has intelligence to carry out actions independently of the *user*.
Basic Support	A type of *generic design property* where the *virtual environment* fulfils basic, fundamental requirements for *interaction* and task completion.
Cognition	Mental reasoning and decision making, using knowledge in memory and inputs from sensory and perceptual processes.
Cognitive map	Structure in memory representing known information about the spatial layout and contents of a space.
Correspondence rules	Component of the *theory of interaction*, consisting of IF...THEN statements that specify the conditions under which potential *usability problems* are likely to occur.
Direct manipulation	An interface style that promotes *interaction* based on the manipulation of continually represented computer-based objects, such as icons, menus and windows.
DM	*Direct manipulation*
Domain	One of the *knowledge sources* in the *theory of interaction*, which represents known information about the application domain.
Element of user knowledge	Item of information in the *knowledge sources* that is relevant during one or more of the *stages of interaction*, such as the identity of an object in the *virtual environment*.
Element type	Class of component in the *virtual environment*, such as object, action or event, that the *theory of interaction* refers to.
Environment Available	One of the *knowledge sources* in the *theory of interaction*, which represents information from the sub-section of the *virtual environment* currently available for perception.
Environment Model (1)	One of the *knowledge sources* in the *theory of interaction*, which represents known information about the current environment.
Environment model (2)	The graphical design of the *virtual environment*, which is used in displaying the environment to the *user*.
Expert system	Rule-based computer system used to implement the *correspondence rules* in the *theory of interaction*, for predicting *usability problems*.
Explore navigate model	One of the *models of interaction* in the *theory of interaction*, which models behaviour in exploring the *virtual environment* by *navigating* around with a target in mind or looking for features of interest.
Flow of interaction	The movement of *interaction* behaviour between *stages of interaction* in one of the *models of interaction*.
GDP	*Generic design property*.
Generic design properties	Component of the *theory of interaction*, consisting of a set of generic *usability* requirements that a design for a *virtual environment* can incorporate to avoid or overcome *usability problems*.
HCI	Human-Computer *Interaction* – the study of all aspects of systems involving people and computers.
Head-mounted display	Device which fits on the head and provides displays an immersive display of the *virtual environment*, using independent screens for each eye.
Hypertext	An interface style that involves an abstract information space, which consists of text fragments connected by access paths and a set of standard operations for *navigation*, such as 'next page'.

Information Providers	A type of *generic design property* where the *virtual environment* provides useful information for *interaction*, that could also be found in the internal *knowledge sources* of the *user*.
Interaction	Exchanges the *user* has with the *virtual environment*, including perception, traversal and manipulation of it.
Knowledge source	Component of the *theory of interaction*, consisting of a set of relevant sources of knowledge, potentially available to the *user* during *interaction*.
Mode of interaction	Set of interactive behaviours, described in one *model of interaction*, that are driven by a particular planning strategy. There are three modes in the *theory of interaction*: *task action*, *explore navigate* and *system initiative*.
Mode transfers	The movement of *interactive* behaviour between different *modes of interaction* in the *theory of interaction*.
Models of interaction	Component of the *theory of interaction*, consisting of models of mental and physical behaviours for different *modes of interaction*.
Motion sickness	Nausea and related problems resulting from the *user* receiving unexpected/unfamiliar sensory information, concerning the orientation and movement of their body.
Multi-modal	Incorporation of more than one mode of communication, such as image, sound and touch.
Navigation	Movement and directing of movement through a space, such as a *virtual environment*.
Norman model of interaction	Established model of interaction for general systems, developed by D.A. Norman, which has been used in the development of the *models of interaction* in the *theory of interaction*.
Orientation	The understanding and adjusting of the *viewpoint* from which the *virtual environment* is represented, so that it is aligned as required.
Other Environments	One of the *knowledge sources* in the *theory of interaction*, which represents known information about other *virtual environments* used.
Presence	The experience of being present inside and participating in the *virtual environment*.
Real World	One of the *knowledge sources* in the *theory of interaction*, which represents common known information about general real world phenomena.
Realism	Correspondence between the *virtual environment* and aspects of the physical world.
Self object	Component in the *virtual environment* that represents the *user* and through which the user *interacts*.
Stage of interaction	Coherent *interaction* activity that has links with related activities, as described in the *models of interaction*.
Stage transition	Actual movement of *interactive* behaviour from one observed *stage of interaction* to another, in the *user* studies.
System control	Activity carried out by the underlying system which directly affects the *user*'s ability to *interact* with the *virtual environment*, such as controlling *navigation* for the user.
System events	Events that occur in the *virtual environment* which are initiated by the underlying system, rather than the *user*.
System initiative model	One of the *models of interaction* in the *theory of interaction*, which models the *user*'s reactive behaviour to *system events* and periods of *system control*.
Tactile interaction	*Interaction* involving the sense of touch.
Task	One of the *knowledge sources* in the *theory of interaction*, which represents known information about *user* tasks for the application domain.

Task action model	One of the *models of interaction* in the *theory of interaction*, which models behaviour in carrying out planned actions as part of the *user* task or current intention.
Texture	The detailed patterns and colours of object surfaces that can be graphically modelled.
Theory of interaction	A structured set of hypothetical ideas about how *users' interact* with *virtual environments*. It comprises: *models of interaction*, *knowledge sources*, *generic design properties* and *correspondence rules*.
Think-aloud protocol	The *user's* talking aloud of *cognitive* processes, as they are being undertaken.
Usability	A measure of how well the system supports the *user* in carrying out their task or fulfilling goals. It centres on how comprehensible, easy to use, efficient and pleasant the system is for the end-*user*.
Usability problems	Critical incidents or breakdowns which interfere with the *user*'s ability to efficiently and effectively *interact* and complete tasks.
User	The person *interacting* with the *virtual environment* to achieve a set of goals.
User object	See *self object*
User representation	See *self object*
VE	*Virtual Environment*
Verbal protocol	See *think-aloud protocol*
Viewpoint	Position in the *virtual environment* from which the *user* receives environment output, such as the image and sounds.
Virtual Environment	A type of computer interface that involves a 3-dimensional graphical model of some structure or place that the *user* can *interact* with and *navigate* through.
Walk-up-and-use system	A computer system that needs to have a fast learning time, allowing *users* to be successful from their very first attempt at using it, such as museum information systems.