Jokko Juhana Korhonen

Interactive Television Usability-Case: TELE Omavisio

Master's Thesis in Information
Systems Science
14.07.1998

UNIVERSITY OF JYVÄSKYLÄ
Department of Computer Science and Information Systems
MSc Programs in Information Technologies
Digital Media

Abstract

Korhonen, Jokko Juhana
Interactive Television Usability – Case: TELE Omavisio
Jyväskylä, University of Jyväskylä, 1998.
136 pages.

Master's Thesis in Information Systems Science

This thesis is a usability study of Omavisio, a prototype interactive television system. Interactive television (iTV) has been under development for three decades. After numerous technical obstacles developers are now in a position to launch the next generation of television. From a Human-Computer Interaction perspective the most important determinant of iTV market penetration is usability. Usability is important because it affects user satisfaction, efficiency and can lead to significant cost savings for both users and developers. Usability is equitable with user interface quality. The user interface is critical to any interactive system because it allows interaction between the user and the system. It provides the controls and information the user needs to accomplish tasks. What constitutes user interface quality depends on the system domain and is determined through usability testing.

Usability tests were conducted to determine Omavisio user interface quality. The Concurrent Verbalization method was employed to generate usability data which was analysed with the Command Language Grammar framework.

The results show that Omavisio is not satisfying to use. The user interface is not efficient, learnable or memorable. An inadequate system conceptual model and numerous interface design errors lower Omavisio usability. Navigation, the primary user task, is not adequately supported in the interface. Inconsistency and lack of feedback make it difficult for users to form an accurate representation of the system image. These results support existing Human-Computer Interaction paradigms about usability. Findings about the system conceptual model are applicable as guidelines to the design of all iTV systems.

KEYWORDS: Usability, User Interface, Interactive Television,
Concurrent Verbalization, Usability Engineering,
Command Language Grammar.

1. Introduction

The first chapter introduces the field of this work. The aims and scope, methodology and results of the research are stated. The organization of the report is presented at the end of the chapter.

1.1 Field of this Work

1.1.1 Central Concepts

Human-Computer Interaction

Human Computer Interaction (HCI) is a multidisciplinary field, where knowledge from *cognitive psychology*, *computer science* and *graphical design* is used to create design methodologies that help design usable systems. Given the large body of knowledge that can be drawn from in the design of user interfaces, it is not surprising to find that different user interface design methodologies exist. Despite the various approaches to interface design, it is generally accepted that an *iterative process*, composed of prototyping, evaluation and redesign improves system usability.

Usability

The user interface is the single most important aspect of an interactive system. It is especially important in the use of embedded systems, camouflaged as consumer products, because end-users are not required to have any training prior to the use of the system. A well designed user interface is functional, useful and pleasant to use. When designing interfaces the designer has to account for the user mental model. User comprehension is often characterized as the construction of a mental model that represents the objects and semantic relations described in the interface. The goal of the designer is to influence the user model of the system in a manner which will make it easier for the user to comprehend the system. The designer can do this through the system image, with metaphors and choice of interface styles. However, to arrive at the right design decisions, i.e. which metaphors and styles to use, context specific usability of the system has to known. The designer has to know the user needs, skills, and types of uses for the system. Without this knowledge, goals for the interface cannot be set and without goals an iterative design process like usability engineering cannot be employed efficiently. In the case of new systems, which lack competitors, the discovery of usability determinants is crucial to the development of the system. The best way to arrive at usability determinants for novel systems is to conduct usability tests.

Usability engineering is a methodology which espouses the idea of iteration, advocating usability heuristics and *usability testing* as the tools for creating good user interfaces. While heuristics support design decisions, in the case of novel systems usability testing supports the validity of the applied heuristics. The emphasis in usability engineering is on discovery: experiments as formalized decision processes are rare. Usability testing is a qualitative methodology, at the core of which is the discovery of characteristics and problems, not the discrimination of theoretical hypotheses.

1.1.2 The Research Domain

In the near future the public at large will communicate directly with companies and purchase products via software interfaces, the same interfaces that will also allow them to screen television broadcasts, play games and order movies. This is the vision that drives interactive television developers to develop novel technologies applicable to the home delivery of multimedia services via the television set. With development already in its third decade, interactive television has been a long time coming. However, the recent breakthroughs in digital video compression and expansion of broadband network capacity indicate that interactive television which has thus far existed only at the trial-level is ready to go mainstream. Hence, the emphasis in the development of interactive television needs to shift from the "traditional" technical network issues to user centred usability issues.

The user interface plays an important role in the success of interactive television. It is believed that users will use interactive services only if these services can be easily integrated into their already information filled lives. Actually, the interactive television user interface has a dual function: it dictates the quality of human-computer interaction and is a carrier of the system's attractiveness and purchasing appeal.

HCI research has concentrated on applications for which the computing systems are clearly present and usable to the trained (Brouwer-Janse, 1992). There is an ever increasing need to research the problematic issues involved in the design of embedded system user interfaces. It is important to discover the design requirements that arise as a result of the diverging computer skills in the wide heterogeneous user population. In all, the interface community needs to understand the opportunities afforded by the merging of TV and multimedia, in order to create "a successful bridge between them" (Mountford, 1992 p.227).

1.2 Scope and Objectives

This is a usability study on the Omavisio iTV prototype user interface. This study has a functional and a fundamental objective. The functional objective of this study is to generate product specific usability information about the Omavisio user interface. In addition, the Omavisio concept is evaluated. Overall, the objective is to learn about the system's attractiveness and purchasing appeal. The fundamental objective is to illuminate some of the issues involved in designing general purpose user interfaces that are displayed on television sets. The results of the study may be of use to other domains of HCI that are interested in the design of similar applications.

This study does not evaluate the range and types of services offered on the Omavisio channels. Neither does this study present or evaluate the design process used by the designers of the Omavisio prototype.

1.3 Methodology

The present study is driven by user data gathered from observing user behaviour and cognitive processes. An understanding of the user's cognitive processes is essential in determining the domain and especially product specific usability attributes. *Verbal protocol analysis* is the usability testing methodology used in the present study. The analysis of the data reveals what usability means in the context of the Omavisio interface. Ultimately, this knowledge can be used by designers to embed usability into the Omavisio system through the iterative design methodology.

1.4 Results

The knowledge presented to the user and the knowledge expected and required by the user did not match in amount and quality. The complexity of the system is not represented clearly in the user interface. The purpose or use to which the users employ the system had not been predicted accurately. Users wanted to use it resourcefully by researching and exploring the information space. The user interaction which demands the most support is navigation.

1.5 Organization of the Report

Chapters 2, 3, 4, 5, 6 clarify concepts relevant to the present study. These concepts are interactive media, iTV, the user interface, interactivity, usability and the Omavisio system.

Understanding usability requires that the application domain is familiar. Therefore, the fundamental differences between interactive media and traditional media are pointed out in chapter 2. In particular Attention is paid to the iTV domain.

Chapter 3 on the user interface covers the important concept of interaction style. *Interaction style* dictates how the user controls the system as well as how system functionality is visually represented to the user. Although user interfaces may differ in terms of style, they can all be described in terms of *structure*. User interface structure is elucidated by the Command Language Grammar (CLG) which is presented at the end of the section.

Next, *interactivity* is considered from a *human information processing* perspective. It is shown that interaction is premised by user knowledge which is acquired from the user interface or is summoned from prior experience. The section explores the cognitive processes of the user and the types of knowledge about the system the user needs in order to successfully accomplish goals.

Chapter 5 presents the concept of *usability*. Usability in the context of a particular application is defined in terms of the attributes which affect the user's understanding of the system. Usability testing allows the designer to discover the components and attributes of the user interface that contribute to the user's understanding or lack thereof.

Chapter 6 presents the Omavisio user interface. The user interface is described in terms of the Levels in the CLG framework.

Chapter 7 reports on the user interface usability tests. Test-users were instructed to think aloud while using the Omavisio system. Their verbalizations were recorded and the test sessions were video taped. The test report is preceded by an overview of the test methodology. The chapter concludes with an analysis of the test data. Analysis consisted of abstracting and reconsidering the data till important usability attributes emerged. The results are categorized according to the Levels of the CLG framework.

Chapter 8 is a discussion of the results. In this chapter, changes to the Omavisio user interfaces are proposed. The enumerated usability attributes will be collated into a set of design guidelines. An attempt will be made to generate iTV user interface design heuristics by generalizing the proposed guidelines.

The final chapter of the present study summarizes the results. Furthermore, in chapter 9 the applicability of the results for generalizations about human-computer interaction is evaluated. Finally, possible future work is outlined.

2. Interactive Media and iTV

Chapter 2 introduces the concept of interactive media and examines how interactive multimedia services will continue transforming television. The chapter begins by elucidating the concept of interactive media. This is followed by an overview of iTV and its development history. Finally, the interactive Media-On-Demand (herein IMOD) project is presented.

2.1 Interactive Media

According to the Webster's New World Dictionary (1982), *media* is a means of communication which can be used to reach the general public. Organizing media in terms of human sensory abilities gives us three main forms of media: visual, aural and tactile (Rada, 1995). These forms can be broken down into specific instances, with text, images and video grouped under the rubric of visual media, while sounds are an instance of aural media and gestures, tactile media.

When several media are synchronized and employed simultaneously the resulting communication form is *multimedia. Hypermedia* is multimedia with links between the media components and a mechanism for moving along the links (Rada, 1995). Hypermedia allows the user to actively specify, free from temporal restrictions, which media components are used in the multimedia presentation of information. With hypermedia the user is no longer bound by predetermined configurations of media but instead is able to select as well as respond to the information, the multimedia content, through a *user interface*. The terms *new media* or *interactive media* are used to represent such ”responsive” two-way media.

2.1.1 Interactive Media and the User

> *"The nature of interactive media is to allow divergent, personalized paths without necessarily providing the reassuring consensus view of broadcast television."* (Mountford, 1992 p.227)

McLuhan (1964) classifies media into "cool" and "hot" media. McLuhan defines "cool" or interactive media as media that demands active participation and an "involvement in process" while "hot" media, like print, encourages passive consumption. A similar view on the contemporary dichotomy of media is presented by McAdams (1995). Old or "hot" media represents the delivery of information via a broadcast on a one-to-many model, while new or "cool" media involves the exchange of information on a one-to-one or one-to-many model.

Interactive media can be as complicated as it is exciting to the user. Since linearity is broken up in interactive media, the user is able to create *information mosaics* according to his or her needs (McAdams, 1995). The creation of information mosaics, analogous to information selection, happens through user interaction: a set of actions which lead to a desired outcome. Driven by the exchange and processing of information, interaction in itself can be complicated for the user. These complications are in part due to the nature of information.

Broadly defined as everything that enters our sensory organs, most people consider *information* as something that makes sense or has significance. Information to one person can be *noise* to another. Since noise problematizes the search and interpretation of information, it also complicates interactivity. To avoid noise, interactivity or the use of information involves the processes of exploring, discovering, seeking and finding. Interactive systems should support these user processes, thus enabling users to accomplish tasks, i.e. information selection and processing, more easily. More importantly, user understanding of the system itself should be promoted in design since the system is the means of accessing information. Interactivity, the user actions involved in the use of a system, is the focus of chapter 4

2.2 iTV Past and Present

While McLuhan considered television to be "cool" media, breaking up the linearity of print culture, new or interactive media takes this break-up to another level. In comparison to interactive media, television can be considered a linear media (McAdams, 1995). In the present study, contrary to McLuhan's view, TV is considered to be old electronic media. The new media "upgrade" of TV is *iTV*. iTV is the synergistic merging of TV and computer multimedia (Mountford, 1992).

The concept of iTV existed prior to the advent of digital multimedia technology. The first experiences with digital iTV using two-way cable systems are from the early 1970's (Richter, 1983). Two-way interactive cable (TV) was considered in the 1970's as the means to develop local communications networks of economic and social importance. The first study on the social benefits of interactive two-way cable was conducted in 1974. Back then the National Science Foundation (NSF) in the United States funded three experiments to empirically determine the relative costs and benefits of two-way cable. Lewis (1983) reports that the experiments demonstrated the applicability of interactive two-way cable to at home education and showed how information delivered via the cable system increased community awareness, especially awareness of social programs.

Two and a half decades later, numerous local and global networks provide information and services similar to those tested in the NSF experiments. However, against expectations the TV set has not become the primary means by which these networks and the information contained within is accessed. Actually, the infrastructure and content of networks has materialized as a result of the advances made in personal and network computing. The popularity of the Internet and the Personal Computer (inhere in PC) signify how indispensable information technology and networks have become.

Presently, the differences between computing and entertainment industries are beginning to blur, an indication that the age of multimedia services delivered to homes via networks is fast approaching. Referred to as *convergence*, the fusion of computer and television is the one of the most significant contemporary trends in the computing and telecommunications industries. The introduction of multimedia PCs bears witness to this trend. Similarly, low-level interactivity has been introduced to television culture in the form of "ring-in" talk shows and digital satellite television. Moreover, Internet Set-Top-Boxes, stand-alone boxes which display World Wide Web pages on TV sets using phone lines are one of the first "true" convergence products to reach the consumer market. The convergence trend is expected to result in the development of a new type of application that unites the offerings of the computer, entertainment and communications industries (Arnum, 1995).

2.2.1 Multimedia Services: the Fruits of Convergence

Convergence and the nature of the emergent new media is best understood by examining the features and characteristics that each "parent", computing and television, contributes. Computing adds to television the attributes of democracy, flexibility, simulation and personalization. With computing television will no longer be a broadcast, time based media. Meanwhile, television adds to computing shared viewing, moral issues and accessibility. The later of the attributes places the most usability constraints on the new media. The merging of these media attributes is already visible in the proposed services that will form the economic backbone of the new media in question. Some of the basic services delivered to the home via iTV will include bank-at-home, data retrieval, video-on-demand, shopping and polling or voting on issues. The types of public and institutional services also expected to be offered to homes include education and instruction and teleconferencing. In short, the next generation of television brings interactive multimedia and hypermedia to the home.

A classification of iTV services in terms of interactive methods gives a more comprehensive view on the types of services and interactivity that are available to user. There are two sets of interactive methods introduced to the home by iTV. The first set of methods consists of *navigation, selection* and *reception* of video content, usually referred to as *video-on-demand* (VOD), where the user is able to select and view video programs on a demand basis. Navigation is especially important since without it the user is not able to effectively and freely select from all the services and videos that are offered.

The second set of methods includes methods needed to actively participate with video content. This type of service allows the user to interact directly with programs and commercials: using an input device such as the remote control the user can play along, learn, purchase products and services.

A third type of service that the iTV platform offers is the provision of access to on-line services. The user is able to interact with Internet content using a remote control or an infra-red keyboard. In this sense iTV can also be considered a sophisticated Internet Set-Top-Box that can also access proprietary media networks.

2.2.2 Technical Components of iTV

Digital technology, similar to technology used for networked multimedia applications, is required for the provision of the above mentioned services. First of all the content or services have to be digitized. The digitized content has to be compressed in order to minimize storage space required from the *storage hierarchy and control system*, also referred to as the *multimedia server*. These servers must be able to service all in-coming requests. A *transport network* is required to deliver the vast amounts of information in a timely manner. This network may consist of *multimedia distribution nodes* which are connected to Local Area Networks or Wide Area Networks and act as "session" managers between the servers and the *Set-Top-Box* in the user's home. A *return path* is needed to carry the user's requests back to the multimedia server. This path is usually integrated with the transport network. The Set-Top-Box (herein STB), an addressable communication box, decodes the encoded compressed digital data it receives and converts it to analog signals displayed on the television. The STB also receives commands from the user and transmits the commands back to the servers. Users need a *user interface* (UI) which allows them to choose between the services offered and communicate their requirements and responses to the multimedia server. An input device, usually an infra-red remote control is used to issue commands to the user interface. These components are likely to be tightly integrated because the lack of industry standards has not made inter-operability a priority among developers (Sweeney, 1995).

2.2.3 iTV Trials

The formation of partnerships and alliances has characterized the development of iTV and is a result of firms spreading the risks of innovation and acquiring expertise in different sectors. In 1993, there were over 500 digitally driven alliances between media, carrier and computer companies (Sweeney, 1995). Out of all the "players" involved in realizing iTV, the telephone industry has emerged as the leader, ahead of cable companies.

The Benton Foundation report (Krasilovsky, 1995) on ITV Testbeds describes 17 telephone and cable industry interactive TV testbeds. The report shows vigorous investments being made in iTV by content owners and the technology and telecommunications industries. The testbeds described in the report diverge in the technology employed for the delivery of services to the homes as well as in the composition of core services. The lack of a standard definition or platform for iTV has been a result of the constant redefinition of iTV technology by firms, alliances, regulators and end users. The development of iTV has been punctuated by firms switching their priorities, allegiances and by the discovery of new uses for old technology.

iTV is best understood as an example of *networked multimedia* applications. The challenge in the technical development of iTV has centred around the network issues of latency, reliability and security. Most of the development work has been conducted on the physical network, communications protocols and platforms. In fact, iTV is often defined in terms of the technology involved, i.e. the transport network, and bandwidth, instead of the degree of communication afforded to the end user and the actual content. Other key technologies under development and usually tested in trials are databases, interfaces, servers and set-top-boxes. Content development, the economic backbone of the industry, has not been as actively pursued mainly due to the lack of a standard platform (Rahmat, 1995).

The present study aims to further the development of a specific iTV platform so that it can be tested in trials set to run at the end of 1997. Next, the iTV platform in question is introduced.

2.3 Interactive-Media-On-Demand Project

Interactive-Media-On-Demand (IMOD) is a joint venture between Telecom Finland Ltd. and its co-operation partners. The aim of the project is to develop an iTV platform and broadband TV environment and upon completion to field test the resulting applications. The project has five main objectives:

1. to demonstrate the functionality of the On-Line Media iTV platform in a controlled lab environment,
2. to determine the technical boundaries of the platform,
3. record user feedback on the quality of the service (including the user interfaces),
4. make recommendations for the future service and
5. create partnerships with content providers.

The project is funded by TEKES, the Finnish Technology Development Center, and is a part of the Finnish national multimedia program.

2.3.1 The Technical Broadband TV Environment

The project aims to create a broadband TV environment that allows the delivery of multimedia services to the home using the television set. The technical environment is composed of the television set, a Set-Top-Box, a multimedia server and a high-speed transport network, which also has a return path for the user signals.

The television set is connected to a high-speed network by way of a Set-Top-Box (STB), or subscriber terminal. The STB is supplied by UK-based On-Line Media and it handles the decoding of digital video streams. An advanced WWW-browser has been added to the STB, allowing the user to access the Internet from the same high-speed network. The STB is equipped with an infra-red remote with which the user is able to communicate with the system. It is possible to use other input devices such as an infra-red keyboard and joystick.

The user receives multimedia services from a multimedia server. The server used in the project is a Digital's Mediaplex server, a 64-bit AlphaServer computer that handles the storage and distribution of MPEG 2 digital video streams. MPEG 2 is a standard developed by the Motion Picture Experts Group for the transmission of digitized video and audio for iTV network. MPEG 2 video is compressed at 30:1 and requires 6 Mbps of bandwidth for transmission (Microware Systems Corporation, 1995).

The IMOD transmission network uses ATM (Asynchronous Transfer Mode), an internationally standardized technique for information-transfer enabling integrated live picture, sound and data. The bandwidth of the network is 6 Mbps .

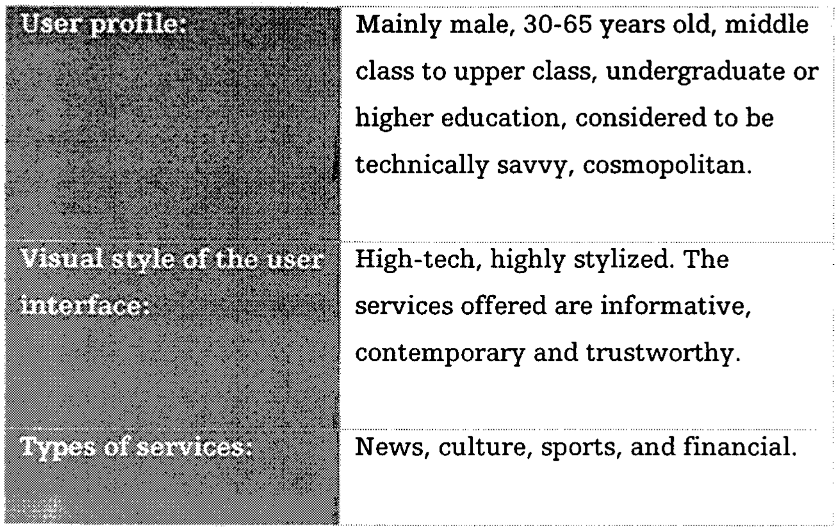
2.3.2 Tele Omavisio Concept and Structure

The Omavisio application is intended for households that have not yet embraced the PC because of its costliness and complexity, yet are looking to access the Internet, as well as to benefit from interactive multimedia content and services as they become available. It should be noted that Omavisio does not compete with the PC, it is an alternative means of accessing the World Wide Web and other Internet applications, but also provides services not yet available to PC users such as VOD.

Omavisio users can order various multimedia services, such as news and home-shopping, games and Video-On-Demand. The Omavisio services are classified according to three core user profiles. These profiles are based on general demographic attributes such as age and income.

Each profile has a visually distinct user interface. A channel *metaphor* is used to represent the grouping of services. The three Omavisio channels are OMA1, OMA2 and OMA3.

OMA1 The "Urban" channel.



User profile:	Mainly male, 30-65 years old, middle class to upper class, undergraduate or higher education, considered to be technically savvy, cosmopolitan.
Visual style of the user interface:	High-tech, highly stylized. The services offered are informative, contemporary and trustworthy.
Types of services:	News, culture, sports, and financial.

OMA2 The ”Family” channel.

User profile:	Families, 5-95 years old, all social classes, unemployed, employed or retired, considered to be conservative with little interest in technology.
Visual style of the user interface:	Peaceful, cheerful, plain. The services are informative, local, contemporary, entertaining and involving.
Types of services:	Entertainment, games, local events and services, hobby and quality of life related.

OMA3 The “Youth” channel.

User profile:	Mainly male, 15-25 years old, currently in school, considered to be social, technically savvy, urban, active.
Visual style of the user interface:	Loud, colorful, exciting, compelling. The services offered are contemporary, entertaining, involving.
Types of services:	Entertainment, games, popular culture, sports, hobby related..

Access to the Internet is also dictated by user profile: each channel has its own navigation starting point to the World Wide Web (WWW). All Omavisio users have the same Internet services available to them. These services include e-mail, news groups and data banks (WWW).

Additional profiles can be added to Omavisio. Content developers are free to create channels to accommodate their mix of services in the Omavisio system.

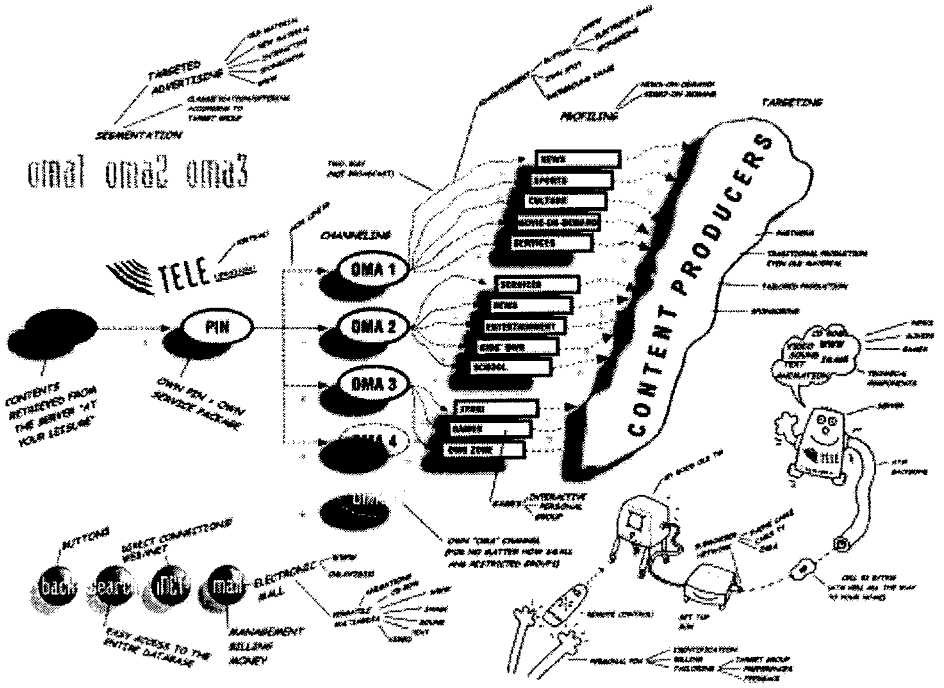


FIG. 2-1 *Omavisio service structure, prototype Spring 1997*

The functionality and features of all three Omavisio profile user interfaces are identical. The Omavisio user is provided with interactive methods to navigate, select and receive video content, actively participate with the video content as well as access to the Internet. A detailed description of the Omavisio user interfaces can be read in chapter 6.

3. User Interface

General knowledge of user interfaces is critical for the present study. The following chapters, through chapter 6, deal with the user interface. In the subsequent sections the concept of the user interface is illuminated by considering its significance from the user's point of view - interactivity and usability - as well as the designer's point of view - usability engineering.

The aim of chapter 3 is to clarify the user interface concept. This is done by examining the important components and aspects of an interface. At the beginning of the section a general definition for the user interface is given. This definition is refined in the following sections. Next, a review of the different interaction styles, the consistent interaction techniques imposed on the user, is presented. In particular, two styles one frequently used in embedded system interfaces are covered in detail. It is argued that the rules for choosing an interaction style are generally not practicable and therefore, the choice of style has to be discovered through iterative testing and redesign. Finally, and most importantly, the structure of user interfaces is examined with the aid of Moran's Command Language Grammar representational framework.

3.1 A General Definition for the User Interface

The *user interface* is the boundary between the user and the system; it consists of the physical, perceptual and conceptual aspects of the system that the user of a system comes in contact with (Moran, 1981). According to Butler (1996) the purpose of the user interface is to provide the user with the visibility and control needed to effectively exploit the system's functionality and data. The user interface allows humans and computers, or humans and interactive systems to interact and therefore, determines what the experience of using a system is like for the user. In sum, the user interface is often the single most important factor to the success of an interactive system (Baecker and Buxton, 1987).

3.2 Interaction Styles

3.2.1 The Development of User Interfaces and Interaction Styles

Nielsen (1993) identifies five *generations* of user interfaces. A *generation* is the period from when early adapters start using a particular user interface to when they start using the next new one. The first or *Pioneer* generation of user interfaces emerged in 1945. The user interfaces from this generation lacked interactivity, and were based on either the batch and programming paradigms. With the invention of time-sharing around 1955, computers became information processors and a new *Historical* generation of line-oriented interfaces, founded on the paradigm of a command language, followed. Line-oriented interfaces were one-dimensional with the user interacting with the computer by typing commands on a single command line. Full-screen interfaces added a second dimension to the interface and is called the *Traditional* generation (1965 - 1980). Interaction was based on hierarchical menus and form fill-in. The fourth or *Modern* generation of interfaces introduced the now prevalent paradigm of the graphical user interface with its Windows, Icons, Menus and a Pointing device. The current generation, from 1995 onwards, called the *Future* generation by Nielsen (1993), is composed of networked single-user system and embedded computer systems like iTV. Nielsen claims that non-command-based interfaces will come to dominate this generation.

It is worth noting that the historical development of user interfaces has been characterized by each new generation containing the previous ones as special cases (Nielsen, 1993). This phenomenon is clearly evident in the following discussion on interaction styles.

3.2.2 Interaction Style

Today a variety of interactive systems exist. The arrival of embedded systems, where the computer is "hidden" from the user, has added new types of user interfaces, to the multitude of computer system interfaces that already existed. Usually interactive systems are classified by application type, e.g. text editing, graphics, etc., however, this dimension is not useful for categorizing user interfaces. This becomes evident from the fact that diverse application types can have similar user interfaces, and different user interfaces can be used on the same application type (Moran, 1981). In the above overview a classification of user interfaces i.e. batch, line-oriented, full-screen and graphical, was made based on the dimensions of *display use* (one-, two- or three-dimensional) and technological development in general. A more accurate dimension, one which describes the way interactive methods are implemented in the user interface, is dialog style.

The concept of *style* is the foundation for understanding interactive systems. *Dialogue style*, also called *interaction style*, is the unified and consistent set of interaction techniques imposed on the user (Baecker and Buxton, 1987). Examples of interaction techniques are menus, form filling dialogues, undo commands, icons, etc.; these are the aspects that are commonly cited when differentiating among user interfaces. Next, two different classifications of interaction styles are presented. Knowledge about interaction styles is applied in this study in the analysis of the test results as well as in the discussion part of the report where recommendations are made regarding which style to use in an iTV user interface.

3.2.3 Two Classifications of Interaction Styles

Over the years as computers started taking advantage of the modifiable nature of the screen, progressing from a modifiable single line in line-oriented systems to a dynamic full-screen in graphical user interfaces, the dimensionality of user interfaces and number of interaction techniques has grown. This evolution has augmented the user's feeling of being able to control the system (Nielsen, 1993). The present proliferation and diverging use of interaction techniques makes the categorization of styles problematic. Nonetheless, an organization of interaction techniques in terms of nine major general categories of interactive style is performed in Baecker and Buxton (1987). A similar, more dissecting organization of interaction techniques is made in Nielsen (1993).

The following holistic organization of interaction techniques into the main interaction styles is presented in Baecker and Buxton (1987).

- **Command line**

  The user issues instructions to the system by typing in formally defined commands. The dialogue between user and system is composed of questions and answers. This is the basic command language system.

- **Programming language**

  The user is able to extend the command language of the system by defining procedures himself. Nielsen (1993) considers this a subset of the command line style.

- **Natural language**

  The dialogue between the user and system is based on a well-defined subset of some natural language such as English.

- **Menu**

  The user issues commands by selecting choices from among a menu of displayed alternatives.

- **Form filling**

  The user formulates and issues commands by filling in fields in forms.

- **Iconic**

  The system commands are represented on screen as graphical symbols or pictograms. System responses to commands are also communicated in graphical form.

- **Window**

  The different functions of the system are grouped and presented on-screen in rectangular areas or "windows." Each window can also be "virtual terminal."

- **Direct manipulation**

  The user interacts with the system by manipulating a graphical representation of the underlying data. The manipulation occurs through button pushes and movements of a pointing device such as a mouse.

- **Graphical interaction**

  This type of interaction style is application specific. The user defines and modifies sketches, diagrams, renderings, and other two-dimensional and three dimensional images and pictures.

It becomes apparent from the above list that interaction style dictates the way actions, objects or choices are presented in the user interface.

Nielsen (1993), as mentioned earlier, has also presented a list of what he considers to be the main interaction styles. These styles are presented in TABLE 3-1. Nielsen's classification, has five styles in common with Baecker and Buxton's classification. The other four, Programming language, Iconic, Window and Graphical interaction styles it replaces with the *Function Keys*, *Non-Command*, *Question-Answer* and *Batch* styles.

TABLE 3-1 *Summary of the main interaction styles (Presented in Nielsen, 1993)*

Interaction Style	Mainly Used In	Main Characteristics
Batch	Batch Processing, email servers	Does not require user intervention, works even when user and computer are in a different time and place
Question- Answer	Line-oriented	Computer controls the user, so suited for casual use.
Command Language	Line-oriented	Easy to edit and reuse command history. A powerful language a can support very complex operations.

Function Keys	Full-screen, WIMP	Fast entry of a few standard commands, but limited flexibility.
Form Fill-in	Full-screen, WIMP	Many fields can be seen and edited at once.
Menus	Full-screen, WIMP, Telephone-based interfaces	Frees the user from remembering options, at cost of potentially being slow or having confusing hierarchy.
Direct Manipulation	WIMP, Virtual Reality	User in control. Enables metaphors from real world. Good for graphics.
Non- Command	Future systems, Virtual reality	The user is freed to concentrate on the domain and need not control the computer. Computer monitors users and interprets their actions, so suited for cases where misinterpretations are unlikely or without serious consequence.
Natural Language	Future systems	Ideally, allows unconstrained input to handle frequently changing problems.

While complements of each other, even together these two classifications are by no means definitive or comprehensive. The differences between the classifications indicate that the dimension of interaction style lacks formality. In sum a user interface cannot be described in terms of style alone.

3.3 Strengths and Weaknesses of the Menu and Direct Manipulation Interaction Styles

A brief description of the strengths and weaknesses of interaction styles is presented in Nielsen's table of styles. The most relevant styles from the perspective of this study are the menu and direct manipulation styles. An in-depth examination of the strengths and weaknesses of these two styles is presented next.

Menu Interaction Style

First, a parallel has to be drawn between the two classifications of styles. While Baecker and Buxton consider the use of menus as an interaction style, Nielsen, whose categorization of styles is more dissecting, seems to consider menus an interface type where the function-key style is used as the primary interaction style. The function-key is often called an *accelerator.* Despite the difference in point of view, the basic interaction style paradigm is the same in both classifications: the packaging of commands into a single user operation. When numerous packaged commands are presented simultaneously they form a menu. The grouping of the commands into a menu is generally done to reflect a certain mode, a specific context of system functionality.

The menu interaction style is ideally suited for systems where the user has not had any prior training in the use of the system. Hence, everything the novice user needs to know about operating the system is visible in the user interface. Essentially, menu interaction relies on the user's ability to recognize desired inputs in the menus as well as the on the user's interpretation of system responses to selected menu items (Habinek and Savage, 1984).

The menu style is commonly used in graphical user interfaces where it is implemented as pull-down menus. Another common use of the style is in full-screen interfaces where the whole screen is used to display a menu. Each new screen contains another menu. These menus, presenting interaction choices to the user, are usually nested and consequently form a hierarchy. This type menu style is used in the Omavisio system.

Menus become difficult to use when the number of options made available to the user is very large. The result can be a menu hierarchy that the user has to learn to navigate. Navigation is undesirable since it requires the user to assimilate a new interaction technique on top of the menu style. Furthermore, it becomes increasingly difficult to find information as the number of menus grows. To cut down the number of menus it is possible to try and squeeze as many options into a single screen. However, the more choices the user has to choose from the longer it will take for the user to decide; the pace of the interaction slows down. The trade-off between navigation and the number of selections is represented by the paradigms of menu *breadth* and *depth*. These menu hierarchies are presented in FIG. 3-2.

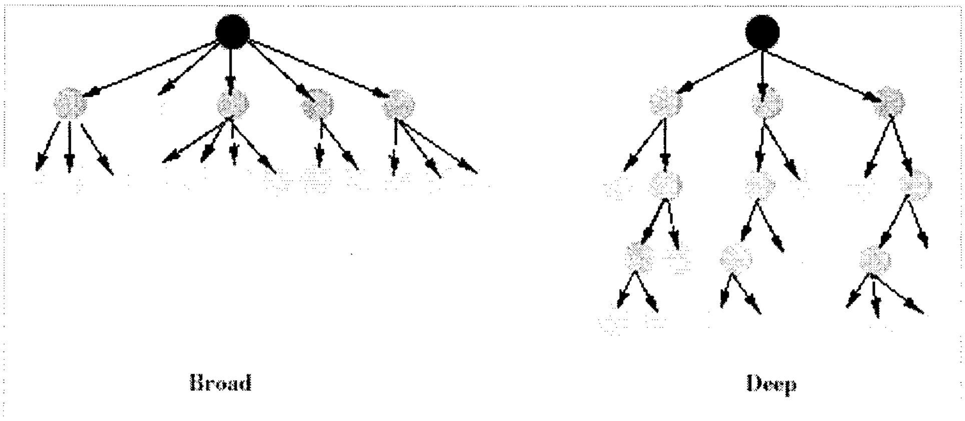


FIG. 3-2 *Example of broad and deep menu hierarchies, both providing the user access to 13 commands. (Presented in Nielsen, 1993)*

As mentioned earlier the menu style is suitable for novice users and thereby it is commonly used in embedded systems like iTV. Experienced users on the other hand may find the style tedious. Progressing from menu to menu becomes tiresome when the system is familiar to the user. To accommodate experienced users, if they are considered to be among the system users, short-cuts to system functions have to be incorporated in the user interface (Habinek and Savage, 1984). Nielsen calls these short-cuts accelerators.

While cumbersome, menu hierarchies are necessary in systems that include many features. Extensive research has been done on the design of menu interfaces (Savage and Habinek, 1984, Perlman, 1985). Navigation, a by product of the menu interaction style has been addressed in research conducted on hypertext and hypermedia (McAdams, 1995, Nielsen, 1990). Navigation of information spaces is covered in section 2.3, where in particular, user knowledge associated with navigation is examined.

Direct Manipulation Style

The direct manipulation style is also used in graphical user interfaces where it is the primary style. This style is implemented to a certain extent in the Omavisio user interface. The following expands on the definition of direct manipulation presented in section 3.2. According to Shneiderman (1983) the direct manipulation style is characterized by several unique attributes that lend the interaction process a feeling of directness. One attribute of the style is an on-screen cursor, an arrow, underscore or a box, that directs the attention of the user to the place in the interface that can be acted upon. The cursor can be controlled in an intuitive and obvious way with arrow keys or mouse or other devices. Some of the typical objects that are acted upon are buttons, sliders, icons and windows. Another important attribute is that the result of an action is immediately displayed to the user; for example, when the user presses an on-screen button, following the logic of the real world, it is instantaneously shown to be depressed. Direct manipulation requires the display and the objects to be continuously updated to reflect the user's actions. Subsequently, the user is able to interpret from the display the status and state of system objects. The last major characteristic of the direct manipulation style is that the user is able to undo or reverse actions or commands.

There are several strengths in the direct manipulation style. First, the style is easily taught to a novice user through demonstration by an expert user. Secondly, the style allows expert users to work efficiently and to customize the interface according their preferences. Error messages are rarely needed with the style since the user can infer from the appearance objects and their behaviour if something does not work and consequently change their actions accordingly. Finally, users feel at ease when operating the interface since they can reverse their actions easily if they make a mistake.

The merits of the direct manipulation style have been scrutinized in Nielsen (1993) and Hutchins *et. al.* (1986). One argument against the use of the style is that not all tasks should be done directly, especially repetitive ones. Another weakness of the style is the burden of accuracy involved in the use of a pointing device which has to be constantly tracked and finely controlled to carry out operations. Nielsen observes that the direct manipulation style can be abductive, in the sense that users, engaged on interacting directly, ignore or are unaware of alternative means of interaction, i.e. using the keyboard and accelerators, available to them. Subsequently, the eclipsing of other styles teaches the user the wrong conceptual model of the system. Ultimately this may hinder the user's ability to operate the system. A final criticism of direct manipulation is that the style relies on visual representation of objects whose meanings are at times left to the subjective interpretation of the user.

3.4 Rules for Choosing Interactive Styles

Given the strengths and weaknesses of interaction styles it is relatively simple to draw common-sense rules for choosing an appropriate style for a system. For example, it could be argued that a menu style is appropriate for iTV because it allows unskilled users to operate the application immediately and furthermore, is frees the user from remembering the choices which are abundant in the iTV information space. In fact, such rules have been proposed by Shneiderman (1991). Yet, Nielsen (1993) writes that it is impossible to give firm rules of this kind. According to Nielsen there are too many exceptions to the rules.

The notion of rules is altogether overlooked by Baecker and Buxton (1987, p. 433) who write

> *"The choice of the "best" interaction style is and will remain a complex function of the task, the users who are to carry out this task, the environment within which they will work, and the tools with which they do their job."*

In the end designers seem to be more influenced by the detailed characteristics of the system domain - elucidated through usability testing - the users and their tasks than by common rules. The result of this can be seen in user interfaces which incorporate elements from several styles, e.g. the *WIMP* interface. The name WIMP comes from the interaction styles used in interface: Windows, Icons, Menus, and a Pointing device (Nielsen, 1993).

Furthermore, while style is important to the understanding of user interfaces, it is a "superficial approach" to comprehending the common structure of user interfaces, emphasizes Moran (1981). As suggested most user interfaces are compositions of stylistic features, and therefore, a taxonomy of "pure" styles can not be used to accurately describe a user interface or its underlying structural organization. To learn about the structure and foundation of the user interface we have to turn to Moran's Command Language Grammar.

3.5 Command Language Grammar

Command Language Grammar (herein CLG) is a representational framework introduced by Moran (1981) for describing the user interface aspects of interactive computer systems. Moran's motivation for developing this framework parallels the motivation for the present study: to understand the composition and functionality of user interfaces and to account for the knowledge that contributes to the user's mental model of the system.

The intended application of CLG is in the development of command languages and command language or line-oriented interfaces. Nonetheless, as Moran himself argues in his three views of CLG, the framework can be applied to the study and design of user interfaces in general. The CLG is the main theoretical framework used in the present study. In chapter 6 the basic CLG framework is applied to describe the Omavisio user interface. The psychological view of CLG, presented in section 4.3.1, is drawn upon in the analysis of the usability testing data collected about the Omavisio user interface.

3.5.1 Components of the User Interface

According to Moran (1981), CLG can be used to describe a variety of interactive system interfaces. The framework allows for the common structure of all interfaces, including those of new systems like iTV, to be abstracted; herein lies the importance of the CLG. The framework stratifies the user interface for purposes of abstraction into a sequence of components composed of different Levels. Each Level, a refinement of the previous Level, is a distinct and complete description of the system at that level of abstraction. The progression through the Levels is analogous to going from abstract to the concrete and from gross structural features to fine details.

On a meta level, the CLG allows the user interface to be partitioned into three components. The resulting components, composed of detailed Levels, are the Conceptual Component, the Communication Component and the Physical Component, see TABLE 3-2.

TABLE. 3-1 *Level structures of CLG. (Presented in Moran, 1981)*

Conceptual Component	Task Level Semantic Level
Communication Component	Syntactic Level Interaction Level
Physical Component	(Spatial Layout Level) (Device Level)

The *Conceptual Component* contains the abstract concepts around which the system is organized. The Levels that make up the Conceptual Component are the *Task Level* and the *Semantic Level*. The *Task Level* lists the set of tasks the system is intended to accomplish. The *Semantic Level* specifies the representation of tasks in the system and the conceptual actions which can be performed.

The *Communication Component* contains the command language of the system, which enables the command-execute cycle, and forms the basis for the user's dialog with the user interface. The Levels that make up this component are the *Syntactic Level* and the *Interaction Level*. The *Syntactic Level* defines the command language used to communicate with the user interface. The *Interaction Level* describes the interface dialog structure as well as the physical actions associated with the command language.

The *Physical Component* contains the physical devices that the user sees and uses while interacting. The *Spatial Layout Level* and the *Device Level* define this component. The *Spatial Layout Level* describes the arrangement of the input and output devices and the graphical display. The *Device Level* specifies "all the remaining physical features". These two Levels are not covered in Moran's presentation of the CLG, albeit they are also essential to the accurate description of the user interface.

3.5.2 The CLG Levels

Next, the first four Levels of CLG are presented. Aspects of the Levels which pertain directly to the construction of a command language are omitted, since they are outside the scope of this study.

Task Level

Moran defines *tasks* as "the specific goals that the user will set for himself and will attempt to accomplish, usually with the help of the system" (1981, p.29). The accomplishing of goals or tasks usually requires *procedures* (means) while *methods* associate tasks with procedures. The environment the task is performed in may contain certain *entities,* which are related to the task at hand. All together the tasks, task entities, task procedures and task methods form a *task structure.*

At the Task Level, the highest and most abstract Level of the top-down CLG framework, the task structure of the tasks and sub-tasks performed with the system is described. From the perspective of the user this Level answers the question, which tasks can be performed with the system. From the designer's perspective this Level is the initial stage of designing a system. At this Level decisions are made as to what type of a system will be built.

Semantic Level

The Semantic Level describes how the real task structure derived from the previous Level is transformed into a virtual structure with conceptual components in order to replicate it in the system environment. Task entities are represented by *system entities* or objects. The system can perform certain *system operations* or conceptual actions relating to the tasks. The user of the system can also perform operations: *user operations.* The procedures and methods, in the task structure, are also conceptually represented as *semantic procedures* and *semantic methods.* Semantic procedures, the composition of user and system operations can have an input parameter and an output result and a failure condition. Semantic methods, like methods in the task structure, associate procedures (semantic) with the tasks. The end result is the definition of the system's functional capability and ultimately its task domain.

From the perspective of the user this Level answers the question about how the tasks and task operations are represented and implemented in the user interface. At this stage in the design process the designer is faced with decisions relating to conceptual representation of the task with the given limitations of the system environment. A usual design concern at this stage is what metaphor to use in the user interface.

Syntatic Level

While the previous Level was concerned with how the performance of tasks is conceptually represented in the system in terms of functionality - the operations that can be performed on conceptual task related objects - the Syntactic Level is concerned with how these operations are evoked and initiated.

This Level illuminates the types of interaction methods and physical actions that the user needs to engage in when using the system. Therefore, this Level translates to the stage in the design process where system commands, their meanings and the context (mode) in which they are valid, all have to be defined.

Interaction Level

This Level specifies the actions, in terms of specific key-strokes, needed to issue the different commands described in the Syntactic Level. The system responses, i.e. feedback, are also specified at this Level. The limitations of the system hardware affect how the designer implements this Level in the system.

An examination of the user interface at this Level of abstraction reveals the purpose of the input device buttons. In other words what does the user have to do to operate the system.

In summary, being a top-down abstraction of the user interface, each Level of the CLG framework, beginning with the Task Level, is mapped onto the components and entities of the subsequent Level. This will be evidenced in the description of the Omavisio user interface.

4. Interactivity - Mental and Conceptual Models

This section tackles from three complementary perspectives the user actions involved in the use of an interactive system. The objective of this section is to present a comprehensive view of all levels and aspects of interactivity. A model of interactivity is necessary to analyze usability test results. First, the stages of activity that constitute interactivity are presented. It is shown that interactivity consists of user tasks which are prefaced by user knowledge. Next, the cognitive process that leads to user understanding of the system is covered. Special attention is paid to mental models and their role in user understanding and problem solving. This is followed by an explication of the types of user knowledge involved in the successful completion of tasks with an interactive system. Finally, the importance of user understanding in interactivity is demonstrated through a knowledge analysis of information space navigation.

4.1 Action Theory

In the context of interactive media, *interactivity* is a term used to describe user activity. Interactivity has been the focus of numerous dissecting HCI studies which have enumerated the activities it consists of. In general interactivity has been approached from a *human information processing* perspective. In the tradition of cognitive science, an attempt has been made to elucidate user behaviour through cognitive modelling (Green, 1991). Several action-based user models have been developed by researchers with the intention of generating design principles conducive to good interface design (Young, Howes and Whittington, 1990). Next, interactivity will be examined through Norman's (1986) *descriptive analysis of user activity.*

According to Norman an understanding of user action is essential when addressing the fundamental difficulties that people generally have in understanding and using most complex devices, including interactive systems. From the perspective of this work, Norman's action theory is critical to the understanding of usability and serves as an introduction to the discussion on mental and conceptual models. The descriptive analysis of user activity is a theoretical tool devised by Norman to help develop engineering principles of design that are based on fundamental principles behind human action and performance. Norman's analysis demonstrates that the user interface is a gulf, the bridging of which is a cognitive problem solving task for the user and a design challenge for the user interface designer.

The premise of the theory is that a discrepancy exists between the user goals and the system used to attain the goals. The user's goals and intentions are *psychological* variables, existing only in the mind of the user. To use the system effectively, the user has to interpret his or her intentions and goals in terms of the system's *physical* controls and states. From this premise we arrive at the view of interactivity where the use of a system is an active process consisting of relating psychological variables to physical ones. In the reverse case where the user has to react to system behaviour, the user is obligated to evaluate the physical variables in terms of the psychological ones. In sum, goals and system state differ in form and content: goals are expressed in psychological terms which are relevant to the user, while the system's functions and states are expressed in terms relative to it.

4.1.1 User Activities in Interactivity

Norman (1986) captures the aforementioned discrepancies in the paradigms he calls the *Gulf of Execution* and the *Gulf of Evaluation,* see FIG. 4-1. The Gulf of Execution exists between the goals formed by the user and the actions required in order to achieve them. The Gulf of Evaluation exists between the system's overt response and the user's relating that response to the original goals.

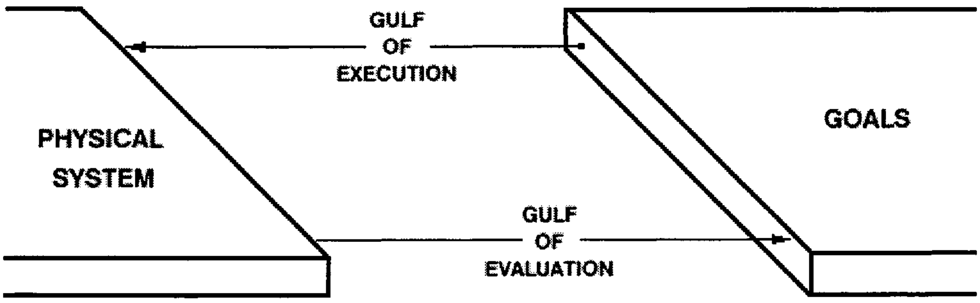


FIG 4-1 *The Gulfs of Execution and Evaluation (Presented in Norman, 1986)*

The Gulf of Execution, the gap from goals to physical system is bridged by four user activities. The bridging of the gulf begins as a mental exercise where the user initially forms an intention. Subsequently, the intention, which exists in the user's mind as psychological goals, is converted to into the physical variables of the system. Hence, the user specifies an action sequence by determining which physical system variables have to be manipulated in order to satisfy the intention. In turn, this requires that the user knows how the variables react to manipulation, that is, the user has to be knowledgeable about the resulting variable states. After all these mental processes comes the first physical action of the sequence: the user executes the action. The user physically communicates the action sequence to the system with an input device. The input device dictates the type of physical action the user ends performing, i.e. pressing a button on the remote control.

The bridging the Gulf of Evaluation requires three user activities. Once the action has been performed, or executed by inputting a command to the system, the user determines the resulting system state. This is done by perceiving and subsequently interpreting the state of the physical variables that the user has acted upon. The state of these variables which is visible to the user in the full-screen graphical display is the *system output*. The user compares the system output to the goals and intentions, a process which enables the user to establish whether or not the correct action was performed: whether or not the gulf of execution was bridged successfully.

The sequences of activities described above are a simplified account of what the user has to do in order to be able to use an interactive system. By enumerating the stages in the bridging of the gulfs, Norman arrives at the sequence of actions which constitute interactivity. In summary there are seven stages of user activity:

1. Establishing the Goal
2. Forming the Intention
3. Specifying the Action Sequence
4. Executing the Action
5. Perceiving the System State
6. Interpreting the State
7. Evaluating the System State with respect to the Goals and Intentions

Norman points out that in some instances of interactivity goal formation is not the primary stage, but results from a need to react to the system output. Typically such behaviour occurs when the user's task is to monitor system state. The primary activity in such reactive behaviour is perception of the system state which is followed by interpretation of the system state. A response is generally expected as a result of the change in system state. The user's response is often goal-oriented which in turn means that the sequence of activities which started mid-way has returned to the primary stage of user activity. Obviously, in instances where the response has to be instantaneous, i.e. almost reflexive, the stage of goal formation is superseded by reactive behaviour.

All things considered actual interactivity may not always follow the sequence enumerated by Norman. In fact, stages in the sequence may be skipped and as shown the sequence may start mid way through with user perception as the initial activity. Nonetheless, it is important to note that no matter what the stages or their sequence, interactivity is a complex sequential process consisting of the types of activities presented by Norman in the seven stages of activity.

4.1.2 Problems with Bridging the Gulf of Execution (A Designer's Perspective)

The bridging of the gulfs can be problematic for the user because it involves mental processes akin to those used in problem solving. Starting with the gulf of execution, the premise of user action is the perceived utility of the system. If the system can not perform the tasks the user has set out to complete then the system is not of any use to the user. Once the system is established to be useful, the user has to be able to figure out how to go about completing the task: forming mappings between physical controls displayed in the user interface and psychological intentions which reflect the user's perception of the task structure. The designer has the obligation to help the user bridge the gulf by designing an interface that matches the user's needs in a form that can be readily interpreted and manipulated (Norman, 1986). This, however, can be extremely challenging for the designer.

Designing a user interface is rendered problematic by the meta-level differences between the users, in their knowledge, skills and needs. More importantly, a single user's requirements for one stage of activity may differ from the requirements for another stage. To bring the user closer to the system the designers have to address these issues. Often this requires an iterative process where usability testing serves as a means to measure the success of the approach taken by the designer.

There are several approaches available to the designer to help the user bridge the Gulf of Execution. The two most common approaches taken by designers are 1.) to provide the user with extra information and 2.) to apply several interaction styles in the user interface.

Presenting the user with extra information to facilitate the crossing of the gulf can easily lead to a confusing and "restless" user interface. While helpful to the user to a certain degree, too much information may lead to information overload, further complicating user activity.

Neither is, supporting various stages of activity by providing multiple interaction styles a panacea to the issues that arise from user differences. A trade-off exists in the use different interaction styles to support different user requirements for action. The use of numerous interaction styles in a user interface can complicate interactivity because the user has to process more mappings between different inputs and outputs than with a single style. Usually designers commit to supporting certain stages of user activity by championing a particular style over others. In the case of the Omavisio interface, the menu interaction style is championed since it is considered to be ideal for general purpose systems, allowing the user to use the system without prior training. In summary, the bridging of the Gulf of Execution translates for the user into an issue of being knowledgeable about the system. For the designer of the interface it translates into the task of helping the user acquire the necessary knowledge.

While the bridging the Gulf of Execution is analogous to bringing the system closer to the user, by accounting for user needs in the design of the interface, bridging the gulf of evaluation means bringing the user closer to the system. Bridging the gap between the system and goals is outside the scope of this study because it involves teaching the user to use system through manuals and lessons. As mentioned earlier in the chapter on iTV, the system must be usable without any lessons or training. This is prerequisite for the market penetration of iTV. The rest of this chapter concentrates on the initial and often the hardest task the user faces in using a system: the bridging of the Gulf of Execution. Next, in elucidating the processes involved in the attainment of user understanding, emphasis is placed on how the system can be brought closer to the user through a good conceptual model.

4.2 Mental and Conceptual Models

Norman's theory of action demonstrates that interactivity is actually an episode of problem solving where the user engages in information processing. The importance of human-information processing is recognized in the field of HCI where it is considered an applied problem which needs to be understood. In HCI, knowledge-based modelling has been used as a means of clarifying the user's mental architecture for the purpose of developing design principles. *Mental models,* a contribution of cognitive science which is a field concerned with elucidating the mental architecture of humans, feature prominently in knowledge-based modelling and are at the core of user-centred design (Green, 1991).

Mental models play a critical role in the use and design of interactive systems and therefore, are a part of building usable systems. Norman (1986) considers mental models internal conceptualizations that aid people to understand and explain their own behaviour and the things that they interact with. Hollnagel (1991), views the mental model as a conglomerate of knowledge or a set of representations of something, part of which may be in the form of a model.

Mental models are formed as a result of interacting with the environment, other people and artifacts of technology (Norman, 1986). They serve to predict and explain the interaction itself. Needless to say mental models are considered to heavily influence user performance in the use of interactive systems (Moran, 1981, Norman, 1986). Hence, they can be determinants of the quality of interaction and subsequently, an understanding of them is fundamental when designing interfaces and conducting usability tests.

4.2.1 Types of Mental Models

According to Norman (1983) when considering mental models in the context of interactive systems there are actually four things to consider: the *target system*, the *conceptual model* of that target system, the user's *mental model* of the target system, and the *scientist's conceptualization* of that mental model.

Target System

This is the system that the user is interacting with.

Conceptual Model

This is the appropriate, accurate, consistent and complete representation of the target system. The conceptual model is also a tool devised for the purpose of understanding or teaching physical things. The conceptual model is used in the designing of interactive systems and therefore, it is also called the design model.

Mental Model

This is the user's model, the knowledge structure that is applied in the interaction with the system. The mental model evolves with system use. The user may use several mental models to understand a system better.

Scientist's Conceptualization of the Mental Model

This is a model of the user's model.

4.2.2 Mental Models and Quality of Interaction

Norman's "models" relate to each other in a way that is deterministic of the quality of interaction. The design model is the designer's conceptual model of the system. The conceptual model is the tool with which the system is designed; it is the "blueprint" for the system and represents the designer's understanding of the system. In fact the conceptual model can be considered to be the user interface, a point clarified later in the section. The *system image* is the means through which the conceptual model is conveyed to the user. This image consists of the physical image of the system, as well as documentation, instructions, and labels that deal or are a part of the system. The system image is the only means the designer has of teaching the user how to use the system, i.e., what the underlying conceptual model is. The user model is the user's mental model, an actively constructed representation of how the system works. The mental model changes during interactivity as a result of the observations the user makes about the system image, particularly the user interface. The observations are in fact instances of information processing where the user fixates some aspects to memory and recollects others from it. If the system image does not make the design model clear and consistent, the user acquires the wrong mental model(s) of the system and ultimately, will not learn to use the system effectively. In all, a user who has the wrong mental model of the system does not and will not understand the system.

The convergence of the models, mental and conceptual, is captured by the concept of *conceptual integration* (Moran, 1981) which stands for user understanding. A similar view on the importance of conceptual integration has been presented by Norman (1986) who states that the quality of interaction between the user and the system is dependent on the convergence of the user's perception of the task with the model of the task presented in the system. Lim *et. al.* (1996) have arrived at the same conclusion. They report that an understanding of the system is formed when the semantic and articulatory distances between the human thought and a system's operation are non-existent. In light of these two views, it can be theorized that user understanding of the system and the quality of interaction both depend on the degree to which the mental model matches the conceptual model - this point is re-iterated in section 4.3.

4.2.3 Mental Models and Designing the User Interface

The definition given in section 2.2 for the user interface - a boundary between user and system which allows interactivity - can be broadened to include the paradigm of mental models. As pointed out above, the user interface is the sum of the system aspects which affect and help build the user model. The user model is revised during interaction to reflect the conceptual model or possibly some other model that is similar to it. When the user and design models are "identical" the use of the system becomes practically automatic for the user, in other words, the user understands the system completely. Considering the above, it is possible to draw a parallel between the user interface and the conceptual model. In van der Veer (1991, p. 71) Rohr and Tauber do exactly this:

> *"The specification of the user interface can be regarded as a conceptual model, intended to be a complement to the intended mental model of the user."*

The implication of mental models on the design of the user interface is that

> *"to design the user interface is to design the user's model"* Moran (1981, p.4).

In theory, the user interface ought to be based on the user model since, as Moran and Norman have pointed out there should be a direct and simple relationship between the conceptual model and the user model. However, creating a model of the user model - the scientist's conceptualization of the user model - is extremely difficult due to the inherent nature of mental models (Norman, 1983). Therefore, the only feasible approach to conceptual integration is to create a conceptual model that the user is able assimilate through training or induce through use.

Norman (1989) writes that the first principle of designing for usability and understanding is to provide the user with a good conceptual model of the system. He claims that the conceptual model must fulfill three criteria. These criteria are applicable to the system and especially to the user interface:

1. Learnability
2. Functionality
3. Usability

These criteria are explained in Chapter 5 which covers usability.

4.3 User Knowledge and Understanding

Thus far in this section literature has been presented which shows that interactivity is a problem solving episode, driven by the user's need to understand the system.

Furthermore, user understanding has been equated with conceptual integration. Next, the degree of conceptual integration necessary for successful interactivity is explored. First, however, the different levels of the user's mental model are explicated using the CLG framework.

4.3.1 Psychological View of CLG

In section 3.5 it was shown that Moran's CLG is a framework for describing the composition and structure of a user interface. Since the specification of the user interface is regarded as a conceptual model (see section 3.2), CLG can also describe the components of the intended mental model of the user. The *psychological view* of CLG presented by Moran (1981) is a psychological model which does exactly this: it is a *knowledge analysis* of interactivity.

The psychological model of CLG divides knowledge about human-computer interaction into four levels (van der Veer, 1991). In the present study the psychological view is used as a framework through which user knowledge of the Omavisio system is examined. The framework is also helpful in localizing the sources of the problems experienced by test users. This can be accomplished by matching the observed problematic user activity with the level at which it occurs.

The knowledge structure of interaction is composed of the following types of knowledge or in other words the conceptual model is composed of the following component models:

Task Level

Task level knowledge is knowledge the user has about the tasks and sub-tasks that can be performed with the system. The way the user acquires this knowledge is through instruction and manuals. However, if the task is familiar to the user from a previous context then this knowledge is already part of the mental model the user applies in the initial use of the system. When the user lacks prior training in the use of the system, this knowledge is *interpreted* and *induced* from the visual representation and behaviour of the system.

Semantic Level

Semantic level knowledge is conceptual or meta knowledge the user has about goal-driven system commands, and the sequences of commands needed to carry out certain procedures. Semantic knowledge contributes to the user's understanding of the system's functionality. This knowledge is learned from the system, i.e. the user's mental model is revised as a result of interaction. Positive transfer of knowledge helps in learning systems and occurs when with systems that are perceived to have the same semantics as those that are known.

Syntatic Level

Syntactic level knowledge allows the user to formulate the correct syntax of the commands available for accomplishing conceptual operations. This knowledge is comprised of the syntactic rules: definitions of commands, their meanings and the context in which the commands are valid. Included in the syntactic model, or structure of syntactic knowledge, is the organization of the output devices and knowledge about the different possible states of system variables. While semantic level knowledge is transferable from one system to another, syntactic knowledge is platform specific. To capitalize on the user's positive transfer of semantic knowledge between systems that are perceived to be similar, designers often use metaphors to link the syntactic level to the semantic level. The conceptual integration of knowledge on a semantic level and a syntactic level increases when syntactic rules are consistently applied throughout the user interface (van der Veer, 1991).

Interaction Level

Interaction level or *key stroke level* knowledge is knowledge consisting of mappings between input device buttons or device manipulations and syntactic level commands.

The user acquires this knowledge by inducing it from system behaviour, the product of user action. Knowledge about system responses is part of this level.

4.3.2 Implication of the psychological view

The Psychological View bears witness to the paradigm of layered user knowledge: the intended mental model of the user consists of both *declarative* and *procedural* knowledge. Declarative knowledge is knowledge about the various kinds of conceptual and syntactic entities in the system. Procedural knowledge is knowledge about how to use the entities. This knowledge can be broken down into strategic knowledge about overall task structure and low-level knowledge about sequences of button pushes. The implication of user knowledge being layered is that the user can know something at one level, yet not at others.

The framework presented by the Psychological View is useful in the analysis of interaction as well as in the assertion of the user's mental model. Instances of system use can be reported and decoded in terms of this framework. For instance, the user may know roughly how to do a task from prior experience with other systems, but not how to do it with the system that has to be used. In such a situation the user has task level knowledge but does not have semantic level knowledge. In another situation the user may know the semantic method, i.e. has semantic knowledge, but not the syntactic method, which results in the user having an "idea" of how to do something but not knowing what commands to use. Likewise, the user may know the commands, but not the actual buttons which need to be pushed to evoke them - an instance of the user knowing the syntactic method but not the interaction method. Finally, the user may know the interaction method but not the task method and semantic method.

4.3.3 Knowledge Acquisition and Transfer

Internal and external knowledge combine during interactivity to determine user behaviour (Young, 1990). The user interface designer can help the user to use internal knowledge by implementing metaphors as well as acquire external knowledge by providing adequate feedback.

Internal knowledge exists prior to the use of the system. It is the state of the user model before the user has been exposed to the system. The user converts prior knowledge into knowledge which is applicable to the use of a system through the act of positive transfer. Transfer of knowledge is supported by the implementation of *metaphors* in the user interface. Metaphors are representations that allow the user to make a connection between system knowledge and general knowledge: to take knowledge of the familiar, concrete objects and experiences and use it to give structure to more abstract system and task concepts. Since metaphors give meaning to the interaction space the user does not have to deal directly with computer semantics. This can help the user bridge the semantic and articulatory distance between the system's operation and enables the user to find the correct expressions and actions for their intentions in the actual task world. In other words metaphors aid the user to acquire both procedural and declarative knowledge about the system.

External knowledge exists in the system and is usually displayed to the user, on the screen. The user acquires external knowledge by interpreting and inducing system behaviour. Interpreting and inducing knowledge during interaction is supported by *feedback*. Feedback is system information communicated through system behaviour which allows the evaluation of actions against goals (Payne, 1990). Moreover, feedback can also be used to communicate the state of the system independently of user actions.

Feedback is critical to conceptual integration. Through feedback the user's models of the component models, described in the CLG framework as Levels of knowledge, become more robust as a result of implicit assumptions becoming explicit (Riley, 1986). Feedback allows the user to make mappings between the different levels on knowledge. From the designer's perspective feedback is a means to bring the user model closer to the design model.

4.3.4 Interference

In situations where the user is not expected to have any prior training, like in the case of iTV, the only way the user is able to acquire a conceptual model at all is to induce it through interacting with the system. However, the acquisition of knowledge by induction can be rendered problematic by *interference* (Moran,, 1981). Similar to negative transfer, interference occurs when the conceptual model is very close, yet very different from one of the user's existing mental models. When prior knowledge interferes with the user's acquisition of the design model the result is that the two models become entangled and confused. An example of this is when the user confuses commands with similar meanings, a phenomenon that does not occur as frequently with commands that have different meanings (Moran, 1981).

4.3.5 User Knowledge and the Importance of Conceptual Model Acquisition on User Interface Design

Moran (1981) and Riley (1986) have both conducted *knowledge-based analyses* of interactivity where they have discussed the aspects of the conceptual model that the user's mental model should consist of. Both consider complete conceptual integration impossible to attain. They observe that regardless of whether integration occurs or not successful user engagement with the system can still occur. In other words, the user does not need to know in detail how to the system works in order to operate it. This is demonstrated by the Psychological View of CLG. Ultimately, however, Riley argues in favour of using a conceptual model in the user interface. She writes that user understanding is still a prerequisite for long-term skilled use of a system and therefore, it should be facilitated through design.

According to Moran (1981) the user does not need to have knowledge about all the methods in order to operate a system. The user model does not have to be complete at all Levels. Van der Veer (1991) takes this view and applies it to the design of interfaces. He claims that the system image which "teaches" the user how to operate the system should be designed according to the type of user expected to use the system. In the case where the user is expected to first and foremost to understand the functional properties of the system, task and semantic level knowledge should be taught to the user through the interface. On the other hand if the user is expected to become an expert user then emphasis should also be placed on teaching syntactic level knowledge. However, if the user is expected to become an operator of the system who is supervised by a task expert then only interaction and syntactic level knowledge needs to be taught. Thereby, each level of user knowledge supports a different learning strategy and ultimately, has different repercussions on the design of the user interface, in particular, on the interface style used.

Riley claims that ultimately user understanding prefaces all successful interaction. According to her users need to have more knowledge about a system than simply a list of procedures for accomplishing specific tasks. She believes users need to understand the system; they need to comprehend the Semantic Level of the user interface. In sum, understanding is important regardless of the amount of experience the user may have or the type of system they are using.

Understanding is needed to distinguish between functioning and non-functioning behaviour of the system. It is also needed in the transfer of skills between systems and it is a prerequisite in interpreting system feedback. Understanding allows the user to continue operating the system despite running into difficulties such as when the user or the system makes a mistake, that is non-routine and possibly unexpected. Furthermore, understanding is needed when procedures need to be regenerated in a situation where they have been forgotten due to disuse. The use of existing procedures in the construction of new methods is similar to the regeneration of procedures: it also requires user knowledge about the semantic level of the user interface or ”operational” understanding. Based on the types of activities understanding is needed for, it can be concluded that user understanding consists of having a mental model which captures task and semantic level knowledge about a system, i.e. declarative knowledge. In conclusion it should be noted that the user model does not need to be identical with the conceptual model. The mental model is adequate when it allows the user to perform the types of activities described above.

4.3.6 User Understanding and iTV

In her research on how much understanding is needed to become a skilled user of a system Riley (1986) explains that the amount of understanding needed is negatively correlated with the immediacy of the user tasks and positively correlated with the degree to which users will explore the system. Paradoxically, Riley implies through her postulation that the extent of conceptual integration required for user understanding may not be needed for systems that are simple, straight forward, routinely used and where the lack of system functionality minimizes the occurrence of user errors. Green (1991) supports Riley's view of simple systems and user understanding. Green maintains that teaching the user the conceptual model, be it through training or the interface, may defeat the purpose of an easy-to-use system, such as iTV. In other words, the knowledge required by the user about the system is dictated by the system's complexity and purpose. The present study attempts to shed light on whether or not understanding is important in the use of embedded systems like Omavisio which are aimed at the general public. Ultimately, the interface style and conceptual model are evaluated.

4.3.7 Overview of User Knowledge Required in the Navigation of Information Spaces

In the last section of chapter 4, user knowledge needed in the use of interactive systems like iTV is briefly enumerated. In the first section of this report it was reported that hypermedia systems present the user with an information space. Menu based interactive systems like Omavisio also present the user with an information space where the user creates information mosaics by selecting from an array of programming choices and services. Hence, navigation is conceptualized to be one of the most important user activities in information appliances since it allows the user to browse and search for information. Being lost in the space is one the most common user problems and bears witness to difficulty of designing navigable spaces. The following is an analysis of navigation conducted within the frameworks of layered user knowledge (CLG) and user understanding.

The psychology of navigation has been studied in Dillon, McKnight and Richardson (1990). In the context of hypermedia systems, the user model is a schemata of the environment the user is in. The acquisition of knowledge - procedural and declarative knowledge - translates into the acquisition of a *cognitive map*. The complexity of this map reflects the types and levels of knowledge the user has. For instance, procedural knowledge in the information space translates into *landmark* and *route* knowledge. When landmarks are part of the cognitive map, the user is able to recognize position in terms relative to the landmarks. Using this landmark knowledge the user is able to navigate from one point (landmark) to another, hence, the user has route knowledge. Finally when the user possesses a fully developed cognitive map, i.e. has acquired *survey* knowledge, the user is able to give directions or plan journeys along routes. This is the analog of having declarative knowledge.

From the above analysis of navigation parallels can be drawn with Riley's postulation on the importance of user understanding. The implication of different types of navigation knowledge is that a user equipped with elementary landmark knowledge alone will not be able to perform complex navigation (skilled operation of the system). Complex navigation requires that the user has an overview of the space, that is the user knows how it is organized and structured. This is manifested in the behaviour of users with limited knowledge who tend to return to the start of the path - simple navigation - when they get lost rather than try to retrace their way back to where they strayed - complex navigation.

For the user to be able to form an overview of the space, the space has to be well designed and the user must be able to explore it. Exploring is a way of organizing information that as McAdams points out is a means of understanding. A well designed information space is one that is consistent, predictable and transparent. The user must be able to come into the full space, or collection and find a form of organization that makes sense in order to able to move around comfortably in it. When the user can explore, moving easily from one topic to another, or from one source to another, he or she is able to build mental connections that reinforce learning and comprehension. Therefore, navigation reinforces complex navigation, and is conducive to user understanding.

As pointed out above, it is of paramount importance that the user interface designer supports navigation in the user interface. Supporting navigation means providing the user with a variety of navigational tools and options. These tools and options should accommodate the different goals and desires of users when they enter and while they are within an information space (McAdams, 1995).

5. Usability

The importance of usability has been widely recognized in the development of products and systems. However, there has been confusion over the meaning of the term. Furthermore, the operationalizing usability has caused considerable difficulty for designers. Consequently, numerous design methods have been proposed in the field of HCI. This section of the report aims to clarify the meaning of usability and to address its operationalization. Thereby, the motivation for this usability study is explicated. Furthermore, an understanding of usability is important for the analysis of the usability test results. First, the question what is usability will be answered. Next, an overview of general usability attributes is presented and the need for context specific ones is introduced. The following section on usability engineering extends the paradigm of system specific attributes and their importance in the design of interfaces. Usability testing is the means through which these attributes are discovered.

5.1 What is Usability?

In recent years people have started to pay attention to the issue of *usability* and it has become a sales angle for system developers. Yet, the definition of usability has remained rather elusive. Nielsen (1993) reports that the most common semantic definitions for usability include “easy to use”, “intuitive”, “easy to learn” and ”transparent”. The most popular term used to describe usability to the users has been “user-friendly”. However, when spoken of in such broad and misleading terms, it is impossible to quantify and operationalize usability in the design of user interfaces. To begin with the term ”user-friendly” is misleading because systems do not have to be “friendly” to users in order for them to be usable. The purpose of this section is to address the concept of usability. This is accomplished by presenting three definitions of usability. These definitions originate from three different views of how usability should be measured: *user-oriented*, *product-oriented* and *user performance-oriented.*

5.1.1 Definitions of Usability

The reason for the emergence of usability as an issue in systems development is its close relationship and correlation with *system acceptability.* System acceptability is a broad issue where the basic question is whether the system satisfies the user's needs and requirements, as well as those of others' who are associated or dependent on the user using it. The *user performance view* is based on this paradigm.

According to the user performance view usability is a part of *system acceptability.* FIG. 5-1 shows that system acceptability is a composite of *social-* and *practical acceptability.* Social acceptability is determined by how favourably the system is perceived by those who come in contact or are affected by it in some way: the public. In turn, this is influenced by the connotations of the system and the impact it has on people's lives. Once the social acceptability of the system is deemed to be sufficient, an analysis of the system's practical acceptability provides the answer to what extend the system will penetrate the potential user population, i.e., the level of system acceptability. The issues of cost, compatibility with existing systems, reliability and support for the system all fall under practicality. The usefulness of the system which is a part of the practical acceptability is determined by whether or not some desired goal can be achieved using it. Usefulness is separated into two categories: *utility* and *usability.* Utility is the principal of whether the system has the functionality to do what is was designed for. Usability is a description of how well the users can use the system's functionality; in other words it describes user performance.

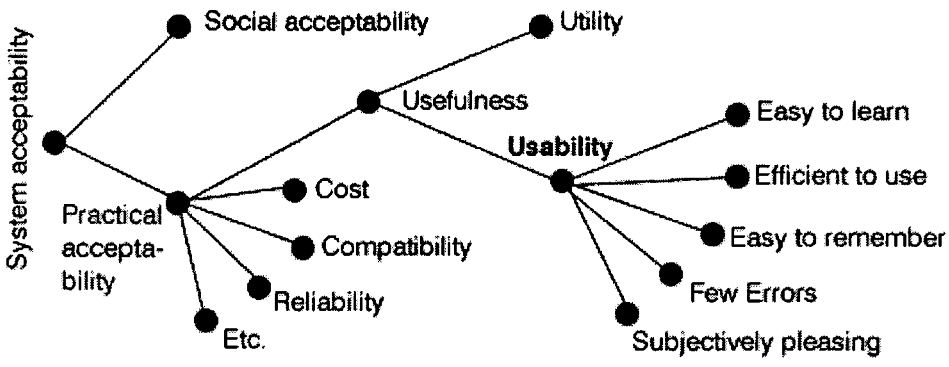


FIG. 5-1 *A model of the attributes of system acceptability (Presented in Nielsen, 1993)*

The view of usability presented above has been criticized because within that framework a system can be considered usable but not useful.

The user- and product-oriented definitions of usability are broader than the performance-oriented view which nests usability within usefulness, within practical acceptability, within system acceptability. Central to these definitions is the paradigm that usability stands for the "quality" of the user interface and of system use.(Bevan, 1995).

Product oriented views of usability are based on the concept of interface quality. For instance, Bevan (1991) defines usability as the user's view of software quality. To Bevan interface quality means the extent to which the interactive system meets the needs of the user. User interface "quality" can be quantified in terms of attributes which dictate whether the interface is congenial or hostile, easy or difficult to learn, easy or difficult to use, responsive or sluggish, forgiving or intolerant of human error (Baecker and Buxton, 1987). Similarly, Vainio-Larson (1990) refers to usability as the extent to which a system matches the end-user's characteristics and skills for the task concerned.

The standard for software qualities (ISO 1991b) states quality to be:

> *"a set of attributes of software which bear on the effort needed to use and on the individual assessment of such use..."*

User-oriented views of usability center on the concept of interaction. Butler (1996), for instance, considers usability as the effectiveness of interaction between human operators and their machines.

A parallel can be drawn between user-oriented views of usability and user understanding. In previous sections it was reported that interaction consists of creating mappings between intentions and physical controls. In particular, according to Norman's theory of user action, effective interaction was equated with the successful bridging of the Gulfs of Execution and Evaluation (1986). In Moran's CLG framework effective interaction was represented by conceptual integration (1981). Section 2.3 on interaction, concluded with Riley's view of conceptual integration where the concept stood for user understanding of the system. Thereby, the paradigm of conceptual integration is most prominently incorporated in the user-oriented view of usability. It is through this view that conceptual integration is brought into the arena of user interface design.

In conclusion, the above definitions, including the user-performance view, imply that usability is not a single one dimensional property of the user interface. A product is not itself usable or unusable, but has *attributes* which will determine the usability for a particular user, task and environment.

5.1.2 Usability Attributes

The definition for usability attributes used by ISO TC159/SC4/WG5 subgroup 2 is:

> *"Usability attributes: the features and characteristics of a product which influence the effectiveness, efficiency and satisfaction with which particular users can achieve specified goals in a particular environment."*

According to Bevan *et. al.* (1991) usability attributes include ergonomic as well as other characteristics of the product which influence quality of use. Included in this set of attributes are those aspects of software quality which affect ease of use.

Nielsen (1990) has enumerated the five attributes most commonly associated with usability:

1. **Learnability:** the ease with which a user can learn to use the system.

2. **Efficiency:** once the user has learnt to control and utilize the system, the system has to promote efficiency in performing tasks.

3. **Memorability:** the system has to be easy to remember so that the users who have already learned to use it once will not have to relearn it if they use it infrequently.

4. **Errors:** the system must be able to anticipate human errors and respond to them in a constructive manner that is useful to the user. The system should be designed in such a way that users cannot make irrevocable errors or errors which affect the system critically.

5. **Satisfaction:** the system should be pleasant to use especially if it has been designed for non-work related interaction.

These attributes can be regarded as components of usability. They give the abstract concept a shape. Conceptualizing usability to have components allows designers to take an engineering approach to improving user interfaces. One such approach is *usability engineering*. It is a design methodology which aims to systematically evaluate and improve these components through a cyclical process.

5.1.3 Apparent Usability

While the above views are concerned with enumerating the concept of usability, Kurosu and Kashimura (1995) have distinguished between two different kinds of usability: *apparent usability* and *inherent usability.* In doing so they argue that the visual representation of the user interface has a usability of its own. According Kurosu *et.al.* the first thing a potential user sees is the visual edifice of the system: the user interface. Upon seeing the user interface for the first time the user forms an opinion about system's usability. The user is not evaluating the inherent usability of the system since no interaction has occurred, instead the user forms an opinion of the system's *apparent usability.*

Apparent usability is the general impression about system usability presented in the layout and aesthetics of the interface and it is a very important part of the overall usability of the system. Kurosu and Kashimura report that users are often initially drawn to a system because of apparent usability and not because of inherent attributes. They argue that usability, i.e. inherent usability, is meaningless to the user if he or she does not perceive the system to be sufficiently appealing to begin using it in the first place. This is particularly true with iTV which is primarily meant to be used for entertainment. It is generally believed that if the iTV interface is not aesthetically pleasing and does not render the feeling of being easy to use, the system will be rejected by the public. Therefore, apparent usability is a factor in the over all usability of systems like iTV. For this reason, the present study also concentrates on evaluating the apparent usability of Omavisio as well as "inherent" usability.

5.2 Usability in Context (Domain Specific Usability)

Brooke, Bevan Brigham, Harker and Youmans (1990) point out that usability cannot be accurately defined in terms of general attributes, mainly because application domains, the unique environment and interaction created by the application, have specific requirements and constraints for the user-system interaction. According to Brouwer-Janse (1992) systems have unique usability attributes and users of these systems have system specific criteria for judging system usability. Subsequently, to define usability one must consider the *context* of system use.:

1. **Users** who will use the system.
2. **Tasks** for which the product will be employed.
3. **Conditions** under which those users will perform those tasks.

Thereby, criteria for usable systems come from specific user requirements, tasks and the environment of use. When context specific criteria is known it can be applied as *domain knowledge* in the design process. By incorporating domain knowledge at the earliest possible stage of the design process as well as, in the methods of analysis and evaluation and moreover, by manufacturing formal representations of this knowledge, user interface design can be enhanced. In practice this entails introducing domain specific style guides where domain knowledge is factored into the elements of the user interface enumerated in the guidelines. For a new application the criteria has to be discovered despite the fact that numerous usability guidelines and design heuristics already exist. The discovery and use of context specific criteria in the design of user interfaces is emphasized in usability engineering, a methodology which serves the objectives of this usability study. The objective of this study is to help designer's gain a better understanding of the usability aspects of Omavisio and to establish context specific usability principles that will guide development in the future.

5.2.1 Usability Engineering

Usability has two complementary roles in design: it is an attribute which must be designed into a product and it is the highest level quality objective which should be the overall objective of design (Bevan, 1995). While this view is generally accepted, there is disagreement on how to operationalize usability in systems and integrate it into the software design process (Bevan, 1991). Numerous approaches exist in the field of HCI which try to achieve the aforementioned objectives. Out of these various methods iterative design methods that account for context of use have been found to improve user interface usability the most (Bailey, 1993). *Usability engineering* is an iterative design methodology which emphasizes the discovery and use of context specific usability criteria. (Wixon, 1995).

Usability engineering is defined by Butler (1996) as a technical discipline for developing computer user interfaces that can be readily comprehended, quickly learned and reliably operated. The discipline provides a methodology and heuristics which when included in the design process help the designer to focus on the usability of the interface. The designer is able to operationalize usability principles and integrate them into the design process. The methodology assures that the designer pays attention to the essential aspects behind a quality interface. This is achieved through a *user-centred* iterative design cycle. The cycle revolves around designing, building, evaluating and analyzing the user interface and it is guided by heuristics. This cycle is sometimes called *spiral design* or *iterative prototyping* (Butler, 1996), see FIG. 5-2.

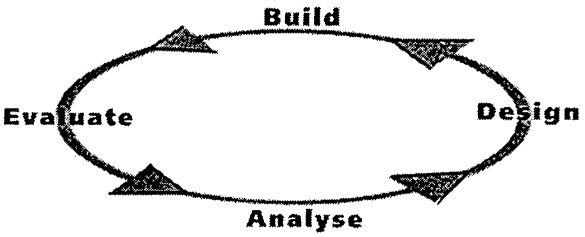


FIG 5-2 *The Usability Engineering Paradigm (Presented in Butler, 1996)*

5.2.2 Stages of the Usability Engineering Lifecycle Model

The engineering cycle is composed of stages. These stages are presented in Nielsen (1993):

1. **Know the user**
    a. Individual user characteristics
    b. The user's current and desired tasks
    c. Functional analysis
    d. The evolution of the user and the job

2. **Competitive analysis**
3. **Setting usability goals**
    a. Financial impact analysis
4. **Parallel design**
5. **Participatory design**
6. **Co-ordinated design of the total interface**
7. **Apply guidelines and heuristic analysis**
8. **Prototyping**
9. **Empirical testing**
10. **Iterative design**
    a. Capture design rationale
11. **Collect feedback from field use**

Nielsen observes that usability engineering can be successful even if it does not include every possible refinement at all the stages or for that matter every single stage of the lifecycle model. Actually success depends on the degree to which those stages that account for context are included. Next, the stages of usability engineering that account for context, are explicated. The importance and relevancy of these stages depends on system requirements which in turn depend on each part of the overall system including hardware, software and users (Bevan, 1995).

Know the user

Like all user-centred methods the design process begins and ends with the user. Knowing the user entails that the designer conducts *user* and *task analyses*. Studying the *end users* and their working habits allows the designer to develop a feel for how the system will be used. Moreover, establishing the needs of the users through analysis allows the designer to list the requirements for the behaviour of the system (Bevan, 1995). In fact, for designers who practice usability engineering, an understanding of the user and the task is more valuable than knowledge of general models of human-computer interaction.

Competitive analysis

Competitive analysis is another means of becoming familiar with the context of the system being designed. It involves studying existing competitive systems. From such analysis the designer is able to learn more about the requirements of the system under development. Ultimately, these requirements are refined and can be adopted as interface guidelines or heuristics. Additionally, competitive analysis is a means for the designer to determine those features and aspects the system must incorporate as well as those that can be improved upon with regards to the competition.

Setting usability goals

The requirements for the behaviour of the system, derived from user and competitive analysis, are expanded and become *usability goals.* These goals serve as measures by which the success of the design can be judged. Satisfaction of criterial values or metrics tell the designer when the iteration can end. In all, it is objective measurement that enables usability engineering and therefore, usability goals are essential when designing an interface (Butler, 1996).

The broad definition of usability measures used by ISO subgroup 2 is:

> *"Usability measures: the effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in a particular environment"*

Once usability goals have been set, to achieve them the designer has to specify the internal attributes of the system as measurable internal requirements. Usually specification begins with the requirements being expressed in terms of general principles, specific interface details, or through the use of style guides. Butler (1996) identifies *learnability, throughput* and *user satisfaction* to be common measures of usability and the suitability of design. Nielsen's (1993) usability attributes, presented in section 5.1.2 are the most common measures. Subsequently, these attributes can be evaluated to produce internal metrics verifying how closely the internal requirements have been met in the design. In practice this entails establishing criterion values for instances of system use, i.e., accomplishing a particular task.

As with the definition of usability, the specification of usability measures is very subjective. In fact, Bevan (1995) reports that no objective rationale for their selection has been proposed. The most important criteria for choosing metrics is that they reflect context specific usability.

5.2.3 Usability Testing

Measurement of usability reveals the success or lack thereof, of the user interface design. Usability can be measured through *usability testing* and *usability inspection.*

In general, *usability testing* refers to experiments where test-users are given a task to perform with a system. Observations are made on the way the users accomplish this task. The observations serve as data from which the usability of the application can be determined by comparing the results with the usability goals set before hand. Therefore, the usability issues of the interface have to be known or predicted prior to a test situation. The results of the actual test consequently confirm or dismiss the expected usability issues of the interface. Additionally, tests may also be used to discover usability issues.

Usability inspection or *heuristic evaluation* is an evaluation methodology where usability experts evaluate the system with a given set of heuristics that are deemed to be representative of what usability is in that instance. In the case of usability inspection methods usability goals serve as usability inspection criteria. Regardless of the evaluation method used, a major result of any form of usability evaluation is always a list of usability problems in the interface.

Usability testing has turned out to be the more effective of the two in finding severe problems in the user interface. While usability guidelines and checklists are useful aids for design and can be applied to make quick expert assessments of the user interface design, they are not a reliable means of determining whether a product is usable. The reason for their shortcoming is the high variance of users and tasks (Bevan, 1995). Similarly, Nielsen (1996) reports that users of usability inspection methods are likely to overlook usability problems in instances where the system is highly domain-dependent and the users themselves have little domain knowledge. Consequently, to accurately assess usability the actual interaction between the user and the system must be studied (Bevan, 1991). This can be done only with usability testing methods.

There are numerous usability testing methods that can inform about design decisions. Common to all testing methods is the collection of user feedback from which it is possible to recognize usability problems and cumulate other knowledge useful for improving the design. For example, user feedback is a rich source for hints about which types of system features support successful user strategies. Nielsen (1993) points out that in analyzing user feedback the designer has to be careful not to confuse usability problems with quirks that result from user differences. For an in-depth explication of user differences see Nielsen (1993). Along with user feedback, usability testing can be used for the evaluation of the underlying metaphor in the interface. In addition, usability tests can generate data about different interface elements and aspects such as the nuances of word choices in menus or the general readability of the display.

While more accurate in measuring usability, usability testing is also more complicated to conduct than usability inspection. To be effective testing has to be conducted in informal natural settings such as the living room. It may involve the simulation of multiple tasks that users are expected to accomplish while operating the system (Brouwer-Janse, 1992). Moreover, usually it is necessary to evaluate a product separately for different user groups, especially if the system accommodates different user profiles as in the case of Omavisio. Given these conditions for effective testing it is essential to identify the intended context of use before carrying out any tests.

Discovery of Attributes

When designing usable systems, usability has to be enumerated in terms of usability metrics that stand for the successful operationalization of usability within the appropriate context. These metrics are reasonably easy to set for new versions of existing systems or systems with clearly defined competition on the market (Nielsen, 1993). For such systems the attributes or metrics can come from the initial task and user analyses or as Nielsen suggests, they can be derived from competitive products. Other sources for such metrics are the numerous user interface style guides and guidelines. However, the introduction of product concepts that provide entirely new functionality into the mass market creates extra burdens for the design of user interfaces (Brouwer-Janse, 1992): ultimately, the usability issues of these new systems have to be discovered before, development can proceed.

Guidelines are not a reliable source for usability metrics: they have well known problems associated with them deriving from their breadth at the expense of context-specific depth (Bevan, 1995). In light of the high degree of task dependency in the design of usable interfaces the lack of context-specific depth renders general guidelines virtually unusable. Furthermore, user and task analyses cannot tell everything about the usability of a new system. This is especially true of a new system of the likes of which the users have never heard or thought about. Likewise, it is impossible to conduct competitive analyses when competitive products do not yet exist on the market.

iTV is a product which is thought to generate a whole new mass market. It is a new media that may completely change our lives. At this early stage of its lifecycle iTV fits Wixon's (1995) classification of a domain where discovery of usability attributes is a prerequisite to more decision- and measurement-oriented parts of the process. This usability study is founded on the necessity to establish the appropriate usability metrics for Omavisio so that the interface can be subsequently developed according to the stages set by usability engineering.

Prototyping and Discovery of Usability Issues

Wixon (1995) considers usability engineering to be a discovery oriented design process where, experimentation not only serves to evaluate usability goals, but is also a valuable data gathering method. What renders usability engineering appropriate for discovery is it advocacy of such discovery-oriented methods as prototyping in the early parts of the development process.

Nielsen (1993) recommends the use of prototypes in early usability evaluation. The use of *prototypes* saves time and cost when developing systems to be tested with real time users. A prototype differs from the full system in the proportion of implementation. A prototype is generated by reducing the level of functionality or the number of features of the final system. Prototypes are either *horizontal* or *vertical* depending which dimension has been reduced.

A horizontal prototype includes the entire user interface to a fully-featured system but with no underlying functionality since the level of functionality has been reduced. It is a simulation of the interface; users cannot perform any real tasks on the system. The horizontal prototype is suitable for assessing how the interface works as a whole; for instance, interface navigation can be studied.

A vertical prototype is a dissection of the real system. The number of interface features has been reduced but those features that are included have in-depth functionality allowing the user to perform tasks on the system. Consequently, vertical prototypes are tested under realistic circumstances and with real users performing tasks.

When both the different features and the level of functionality are reduced the prototype becomes a *scenario.* A scenario has only one predetermined path which simulates the functionality of the system. The test-user is instructed to follow this previously determined path during testing.

A scenario is an encapsulated description of

- an individual *user*
- using a specific set of computer *facilities*
- to achieve a specific *outcome*
- under specified *circumstances*
- over a certain *time interval.*

Scenarios are useful early on in the design process to present and understand the interactivity of the final system. They are also a means of testing the user interface and getting feedback in the initial stages of the process. The scenario is testable if it offers some interactivity and flexibility in the narrative or predetermined path.

The Omavisio system under investigation is a mix between a horizontal and vertical prototype. As far as features the prototype is representative of the final system. Some of these features have been developed to the point where the test-user is able to use the system as in the way it is meant to be used. A more in-depth description is given in chapter 6.

6. The Omavisio User Interface

The present study concentrates on the Omavisio system. Omavisio is a brand name which translates into English as “My own (tele)vision.“ The complete Omavisio platform consists of the Omavisio system, a media server and a network through which content is delivered. The components of the system are a television set, a STB and two input devices which are an infrared remote control and keyboard. This section presents an overview of the interface of the Omavisio system. The description is structured according to the CLG framework and therefore, proceeds from an overview of the system to a detailed explication of the interface elements. The design decisions made by the Omavisio designer’s are not discussed as they are outside the scope of this report.

6.1 The Omavisio Prototype

The system used in this study is the first prototype of the Omavisio system. The primary motivation for constructing the prototype was to demonstrate to potential Omavisio content providers and advertisers the concept of profiled information services delivered to users via a television set. The prototype features all of the main types of Omavisio applications; however, some of them are not fully functional. For instance, in the shopping mode it is possible to browse through a “catalog“ of goods but ordering and payment transactions have not been implemented. Thereby, only certain “paths“ in the home shopping hierarchy are navigable. The Omavisio prototype does not fit under any of the rubrics found in Nielsen’s (1993) taxonomy of prototypes. It is closest to a *vertical prototype* however, overall functionality has been extended to almost all modes instead of limiting it to only one.

The prototype supports two different types of *run-time engines.* Each engine runs one general type of application on top of the STB Acorn PC operating system. Navigator is the main Omavisio application. The Navigator run-time engine displays menu pages which serve as a front-end to different services, videos, and games as well as to the second type of application, the WWW browser. The menu pages are created with platform proprietary authoring tools. The authorware includes software for creating pages, a runtime player for testing and a utility which compresses and stores images in *JPEG* (Joint Photographers Expert Group) format. The second type of application, WWW browser, displays Hyper Text Mark-up Language (HTML) pages. Since many tools exist for the authoring HTML pages the Omavisio authoring environment does not support HTML page design. The only time the user is able to see the underlying STB operating system is when the WWW browser cannot perform the user specified action. Whence, the user is informed of the failed operation with a STB operating system dialogue window which has to be confirmed before browser operation can resume. The following description of the Omavisio user interface concentrates on the profile menu pages: the default front-end interface for Omavisio content.

6.2 Omavisio Task Level (Conceptual Component)

In the CLG framework the Task Level is the most abstract and holistic view one can take of the system. The primary purpose of the Task Level view is to elucidate all of the different tasks that can be accomplished with the system. The tasks enumerated next are those that can be accomplished with the first Omavisio prototype.

Omavisio introduces a novel way to perform the familiar tasks of

- browsing television programming choices,
- viewing programs and movies,
- viewing the news,
- shopping for goods,
- researching goods, prices and information of interest,
- playing video or computer games,
- and reserving tickets,

Furthermore, it is possible to accomplish new-media centric tasks such as

- browsing the World Wide Web,

- interacting with Web forms,
- using Internet applications such as e-mail,
- and interacting with CD-ROM's.

6.3 Omavisio Semantic Level (Conceptual Component)

The Semantic Level in the CLG framework describes how user tasks are conceptually represented in the user interface. Moreover, the Semantic Level of the user interface encompasses the semantic procedures for accomplishing the conceptual tasks.

6.3.1 Conceptual Actions

In Omavisio, tasks are performed "virtually." The user *navigates* in and *manipulates* the information space filled with numerous applications, videos and services from which the user can choose. The conceptual actions involved in the use of the system relate to information processing: they are *browsing, searching, retrieving, entering, submitting, selecting, activating, viewing,* and *controlling.*

The semantic operations enumerated above are exemplified in the following description of the operations involved in shopping with the Omavisio system:

1. The user *selects* and *activates* Kauppakeskus, "shopping center" button, the shopping center item on a menu page. In doing so the user makes a choice as to what will be displayed next.

2. Next, the user selects again this time from the shopping menu page which is displayed as result of step 1. The user continues to select from subsequent pages, in other words the user *navigates* back and forth between menus and product pages, all the while *browsing* the selection of products.

3. When on a product page, if the user chooses to purchase the product, the user has to navigate to the order form page.

4. The user enters the amount, specifies attributes and other pertinent product information in a form. Also all information concerning payment of the order is entered.

5. Finally the user *submits* the form and automatically returns to a shopping menu.

6.3.2 Organization of the Information Space

Omavisio content (applications) is organized according to three user profiles into an information hierarchy or "tree." In the hierarchy a node is either a menu page or an application such as a game, a video or some other interactive application. In short, users access or "retrieve" content through menus. User navigation between nodes is supported by a Omavisio directory, the INFO/HAKU page. The directory page presents the user with an overview of the three main paths in the information hierarchy. The user can use the directory to navigate to content which cannot be navigated to from within branch of the hierarchy that the user is in. From the directory the user is able to navigate between menus within a profile as well as between the three profiles. For example, the user is able to navigate from the OMA1 main menu to one of the OMA3 lower-level menus.

6.3.3 Metaphors

The Omavisio user interface borrows from two metaphors. The metaphors which are partially used in the interface are *broadcast TV* and *hypertext system.* The broadcast TV metaphor comes with a rich vocabulary which includes channels, menus, receivers and VCRs all of which are part of the everyday experience of most people. The concept of TV channels is employed in Omavisio. Channels are used to abstract and simplify the concept of profiled services and content. The metaphor draws a parallel between a profile which serves as the basis for grouping content and a TV channel that specializes in broadcasting to a particular audience. Hence, each of the three user profiles in Omavisio has its own "channel:" OMA1, OMA2 and OMA3. The channel concept is represented to the user through the use of a "channel identifier" in the top right-hand corner of each page. The primary functionality of the identifier is to inform the users where they are in the overall Omavisio information space. The "channel identifier" is not visible when the user uses the Internet.

The second metaphor used in the user interface is a less familiar to most users. The hypertext metaphor comes with a vocabulary consisting of terms like link, home, history, back, forward, bookmark and jump. The concepts of back, forward and home are used in the interface to facilitate user navigation. The main menu or starting point from which all users begin their navigation within each channel is called "KOTI" or "home." It is the reference point that users can easily return.

6.4 Omavisio Syntactic Level (Communication Component)

The Syntactic Level in the framework captures all aspects which relate to the system "command language." This entails explicating the methods used to evoke conceptual actions. In other words the Syntactic Level of Omavisio deals with *how* the user navigates and manipulates content. In addition, the different instances and contents of the interface dialogs are described.

6.4.1 Interacting with Omavisio

Omavisio is a user paced passive system. The user tells the system what to do, then the system does it and tells the user what it has done. The user communicates or tells the system what to do by making selections from the displayed menus. This is done by selecting areas on screen and activating them. Some areas can be selected and activated simultaneously by pressing a button on the remote control.

The selectable areas on screen are called *hotspots.* Hotspots are areas on a screen where system commands have been embedded. When a hotspot is selected and activated the system command is executed. With these commands the user is able to move to a different menu, start a video, or access the Internet. In sum the user does not have to learn a command language in order to successfully operate the system. In fact the only time the user needs to type a command to the system is when specifying the URL (Uniform Resource Locator) of a WWW page.

6.4.2 User Interface Dialogue

The most common type of dialogue that the user engages in with the user interface consists of making selections from static menu pages. In some instances such as the log-in, ordering of goods and entering of URL, the user enters information in forms and them submits. The only time the system prompts the user for confirmation of system message is in Internet mode. The STB operating system shows the user a dialogue when the user tries to backtrack from the system homepage.

6.4.3 Omavisio Modes

Omavisio is a one window system. In a one window system the screen is the window: it does not have borders or handles which would allow the user to resize or move it. In such a system only one menu or application can be used and displayed at a time. In a *modal* system like Omavisio the user is forced to use different commands to control the different application types. TABLE 6-1 lists the modes found in the prototype.

TABLE 6-1. *Omavisio modes.*

Mode:	Description:
Navigator or the Omavisio channel	The front-end to all other modes. The remote control button sets that work are the numeric set, the colored button set, the arrow buttons and the system/navigation set. However, not all buttons in each set work. This means that if a button has not been mapped to a command nothing happens when the user presses it. OMA channel menu pages are also modal. There are two ways to activate hotspots on the menu pages. The first is "direct selection" where remote control buttons are mapped directly to hotspots. The second is "direct manipulation" where the area selector is tabbed from hotspot to hotspot and the hotspot is activated by pressing SELECT on the remote control. Both interaction styles can be used on OMA1 and OMA2 pages while only the second style works on OMA3 pages.
Game	The remote control buttons act as game controls. The control mappings vary from game to game.
Omavisio News	Similar to the Omavisio menu mode with the exception that the red and blue color buttons have different functionality. STOP on the remote control stops the current segment of a newscast however, the following newscast begins playing on its own. The user cannot stop a newscast, instead all the segments have to played or at least scanned before it is possible to navigate back.

WWW	Only the arrow buttons and the KOTI, INFO, SELECT and STATUS buttons work. The arrow buttons move the pointer from one link to another and allow scrolling of the WWW page. Links can only be activated with SELECT.
E-mail	Is a sub mode of the WWW mode. Requires the keyboard for writing messages.
Video (VOD)	The video takes up the full screen. No on screen controls are visible. Only the video control set works. No screen visualization of controls.
Shopping	Some of the video control buttons work as navigation controls, moving the user from one menu to another. Otherwise identical to Omavisio menu mode.
Help	A screen with instructions, no controls are visible on screen.

6.4.4 Generic System Commands

Generic system commands are modeless system commands. They carry out the same transaction regardless of the interface mode. There are no generic commands in the Omavisio user interface with the exception of the ON/OFF button.

6.5 Omavisio Keystroke Level (Communication component)

The Keystroke Level specifies the remote control button and keyboard key mappings. Hence, at this Level the functionality of each button on the system input devices is described. In addition, the system responses and feedback to user inputs is also explicated.

6.5.1 Remote Control

The Omavisio infra-red remote control, shown in Fig 6-1, has 5 different groupings or sets of buttons. Starting from the top of the remote control the first group of controls is the numeric button group. The buttons are numbered from 0-9 and they are organized in four rows with three buttons to a row with exception of the 0 button. Also at the top but the to the right of the numeric buttons is the second group of buttons. This group contains the STATUS, HOME, BACK, HELP and SELECT buttons. Some of the buttons are used for navigating the information space. Below and to the left of these two groups is a group of arrow shaped buttons. These buttons, UP, DOWN, RIGHT, LEFT, and four others, one half way between each direction, are organized in a compass like 360 configuration with the arrows pointing away from the center. At the center of the group is a second SELECT button. The fourth group, below the arrow buttons, is formed by the video control buttons. This group consists of standard controls for video content: STOP, PLAY, REWIND, FORWARD, as well as quick find modes of the last two. The last group of buttons on the remote control is composed of a RED, GREEN, YELLOW and BLUE button. Together they form the colored button set. In all there are 32 buttons on the remote control.

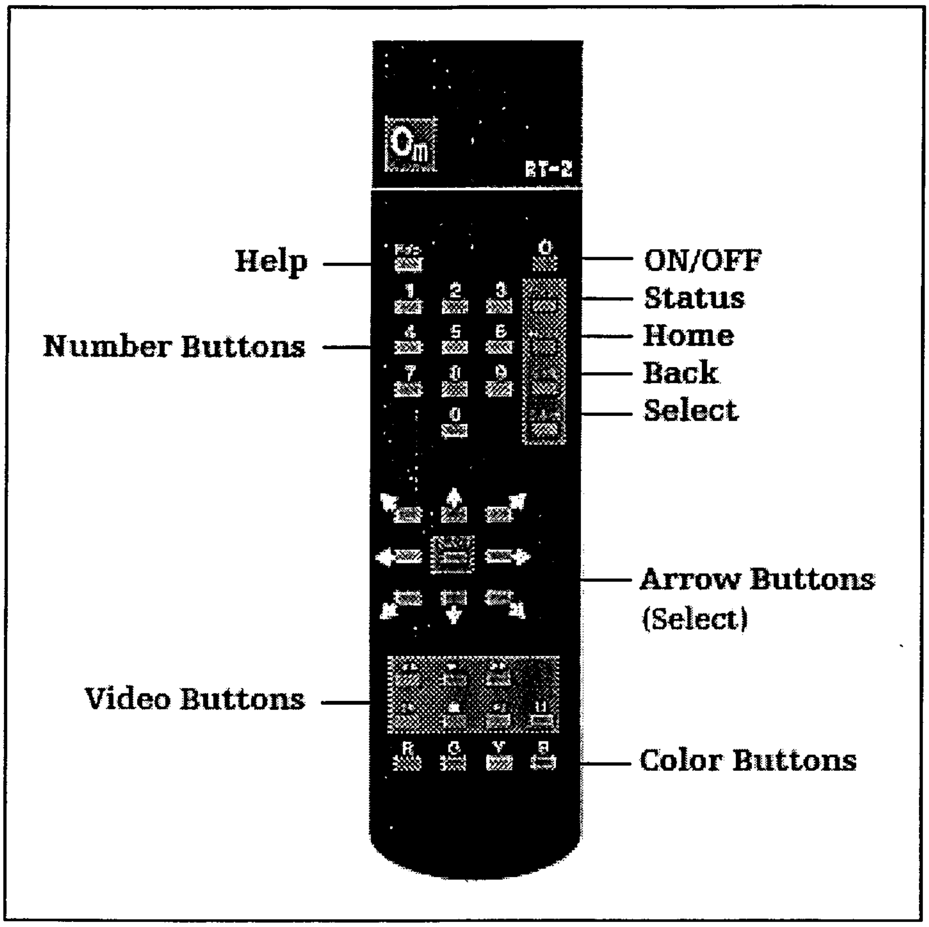


FIG. 6-1. *The Online Media IR Remote Control*

Each of the five groups of controls is used primarily for one of the five basic types commands that the Omavisio interface accepts. The numeric buttons are used for entering numeric information or for selecting numbered items on menu pages. The video control group controls video content. The group of navigation buttons allows the user to navigate to menus as well as to certain fixed locations in the information space like KOTI or home. The arrow buttons are for controlling the pointing device or area selector. The colored button group is the least conceptually cohesive in terms of functionality. In fact it contains buttons with two different types of functionality. The RED and GREEN buttons are used for basic navigation within the menu hierarchy. The YELLOW and BLUE buttons change system mode from the Omavisio menu mode to WWW and shopping mode.

Some of the buttons are modal. For instance, the FF and RW buttons double as navigational controls in the shopping mode. Moreover, when using imagemaps on WWW pages the arrow buttons function in two different ways. In the WWW mode the arrow buttons move the area selector from one hypertext link to another. When an imagemap on a WWW page is selected the arrow button functionality changes: the selector turns into a cross hair and the buttons double as directional continuos-movement controls.

The Default Mappings of Remote Control Buttons

The default mappings of the remote control buttons is shown in TABLE 6-2. The numeric buttons do not have default mappings. Hotspots are mapped to different numbers which allows users to make selections without moving the selector.

TABLE 6-2. *Remote Control Mappings*

Buttons:	Function:
HELP	**Navigator command:** Branch to the application's default help page. Displays the general or overall help page as well as a context specific help page. In the WWW browser mode it displays the profile help screen and not an HTML page.
HOME	**Navigator command:** Go to the page at the top of the hierarchy (stack) Displays the user's main menu page or the profile WWW home page if in WWW browser mode. In all it is the applications

BACK	**Navigator command:** Go back to previously-displayed page. Displays the previous menu screen or WWW page.
STATUS	**Navigator command in WWW mode:** Display a graphical scroll bar. From the scroll bar the user is able to interpret how much of the WWW page is visible on-screen and how much is left to scroll.
SELECT	**Navigator command:** Run the embedded command. Functions as the "action" button; used in activating hotspots.
ARROW BUTTONS	**Navigator command:** Goto next command. Move the area selector, up, down, left, right, down and left, down and right, up and left, up and right. The selector moves to the next selectable area of the screen or hotspot. Also functions as up and down scroll buttons on WWW pages that do not fit on the screen. When scrolling a WWW page the first hotspot in the new screen is automatically selected.

RED On-screen: (OMA1 = KOTI) (OMA2 = PALUU) (OMA3 = KOTI)	**Navigator command:** Run the embedded command which in this case is to go to page at the top of the stack. Same as HOME. Displays the profile main menu screen or the profile WWW home page if in WWW browser mode.
GREEN On-screen: (OMA1 = INFO) (OMA2 = HAKU) (OMA3 = INFO)	**Navigator command:** Goto to the specified page (directory page). Displays the INFO page, a graphical table of contents. Shows the different menus in the Omavisio information space.
YELLOW On-screen: (OMA1 = iNET) (OMA2 = iNET) (OMA3 = iNET)	**Navigator command:** Run the specified command, taking over the Navigator's wimpslot. When it exits, Navigator will be restarted at the current place. Activates the WWW browser and displays the profile WWW home page. Changes interface mode.
BLUE On-screen: (OMA1 = KAUPPAKESKUS) (OMA2 = KAUPPAKESKUS) (OMA3 = SHOP)	**Navigator command:** Goto to the specified page (homepage of hierarchy). Activates the home shopping center or virtual mall. Changes interface mode.

6.5.2 Keyboard

The Navigator Omavisio user interface is designed so that the user does not need to use a keyboard. There are no default keyboard mappings. However, all of the remote control commands are mapped to keyboard keys.

The keyboard is used primarily for typing URLs (Uniform Resource Locator). The input field into which the URL is entered is opened by pressing a key on the keyboard. Likewise the URL of the page is not shown to the user by default, instead it can be viewed by pressing STATUS on the remote control or another keyboard key. As the status bar can be hidden by pressing STATUS again. Additionally, the keyboard is used for certain games which cannot be controlled with the remote control alone.

6.5.3 Omavisio States and Feedback

The Omavisio system can be in several states during the course of use. These system states are communicated to the user via the user interface. The main user interface states are

- **Loading:** The STB is loading from a server or the WWW. No feedback is given by default. In the WWW mode a graphical status bar showing the percentage of the page content that has been downloaded can be opened by pressing the STATUS button.
- **Waiting for input:** The STB is ready for the user's input. The user interprets this state from the appearance of the area cursor that automatically frames the first hotspot on the screen. This is an indication that the system is ready.
- **Streaming:** The STB is receiving a video stream, which it displays to the user. The user sees the video and therefore, knows that video is being streamed.

The only direct feedback associated with user actions is an a single *earcon.* Earcons are symbolic sounds that inform the user of some action carried out by the system. In Omavisio the earcon is a "clicking" sound which is audible when a hotspot is activated.

6.5.4 Hotspot

There are three ways, all visual, which are employed to communicate the presence of a hotspot to the user. A hotspot is either a "soft" button, a numbered item on a menu screen or a picture (company or brand logo). When the hotspot is an on screen representation of a remote control button, i.e. a colored, video control or a numbered item, within the 1-9 range, the user can directly select it with the corresponding "hard" button on the remote control. When the hotspot is a picture with no corresponding *accelerator* it must be first selected with the pointer before it can be *activated.* In all the user has two ways of activating a hotspot: by pressing the SELECT button when the hotspot is selected, or by pressing the corresponding color or number button.

Hotspot visibility is reinforced by the *pointing device*. The pointing device can only move from one hotspot to another. In other words, the pointer cannot be moved to the right of a selected hotspot if another one is not located there. Instead the pointer will move to the next hotspot even if it is to the left or below the activated one. When over a hotspot the pointer frames the hotspot with a different colored border or causes the hotspot to change in some way; color, shape or both. This feedback tells the user which hotspot can be activated with the SELECT button. In all the pointer could be considered to be a hotspot *locator*, a tool which helps the users interpret the screens better.

6.5.5 Transition Effects

One of the following visual effects is used each time a new page is displayed.

- **Cut:** Simple cut between two pages.
- **Wipe:** The new page drops down from the top of the screen, with a logo bar over the join between the previous and new pages.
- **Fade:** The previous page fades to black and the new page fades up.
- **Scroll:** The new page drops down from the top of the screen over the top of the previous page, unrolling like a scroll of paper.

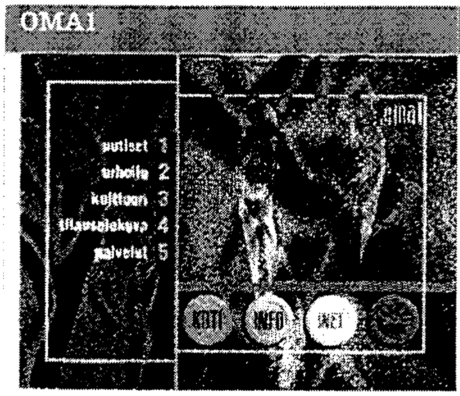
6.6 Omavisio Spatial Layout Level (Physical Component)

The Physical Component was not presented in section on the CLG. This component encapsulates the input and output devices of the system. It contains the Spatial Layout Level which is the description of the interface layout.

Since, Omavisio is a single window interface where each application that commands its own run-time engine from the STB changes interface mode. Consequently, the layout varies depending on the media or application the user is viewing. When the STB is streaming a video no on-screen controls are visible. Likewise, when the WWW browser is activated, the STB only displays the HTML page.

6.6.1 User Interface Layout

As mentioned in the beginning of the report each of the Omavisio profiles channels has its own visual look, see FIG 6-2. In addition, apart from the on-screen color button group located in the lower right-hand corner of the page the OMA3 page layout differs from OMA1 and OMA2.

OMA1	Interface Elements:
	Channel Logo Menu Items RGYB (Color) Buttons

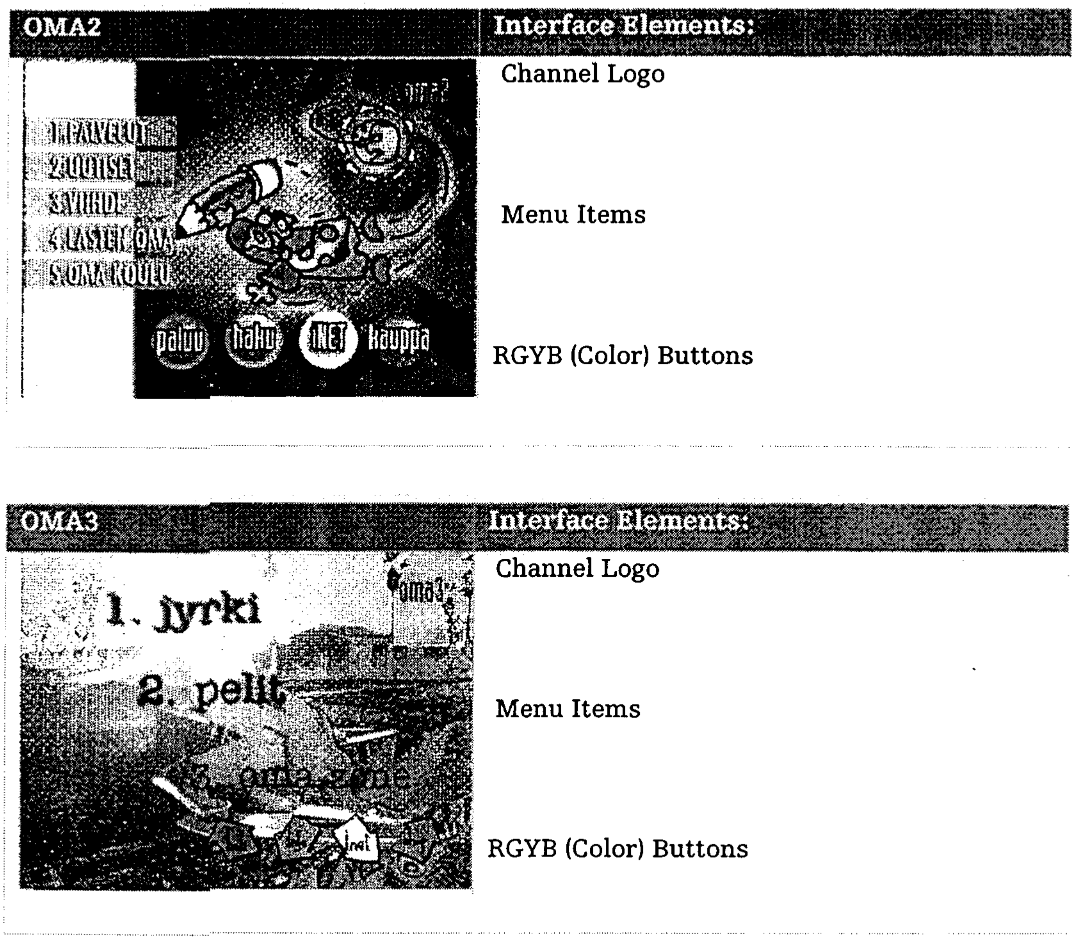


FIG 6-2 *Omavisio Navigator User Interfaces (Omavisio Channels)*

In OMA1 pages blue and black are the prominent colors. Photographs, shown in a blue, green duotone are used as page background images. They communicate to the user the theme of the page, i.e. the category the menu page belongs to. The pictures are pixelated and coarse. The overall color value of pages is dark. The area selector has yellow borders, which contrasts with the dark blue background. White text is used in the menus. The same font is used throughout the pages.

In OMA2 the overall luminance of the pages is much lighter. Light blue, white and red form the page palette. Cartoon figures are used to communicate the menu category. The area selector is a grayish blue, and does not contrast with the blue pane, where the menu choices are located. White text is used in the menus. The a different font is used on the OMA2 pages and it too is used consistently through out the pages.

OMA3 pages do not have a single uniform color palette. Each page has its own palette. In general many colors and visual styles, including fonts are used on the pages.

6.6.2 Display Characteristics

The interface layout and especially element scaling is dictated by the limitations of television. The standard Omavisio display is a PAL 27" system television set. The ratio of the sides forming the screen is 4:3. The resolution of the set is 768 x 576 pixels . The designer has to account for a 5% safety rail area when placing elements near the borders of the display. Due to the low resolution, images appear pointilized and thereby, elements do not have sharp outlines.

In the PAL supported color palette, RGB (Red, Green, Blue) values cannot exceed 225. This results in the lack of pure colors like white, red, blue and green. Since, colors on television are created by adding and subtracting light it is often not possible to display realistic colors associated with earthly (natural) things formed by subtracting light.

7. Experiment and Results

In this chapter the method of evaluation and discovery is explained. The explication includes an overview of various usability methods and the selection criteria for choosing a method. Concurrent Verbalization, the method used in the present study is described in detail. Next, the experimental procedure, i.e. the subjects, setting and tasks is explained. Finally, the experiment results are categorized under the user interface Levels described in the CLG.

7.1 Methodology

7.1.1 Choosing a Usability Testing Method

Various usability methods are presented in Nielsen (1993). TABLE 7-1 is a summary of the most important methods:

TABLE 7-1 *Usability Evaluation and Testing Methods (Presented in Nielsen, 1993)*

Method Name	Lifecycle Stage	Users Needed	Main Advantage	Main Disadvantage
Heuristic Evaluation	Early design, "inner cycle" of iterative design	None	Finds individual usability problems. Can address expert user issues.	Does not involve real users, so does not find "surprises" relating to their needs.
Performance measures	Competitive analysis, final testing.	At least 10	Hard numbers. Results easy to compare.	Does not find individual usability problems.
Thinking aloud	Iterative design, formative evaluation	3-5	Pinpoints user misconceptions. Cheap test.	Unnatural for users. Hard for expert users to verbalize.
Observation	Task analysis, follow-up studies	3 or more	Ecological validity; reveals user's real tasks. Suggests functions and features.	Appointments hard to set up. No experimenter control.
Questionnaires	Task analysis, follow-up studies	At least 30	Finds subjective user preferences. Easy to repeat.	Pilot work needed (to prevent misunderstandings),

There are numerous criteria for selecting the appropriate method. The most important criteria for selecting a method is the number of available *test-users*. In case there are only a few test-users available Nielsen advocates the thinking aloud method along with all of the heuristic (inspection) methods. Another criteria for selecting a usability method is the experience of the experimenters or of the usability staff. For novice experimenters Nielsen recommends the thinking aloud method because it is one of the simplest methods to use.

A third criteria for method selection is the novelty of the system under development. When testing prototypes the experimenter is concerned with discovery of characteristics and problems that will guide later development, not the discrimination of theoretical hypotheses. According to Wildman (1995) one of the keys to designing and evaluating software usability is to know what the users are thinking. Qualitative methods of study are especially good for evaluating new paradigms because they aid in detecting, localizing and diagnosing confusion users experience while interacting with a system (Wixon, 1995). One of the most popular qualitative methods which allows the designer to know what the user is thinking is the verbalization method of thinking aloud also referred to as concurrent verbalization. Thinking aloud is used in the present study and it is explained next.

7.1.2 Concurrent Verbalization (Thinking Aloud)

There are different types of verbalization procedures. Common to all procedures is the subject's oral response to an instruction or probe. Verbalization procedures are usually classified according to the length of the interval between acquiring information and recalling it. Based on this dimension procedures can be grouped into two distinct types: *concurrent* and *retrospective*. Verbalization can also be categorized by the generality or particularity of events that are to be reported as well as by the comprehensiveness of the topics to be reported. The *concurrent verbalization* is the methodology used in the present study.

Concurrent verbalization, also referred to as thinking aloud, involves performing a task under study and verbalizing about it concurrently (Ericsson and Simon, 1980). In the case of studying a user interface a test-user is given a task to perform with the system. The goal of the method is to establish the overall usability of the interface. Nielsen (1993) calls this type of evaluation *formative evaluation* and considers it essential in the iterative process of interface design. Despite its name the method does not try to find out what the user thinks about the interface, which can also be determined, but strives to discover the degree to which the cognitive model of the application matches that of the typical user. The experimenter analyses the verbal data from which it is possible to enumerate the user's cognitive processes and identify misconceptions about the system. Moreover, the experimenter is able to understand system specific usability through the user's cognitive processes: an understanding of usability requires understanding the cognitive processes which enable users to achieve their goals in a satisfactory manner (Bevan, et. al , 1993). In sum the thinking aloud method permits the discovery of a system's usability attributes and for this reason it is a suitable method for the present study.

Thinking aloud was introduced as a means of revealing in greater detail the intermediate stages of cognitive processes. With the thinking aloud method conclusions about the application under study are based on analysis of test-user observations regarding the interface and usability. Test-users, chosen on the basis of being representative of the typical end-user are instructed to verbalize their actions, impressions and observations while they are engaged in a task scenario using the system. The resulting verbalization is recorded on video. This allows the observation of the test-user's behaviour as well as a precise documentation of the verbalization.

The premise of the method's validity, as with all verbal protocol analyses, is that the subject's verbal responses are veridical in standard laboratory paradigms. In other words, the test-user responses are considered and treated as valid scientific data. This issue has been the center of much debate and the cause for criticism against the method. However, most of this criticism has been satisfactorily addressed by Ericsson and Simon (1980), who have concluded that verbalization does indeed constitute a scientifically valid methodology of data collection.

7.1.3 The Processing Model

The strength of the thinking aloud method is in the immediacy of the data related to the behaviour of the subject: the temporal density of the data is high. It is assumed that the user's verbal reports describe directly what the user is doing and thinking at that instance. This assumption is supported by cognitive processing theory which states that information recently acquired and attended to by the human central processor is kept in the *short-term memory* (referred to as STM) and is directly accessible for further processing (Ericsson and Simon , 1980). Hence, the method enables the experimenter to directly determine what information the subject is attending to or processing at that given moment of system operation. The experimenter is able to assess the internal stages of the subject's cognitive process with the aid of a cognitive *processing model.*

Ericsson and Simon have proposed a specific processing model for the decoding of the verbal data (1980). The model aids in the interpretation of the verbal data and the relation of the verbal data to the behaviour of the subject. The basic assumption in the model is that humans store information in several memories each having different capacities and accessing characteristics. The smallest and most vulnerable memory is actually a collection of several sensory stores of short duration. The *short-term memory* (STM) has a limited capacity and/or intermediate duration, while the *long-term memory* (LTM) has a larger capacity for holding information as well as relative permanent storage but with slower *fixation* and *access* times compared with other memories.

A key assumption in the information processing framework is that information recently acquired and attended to by the *central processor* is kept in the STM. This information is directly accessible for producing verbal reports. Information in the LTM has to be first retrieved or transferred to the STM before it can be reported. Due to the limited capacity of STM only the most recently heeded information is accessible directly. The implication is that a link exists between verbalization and the user task, i.e. use of the system. Next, the main cognitive processes and components are briefly described.

7.1.3.1 Cognitive Processes and Components

Recognition

Information received from sensory organs is stored in the STM and is directly recognized and encoded with information already stored in the LTM. This means that the stimulus, i.e. system behaviour, is associated with existing patterns in the LTM. Associations can give the designer of the interface clues about metaphors that could be tried in future iterations of the design cycle. Additionally, recognition is indicative of user experience.

Short-term memory

Only a limited amount of information, counted in *chunks*, can reside in the STM. Chunks are familiar patterns of data. As new information is heeded information previously stored in the STM may be lost. Information about automated processes is not stored in the STM. The implication of the limitation of STM is that when the user is exposed to a barrage of new information in the user interface at one time, the STM is overloaded and some previously stored information is lost. The phenomenon of *cognitive overload* is easily observed since users stop verbalizing when they are under high cognitive load. Therefore, instances of low or incomplete verbalization indicate that the user is under high cognitive load. Subsequently, information in the STM may be destroyed which in turn lowers the occurrence of fixation.

Fixation

The process of creating a LTM representation of STM information through associations by coding and imaging is called *fixation.* This is also called *learning* and usually takes from 8-10 seconds.

Automation

When intermediate steps are carried out without being interpreted and without their inputs and outputs using STM, then *automation* has occurred. Automation is not manifested in verbal reports but is observed in action. Hence, it is important to observe the user and record the verbalization and behaviour. Observation actually compliments the verbal report. Automation results in frequent meta-statements about the processes and less about the inputs and outputs in the protocols.

Control of attention

Interruptions occur during the user's information processing. They can be caused by sudden movements in peripheral vision, loud noises and emotions. The implication of this occurring is that verbal reports before or after the interruption are less complete than had they been conducted orderly, exposing successive contents of the STM. Interruptions suggest a breakdown has occurred in the communication between the user and the system.

7.1.3.2 Recoding Before Verbalization

For the verbalization to be a significant and accurate source of information about the user's cognitive processes, the test situation and the verbalizing process the user engages in have to be subordinate to the actual cognitive process being observed, i.e., the performance of the task. If the test-user is not at ease, and the test is not conducted in the proper context, the results may be misleading. Even when the verbalization is concurrent there can be various intervening processes, each an instance of *recoding*, which affect the resulting verbalization. An intervening process or recoding occurs in-between the time information is acquired and stored in short-term memory and the time a corresponding verbalization is given. Ericsson and Simon (1980) have distinguished between direct or *level 1* verbalization where no recoding is involved and *level 2-3* verbalization where the type of recoding involved prescribes the resulting level. It is important to note that when intervention occurs between acquisition and verbalization the information about the process is modified.

- **Level 2** verbalization involves the recoding of the subject's non-verbally encoded mental representation of the information into verbal form. This type of recoding involves determining correct words and idiosyncratic referents; both processes are dependent on past experiences and knowledge. Furthermore, the subsequent communication of the recoded representation(s) to another person requires further recoding by the subject who has to find understandable referents to ensure the communication is comprehended.

- **Level 3** verbalization describes the necessary additional recoding which takes place when the subject has been instructed to verbalize only about a certain aspect of the task. This restriction on the verbalization requires constant "testing " by the subject to determine whether or not the heeded information matches the desired information. Level 3 verbalization can also describe a situation where the subject has been instructed to verbalize information which is not usually acquired during the performance of a task.

Abstraction and inference are the basic intermediate processes described above. The outcome of these processes is verbalization.

The expected level of verbalization can be determined by the type of information which has been requested from the subject. Information about motor activities, descriptions about what objects the subject moved and where and what the subject is looking at, is usually not directly stored in short-term memory and thus requires recoding.

In summary the concurrent verbalization method is exceptionally good for collecting a wealth of qualitative data about the interface. With this in mind it is important to be able to filter the relevant information from unimportant data. According to Nielsen (1993) test-users will often provide inconsistent information about certain irritants, aspects which they like or dislike. Therefore, adjustments to the interface should not be made on the basis of a single user's comments. The experimenter has the responsibility to interpret the user comments in such away as to determine which are relevant to the larger interface design perspective.

7.2 Procedure

7.2.1 Test-users

The test-users in a usability test must be representative of the actual end-users, see TABLE 7-2. The test-users for the present study were chosen on the basis of the profiles, used to categorize Omavisio content. Anyone who fit one of the three "broad" profiles was a potential test-user. Ten test-users were chosen for the experiment. The traditional categorization of users according to *novice* versus *expert users* did not apply in the experiment. Due to the novelty and uniqueness of the system there were no expert users to select as test-users. Users were not asked if they had prior experience with PCs. The OMA3 test-users were all chosen on the basis on having experience with computer games. In all the test-user population was representative of the three profiled end-user groups, both in terms of user demographics as well as experience with interactive media.

TABLE 7-2 *Breakdown of test-users by channel and age group*

Channel Age Group	OMA1	OMA2	OMA3
17-25 years old			1 Female 2 Males
25-35	2 Females 1 Male	1 Female	
35-45	1 Male	1 Female	
45-55	1 Male		

The uneven distribution of the test-users among the three groups is reflective of the anticipated initial distribution of users.

7.2.2 Experiment Setting

The experiment was conducted at Telecom Finland's research lab in Helsinki, Finland. The set-up of the experiment room is depicted in FIG 7-1. The experiment "room" was furnished to resemble a typical living-room. The purpose of furnishing the room was to create the look and feel of the living-room; the place where people usually watch television. Paintings were put on three walls and the wall behind the iTV appliance was covered with a large imprint of a background image from the OMA1 user interface. The room had two couches which were placed to face each other. In-between the couches was a low coffee-table which was the same length as the couches. The Omavisio remote control and keyboard were placed on the table. The television set and STB were located at a 90 degree angle approximately five feet away from the couches. The STB was placed on top of the 26 inch television set. A video camera was placed behind the couch on the right, directly across from the TV. The camera angle was set so that the test-user and the TV set fit into the same shot. The noise level around the test area was kept to a minimum in order not to divert the user's attention away from the screen during the actual test session.

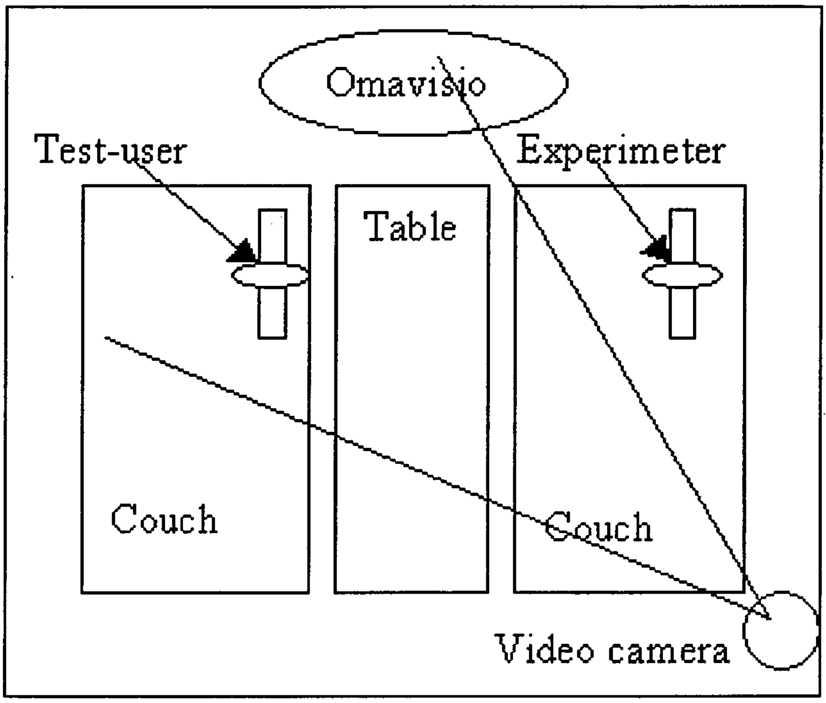


FIG 7-1 *Experiment room set-up*

7.2.3 Experiment Tasks and Procedure

Usually the test-user is given a task or tasks. In this experiment no particular tasks were given to the test-users since, the system being tested is a consumer product. An objective of the experiment was to observe whether or not the test-users could generate specific tasks for themselves, based on the user interface alone. The goal was to see to what extent the test-users saw the user interface as a resource for action: to see if the users would become mesmerized by the system. If they could entertain themselves with it. This included observing if the test-users could logon to the Omavisio system and if they understood the structure of the information space.

When a test-user arrived in the test area he or she was asked to take a seat on the couch across from the video camera. The TV set was already switched on and displayed a blank blue screen. The volume of the TV set was turned down. The blank blue screen is the default state of the display when the STB is connected to the TV but not switched on and the regular TV antennae is not plugged into the STB.

Since the test-users were not familiar with either the system being tested or the actual the test methodology each one was briefed quickly. Each test-user was told that the system being tested was an iTV prototype. The Omavisio brand name was not emphasized in the briefing and in some cases it was completely omitted.

Following the briefing the test-user was instructed to turn on the STB and begin using Omavisio. Also at this point the user was instructed to speak out loud while using the system and verbalise all actions taken during the session. With the exception of the user registration page, where a profile specific password was given, no other instructions or guidance was given at any time during the test-session.

The test-user controlled the progression of the session. He or she created tasks for him or herself, mostly exploring the contents of the system. Each session lasted approximately 60 minutes. Since the test-users were not given a task to complete, the experimenter had to decide when to end the session. The sessions were ended when all of the features and applications had been tried at least once or if the test-user lost interest and wanted to end the session. The later happened with only one test-user.

The complete session was video recorded. One camera angle was used for all sessions. The shot showed both the test-user, sitting on the couch, and the Omavisio system.

7.2.4 Thinking Aloud

During the experiment all of the test-users had to be often prompted to continue to think aloud. Most of the prompting had to be done at the beginning of the experiment.

Only open questions were asked because it is important not to direct the user's way of thinking.

Typical prompts that were used included:

- What are you thinking now?
- What would like to do now?
- What do think is happening?

All questions posed by test-users during the test-sessions were answered with one of the following questions:

- What do you think?
- What do you think will happen if you do it?

7.3 Results

The test sessions were video taped. Approximately eleven hours of video material was reviewed and analysed. The following results are based on observations made during the test sessions and on review of the test-session video. Observations about test-user verbalizations and experimenter observations about test-user and system behaviour has been abstracted to arrive at the following results.

7.3.1 Test-user Verbalizations

The test-user verbalizations dealt with all aspects of the Omavisio system. The verbalizations included opinions about the interface graphic design, system functionality and content, i.e. the games and videos. Some test-users evaluated the variety and quantity of selections that were available. Others expressed what they had expected from the system. At times, users asked for help in carrying out an action they believed was possible to do with the system. Most asked for help when they failed to accomplish what they intended to do. Additionally, the test-users verbalized the difficulties they were experiencing using the system. Occasionally, some users offered explanations for system behaviour . Still others explained away errors they made. Furthermore, suggestions were offered on how to improve the interface, i.e. which features were lacking.

Most of the test-user verbalizations can be considered to be of Level 2 type. Since, no specific task was given, the verbalizations made during the tests concentrated on those aspects of the system that grabbed the user's attention. Thereby, most of the time the test-users were engaged in determining the correct words and idiosyncratic referents to express.

Observations made about user behaviour that accompanied the verbalizations revealed problems the users experienced. In addition, patterns of user behaviour were noted. User verbalizations and behaviour was analyzed in light of observations about system behaviour at the time of the verbalization.

7.3.2 CLG Classification of Results

By classifying the results according to the CLG framework the usability of the current interface conceptual model can be better understood. The CLG classification gives an indication of the severity of the interaction problems the users encountered in the tests. Problems which adversely affect user knowledge on more than one Level of the interface are considered severe. They result in the wrong user model and prevent conceptual integration.

7.3.2.1 Task Level

The users were not familiar with the Omavisio system. They had no prior experience with iTV applications. Test-users were not asked prior to the tests about which services they would like to be offered through iTV, i.e. no market research was conducted. Consequently, very little was verbalized about the tasks that could be accomplished with Omavisio. Users were more occupied with the interaction aspects of the system, i.e. controlling the system.

The most notable result at the Task Level was the user expectation that Omavisio could be used to research information about products. A case in point was the user reaction to a commercial which began playing when users activated an advertisement image on the OMA1 main menu page. Test-users had expected to be presented with detailed product information instead of the commercial (video) that began playing. Based on the majority of user comments commercials should not be shown on iTV, they do not belong to the media. Users want to see product information, they want iTV to support purchase decisions. The creation of meaningful information mosaics was a common user goal.

7.3.2.2 Semantic Level

User Model of System Operation

The results show that users perceived Omavisio as an information space where navigation is the primary mode of interaction. All users referred to the menu pages as places in a space. Some common verbalizations were "I'm going to the next menu" and "I don't know where I am?". One user compared using the Navigator menus to browsing on the WWW and subsequently, expressed a fear of getting lost like on the WWW. In sum, users equated learning to operate Omavisio with learning to navigate between the menu pages.

The intuitiveness of the navigation model became clear from observing users in video mode. All users tried at first to navigate away from a video they had been watching when they wanted to do something else. Users pressed the BACK and HOME buttons on the remote repeatedly. One user threatened to turn the STB off after frustration set in when neither button functioned as expected. Though all users "discovered" and used the STOP button successful, they still initially tried to navigate away from the video the following time they came across video content. Two users made the same mistake every time they wanted to exit video mode. This behaviour suggests that users considered navigation to be more intuitive than the use of the video controls. Interference from the predominant navigation model lowered fixation of the video model.

Organization of the Information Space: Koti and Info Pages.

Test-users actively interpreted the hierarchical structure of the Omavisio menu pages. Users expressed their views of which page was where in the information hierarchy. For instance, one user thought that the top level of the hierarchy consisted of the pages represented by the row of colored buttons: Koti, Info, iNET and Kauppa Keskus. Others based their view of the Omavisio information hierarchy on the assumption that the first menu they saw after the registration page was the top node in the tree hierarchy. This initial assumption caused users to misunderstand the OMA concept of three channels and the purpose of the Info page.

The concept of profile specific Koti, or home, confused all test-users at the beginning of the test. For some the functionality of the Koti button caused problems all the way through the test. At the beginning of the test, before trying the Koti button, users interpreted Koti to be a different page than the main menu page. Although, most test-users eventually equated the two in their own channel, i.e. within one branch of the hierarchy, they did not grasp the concept of each channel having its own home page or main menu.

All users held the belief that there was only one home or main menu in the Omavisio information space. This was evidenced each time users wanted to return to the page from which they had begun navigation of the information space. For example, one user wanted to return to the OMA1 main menu from a OMA2 menu screen, to accomplish this goal he pressed HOME on the remote control. Since HOME is modal, i.e. relative to the channel the user is using, the user ended up navigating to the OMA2 home. Realizing that HOME had not worked as he had expected the user proceeded to select and activate the red soft button labelled Koti. The user behaviour and verbalizations indicated that the user did not revise his model of only one home based upon the observed system response. In-fact most users associated different functionality to the HOME and Koti buttons.

Several test-users called the Omavisio structure misleading. They came to this conclusion usually toward the end of the test session. While users verbalized that they were beginning to understand the structure, their behaviour and the comments about the Info page suggested otherwise. In conclusion test-users had difficulty understanding the overall information hierarchy despite an overview of the information hierarchy provided by the Info page.

User Model of Interface Style

The profuse use of computer terminology by all users in their verbalizations indicates that the initial user model of the interface style was based on prior internal knowledge of personal computers. The direct manipulation style and concepts relating to the browsing of the WWW were particularly prominent in the verbalizations.

Most users clearly expected the Omavisio interface to be a direct manipulation graphical user interface. One user enquired at the start of the session while the user log-on page was loading if he needed a mouse to interact with the system. Another user wanted an undo command to replace backtracking in the information hierarchy. Furthermore, the test sessions demonstrated that users expected higher interface persistence in the form of confirmatory dialogues. Most test-users anticipated that they would need to confirm the PIN code they had entered on the user registration page. These users sought for an ENTER button on the remote control. The expectation of having to confirm input with an ENTER command, resulted in users taking a surprisingly long time to learn the functionality of the SELECT command.

The use of a computer model was particularly prominent in WWW mode. One user interpreted the hand shaped cursor which is displayed briefly when Navigator mode switches to WWW mode, to indicate that she needed to switch input devices from the remote to the keyboard. Another user wanted to resize and move the browser "window" so that the particular WWW page would fit horizontally on the screen. More significantly, most users remarked that the Omavisio browser window was missing a navigation tool bar; the toolbar is a standard feature in all PC browsers.

User model of Omavisio

The TV model influenced user behaviour at the outset of Omavisio use.
Most test-users considered the TV and STB to be a fully integrated system. The first buttons that most of the test-users pressed when trying to switch on the STB were the number buttons. This behaviour is analogous turning on a TV set directly to a particular channel by pressing the channel number. Furthermore, two test-users spent considerable time looking for the TV volume control on the STB remote. Another user interpreted the REWIND and FORWARD video control buttons to be the aforementioned volume control.

7.3.2.3 Syntactic Level

System Modality

The modality of the Omavisio system caused major problems for test-users. Test-users constantly tried using commands from one mode in another mode. One user tried to navigate between the Navigator menus using the REWIND and FORWARD video control buttons. Another user tried substituting BACK in Navigator mode with the keyboard ESC button after she had used it to backtrack from the WWW mode to Navigator mode. Another common mistake made by users was the generalization that the BACK and HOME buttons were generic and thereby, modeless. When the buttons did not work as expected the users firmly held on to their belief and proceeded to try them in the next mode. The model of modeless or generic commands persisted.

Unclear mode boundaries also caused problems for users. Users did not perceive Omat Uutiset to be modal. The blue button on the Omat Uutiset pages, used for navigating to the page where the user's news profile can be configured, was overlooked by all users although most anticipated and looked for a way to personalize the news service. Furthermore, the modality - different functionality - of the video controls on Omat Uutiset videos prompted users to generalize that the video controls did not function properly. In addition, the boundary between Navigator and WWW browser mode was unclear to some of the users.

It is not surprising that test-users considered lack of modality desirable. For instance, users appreciated the presence of the colored buttons on all Navigator pages because they allowed modes to be switched effortlessly. Furthermore, the user behaviour on the Omat Uutiset pages demonstrated that users believed that the colored buttons would work uniformly on all of the Navigator pages.

Navigator and WWW Browser Feedback

System feedback was completely missing, inadequate or inappropriate. The lack of feedback often left users confused and above all decreased system learnability. Additionally, users expected feedback even in situations where they clearly knew the system state or what the system was doing.

At the start of the test-sessions, while waiting for the user registration page to load, several users remarked that they expected a system message to appear informing them to please wait while Omavisio pages were being loaded.

Some users misinterpreted system related as well as non-system related behaviour to be feedback. One user interpreted the arrow cursor, an element of the STB operating system interface that appears during the transition from Navigator mode to WWW mode, to be a system signal for pages being loaded. Another user mistakenly associated the HOME button with TV volume control because when the system is in video mode it mutes the TV. A third user misinterpreted an animated GIF image on a WWW page to be some sort of system feedback indicating that the system is loading the page.

Inappropriate feedback was also a problem for users. The fade effect used between pages was often misinterpreted. Well into the test-session users still thought that the system had failed or the network connection was broken each time the fade in and out of black transition between pages took longer than usual (about 2 seconds). In fact, the longer the effect took, the more worried the users became. Inappropriate feedback was also a problem in WWW mode. Users had difficulty understanding the meaning of the check mark which is displayed in the graphical status console. In Finnish culture the checkmark is associated with school where it is used to mark incorrect answers on tests. In Omavisio the check mark is used to inform the user that a page is being loaded from the network.

Feedback about system state and mode was inadequate or completely lacking. In particular, users had problems with the WWW mode. All users enquired whether the mode had changed the first time they navigated to the WWW mode. Some test users, OMA1 users in particular had trouble determining what action to take on the first WWW page that was displayed. The view the user sees of the first page that loads when iNET is pressed in OMA1 Navigator mode does not contain any hypertext links. The links on the page are not visible because they are at the bottom of the page. Moreover, the system does not automatically give feedback about how much of the current WWW page is visible on the screen. There is no scroll bar which communicates this information. Hence, the users did not at first realize that it was possible to scroll the page down.

STB Operating System Feedback

The only time the system does give direct feedback as a result of a user action is when the user tries to backtrack from the first WWW page. The STB superimposes an operating system dialogue on the single WWW browser window. The dialogue window displays the message "No More Docs. in History." The apparition of this message surprised users. The unexpected and unfamiliar looking dialogue window was interpreted to have appeared as a result of some illegal operation. One user turned the STB off in the belief that he had made a critical error which had caused the system to fail. Others thought that they would be able to confirm the dialogue window, press the OK soft button, with only the keyboard. After this incident the users who were presented with this dialogue window only used the keyboard in the WWW mode.

Navigation Feedback

The lack of navigation feedback resulted in users misinterpreting the functionality of commands. Several test-users concluded that Koti, the red-soft button, did not work because activating it had no noticeable effect on the system. In actuality the command did function as users were to discover later in the test. The reason why the system did not appear to respond was because the users were already on the Koti page.
The OMA1 main menu, Koti and all of the OMA2 and OMA3 Navigator pages including Info page lack page titles. Thereby, users were obligated to induce from the contents of the menu what page they were on. This prevented users from forming an accurate model of the Omavisio information space. Users could not immediately tell from a page where they were in the overall Omavisio information space. Furthermore, lack of page naming complicated user tracking of visited pages. Since unique page titles were missing users were forced to rely on recognition and interpretation of numerous menu items in-order to be able to determine what page they were on and subsequently where they were in the whole information space. Expectedly, one user suggested that the system should provide real-time information about where each hotspot leads to; especially, information pertaining to contents of the subsequent menus.

Inconsistent Naming of Commands and Pages

Inconsistent naming of commands and poor choice of command names lower quality of use. Though buttons looked similar and were the same color, users still expected functionality to be different because the names on the buttons were different. OMA1 and OMA3 users expected the red and green buttons on the OMA2 pages to function differently because they were named differently. The green button on OMA2 pages is called "Haku" while on the other Navigator pages it is "Info". Info is short for information and Haku means "search". The different names suggested different functionality to the users. Later in the test users discovered that both display the Omavisio service directory.

The red button on the OMA2 pages is named “Paluu” while on the other Navigator pages it is “Koti”. “Paluu” means “return back” and “Koti” means “home”. Users expected the two buttons to function differently. Users used the button when they wanted to backtrack one level in the OMA2 hierarchy. The only time the button worked as expected was when the user was on the second level of the hierarchy: BACK and KOTI both took the user to the OMA2 main menu page. Users considered the other instances, when Paluu did not work as expected, to be the result of a system error which they attributed to fact that the system was a prototype.

Misleading Naming of Commands and Pages

The choice of command and page names caused some problems. Users associated the “Info” name with boring descriptive information about the system and therefore, many explored its contents at the very end of the session. One user repeatedly confused Info with HELP. Each time the user was experiencing difficulty she pressed the green button in the hope of getting instructions on what to do next. The general impression among test-users was that the Info name was misleading; it did not accurately reflect the purpose and functionality of the page.

Quick Exit from Modes

When test-users experienced difficulty with a particular mode they looked for a quick exit from that mode. The tolerance for inexplicable and incomprehensible interface behaviour was extremely low. Some expected HOME on the remote control to be a generic exit that allowed backtracking to the Navigator main menu page from the other modes. Two users resorted to the STB power switch when they could not exit or proceed in the WWW mode. Booting the system acted as an exit from the WWW mode. It was also a means of navigating away from a WWW page that the user was stuck on because he was experiencing problems controlling the scrolling of the page.

7.3.2.4 Interaction Level

Direct Manipulation and Mapping of Selector Movements

Users had trouble understanding how the area selector worked. Initially, all users expected to be able to manipulate the area selector freely on the page. However, the selector could not be manipulated in the manner users expected. The movement of the selector is programmed to tab from one hotspot to another. Thereby, movement of the selector is primarily controlled by hotspot placement and not by the user. Consequently, at times during the tests the selector moved in a different direction than what users had specified; for instance, it would move down instead of to the left. Users felt they had lost control of the interface.

Despite the illogical movement of the area selector users seemed to prefer the selection and activation style of interaction to the use of the accelerators. This was especially the case with the Navigator page color buttons. The affordance of the button shaped color buttons invited users to act upon them rather than directly activating them with an accelerator.

The appeal of the direct manipulation style was evidenced in WWW mode where it was a source of many user problems. Users tried to use the area selector to scan the text like they would with a mouse. The selector moves from one link to another and in the process it automatically scrolls the page if the next link is not in the part of the page that fits the screen. This system behaviour caused users to scroll the page unintentionally, often making them lose their place in the text they were reading. Comprehension of area selector functionality in WWW mode was further complicated by the doubling of the arrow button control set as page navigation short-cuts. The top right and bottom right arrow buttons act as page up and page down controls or up and down anchors, while the other arrow buttons move the area selector from link to link. In particular, one user found the dual functionality of the arrow buttons impossible to comprehend. He was not able to navigate to another WWW page because he kept navigating from top of the page to the bottom and vice versa. Instead of trying the other arrow buttons he turned the STB off.

In Navigator mode, particularly on the OMA3 pages, comprehension of area selector functionality was impeded by inconsistent grouping of hotspots and lack of consistent page structure. Users described the movement of the area selector on the OMA3 pages as illogical.

Slow System Response Time

All test-users complained about the slow response and processing time of the system. Delays in system response left the causality of actions unclear for the users. This led some users to deduce wrong rules of operation. One user asked if the inputs had to be confirmed with an ENTER since, the system did not seem to respond to the input. Unsure whether the system had registered their input, users often pressed the same button till they perceived a reaction from the system. Some users thought for a good part of the test-session that some commands had to be entered twice before the system registered them. Others thought that there was something wrong with the remote control. Entering the same command several times in a row slowed the system response time even more making matters worse. Consequently, there were times during each test-session when the user was out synch with the system feedback: user did not know which action caused what subsequent system reaction. Needless to say, the system response time affected the pace of interaction: test-users were often left waiting impatiently for the system to process their input.

7.3.2.5 Spatial Level

Users experienced problems with several aspects of the visual representation of the interface. Users found text difficult to read. Furthermore, they had trouble interpreting the page layout and the affordance of page elements.

Legibility

Legibility was a common problem for all test-users. All found the text on some pages nearly impossible to read. The font used to label the OMA3 color buttons was illegible because it was too small. The use of a different font-family also caused problems for OMA1 and OMA2 users. The end result was that all users misread “Koti” as “Koli”. This was a significant problem especially for first-time OMA3 users because unlike “Koti”, “Koli” has no particular meaning. The functionality of the button was concealed due to the poor choice of font.

Affordance

Determining the affordance of interface elements was problematic for test-users. Hotspots on a page were overlooked because they did not contrast with the page background or the hotspot functionality of the element was secondary to purpose of the element. For instance, practically all users initially failed to comprehend the functionality of the Info page because they did not perceive the listed items to be hotspots. The presence of hotspots was not efficiently communicated to the user, i.e. the items in the columns where not numbered nor were they made to look like buttons.

Moreover, the poor communication of element affordance can be attributed to the poor visibility of the frame created around a hotspot by the area selector. Lack of affordance can also be explained by the inconsistent representation of hotspots in the Omavisio interface. In most OMA1 and OMA2 menus hotspots are numbered, though there are some, apart from the colored buttons, that are not. The numbered hotspots were obvious to the user while the unnumbered ones were rarely spotted. Few users discovered that by selecting and activating the ad image on the OMA1 main menu page they could watch a commercial about the product. The majority of the users wondered if the image was also a hotspot, a link to information about the product being advertised.

In general visibility of hotspots was considered better when the whole menu item instead of just the item number was framed by the area selector. To compensate for the poor visibility of hotspots one user suggested that on-screen instructions be provided: "Choose from one of the following."

The importance of consistency and intuitiveness in hotspot representation became evident from the behaviour and verbalizations of those test-users who navigated to the OMA3 channel after having explored one or both of the other channels. These test-users immediately understood that the arrow shaped interface elements could be activated with the color buttons on the remote. However, it took much longer for OMA3 users to understand how to activate them - the area selector cannot be tabbed to these buttons. In-fact, OMA3 test-users had considerable problems making the connection between the on-screen color soft-buttons and the remote control color buttons. In sum, the affordance and mapping of the color buttons on the OMA1 and OMA2 pages is more natural and therefore, more easily fixated to LTM.

The reverse phenomenon was observed with menu item hotspots. Users accustomed to numbered hotspots in OMA1 and OMA2 had difficulty making selections in OMA3 menus where the hotspots were not numbered. Since the hotspots are not numbered the only way of activating a hotspot is by framing it with the area selector and pressing SELECT on the remote.

All test-users had trouble making selections on the OMA3 shopping page where hotspots are placed randomly on the page. First, test-users found it extremely difficult to determine which elements were hotspots. Second, test-users found the use of the area selector to more difficult than on other pages. Tabbing the area selector to a hotspot became a cognitive task because users could not easily determine which arrow button to press in-order to reach the next hotspot. The task was complicated by the illogical ordering of hotspots (see the Interaction Level results).

Layout

Each time the users came across a new layout the amount of verbalization decreased; an indication that varying page structure resulted in cognitive overhead. In general, users ended up making judgements about the consistency of interface functionality based upon the appearance of pages. Users interpreted the different page layout between Navigator pages and WWW pages to mean that the functionality of pages was different. Moreover, users predicted that the different layouts used in OMA1 and OMA2 page and OMA3 pages also meant that the functionality of the area selector on these pages had to be different.

Layout also affected the ease with which users were able to tab the selector around the page. Users had trouble with the inconsistent placement and grouping of hotspots on menu pages. On WWW pages the location of links within a body of text and on image-maps made it difficult to make selections with the Omavisio area selector. Throughout the test sessions the users seemed to expend more energy on solving how to get the selector to a particular hotspot than on deciding where to navigate. Framing a hotspot and activating it proved to require a high degree of cognitive resources.

Graphic design

Test-users commented on the quality of the Omavisio graphic design. One user found the OMA1 graphic design to be too dark and messy. One OMA2 user thought that OMA3 graphics were extremely messy, too colorful and confusing. This user was put off by the visual appearance and did not want to explore the channel at all. Another test-user thought that OMA2, the family profile, was aimed at children because pastel colored cartoon figures were used for page backgrounds.

8. Discussion

In this chapter the test results are analysed. The CLG Psychological View is used to examine the usability of Omavisio. The findings from the analysis are extended to the iTV domain, and usability of iTV systems in general is discussed. Finally, based on the usability requirements specification, recommendations and guidelines for the next round of Omavisio interface design iteration were presented.

8.1 CLG Analysis of Results

The CLG framework encompasses all the dimensions or components of a user interface. By examining how the design of the interface supports the different components and how the components are linked through design, an understanding of system usability can be developed. The following sections examine the user knowledge of the interface components as well as how the design supports the mapping of the components to each other.

8.1.1 Overview of Results

The Omavisio test-users were not familiar with the task structure - the tasks, task entities, task procedures, and task methods - of the system. According to the Psychological View of the CLG framework, when a user lacks prior training in the use of the system, task knowledge is interpreted and induced from the behaviour and visual representation of the system. Therefore, the Omavisio test-users had to acquire this external knowledge about the Task Level from the Communication Component, i.e. Syntactic and Interaction Levels, as well as the Spatial Level of the user interface. It was found that the implementation and comprehension of the Communication Component and Spatial Level affects user comprehension of the Conceptual Component. It was also found that the test-users employ internal knowledge, i.e. models of the Conceptual Component, in their interpretation of the Communication Component and Spatial Level. The results demonstrate that internal knowledge held by test-users causes interference and hinders conceptual integration. It was also found that the mental models test-users form on the basis of the Omavisio system image are not usable. There were several short-comings in the system image, conceptual model, which result from common interface design errors. The most notable error is interface inconsistency. As a result of the errors the Omavisio information space is not communicated clearly to the users; the test-users' models of the space did not accurately represent the information space as it is designed. Moreover, the test-user models of the system reveal functional and conceptual short-comings in the Omavisio conceptual model: the system does not live-up to user expectations.

8.1.2 User Understanding of the Conceptual Component

Test-user understanding of the Conceptual Component is based on knowledge acquired from the Communication Component and the Spatial Level. However, test-user comprehension is negative affected by design shortcomings and errors. Users do not acquire critical knowledge that would allows them to construct a usable model of the information space There are two reasons for this. First, the information space is not well designed. Second, several interface design errors on the Communication Component and Spatial Levels impede users from acquiring accurate structural knowledge.

Information space

The tests show that test-users understand fairly quickly the main conceptual operations involved in interaction with Omavisio. They recognize that the variety of choices presented in the Omavisio menus form an information space; i.e. the workspace. Understanding the structure of the information space is a user goal. This observation is consistent with McAdams (1995) who wrote that organizing information into meaningful components is a means of understanding.

Users try to comprehend the information space consisting of services and videos, by giving it a structure. Dillon, McKnight and Richardson (1990) call this structure the cognitive map. When users have survey knowledge, they know landmarks that allow them to position themselves in the space, they have the correct cognitive map. With the aid of the cognitive map users can explore the information space, without getting lost. When users can explore, moving easily from one topic to another, or from one source to another they are able to build mental connections that reinforce learning and comprehension. The Omavisio interface does not adequately support navigation. Consequently, users are not able to construct the correct cognitive map.

System modes affect the information space structure and nature of the workspace. The Omavisio information space is structured, information is classified, according to the profiles used for the OMA channels. While, the test-users understand the different categories of information, i.e. News, Culture, etc., in general they do not comprehend the Omavisio classification of information according to profile or modes: the Omavisio concept of user profile based information hierarchies is not clear to the test-users. The several times during the tests that test-users could not orient themselves in the information space, i.e. lost, they chose to return to the start of the path. The test-users logged in to the system again, by booting the STB, instead of attempting to retrace their way back to the point where they strayed. Such behaviour is common in situations where users have limited knowledge about the system. Without the correct cognitive map of the information space, it is impossible for the users to understand the system and engage in problem solving. In conclusion, systems that are based on navigation should make the information space easy to understand and provide the user with the right tools needed to form a cognitive map of the space.

Feedback

According to Riley (1986) knowledge cannot become more robust if implicit assumptions do not become explicit. The only way this can happen is if the interface behaviour is meaningful, distinct, and above all observable. However, the quality and quantity of feedback provided by the interface does not allow evaluation of actions against goals: it does not fit the criteria for adequate feedback.

Namely, the lack of navigational feedback prevents test-users from forming an accurate cognitive map. Users are forced to interpret interface mode and location in the information space from appearance and contents of pages instead of being able to recognize it from a page title. The interface does not make this important knowledge explicit.

Lack of feedback when the system is in a busy state is another problem for the test-users. Users have to interpret system state during the times the system is processing a command, i.e. retrieving a page, a game or video because it does not inform that it is busy. The cognitive load on the user which results from the need to interpret the system state contributes adversely to the learnability of the interface model. According to the Processing Model when the user is under high cognitive load information in the SMT may be destroyed which in turn lowers the occurrence of fixation. More importantly, because of the slow system response time, test-users conjured non-applicable rules of operation. Thereby, test-users end up revising their mental model of the system based on inaccurate observations made about the system Syntactic and Interaction Levels.

Furthermore, some of system feedback works counterproductively, actually making the system more difficult to comprehend. A case in point is the STB operating system dialogue in WWW browser mode which renders the system model more complex and hence, more difficult to understand. The dialogue introduces the concepts of multiple windows and the presence of an underlying operating system to the user's mental model. These concepts are not consistent with the single window model initially taught to the user in Navigator mode.

Inconsistency

Conceptual integration of knowledge on a Semantic Level and Syntactic Level increases when syntactic rules are consistently applied throughout the user interface (Van der Veer, 1991). Consistency is important in the design of a usable information space (Riley,1986). However, inconsistency is present on all levels of the Omavisio interface. It hinders user understanding of the Omavisio Conceptual Component.

Inconsistency at the Spatial Level affects the user's mental model. The different page layouts used to accentuate the profile modes unnecessarily complicate Omavisio use. Test-users refrain at first from employing known semantics in interaction when they were presented with a new layout. Test-users interpret the different lay-outs to represent different interface functionality. Thereby, lack of a consistent layout hinders positive transference of knowledge between modes.

Inconsistent visual communication of hotspots negatively influences user behaviour. The importance of consistency and affordance of screen elements in the interface becomes explicit from observing users who are first exposed to the OMA1and OMA2 pages. Moreover, it is evident that knowledge acquired from OMA1 and OMA2 interfaces is more applicable to the overall use of Omavisio than knowledge acquired from the OMA3 interface alone.

User interface inconsistency is a result of system modality. Knowledge acquired from interacting with one channel is not easily transferable to interacting with another channel. Hence, the learnability of the system is low. It takes longer for users to understand how the system works. Modality increases the amount of knowledge, both strategic (declarative) and lower-level (procedural), the user has to acquire. Knowledge about the sequence of button pushes is not enough. The user has to have strategic knowledge about system states in order to be able to apply the correct lower-level knowledge. The implication of this is that user understanding, i.e. declarative or conceptual component knowledge, is required in the use of Omavisio: knowing the sequence of button pushes does not allow the user to accomplish goals.

8.1.3 Effects of User Internal Knowledge on Understanding of the Communication Component and Spatial Level

Users interpret the Communication Component and Spatial Level in light of assumptions they make at the Semantic Level. The most prominent test-user model is the PC model. This makes the acquisition of the Conceptual Component model more difficult because user internal knowledge is not directly transferable to Omavisio interaction.

Common User Model

User understanding of the Communication Component and Spatial Level is influenced by internal knowledge. The Omavisio metaphors do not help users use the right kind of internal knowledge. Instead the metaphors introduce complex internal knowledge to the user model which complicates instead of simplifies system use. Users assume that Omavisio functions like a WWW browser. They base their models of the Semantic, Syntactic and Interaction Levels on this assumption. As a result, all users experience interference which occurs when the conceptual model (interface model) is very close yet very different from one of the user's existing mental models. Since the inapplicability of the internal user knowledge is not made explicit to the user, it takes along time for users to realize that the interface does not live up to their expectations of system functionality and interface style.

Navigation

The user model of the Semantic Level influences understanding of the semantic procedures involved in the use of Omavisio. The test-users acquire the semantic model of the information space from the interface. The most prominent semantic procedure that test-users acquire from the interface is the procedure of navigation. This procedure fits the model of the Conceptual Component that they bring to the test. It is the main component of test-user strategic knowledge. The results show that test-users apply this procedure intuitively: navigation is the primary mode of interaction.

User behaviour is heavily influenced by the intention to navigate. Users always first try to backtrack when they want to exit a mode. In VOD mode this is the wrong semantic procedure. In a conventional model of video operation, video plays when the play button is pressed and ends when the stop button is pressed. The conventional model is not utilised by test-users, although the conventional video controls are present on the

Omavisio remote control. In sum, user understanding of the different modal syntactic models is negatively affected by the intuitiveness of the navigation semantic procedure.

Interface Style

Omavisio test-users clearly rely more on the use of the area selector than on accelerators. The user expectation is that the interface style is direct manipulation. This expectation results in a negative transfer of knowledge from the domain of GUI operated systems and prevents users from successfully controlling the system interface. Users expect to be able to directly manipulate the area selector in an intuitive and obvious way with the arrow buttons. In actuality cursor movement is mapped according to hotspots. Therefore, the users' model of the Interaction Level does not match that of the system. Users struggle to find the correct expressions and actions for their intentions and as a result they expand cognitive resources on interface. In sum, users do not have the Syntactic and Interaction Level knowledge required to operate Omavisio.

Furthermore, the direct manipulation style model held by all test-users raises their expectations about the appropriate level of system feedback beyond what is technically possible with the Omavisio platform. As pointed out by Shneiderman (1983) an important attribute of the direct manipulation style is that the result of a user action is immediately displayed to the user. Users expect the interface to be more responsive to their actions. Thereby, users are constantly interpreting the status of interface elements, even when the interface behaviour is not linked to user behaviour. Subsequently users form inaccurate models of the Communication Component.

These findings are consistent with Nielsen who has reported the direct manipulation style to be abductive. Active manipulation of objects is a more familiar model to users than menu based selection. Users prefer to directly manipulate interface objects to the extent that they ignore the other styles supported by the interface.

8.2 Omavisio Usability (iTV Domain Usability)

The CLG analysis of the test results, pointed out the usability short-comings of the Omavisio system. It demonstrated that the Omavisio user interface is not usable, i.e. it is not learnable, satisfying, efficient or memorable. Conceptual integration is impeded by poor interface design and inadequate user support. User satisfaction is low because the system does not live up to user expectations.

Next, the determinants of usability discovered in the Omavisio tests are enumerated. Four of the five attributes enumerated by Nielsen are refined to describe the Omavisio interface context, which has been described through the analysis of the test-results.

Learnability

It was found that the use of iTV involves navigating the information space of choices. Users go from an over-view state to a specific application or chunk of information. Learnability, therefore, applies to the degree to which the user acquires the necessary knowledge to seek and find the relevant navigation styles and paths of the system. This knowledge is acquired through interaction with the system.

Nielsen (1993) reports that users do not wait till they have completely mastered the system before they begin to use it. Most users begin using the system when only a part of the user interface has been learned. The resulting behaviour is *exploratory learning.* Exploratory learning is the primary means for a user to learn to use a system. It contributes to the speed with which the system is mastered.

Exploratory learning is preconditioned by the inclusion of easy to understand error messages, undo actions and systematic prompts of the user before execution of risky commands in the user interface. In addition, the user must be able to accomplish meaningful work with the application when only a part of it has been learned.

Ideally iTV will be as easy to learn to use as television is. However, in light of the trade off that exists between the control of the device and the control of content which is the underlying paradigm in iTV, augmenting system learnability can prove to be challenging. Generally speaking, increases in system functionality add to system complexity which in turn hinders learnability.

While Omavisio designers have approached learnability with the goal of making Omavisio as easy to use as a walk-up and use systems or information kiosks, they have sacrificed efficiency in the process. Walk-up and use systems are intended to be used only once, and therefore, they require an essentially flat learning curve. The zero learning time is a result of the underlying simplicity of the system as well as its narrow scope of functionality. In terms of functionality, iTV falls in-between walk-up and use systems and complex computer systems. The importance of learnability in the Omavisio user interface can be compared to the importance of learnability in WWW sites or other hypertext systems.

Efficiency

Once the user has attained expert-level the system should allow the efficient execution of the task(s) the user desires to accomplish. Efficient use necessitates the inclusion of the optimal functionality in the system for the accomplishment of the given task which in turn adds to the complexity of the system and takes away from its learnabilty.

In the context of iTV efficient use refers to the efficiency of being able to carry out processes involved in the four user needs: browsing, searching, selecting and viewing. These processes can be made more efficient if navigation is supported better in the interface. The user should have navigational aids like history views of visited pages or videos watched as well as a favourites list that allows direct access to any level of the information hierarchy. With such tools users are able to efficiently make choices and carry-out selections much quicker and in a more satisfying way.

Subjective satisfaction:

Satisfaction is the most important usability attribute in systems which are used for non-work activities (Nielsen, 1993). As the successor of television, an appliance with a high user satisfaction rating, iTV has to be especially rewarding to use in order for it to be widely accepted.

Users associate satisfaction with the feelings of directness and control when using a system. Nielsen reports computer systems are perceived to be pleasant to use if they allow the user to feel in complete control. Therefore, interactivity in iTV has to be fluid and according to the terms of the user. In large part the feeling of control experienced by the user is determined by the quality of communication between user and system, i.e. feedback augments user satisfaction. In the case of Omavisio better control of system can be achieved through the implementation of the direct manipulation interface style.

Furthermore, as mentioned earlier subjective satisfaction can also be augmented by the implementation of navigational aides which support user manipulation of information. iTV users want to manipulate the information space of choices and navigate in it effortlessly.

Memorability

Once the use of a system has been learned, even with infrequent use the user should be able to remember how the system was operated. Memorability is a problem with Omavisio and all highly modal iTV systems. Modality increases the amount of knowledge the user has to remember in order to operate the system satisfactorily. Thereby, memorability can be increased by limiting the system modality.

8.3 Recommendations

8.3.1 Severity of Interface Problems

This study has discovered usability problems in the Omavisio user interface. The severity of these problems can be assessed with the aid of the CLG framework. The rule of thumb is that Communication Component problems are less severe than Conceptual Component ones. Design errors, relating to apparent usability, can be corrected by modifications to the Communication Component and Spatial Level. However, modifying the Communication Component will not suffice; even with a new Communication Component the interface will not allow efficient and skilled use to take place. Skilled performance is only possible when a system is inherently usable. Controlled automatic user behaviour is observable in situations where the user understands the system; when conceptual integration occurs. Therefore, to meet user expectations and goals it is recommended that the Conceptual Component be redesigned based on the findings of this study.

8.3.2 Conceptual Component

The organization of the information space is a determinant of Conceptual Component usability. The redesigned system image should make the information space structure and mode boundaries clearer to the user.

It is recommended that the interface be redesigned to incorporate an intuitive metaphor which brings the system closer to the user by meaningfully abstracting the information space. If profiles are used then the relationships between profiles and the user's relationship with his or her profile has to be clarified to the user. The Omavisio approach of assigning a profile does not work. The concept of a personalized service works only when the user can create a unique profile for him or herself.

The type of the information offered to the user regarding the choices in the information space is correlated with the efficient use of the system. For instance, along with the traditional grouping of videos according to genre, videos could be grouped according to duration, cast, language, etc. Thereby, it is recommended that Omavisio support multiple views to the same data. This entails augmenting interface functionality.

Interface functionality has to be redesigned so that user has the tools which allow control of the workspace. The main processes that the user needs are *browse* (to view the menus which contain the choice of services or videos), *search* (to find services or videos that match requirements), *select* (to fix a choice), and *view* (to see the video itself or a clip from it). These processes can be translated to the navigational procedures. Therefore, navigation has to be supported better in the user interface through such features as history lists as well as through effective communication of the information space structure to the user.

8.3.3 Communication Component

System modality should be avoided as much as possible by designing a conceptual model that allows the user to control the different applications with the same conceptual procedures. The benefit of a clear conceptual model is that it can be prescribed to all outside developers with the aim of embedding it in all the new services that are introduced to the Omavisio platform; thereby, ensuring interface consistency.

The recommendations made for the redesign of the conceptual component requires that the interface style and consequently, the input devices be modified. The responsiveness and user control of the interface need to be improved. Greater control can be afforded to the user only by employing the direct manipulation style in the interface. This requires that the Omavisio technical platform be upgraded and a new remote control be designed.

8.3.4 Physical Component

The following recommendations aim to augment apparent usability of Omavisio. These guidelines promote the ease with which the user distinguishes the relevant areas of the screen from the irrelevant areas.

If the technical platform cannot be upgraded then the interface layout needs to be redesigned in order to make the grouping and affordance of elements clearer to the user. The user must be presented with an interface that can be controlled intuitively with only the use of a remote control by tabbing the area selector using the arrow buttons. Therefore, hotspots should be easily distinguishable and preferable aligned along a margin to perceptually support the notion of a distinct group. Image maps should be avoided when designing WWW pages to be viewed with the Omavisio system because selecting embedded links in images is extremely difficult with the remote control.

The choice of font family, lettering type, is very important. Fine details on images are difficult to see because of the low resolution of television sets. In particular designers have to pay special attention to text because letters have fine details and thin lines. It is recommended that sans-serif fonts, no smaller than 18 pt, should be used when designing for television. Using anti-aliasing is recommended since it reduces flickering and adds to readability. When choosing colors for text, use light-colored text against dark-colored backgrounds; television audiences find it easier to read.

Since televisions interlace images, displaying all the odd lines on the screen first and then all the even lines, thin horizontal lines that are only on scan line high flicker. Therefore, the use of thin horizontal rules (denoted with the <HR> tag) should be avoided. Using anti-aliasing solves this problem and is an alternative to those designers who do not want to use thicker lines.

8.4 The Chosen Methods of Study

Finally, the last section of this chapter looks at the suitability of the methods used in this study. It was found that the methods used in this study were adequate and served the objectives of this study.

The usability testing methodology employed in this study has elucidated usability determinants of the Omavisio system. Furthermore, the classification of the test results according to the CLG framework has clarified the major problems in the interface. With the CLG Psychological View framework it has been possible to demonstrate to which degree users are able to comprehend the system. The Psychological View has allowed for the detection of the knowledge the users lack in order to operate the system skilfully. Furthermore, the framework has allowed the interface conceptual model to be described accurately. This description will be beneficial to designers in the next design round.

9. Conclusion & Suggestions for Future Research

Usability is especially important in consumer products. Interactive television is expected to be the successor of television, one of the most popular and usable consumer products of our time. The present study has studied the usability of Omavisio iTV. The usability attributes of the system under study were discovered by conducting a usability test and through analysis of the results with the CLG framework. The findings and recommendations of this study support existing Human-Computer Interaction theory and guidelines as well as introduce new ideas of what iTV usability is.

9.1 Omavisio Usability

The major finding of this thesis is that the Omavisio interface is not usable. The interface does not follow usability guidelines and therefore, the quality of use is negatively affected. User comprehension and learnability of the system is lowered by cognitive overhead which is the result of the user being forced to expend cognitive resources on basic interactivity. Users waste cognitive resources on their orientation in the workspace, on navigation and interpretation of system feedback. Furthermore, the use of the system is complicated by the users themselves because of the internal knowledge they transfer to the use of the system. The use of the PC model results in user expectations of the interface style not matching the style used in the system.
In sum, the user interface does not match user needs in a form that can be readily interpreted and manipulated.

To improve Omavisio usability designers first, need to come up with a conceptual model which lives up to user expectations. This can be done by creating a more advanced browsing environment for the user. The choice of interface style has to be made on the basis of user expectations. It is at this level of the interface that internal knowledge has the most affect on the overall satisfaction that the user gets from using the system. Then, designers should aim to improve user understanding of the workspace. Designers have to assist users in the construction of their mental models by strengthening factors that make the system image clear and by weakening those that don't. Use of landmarks and overviews are means of elucidating information space structure to the user. In addition, system feedback needs to be improved: the goal is for user actions to be in synch with system responses. The pace of interaction should be set by the user rather than the system.

While the above recommendations were based on findings which support existing Human-Computer Interaction knowledge, the following finding adds to the knowledge base of iTV usability. It was found that users see iTV as being more than a VOD system, where they have control of programming. Users draw a parallel between iTV and WWW browsers. They expect the system to support information processing: in addition to the basic functionality of browsing and pointing and selecting from menus, users want to be able to carry out goal oriented searching.

In conclusion, this study has elucidated the usability attributes which are applicable to Omavisio and iTV interfaces in general. These attributes are learnability, satisfaction, efficiency and memorability. Furthermore, this study lends support to the paradigm that the user of interactive system needs to be presented with a clear conceptual model in the user interface. It also supports knowledge about the abductive nature of direct manipulation interfaces. The study demonstrated the applicability of the CLG framework to the analysis of graphical user interfaces. Furthermore, this thesis can be considered a prime example of the usefulness of the concurrent verbalization method in determining usability attributes of novel systems.

9.2 Future Research

In the last section of this study ways to continue this work are briefly discussed. The thesis could be expanded in the following areas.

Affect of User Experience with PCs on the Usability Test Results

Test-users in the present thesis were not presented with specific tasks to accomplish. According to the information processing model proposed by Ericsson and Simon (1980) when users are not concerned with verbalizing a particular task they are expected to accomplish, basic type of recoding occurs. The basic type of recoding, Level 2 verbalization, involves the recoding of the subject's non-verbally encoded mental representation of the information into verbal form. This involves determining correct words and idiosyncratic referents; both processes are dependent on past experiences and knowledge. The user who is verbalizing has to find understandable referents to ensure the communication is comprehended. It would be interesting to find out if the test-user model of the assumed experimenter knowledge affected the internal knowledge they ended up transferring to the use of the system. Did test-users employ computer terminology to show experimenter that they were knowledgeable about interactive systems, or did they actually perceive similarities between Omavisio and PCs? Future work can also concentrate on the discovery of the degree to which the test-users PC influenced mental model, reinforced by the terminology users employed to describe their behaviour and system behaviour, affected their understanding of the system?

Alternative User Interface Technologies for Omavisio

It is likely that developments in the study of Web STB usability will be have potential for application in the domain of iTV. The two applications are converging and it is feasible iTV will actually be implemented through the Internet instead of proprietary broadband networks. Therefore, it is worthwhile studying how an iTV interface could be constructed using either HTML or VRML, both mark-up languages that are used to create WWW pages or worlds.

Information Processing Features of iTV

Since, iTV is an information appliance, the information manipulation and processing requirements of users need to be better assessed. In particular, what is the value of information to the user? Will iTV used for non-entertainment related tasks?

Direct Manipulation with IR Remote Control

The present study demonstrated that users relied more on the area selector than accelerators. This result cannot be explained by the fact that users were novices and that accelerators are only used by expert users because, accelerator use was emphasized in the system design. Therefore, a study should also be conducted on the requirements of a remote control driven direct manipulation interface.

Use of Sound in Usable iTV Interfaces

The use of sound in the interface has not been discussed in this thesis. Future research could concentrate on how sound can improve the usability of iTV and embedded systems in general.

References:

Arnum E. (1995), The Interactive Multimedia Services Market. http://www.sequent.com/news/releases/whpapers/imm/1109wh.html (7.5.1997).

Baecker R. M. and Buxton W. A. S., Readings in human-computer interaction. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1987, p. 1

Bailey G., Iterative Methodology and Designer Training in Human-Computer Interface Design. In: Human factors in computing systems / INTERCHI '93 (eds. Ashlund S., Henderson A., Hollnagel E., Mullet K. and White T.), IOS Press, Amsterdam, 1993, p. 198-205.

Bevan N., Kirakowski J. and Maissel J. (1995), What is usability? http://www.npl.co.uk/npl/sections/us/publications/index.html (8.8.1997).

Bevan N. (1995), Usability is Quality of Use. http://www.npl.co.uk/npl/sections/us/publications/index.html (8.8.1997).

Brooke J., Bevan N., Brigham F., Harker S. and Youmans D.,Usability Statements and Standardisation - work in progress in ISO. In Human-Computer Interaction - INTERACT'90 : Proceedings of the IFIP TC 13 Third International Conference on Human-Computer Interaction (eds. Diaper D., Cockton G., Gilmore D. and Shackel B.), Cambridge, U.K., August 27-31, 1990, Elsevier Science Publishers B.V., North-Holland, IFIP, 1990, p. 357-361.

Brouwer-Janse M.D., Interfaces for consumer products: "How to camouflage the computer?" In CHI '92 conference proceedings : Association for Computing Machinery (ACM) Conference on Human Factors in Computing Systems - Striking A Balance (eds. Bauersfeld P., Bennett J., and Lynch G.), May 3-7, 1992, Monterey, California, Addison-Wesley, Reading, MA, 1992, p. 287-290.

Butler K. A.Usability Engineering Turns 10. interactions, January, 1996, Association for Computing Machinery (ACM), 1996, p. 59-75.

Dillon A., McKnight C., and Richardson J., Navigation in Hypertext: A Critical Review of the Concept. In Human-Computer Interaction - INTERACT'90 : Proceedings of the IFIP TC 13 Third International Conference on Human-Computer Interaction (eds. Diaper D., Cockton G., Gilmore D. and Shackel B.), Cambridge, U.K., August 27-31, 1990, Elsevier Science Publishers B.V., North-Holland, IFIP, 1990, 587-592.

Ericsson A. K. and Simon H. A., Verbal Reports as Data. Psychological Review Volume 87 Number 3 May 1980, American Psychological Association, 1980, 215-251.

Green T. R. G, User Modelling: The Information Processing Perspective. In Human-Computer Interaction, Research Direction in Cognitive Science, European Perspectives Vol. 3 (Eds. Rasmussen J., Andersen, H.B., and Bernsen N.O.), Lawrence Erlbaum Associtates Ltd., Publishers, East Sussex, U.K., 1991, 27-47.

Habinek J. K., and Savage R. E., A Multilevel Menu-Driven User Interface: Design and Evaluation Through Simulation. In Human Factors in Computer Systems (Eds. Thomas J. C. and Schneider M. L.), Ablex Publishing Co., Norwood, New Jersey, 1984.

Hollnagel E., Much Ado About Nothing? In Human-Computer Interaction, Research Direction in Cognitive Science, European Perspectives Vol. 3 (Eds. Rasmussen J., Andersen, H.B., and Bernsen N.O.), Lawrence Erlbaum Associtates Ltd., Publishers, East Sussex, U.K., 1991, p. 198.

Hutchins E. L., Hollan J. D, and.Norman, D. A. Direct Manipulation Interfaces. In User Centred System Design (Eds. Norman D. A.and Draper S. W.), Lawrence Erlbaum Associates, Hillsdale, N.J, 1986, p. 118-123.

Krasilovsky P. (1995). ITV Testbeds. Communications Policy Working Paper #7. Benton Foundation.
http://www.benton.org/Library/Testbeds/working7.html (4.5.1997)

Kurosu M.and Kashimura K. (1995). Apparent Usability vs. Inherent Usability.
http://www.acm.org/sigchi95/Electronic/documents/shortppr/mk_bdy.html (18.2.1997).

Lewis R. J., Meeting Learner's Needs Through Telecommunications. Washington, D.C: American Association for Higher Education., 1983.

Lim Kai H., et. al., The Interactive Effects of Interface Style. ACM Transactions on Computer-Human Interaction, Vol.3, No. 1, March 1996, Addison-Wesley/ACM Press, 1996, p. 8-37.

McAdams M.. Information Design and the New Media. Interactions, October, 1995, Association for Computing Machinery (ACM), 1995, p. 39-46.

McLuhan, M. Understanding Media: The Extensions of Man. McGraw-Hill, New York, 1964.

Moran T. P. The Command Language Grammar: A Representation of the User Interface of Interactive Computer Systems. In International Journal of Man-Machine Studies (15), 1981, p. 3-50.

Mountford S. J., When TVs are Computers are TVs (Panel). In CHI '92 conference proceedings: Association for Computing Machinery (ACM) Conference on Human Factors in Computing Systems – Striking A Balance (eds. Bauersfeld P., Bennett J., and Lynch G.), May 3-7, 1992, Monterey, California, Addison-Wesley, Reading, MA, 1992, p. 27-230.

Nielsen J., Hypertext & Hypermedia. San Diego, CA: Academic Press, Inc, 1990.

Nielsen J., Usability Engineering. Boston, MA: Academic Press, Inc., 1993.

Nielsen (1996). Characteristics of Usability Problems Found by Heuristic Evaluation. http://www.useit.com/papers/heuristic/usability_problems.html (7.6.1997).

Norman, D.A., Some Observations on Mental Models. In Mental Models (Eds. Gentner D.and Stevens A.L.), Lawrence Erlbaum Associates, Hillsdale, N.J, 1983, p. 7-14.

Norman, D.A., Cognitive Engineering. In User Centred System Design (Eds. Norman D.A.and Draper S.W.), Lawrence Erlbaum Associates, Hillsdale, N.J, 1986, p. 31 – 62.

Norman D.A., Design of Everyday Things. Double Day, New York, 1988. [Originally published as The Psychology of Everyday Things].

Payne S., Looking HCI in the I. In Human-Computer Interaction – INTERACT'90 : Proceedings of the IFIP TC 13 Third International Conference on Human-Computer Interaction (eds. Diaper D., Cockton G., Gilmore D. and Shackel B.), Cambridge, U.K., August 27-31, 1990, Elsevier Science Publishers B.V., North-Holland, IFIP, 1990, p. 185-191.

Perlman G., Making the Right Choices with Menus. In Human-Computer Interaction – Interact '84, Amsterdam: North-Holland, 1985, p. 317-321.

Rada R., Interactive Media. Springer-Verlag, New York, 1995.

Rahmat O., Milking the Couch Potato. Interactivity, Vol. 1, No. 4. September/October, 1995, p. 53-66.

Richter J., Interactive Cable: History, Implications and Prospects. In The Interactive Cable TV Handbook (Ed. Kimmel, M. R.), Phillips Publishing Inc., Bethesda, MA, 1983.

Riley M., In User Centred System Design (Eds. Norman, D.A.and Draper S.W.), Lawrence Erlbaum Associates, Hillsdale, N.J, 1986, p. 157-160.

Shneiderman B., Direct Manipulation: A Step Beyond Programming Languages, IEEE Computer, Vol. 16(8), 1983, p. 57-69.

Sweeney J. (1995), An Introduction to iTV. http://www.ursley.ibm.com/misc/xw-itvintro.html (6.5.1997).

Vainio-Larsson A., Evaluating the Usability of User Interfaces: Research in Practice. In Human-Computer Interaction – INTERACT'90 : Proceedings of the IFIP TC 13 Third International Conference on Human-Computer Interaction (eds. Diaper D., Cockton G., Gilmore D. and Shackel B.), Cambridge, U.K., August 27-31, 1990, Elsevier Science Publishers B.V., North-Holland, IFIP, 1990, p. 323-328.

Van der Veer G. C., Human-computer interaction from the viewpoint of individual differences and human learning. In Human-Computer Interaction, Research Direction in Cognitive Science, European Perspectives Vol. 3 (Eds. Rasmussen J., Andersen, H. B., and Bernsen N. O.), Lawrence Erlbaum Associtates Ltd., Publishers, East Sussex, U.K., 1991, p. 59-93.

Wildman D., Getting the Most from Paired-User Testing. Interactions, July, 1995, p. 21-27

Wixon D., Qualitative Research Methods in Design and Development. Interactions, October, 1995, p. 19-27.

Young, R. M., Howes A. & Whittington J., A Knowledge Analysis of Interactivity. In Human-Computer Interaction - INTERACT'90 : Proceedings of the IFIP TC 13 Third International Conference on Human-Computer Interaction (eds. Diaper D., Cockton G., Gilmore D. and Shackel B.), Cambridge, U.K., August 27-31, 1990, Elsevier Science Publishers B.V., North-Holland, IFIP, 1990, p. 115-120.