

HELSINKI UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering

TUOMAS VAITTINEN

**Guideline-supported user-centered design of
multimodal speech-enabled TV-guide**

HELSINKI UNIVERSITY OF TECHNOLOGY	ABSTRACT OF MASTER'S THESIS
Author: Title of the thesis: Date: Number of pages:	Tuomas Vaittinen Guideline-supported user-centered design of multimodal speech-enabled TV-guide February 12, 2003 90 + 3
Department: Professorship:	Computer Science and Engineering Tik-121 Usability Research
Supervisor: Instructor:	Prof. Marko Nieminen Hannu Kuoppala, MA
<p>User interfaces that allow the user to communicate with the system using several modalities, such as voice, gesture, and typing with a keyboard, are called multimodal. A great deal of the research on multimodality concentrates on applications that combine speech recognition and some other input modes. Distinct features of speech recognition have a strong effect on how speech should be used in multimodal interfaces. As with any user interface, multimodal interfaces must be designed very carefully if any naturalness and ease of use is intended. User-centered design is a method that focuses on users throughout the design process. During a design process, a designer may take advantage of existing knowledge about good user interfaces, which is often summarized in design guidelines.</p> <p>The aim of this research was to collect experiences about the special considerations needed in user-centered design of multimodal interfaces. The focus was on speech-recognizing multimodal interfaces and on the role of guidelines in the design process. A list of guidelines relevant for interfaces that combine speech with WAP was gathered. A constructive approach was used, i.e. this thesis describes a design process of a multimodal TV-guide and it reports what was learned during the process. The design process was strongly inspired by Jakob Nielsen's model of a user-centered design process.</p> <p>This research indicated many useful special considerations. Each available modality should be analyzed to find the modalities that are best suited for each task. Some studies should also be made concerning how people talk about the task domain. In addition, traditional methods of usability testing must be adapted slightly for tests of speech interfaces. Guidelines proved to be useful in every phase of design.</p>	
Keywords:	user-centered design, design guidelines, multimodal interfaces, speech interfaces, WAP

Acknowledgements

The design process described in this thesis was a joint effort of several people. I want to thank Sirpa Autere, Johanna Laakko, and Marjut Kytösalmi from Elisa Communications as well as Pekka Kapanen, Suresh Chande, and Katriina Halonen from Nokia Research Center for a close collaboration in different phases of the process. I also want to thank other people participating CATCH-2004 project in different companies. Special thanks to Péter Boda from Nokia Research Center for organizing native speakers of English to participate in the usability tests.

I want to express my gratitude to my colleagues at Elisa Communications for the support and to my superior Kari Lehtinen for letting me choose the subject of the thesis so freely. My sincere thanks to all the people who took part in our studies as test users, diary study participants etc.

The help from my supervisor, Professor Marko Nieminen, has been vital in getting the work on this thesis started. With his support, I was able to create the first drafts of the thesis and find the title for it. Regular meetings with my instructor Hannu Kuoppala shaped this thesis into the form it currently has. Hannu has commented on numerous versions of this document and answered to hundreds of questions that have troubled me. Professor Roope Raisamo and researcher Markku Turunen from University of Tampere have also commented on several versions of this document and provided valuable help in finding material concerning multimodal interfaces. I highly appreciate the time you all have spent in guiding me.

I thank both of my parents for the support during this effort, especially my father who read and commented on an early version of this thesis. For tackling the peculiarities of English language, I have gotten invaluable help from Elizabeth Heap-Talvela and Janne Rovio. Finally, I thank Reetta for her incisive comments as well as for enduring me during this project, despite my constant absent-mindedness and occasional frustration. Without your support, I would not have had the strength to finish this work.

Helsinki, February 12, 2003

Tuomas Vaittinen

Contents

1. INTRODUCTION.....	1
1.1 Multimodal user interfaces.....	1
1.2 Design of multimodal user interfaces.....	1
1.3 Application constructed in this study.....	2
1.4 Relation to existing research.....	3
2. RESEARCH QUESTION.....	6
3. THEORIES AND RELATED RESEARCH.....	8
3.1 User-centered design process.....	8
3.1.1 ISO 13407.....	8
3.1.2 Usability engineering lifecycle.....	9
3.2 Guidelines.....	14
3.2.1 Voice-related guidelines.....	15
3.2.2 WAP guidelines.....	19
3.2.3 Voice and WAP in multimodal interfaces.....	22
3.3 Role of guidelines in usability engineering.....	24
4. REFINED GUIDELINES FOR INTERFACES WITH SPEECH AND WAP.....	26
4.1 Voice-related guidelines.....	26
4.2 WAP-related guidelines.....	30
4.3 Guidelines based on modality analysis.....	32
5. ITERATIVE GUIDELINE-SUPPORTED DESIGN.....	34
5.1 Diary study.....	37
5.1.1 Research subject.....	37
5.1.2 Method.....	38
5.1.3 Results and design suggestions.....	38
5.1.4 Discussion.....	39
5.2 Natural dialog study.....	41
5.2.1 Research subject.....	42
5.2.2 Method.....	42
5.2.3 Results and design suggestions.....	43
5.2.4 Discussion.....	43
5.3 Parallel design.....	45
5.3.1 Research subject.....	45
5.3.2 Method.....	45
5.3.3 Results.....	45
5.3.4 Discussion.....	51
5.4 Heuristic evaluation.....	53
5.4.1 Research subject.....	53
5.4.2 Method.....	57
5.4.3 Results and design suggestions.....	57
5.4.4 Discussion.....	61

5.5 First usability test	62
5.5.1 Research subject	63
5.5.2 Method.....	65
5.5.3 Results and design suggestions	65
5.5.4 Discussion	65
5.6 Second usability test.....	68
5.6.1 Research subject.....	68
5.6.2 Method.....	72
5.6.3 Results and design suggestions	73
5.6.4 Discussion	75
6. DISCUSSION	78
6.1 Special considerations in design process.....	78
6.2 Role of guidelines.....	81
6.3 Relevant guidelines for speech and WAP	81
6.4 Design process.....	81
6.5 Successfulness of our research	82
7. CONCLUSIONS.....	83
7.1 Research questions	83
7.2 Reliability and limitations of results	84
7.3 Future research	85
REFERENCES.....	86
APPENDICES	91
A Tasks used in first usability test.....	91
B Tasks used in second usability test	92
C Outline of interview in second usability test	93

Terms and abbreviations

direct-manipulation interfaces	interfaces that support visibility of objects and actions as well as allow easy manipulation of objects, rather than require complex command syntax
design guideline	recommendations for the designer that summarize existing knowledge of usable interfaces
grammar	when related to speech interfaces, a formal specification of language structure that defines what kind of utterances the system accepts
haptic input	input methods that involve physical contact with the computer, like joysticks, data gloves etc., often accompanied with haptic output, like force feedback
ISO	International Organization for Standardization
modality	a way of exchanging information between humans and computers, see a longer discussion in Chapter 2
multimodal interaction	interaction that comprises more than one input or more than one output modality and use more than one device on either side, see a longer discussion in Chapter 2
product concept	a rough description of the functionality and the form of a product that is created in the early phases of the design process
prompt	when related to speech interfaces, something that the computer says to the user
SMS	Short Message Service, enables the transmission of alphanumeric messages between mobile phones
soft button	a button in a WAP-enabled mobile phone whose behavior can be altered from the WAP service
taxonomy	a classification scheme
unimodal interaction	interaction that comprises only one input and only one output modality
utterance	when related to speech interfaces, something that the user says to the computer
vocabulary	when related to speech interfaces, a set of words that the system is able to understand

VoiceXML	Voice eXtensible Markup Language, a language for describing voice menus
WAP	Wireless Application Protocol, used in constructing services that resemble web pages but are simple enough to work in portable devices
WML	Wireless Markup Language, a language for describing the content and the user interface of WAP services
working memory	when related to human beings, a component of the human cognitive system that allows people to retain information about their immediate past experience

1. Introduction

1.1 Multimodal user interfaces

At the time of writing this thesis, people interact with computers and smart devices mostly using keyboard input, mouse input, and visual output, sometimes also pen input or audio output. These modes of interaction are drastically different from the ones people typically use when they interact with other people. Communication between humans relies mainly on spoken language, facial expressions, and body gestures. It is also very common that computers and smart devices offer only one way for each type of input, for example text may be input either using keyboard or pen. This can be problematic if the device is used in situations that differ a lot from each other, such as while sitting in an office and while walking on the street.

User interfaces that allow the user to communicate with the system using several modalities, such as voice, gesture, and typing with a keyboard, are called multimodal. Since communication between humans combines often at least speech and gestures, i.e. takes advantage of several modalities, multimodal systems are thought to be more natural than traditional computer systems. Multimodal interfaces also offer flexibility that can be very valuable when the device is used in changing environmental circumstances. The user may be unable to use a pen or a keyboard if he or she is carrying something with both hands, although speech would be unaffected (Oviatt et al. 2000). A multimodal interface may also help diverse user groups to get access to an application. A user with repetitive stress injury will probably prefer speech input, but a person with a strong accent might prefer pen input (Oviatt & Cohen 2000).

A great deal of the research on multimodality concentrates on applications that combine speech recognition and some other input modes. One reason for this might be that both multimodality and adding speech recognition capabilities to applications are seen as ways to improve the naturalness of the interaction. However, one should bear in mind that speech recognition applications or multimodal applications are not automatically natural to use. Applications must be designed very carefully if any naturalness of use is intended. In addition, the concept of naturalness is very vague, so a more precise definition of the objectives should be made before starting designing the application.

1.2 Design of multimodal user interfaces

The same design principles that apply to graphical user interfaces also apply to multimodal and speech-based interfaces (Mané et al. 1996). User-centered design and iterative cycles that include designing, testing as well as modifying are valuable in multimodal interface design (Mané et al. 1996). User-centered design is a design methodology that focuses on users of the future product throughout the design process. By concentrating on users, designers can produce applications that people really need and that are easy to use (Gould & Lewis 1985). User-centered design is especially valuable on projects where the product technology is new (Murray et al.

1997). In the design of novel products, one cannot rely on previous experience as often as in the design of more traditional applications.

The previous experience of user interface design is often formed into collections of design guidelines. Guidelines are recommendations that help the designer to create good user interfaces. Collecting and applying guidelines is one activity in most user-centered design processes. Availability of guidelines that take into account latest advances in the technology is limited because it takes time to gather experience and convert it into well-formed recommendations. Although there are not yet many guidelines for multimodal interfaces, there are several guidelines for e.g. graphical user interfaces and speech user interfaces. Guidelines are one way to achieve consistency in the interface and consistency helps users to learn the system rapidly. In speech interfaces, user's speech must coincide with the recognition capabilities of the system. Because this requires learning from users, guidelines might prove to be especially important for interfaces that involve speech.

The work described in this thesis is a part of a larger project that has developed a flexible architecture for multimodal applications and has demonstrated the possibilities of the architecture. Because there is not much literature about user-centered design of multimodal interfaces, we wanted to gather experiences of the process. We wanted to know if multimodality or speech modality creates any special considerations of which designers should be aware. We were also interested to see what kind of guidelines would be applicable for multimodal interfaces and what exactly would be their role in the design process. A constructive approach was chosen, i.e. we created a speech-recognizing multimodal application using a user-centered process and this thesis describes the process and the findings.

1.3 Application constructed in this study

We built a multimodal TV-guide for WAP-enabled mobile phones. Wireless Application Protocol¹ (WAP) is used in constructing services that resemble web pages but are simple enough to work in portable devices. In a normal interaction with a WAP service, people select options using buttons and visual menus. The service provides the information in text on a screen of the phone. In addition to these modalities, our multimodal application allows the usage of speech. The user can use either buttons or speech commands in finding the TV programs he or she is interested and the phone provides the information by speaking to the user and showing information on the screen. Users are expected to see the screen all the time. A unimodal application that does not require visual attention was also built in the project, so in the design of the multimodal application we wanted to concentrate on situations where the screen is available. Due to the challenges related to building

¹ WAP is a protocol that was defined by WAP Forum to allow web-like content to be delivered to small wireless devices. More information about WAP can be found from <http://www.wapforum.org/> [checked December 6, 2002].

speech recognizing mobile devices², we implemented only prototypes of such appliances in our project. The prototypes were not truly mobile. Instead, they displayed a picture of a mobile phone on the screen of a PC and the PC performed the speech recognition.

The design and the implementation of the application were done in the CATCH-2004 project³, which belongs to the 5th Framework Programme⁴ of the European Commission. This application was designed in close cooperation of Nokia and Elisa Communications. The effort of other partners was also vital, especially the work of IBM.

The product concept and the system architecture were not defined solely on the terms of this study and no user-centered methods were used in this phase. The decision that mainly information services were constructed in the project was made already before the project started. Since other applications were also built in the project using the same architecture, needs of the other applications affected the architectural decisions. The concept of a TV-guide was selected because it provided a real challenge for the speech recognition technology. Since the data of a TV-guide has to be updated regularly and since the database contains information in several languages, a lot of valuable experiences could be gathered by building this application. The selection of modalities reflected the interests of participating companies. Multimodality offers many potential benefits for mobile devices, like the possibility to choose the best modality for each situation. Speech is particularly interesting modality for small devices because keyboards take most of the size of many devices. The multimodal TV-guide and the architecture will be described in more detail in Chapter 5.

1.4 Relation to existing research

Many different methods of input and output have been researched in the literature of multimodal interfaces. In addition to the most common ones like direct manipulation and keyboard input, also for instance eye gaze tracking, lip movement tracking, speech, pen, and haptic input have been studied. On the output side, visual display

² Speech recognition requires a lot of processing power, which is difficult to squeeze in a small portable device with a small battery. The device might transport the speech to a more powerful server that would perform the recognition but that would require a simultaneous transfer of voice and data, which is not possible with current mobile phones.

³ Aim of the CATCH-2004 is to develop a multilingual and multimodal conversational system with a unified architecture across different client devices. On top of the architecture, partners will build several applications that demonstrate the possibilities of the architecture. Our multimodal TV-guide is one of these demonstrations. More information about CATCH-2004 project can be found at <http://www.catch2004.org/> [checked December 6, 2002].

⁴ Information about EC's 5th Framework Programme (FP5) is available at <http://www.cordis.lu/fp5/> [checked December 6, 2002]. FP5 consists of seven Specific Programmes and CATCH-2004 is a part of the program User-friendly information society.

has been traditionally the most dominant method, but also speech and haptic output have been explored. Lip movement tracking, speech, and some forms of pen input differ slightly from the traditional unambiguous input modes because they are based on recognition algorithms that can make errors. Algorithms can come up with the most probable interpretation of user input, but there is always a possibility that this interpretation is not the one the user intended. Pen input is unambiguous if it is used in selecting a coordinate pair on a screen, i.e. selecting for example an icon, but recognition algorithms are needed when a pen is used to input hand-written text or other gestures. The above list of recognition-based input modalities is not comprehensive, but it gives an idea of what type of input is based on recognition algorithms. Our application uses one recognition-based modality, i.e. speech.

Multimodal systems can be built so that a combination of two error-prone technologies works more robustly than a system that utilizes only one of the two input modes (Oviatt 1999a). For example, an application that combines speech and pen gesture input can be built so that it takes advantage of both voice recognition results and pen gesture recognition results when it makes a decision of what command the user gave (Oviatt 1999b). A redundant set of input modalities also takes advantage of people's natural intelligence about choosing less error-prone input modality. For instance, users will more likely write foreign names than speak them because pronouncing foreign names is considered potentially difficult (Oviatt & Olsen 1994). The technology we had available did not allow increasing robustness by combining simultaneous input from several modalities to a single command, but it allowed us to provide our users with a redundant set of modalities to choose from.

Nigay and Coutaz (1993) created a classification scheme for multimodal systems. Systems are classified according to how modalities are used and how the input from different modalities is combined. If most of the commands are formed from a simultaneous input of at least two modalities and the interpretation of the input is based on their combination, the system is synergistic. If, on the other hand, the user has a choice between multiple modalities to express each command, but only one modality is used to specify the command, the system is classified to be exclusive. In between these two extremes, Nigay and Coutaz (1993) place two more classes of multimodal systems, alternate and concurrent systems. The multimodal TV-guide presented in this thesis belongs undoubtedly to exclusive class of multimodal systems because the user has a choice between voice and buttons for input, but uses only one of them to give any single command.

Our multimodal TV-guide does not take advantage of the most advanced techniques developed in the multimodal research community. It provides somewhat simple version of multimodal interaction with a TV-guide, but our focus has been more on the user-centered design process and on the guidelines that can be taken advantage of in the design process.⁵ Mané et al. (1996) stated that multimodal systems should be

⁵ We hoped that more advanced multimodal interaction techniques would have been available for us in later phases of the project, but the implementation of the architecture did not proceed fast enough.

designed using user-centered methods, but very few examples of such projects exist.⁶ This is most likely due to the fact that multimodal systems have been designed mainly for research purposes. Real commercial systems that have end users do not yet exist. In the absence of examples of multimodal systems that had been created with user-centered methods, our process has been based on well-known user-centered process from Jakob Nielsen (1992, 1993). His process model, usability engineering lifecycle, is not specific to any type of user interface, so we were prepared to adapt the model slightly when needed.

⁶ One example of such a project might be the Magic Lounge described by Bernsen and Dybkjær (2001). It combines participatory design and some user studies. More information about the project can be found from <http://www.dfki.de/imedia/mlounge/> [checked December 30, 2002]

2. Research question

This work studies user-centered design of multimodal interfaces. The focus is on those multimodal interfaces that take advantage of speech recognition. By designing and implementing a multimodal TV-guide, experiences are gathered from special considerations related to speech-enabled multimodal interfaces. User interface guidelines play a role in user-centered design and the exact nature of this role is studied both in theory and in concrete case study. A list of useful guidelines related to building similar systems is also created.

The research questions that this work seeks to answer are the following.

- What special considerations are related to the user-centered design of speech-enabled multimodal user interfaces?
- What is the role of user interface guidelines in the user-centered design process?
- Which guidelines are applicable to user interfaces that utilize speech recognition and WAP?

Because words 'multimodal' and 'guideline' are used in varying meanings in the literature, both terms are defined below. Before defining 'multimodal', two additional terms 'modality' and 'medium' need to be explained.

In psychology, sensory modality refers to different human senses e.g. hearing and vision (Reber 1985). Some of the researchers who are concentrating on multimodal systems have a slightly different view than the psychologists. Still, even the multimodal theorists lack full consensus. In the following, several views are described. Niels Ole Bernsen (2002) makes a distinction between medium and modality. A medium is the physical realization of some presentation of information. His definition makes medium closely related to sensory modalities i.e. the graphical medium is what people and systems see and the acoustic medium is what people and systems hear. For example, both text and images are graphical media. Since one has to be able to distinguish between text and images, there is a need for another term. Bernsen (2002) uses notion of representational modality and defines it as a way of exchanging information between humans and machines in some medium. Text and graphical images are different representational modalities, although they are both graphical media.

Literally, multimodal interaction is an interaction where participants communicate using more than one modality. One could argue that almost every interaction with a computer is multimodal because one input and one output modality are together already two modalities and furthermore e.g. text and images are different representational modalities. There are several ways to separate multimodal interaction from the interaction with traditional applications. Schomaker et al. (1995) highlight that multimodal interactions must comprise either more than one input or more than one output sensory modality and use more than one device on either side. When multimodal systems and multimedia are compared, another aspect is often emphasized. Both multimedia and multimodal systems use several modalities, but in addition, a multimodal system is automatically able to model the content of the

information at a high level of abstraction (Nigay & Coutaz 1993). If a system must offer the same functionalities with very different input modalities, it is beneficial that the system handles the input as much as it is possible in a modality-independent way. This requires modeling the content at a high level of abstraction. The same need is apparent when the system must present the information in several output modalities. In this thesis, the view of Schomaker et al. (1995) is adopted.

User interface guidelines are generally stated recommendations for user interfaces (Smith 1988). Usually they are accompanied with examples and explanation (Smith 1988). A style guide is a collection of very specific instructions which helps in keeping user interfaces developed by a certain company consistent and usable (Sinkkonen 1996). Some authors call style guides guidelines but, in this thesis, guidelines are seen as general recommendations applying to large group of interfaces, whereas style guides are considered more specific and developed for use of one company only. Guidelines help in achieving consistency, but specificity of a style guide makes it more suitable for assuring consistency of interfaces. Style guides are also sometimes called corporate user interface standards but, in this thesis, standards mean official standards. They are generally stated requirements imposed in a formal way, for example by legislation (Smith 1988). Several national and international bodies have produced standards for user interfaces, like Deutsches Institut für Normung (DIN) and International Organization for Standardization (ISO).

3. Theories and related research

In this chapter, the literature and theories that are most relevant for this thesis are reviewed. There exists several models for user-centered design processes and two of them are described first. Application of guidelines is one important activity in user-centered design and of special interest for this work, so guidelines are reviewed in detail. Special emphasis is placed on speech-related guidelines and WAP-related guidelines, which were important in the design of the multimodal TV-guide. The status of multimodality-related guidelines is considered next and, last, the role of guidelines in user-centered design is discussed.

3.1 User-centered design process

In user-centered design, user feedback is gathered throughout the process and it is taken advantage of in making design decisions. Users are involved in all stages of a design process. Several models of a user-centered design process can be found in the literature and they differ somewhat in the number of phases and actions suggested in each phase. Two models are presented in this chapter. First, an international standard called Human-centred design processes for interactive systems is outlined and then a well-known model, usability engineering lifecycle, from Jakob Nielsen is described in detail.

3.1.1 ISO 13407

The International Organization for Standardization (ISO) has created a standard that is titled ISO 13407: Human-centred design processes for interactive systems. It provides guidance on managing human-centered activities throughout a design process and it is intended mainly for project managers. ISO 13407 divides a design process into four main activities:

- Understand and specify the context of use
- Specify the user and organizational requirements
- Produce design solutions
- Evaluate designs against requirements

The context of use that needs to be understood consists of the characteristics of the intended users, the tasks the users are to perform and the environment in which the users are to use the system. The information about the context should be collected to a document and it should be kept up-to-date during the design process. (ISO 13407)

The user and organizational requirements should be documented along with the functional requirements. They should provide a clear statement of the human-centered design goals and provide measurable criteria, so the design can be tested later against these criteria. It is furthermore important that the requirements are confirmed by the users. (ISO 13407)

Preliminary design solutions can be produced based on existing knowledge about ergonomics, psychology and product design. These solutions should be concretized with prototypes and be presented to users. The prototypes can be paper-based in the early phases of the process but later they should be more realistic. Users should be allowed to use the prototypes and carry out tasks so the design team can observe the difficulties users may have with the system. The design team should gather comments, which they can use in refining the system. Several rounds of iteration provide maximum benefits. (ISO 13407)

Evaluation provides feedback that can be used to improve the design. Feedback helps to diagnose potential problems and improve the interface. Evaluation also allows the design team to judge if the design meets the requirements specified earlier. It is useful to evaluate the final product, in addition to the prototypes. Comments should be gathered from real users of the final system. Help desk and performance data should also be analyzed. (ISO 13407)

3.1.2 Usability engineering lifecycle

Usability engineering presented by Nielsen (1992, 1993) is a set of activities that take place throughout the lifecycle of the product. Many valuable but affordable studies can be made already before any version of the user interface is designed. Nielsen (1992, 1993) divides the usability engineering lifecycle into three phases and furthermore into 11 activities⁷. The activities in the pre-design phase are:

- Know the user
- Competitive analysis
- Setting usability goals

In the design phase, the activities are:

- Parallel design
- Participatory design
- Coordinated design of the total interface
- Apply guidelines and heuristic analysis
- Prototyping
- Empirical testing
- Iterative design

Only activity left in the post-design phase is:

- Collect feedback from field use

⁷ In his article in the IEEE Computer, Nielsen (1992) presents the division of activities to three main phases. In his later book, Nielsen (1993) adds one activity, parallel design, to the lifecycle but does not provide the division to phases. It is nevertheless obvious that parallel design is a part of the design phase.

Although Nielsen (1992) presents the activities sequentially, he stresses that iteration between the activities inside each phase will be needed. Empirical testing might provide suggestions to improve prototypes and a need to apply new guidelines etc. In addition to the activities mentioned above, Nielsen (1992) suggests some meta-methods that should be used throughout the process.

Each of the stages of the model will be explained below.

Know the user

First, developers must study the intended users and the use of the product. They should at least visit the site of their customers so that they are able to get some feel of how the product will be used. An effort should be made to get direct access to real users, not just users' managers. Information about user characteristics and their work environment helps the designer in many ways. Users' work experience, educational level, and previous computer experience help in predicting learning difficulties and setting limits for complexity of the user interface. Knowing, for example, that users will be using the product in an open office environment may lead to a decision not to rely on sound effects in the interface. (Nielsen 1993)

A task analysis is another activity that should be carried out in this phase. Users' overall goals should be studied as well as how they currently approach the task, what their information needs are, and how they deal with exceptional circumstances. Users' model of the task should also be identified since it can be used as a source for metaphors for the user interface. A task analysis provides a list of things users want to accomplish with the system. It further provides knowledge of the steps that they need to perform and the interdependencies between these steps. Their information and communication needs should also be monitored since the user interface should support these needs, if possible. Because some of the features found in the task analysis might be due to limitations in previous technologies, one should analyze what users really want to do. In addition, the designer should take into account that users' abilities with the system will improve after they have used the system for some time. The system might support this process by offering some form of interaction shortcuts. (Nielsen 1993)

Competitive analysis

Existing products can be analyzed heuristically or with user tests. Since the product is already on the market, it is fully implemented. Users can perform realistic tasks with it, making it possible to analyze how the chosen approaches work with real users. Sometimes it might even be valuable to analyze how people use some non-computer interface for the same task that the planned computer product will be used. (Nielsen 1992, Nielsen 1993)

Setting usability goals

Different usability parameters can be expressed in measurable ways so the designer can set usability-related goals for the product. Examples of such parameters might be frequency of user errors and subjective satisfaction. One should at least define the acceptable level for each parameter that allows the release of the product, but even more complex meters can be specified. Usability goals are reasonably easy to set for new versions of existing systems or for systems that have a clearly defined competitor on the market. A possible goal could be e.g. a sufficiently large improvement that induces the users to change the system. For brand new systems without clear competitors, it is more challenging to find appropriate usability goals. (Nielsen 1992, Nielsen 1993)

Parallel design

It is often worthwhile to start the design work so that several different designers work a few hours independently and produce alternative preliminary designs. From the alternative designs, one can combine the best ideas to produce a draft that is pursued further. This is most important for novel systems since very little guidance is available for what interface approaches work best. (Nielsen 1993)

Participatory design

The early studies made during the "Know the user" activity will not provide answers to all questions arising in the course of the design process. Developers' model of the users' task might differ slightly from the actual task, which the users can point out if they are involved in the process through regular meetings with the designers. Users cannot normally come up with design ideas from scratch, but if suggested designs will be presented to users in the form of prototypes they can react to flaws in the design easily. (Nielsen 1993)

Coordinated design of the total interface

Consistency should be maintained across the total interface, like the application itself, the documentation, and the training classes. Consistency should be maintained even across entire product families. Corporate user interface style guides are one way of promoting this goal. Some sort of flexibility should be allowed, however, so that bad designs are not forced for the sake of consistency. One person or a committee should coordinate the various aspects of the interface. (Nielsen 1993)

Apply guidelines and heuristic analysis

Guidelines are well-known design principles for user interface design. They can be presented in a hierarchical fashion: general guidelines, category-specific guidelines, and product-specific guidelines. General guidelines apply to all user interfaces, but

category-specific guidelines are applicable to a certain category of interfaces, like a particular type of speech interfaces. Product-specific guidelines are even further narrowed to concern only the product being designed. Many guidelines can be found in the research literature, but often companies also produce guidelines for only their own use. (Nielsen 1993)

Because of the importance for this work, this stage will be discussed further in Chapters 3.2 and 3.3.

Prototyping

Creating a full-scale implementation based on early ideas about the interface would not be wise. The first design will not be flawless. The idea behind prototyping is to save time and costs and develop something easier, which is realistic enough so that it can be tested with real users. By testing the prototypes, the designer can find the problems in his or her designs. Basically, there are two types of prototypes. A vertical prototype is one with fewer features than the final system, but the features selected to be implemented work the way they are supposed to work in the final system. A horizontal prototype shows all the features of the user interface, but there is no underlying functionality. These approaches can be combined so that the designer reduces both the number of features and the level of functionality. (Nielsen 1992, Nielsen 1993)

Empirical testing

Some evaluation should always be done and it should be based on usability tests where real users use the system. The result of empirical testing will be a list of usability problems and hints for features that would support users while using the application. Because it is normally not feasible to implement all the solutions, prioritization is needed. Prioritization should be done by counting the averages of the ratings that several usability specialists have given for each problem. One should not trust on prioritization done by one person because opinions about a severity of a problem vary a lot. The severity of a problem depends on at least the number of users who encountered the problem as well as the impact of the problem on the users who encountered it. (Nielsen 1993)

Iterative design

Based on the results of the empirical evaluations, a new prototype of the interface can be created. It should then be tested to find the remaining problems, so an even better version can be produced etc. Because some of these iterations might not provide improvement, resources should be reserved for at least three rounds of iteration. Minor problems might not even be found before larger problems have been fixed. In some cases, it is not feasible to test each successive version with actual users. A heuristic analysis can substitute for a user test in some rounds of iteration. If

a design rationale is captured in a document, it helps decision making when changes are made to the interface later. (Nielsen 1992, Nielsen 1993)

Collect feedback from the field use

After the product is released, one should collect usability data for the next version of the product. In this phase, it is possible to study how real users use the interface for naturally occurring tasks in their real-world environment, which is almost impossible to study in a laboratory. Variety of follow-up studies can be made. Examples include marketing studies that gather customer satisfaction data and analysis of help line calls and user complaints. (Nielsen 1993)

Meta-methods

Some methods apply to all activities in the usability engineering lifecycle. For each activity, one should write down an explicit plan for what to do. This plan should be reviewed by a person who is not part of the team that designs the product. All studies should be started with a pilot activity, where only small amount of effort is spent. This helps in finding problems in the test setup and the problems can be fixed before the actual study. (Nielsen 1993)

Prioritizing usability activities

In a real-life project, it is sometimes not possible to carry out all the activities belonging to the usability engineering lifecycle. Nielsen (1992) surveyed usability specialists and found that the six methods that were considered having greatest impact were:

- Iterative design
- Task analysis of user's current task
- Empirical tests with real users
- Participatory design
- Visit to customer site before start of design
- Field study to find out how the system is actually used after installation

Nielsen's (1993) own, somewhat different, suggestion includes:

- Visits to user sites
- Light version of prototyping
- Simple usability tests
- Heuristic evaluation

3.2 Guidelines

As noted in the previous chapter, guidelines play a valuable role in user-centered design. In this chapter, benefits of the guidelines are discussed in more detail and a few examples of guidelines are presented. This chapter pays special attention to some lists of modality-specific guidelines that were relevant to the multimodal application described in this thesis.

The aim of user interface guidelines is to help the designer to develop usable and consistent applications. Guidelines support him or her in focusing attention on agreed design objectives. Guidelines are largely based on expert judgment and accumulated practical experience. Experimental data can also be used in deriving guidelines, but the researcher needs to use his or her judgment when he or she attempts to make a general statement based on a particular study. Guidelines should be written generally enough to apply to variety of systems, but the designers might translate the guidelines to more specific design rules, which apply only to a specific application, before starting the design. This reduces the amount of interpretation needed during the design. In the beginning of a translation work, some guidelines might be discarded because of their irrelevance for this particular application. To help the designer to interpret the guidelines, the guideline author should include some examples and discussion about the rationale of each guideline. Even with the aid of examples, designers may still find it quite difficult to follow a long list of guidelines. Involvement of users is still needed to get real information about usability of the designed application. One should not consider guidelines unchangeable. They record what we currently have agreed upon user interface design, but as we learn more, the guidelines will change accordingly. (Smith 1988, Tetzlaff & Schwartz 1991)

The relation of user-centered design and guideline-supported design will be discussed further in Chapter 3.3. One well-known set of guidelines was compiled by Rolf Molich and Nielsen (1990). They call their guidelines usability heuristics and, after analyzing a large set of usability problems, Nielsen (1994a, 1994c) refined their heuristics to the following list⁸:

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

⁸ Nielsen (1994c) also provides explanation of each heuristic, but those are omitted here.

Another set of general user interface guidelines is the eight golden rules of dialog design by Ben Shneiderman (1987). They are the following.

- Strive for consistency
- Enable frequent users to use shortcuts
- Offer informative feedback
- Design dialogs to yield closure
- Offer simple error handling
- Permit easy reversal of actions
- Support internal locus of control
- Reduce short-term memory load

Shneiderman (1987) also provides many specific guidelines and principles for e.g. direct manipulation interfaces and menu selection. Both Shneiderman's golden rules and Nielsen's heuristics are examples of lists that are kept reasonably short, so the designer can keep the guidelines in mind while designing or evaluating the system. An example of another extreme is the extensive guideline document compiled by Smith and Mosier (1986) that includes hundreds of guidelines.

3.2.1 Voice-related guidelines

There are many distinct features of speech recognition that affect how speech should be used in multimodal interfaces. The speech input is based on recognition algorithms and, hence, prone to errors. Recognition algorithms are also very sensitive to any noise in the vicinity of the microphone that is used to capture the speech.

The number of words the recognizer must be able to handle affects how accurately it is able to recognize the utterances. The smaller the vocabulary is the more accurate the system is (Schmandt 1993). The type of training is also a factor. Recognizers that are trained with each user's speech are called speaker-dependent and they work more accurately than the speaker-independent systems, which work without any training by the user (Schmandt 1993). The tasks that require a very large vocabulary, like dictating a letter, therefore call for a recognizer that can be trained by each user, whereas applications that are not feasible if each user has to train the system first, like ticket reservation, call for design that works with a limited vocabulary.

Although it is very natural for humans to use speech for communicating with other humans, most computer systems do not allow users to use natural spontaneous language. Automated systems do not have all the capabilities of a human listener (Boyce & Gorin 1996). Very often spoken language systems require users to employ restricted language that is easier for the computer to understand. The vocabulary is often restricted and the topics that the computer can handle are usually limited to some predefined set. It is also common that users are required to use fairly short sentences.

Some features of speech output, which are equally present in spoken conversation between two humans as well as between a human and a computer, are also important. Speech is temporal, i.e. if the listener was inattentive when the message was spoken the information is lost. Speech is public, so there are several situations where it is

undesirable that sensitive messages are voiced. Speech is furthermore slow compared to reading, so interaction can be quite tedious if a lot of information is presented to the user using speech. (Schmandt 1993)

Because of the many distinct features of speech as a user interface component, unspecialized user interface guidelines, like the ones described on page 14, are too general to support speech interface design. This has led several authors to create specific speech interface guidelines. Some of them have used guidelines for graphical user interfaces as their starting point and adapted them to speech interfaces. Some others have developed their design principles experimentally by analyzing interaction problems between users and spoken dialog systems. In the following, three lists of voice-related guidelines are examined more carefully. The decision to select these three lists for further examination was based on the clear intent of the authors to find design principles that apply to most speech recognition systems. More informally presented design tips were left out from this discussion but some of the most interesting ones are described in Chapter 4.

Shneiderman (1987) has identified three essential design principles for direct manipulation interfaces. Candace Kamm and Marilyn Walker (1997) have adapted them to several guidelines for speech interface design. Shneiderman's (1987) principles are:

- Continuous representation of objects and actions of interest
- Rapid incremental reversible operations whose impact on the object of interest is immediately visible
- Physical actions or labeled button presses instead of complex syntax

Since spoken language interfaces have to address the same limitations of human cognitive system as any other user interface, there should be a speech equivalent for each of Shneiderman's principles. Literally, continuous representation of all options in the spoken dialog would mean continuous repeating of options, which is not desirable, especially if the list is long. More feasible adaptation of this principle is to provide prompts in two phases. In the first phase, the user is provided a question and only if the user cannot answer it the second phase is spoken, where the options are listed. A frequent user will know the options beforehand, so he or she will answer the question immediately. If the user stays quiet for some seconds after the question, the system can interpret that the user is a novice who needs to hear the options. Assuring consistency across the features of the application is another way to support a sense of continuous representation in a voice-only interface. (Kamm & Walker 1997)

The principle of immediate observable impact is taken into account if the system responds to the user immediately and informs that the user's request was understood. It also requires that the system has a low latency and that it allows the user to interrupt the prompt of the system. Incrementality in the speech interface means that, if the user does not provide sufficient information in his request, the system is able to make a clarifying question to the user. If the user is not aware of all the details he or she should provide in a request, he or she can state them incrementally with the aid of clarifying questions from the system. Because of the limited capacity of a human auditory memory, an observable impact also requires summarization when the system has to present a lot of information. The principle of preferring physical

actions and labeled buttons instead of a complex syntax does not have direct speech-related equivalent, although one could say that a complex syntax is not desirable in speech interfaces either. (Kamm & Walker 1997)

The guidelines proposed by Kamm and Walker (1997) are summarized in Table 3.1. The guidelines are numbered, allowing easier reference.

Table 3.1 Voice-related guidelines from Kamm and Walker (1997). 'KW' before each number stands for the authors to distinguish them from subsequent guidelines.

No.	Guideline
KW1	The designer should use question - pause - options format i.e. he should provide the available options for the user if the user has not responded to the system after few seconds.
KW2	The designer should provide consistency across features in a multi-featured application i.e. he should use a subset of the vocabulary that is always available and that serves the same function regardless of where the user is.
KW3	The system should respond to the user immediately and inform that his request was understood and the system is acting to fulfill the request.
KW4	The system must have a low latency and it should allow interruption of its prompts when possible, so the natural flow of the dialog is preserved.
KW5	If the user does not give sufficient information, the system must make clarifying questions to the user.
KW6	The system should provide summarization when there is a lot of information to present to the user.

Bensen et al. (1998) have compiled another set of voice-related guidelines. They performed series of user tests with a simulated system, where a human operator acted as a speech recognition system. The operator chose the prompts the system presented, so no recognition errors occurred. They validated the guidelines derived from experimental data by comparing them to the literature about human-human conversation. As the last phase, they performed user tests with a real speech recognition system and made some refinements to their guidelines. Their guidelines are summarized in Table 3.2. The authors also include over 30 pages of explanation and examples to help in applying the guidelines, but that discussion is omitted here.

Table 3.2 Voice-related guidelines from Bernsen et al (1998). Authors present their guidelines in a hierarchical fashion, e.g. Be2 and Be3 highlight some specific aspects of Be1. The specific guidelines are marked with a text 'Specifies...'. 'Be' before each number stands for the authors to distinguish them from the previous as well as subsequent guidelines.

No.	Guideline
Be1	Make your contribution as informative as is required (for the current purposes of the exchange).
Be2	Be fully explicit in communicating to users the commitments they have made. <i>[Specifies Be1]</i>
Be3	Provide feedback on each piece of information provided by the user. <i>[Specifies Be1]</i>
Be4	Do not make your contribution more informative than is required.
Be5	Do not say what you believe to be false.
Be6	Do not say that for which you lack adequate evidence.
Be7	Be relevant, i.e. be appropriate to the immediate needs at each stage of the transaction.
Be8	Avoid obscurity of expression.
Be9	Avoid ambiguity.
Be10	Provide same formulation of the same question (or address) to users everywhere in the system's interaction turns. <i>[Specifies Be9]</i>
Be11	Be brief (avoid unnecessary prolixity).
Be12	Be orderly.
Be13	Inform the users of important non-normal characteristics which they should take into account in order to behave cooperatively in spoken interaction. Ensure the feasibility of what is required of them.
Be14	Provide clear and comprehensible communication of what the system can and cannot do. <i>[Specifies Be13]</i>
Be15	Provide clear and sufficient instructions to users on how to interact with the system. <i>[Specifies Be13]</i>
Be16	Take partners' relevant background knowledge into account.
Be17	Take into account possible (and possibly erroneous) user inferences by analogy from related task domains. <i>[Specifies Be16]</i>
Be18	Separate whenever possible between the needs of novice and expert users (user-adaptive interaction). <i>[Specifies Be16]</i>
Be19	Take into account legitimate partner expectations as to your own background knowledge.
Be20	Provide sufficient task domain knowledge and inference. <i>[Specifies Be19]</i>
Be21	Enable repair or clarification meta-communication in case of communication failure.
Be22	Initiate repair meta-communication if system understanding has failed. <i>[Specifies Be21]</i>
Be23	Initiate clarification meta-communication in case of inconsistent user input. <i>[Specifies Be21]</i>

Be24	Initiate clarification meta-communication in case of ambiguous user input. [Specifies Be21]
------	--

In the report from CHI'96 workshop, Mané et al. (1996) list speech user interface guidelines that Caroline Henton has used in her projects and which she presented in the workshop. The report does not include examples or justification of the guidelines. The guidelines are listed in Table 3.3.

Table 3.3 Voice-related guidelines from Mané et al. (1996). 'M' before each number stands for the authors to distinguish them from the previous and subsequent guidelines.

No.	Guideline
M1	Typical scenarios should be painless.
M2	Assume the user does not know the active vocabulary and guide him towards responses that maximize clarity.
M3	Allow for the user not knowing the answer to a question or not understanding a question.
M4	Supply confirmation messages frequently, especially when the cost or likelihood of a recognition error is high.
M5	Assume errors are the fault of the recognizer, not the user.
M6	Design graceful recovery when the recognizer makes an error.
M7	Assume a frequent user will have a rapid learning curve.
M8	Guide users toward natural in-vocabulary responses.
M9	Do not give too many options at once.
M10	Keep syntax and semantics consistent across all prompts in the system.
M11	Keep prompts brief to encourage the user to be brief.
M12	Avoid prompts that are too similar.

3.2.2 WAP guidelines

Wireless Application Protocol⁹ (WAP) was created to allow building of services that resemble web pages but are simple enough to be presented in small portable devices. WAP relies on a deck of cards metaphor, which means that information is organized into a collection of cards that are grouped together into decks. Since WAP-enabled devices have limited-size screens for output and usually either a small keypad or a pen for input, several factors have to be taken into account when designing WAP applications. Need for user interaction must be minimized by designing content specifically for WAP (Schmidt et al. 2001). In addition to the characteristics of input

⁹ More information about WAP is available at <http://www.wapforum.org/> [checked December 6, 2002].

and output mechanisms, the mobility of the device is an important aspect that has to be considered. People seldom use WAP applications in quiet office environments so they may experience many interfering factors that distract their attention (Grimstad et al. 2000).

Grimstad et al. (2000) wrote a guideline document originally for the designers of the WAP portal of Telenor Mobil. They made their document publicly available after they found that other designers were also interested in their guidelines. The authors derived a couple of guidelines based on the characteristics of WAP terminals. Because the user has a limited area to use for orientation and navigation, the application has to be more structured than web applications. The tasks people carry out with WAP-enabled devices are often related to solving problems rather than browsing and passing time. For this reason, the designer needs to put a lot of effort into investigating which tasks are most important for the intended users. The authors also present a large number of more specific guidelines. They are omitted from this thesis to keep the discussion in a relatively general level. The general guidelines from Grimstad et al. (2000) are summarized in Table 3.4.

Table 3.4 WAP-related guidelines from Grimstad et al. (1996). 'G' before each number stands for the authors to distinguish them from the previous and subsequent guidelines.

No.	Guideline
G1	Design WAP applications more structured than web applications.
G2	Make sure that the user is able to solve the tasks they have. To realize this, one normally needs to put lot of effort into investigating which tasks are the most important for the intended users.

Buchanan et al. (2001) derived their own set of guidelines for WAP service providers. Their guidelines are based on usability tests, user questionnaires and case-study work with several WAP applications. The aim of the authors is to show that by employing a user-centered approach current and future WAP services can be made more effective and useful. The guidelines from Buchanan et al. (2001) are listed in Table 3.5.

Schmidt et al. (2001) derived a set of guidelines from their experience in developing WAP applications. They group their guidelines in general ones (S1), ones related to input design (S2 - S7) and ones related to output design (S8 - S12). Their guidelines are listed in Table 3.6.

Table 3.5 WAP-related guidelines from Buchanan et al. (2001). 'Bu' before each number stands for the authors to distinguish them from the previous and subsequent guidelines.

No.	Guideline
Bu1	Develop phone-based WAP services that provide direct, simple access to focused valuable content. Usable and useful WAP services on phones will be the ones that give the user key, summarized information with very few keystrokes or text entry.
Bu2	Trim the page navigation down to a minimum; use simple hierarchies, which are similar to the phone menus that users are already familiar with.
Bu3	Reduce the amount of vertical scrolling to a minimum by simplifying the text you wish to display (avoid wordy messages; go for action oriented keywords).
Bu4	Reduce the number of keystrokes you expect the user to do. You can do this by simplifying navigation and by replacing text input with other types of interaction method (e.g. list selection).
Bu5	Combine theoretical and empirical evaluation to provide further insights.

Table 3.6 WAP-related guidelines from Schmidt et al. (2001). 'S' before each number stands for the authors to distinguish them from the previous guidelines.

No.	Guideline
S1	Identify the benefits for bringing an application to the WAP platform. Supporting these benefits must be the primary goal of the application development.
S2	Use numbers for input whenever possible.
S3	Use common abbreviations like country codes.
S4	Keep the input mechanism in mind if letters are used e.g. prefer first letter on key.
S5	Offer choices (e.g. numbers, list boxes, radio buttons, link lists) or default values when applicable.
S6	Provide labels to hardware buttons where possible.
S7	Use standard conventions on buttons (e.g. back).
S8	Assess the screen size and quality of the target devices with text and graphics.
S9	Reduce the output by customizing to the users need.
S10	Design information chunks that are seen at once on the screen. Larger text blocks (more than 20 words) within on card should be structured.
S11	Horizontal scrolling is easy on most devices, vertical is not.
S12	Use multiple cards in one deck instead of very large cards or multiple decks.

The companies that have built WAP browsers, like Nokia¹⁰ and Openwave¹¹ provide extensive lists of design guidelines for their WAP browsers. Since these guidelines go into very small details and are partly browser-specific, they are not considered here.

3.2.3 Voice and WAP in multimodal interfaces

In the literature, no guidelines exist yet for multimodal interface design in general (Raisamo 1999). One explanation for this is the fact that each modality has unique characteristics. Hence, each combination of modalities has unique features as well. It is very time-consuming, probably impossible, to find guidelines for each possible modality combination. Despite the vastness of the problem of multimodal interface design, some researchers have sketched ideas concerning the issue. Their ideas are not in the form of guidelines, but they are nevertheless developed to help the designer of multimodal systems, so they are reviewed here shortly.

Bernsen (2002) has created a taxonomy of output representational modalities. On the generic level, he sees 20 different output modalities, for example static analog graphics. These can furthermore be divided into atomic modalities, like static graphic images and static graphic maps. His taxonomy provides a basis for the research on characteristics of each representational modality and furthermore on combinations of several modalities. Bernsen and Verjans (1997) suggested a methodology where the designer would use particular information mapping rules to map the requirement specification of an application to a rough interface sketch. The huge number of possible modality combinations and the importance of a usage context led Bernsen (2002) to abandon this methodology. To help the designer to decide whether he or she should choose to use speech in the interface of some application, Bernsen and Luz (1999) studied the relevant properties of the speech modality and created a hypertext tool SMALTO.¹² Although Bernsen's work has not introduced clear guidelines for multimodal user interface design, he has been able to provide some advice on selecting right modalities for a particular application. His taxonomy of output modalities is also a good basis for further investigation of multimodal interfaces.

¹⁰ Nokia includes WAP Service Designer's Guide to Nokia Handsets in the download package of Nokia WAP Toolkit 2.0. The company also offers several phone specific versions of their design guides at <http://www.forum.nokia.com/> [checked December 6, 2002]. A more general document, How to Design Usable WAP Services, concerning user-centered design process is also available at the same site.

¹¹ For example GSM Application Style Guide, which is available at <http://demo.openwave.com/pdf/styleguides/gsm.pdf> [checked December 6, 2002]. They also provide Top 10 Usability Guidelines for WAP Applications at <http://developer.openwave.com/resources/uiguide.html> [checked December 6, 2002].

¹² Available at <http://disc.nis.sdu.dk/smalto/> [checked December 6, 2002].

Even though guidelines in Chapters 3.2.1 and 3.2.2 offer many instructions on how to use speech and WAP separately, they do not include any specific hints on how to combine speech with the modalities WAP provides. The designer needs to decide many issues without the help of any guideline. He or she has to decide what features are offered using which modalities and how modalities are used to support one another. More research on the human brain will be needed before the designer knows precisely how to combine different modalities in such a way that they will support each other rather than interfere with each other (Raisamo 1999). The literature offers some hints for a designer who attempts to combine voice with some visual modality, although it would be exaggeration to call these advises guidelines. They are reviewed in Chapter 4, where a couple of guidelines for our own use are derived based on these advises. In the same chapter, the voice modality and WAP are analyzed shortly to find ways to use WAP to support speech and vice versa.

The voice-related guidelines presented in Chapter 3.2.1 are collected from different sources, so it is obvious that there is some overlap in the lists. For example, guidelines Be10 and M10 both call for consistency of prompts. Also, some guidelines from Bernsen et al. (1998), like Be5 and Be6, are somewhat strange. No designer would deliberately make the system lie to the user. The authors have borrowed some of their guidelines from the literature of co-operative human-human discussion, which explains the peculiarity of for example Be5 and Be6. In addition, the idea of some guidelines, e.g. Be12, is not immediately clear. The authors explain each of their guidelines in detail and provide examples so, eventually, the meaning of all guidelines can be understood. Before taking advantage of the voice-related guidelines, it seems wise to remove overlapping guidelines and reformulate those guidelines whose meaning is not obvious. This process is described in Chapter 4.

There are also some problems related to WAP guidelines that were presented in Chapter 3.2.2. Guideline S11 states that horizontal scrolling is easy on most devices but vertical is not. This statement raises some questions because horizontal scrolling is not possible on many WAP devices. Idea behind this guideline might be that, since most WAP browsers will wrap long lines automatically, there is no need for the designer to think horizontal scrolling. Another, perhaps more likely, explanation for this statement is that authors have confused vertical and horizontal scrolling.

Guideline S12 suggests using multiple cards on one deck instead of very large cards. This contradicts findings by Buchanan et al. (2001), who conducted a user test with three different systems. They found that the interface with cards as long as 27 lines were preferred to the interface that had decks with several shorter cards. The original idea behind WAP has been to reduce scrolling by using several small cards, so it is possible that Schmidt et al. (2000) have emphasized the original approach. The results of the tests by Buchanan et al. (2001) suggest, however, that the users have not accepted the deck of cards metaphor. It seems that users expect the WAP services to work similarly as web pages, so a longer card might be a better option than several cards. Refined versions of WAP-related guidelines for our own use can be found in Chapter 4.

3.3 Role of guidelines in usability engineering

Guidelines summarize the existing knowledge of good user interfaces, so the application of guidelines is one important activity in any usability engineering process. However, guidelines cannot deal with choices that are highly dependent on context (Gould & Lewis 1985). To take into account this restriction, the designer has to keep some things in mind. Rosenzweig (1996) emphasizes that guidelines should never replace prototyping and usability testing. According to her, the user-centered approach is still the best way to ensure that a product successfully incorporates the voice of a customer. Smith and Mosier (1986) stress the same aspect, but they also remind of the dangers in relying solely on prototyping. Prototyping is not a substitute for careful design. Unless the initial design is reasonably good, prototyping may not produce a usable final design. Applying guidelines is one method that helps in creating reasonably good initial designs.

Nielsen (1992) divides his usability engineering lifecycle into three main phases: activities considered before the design, during the design, and after the design. Nielsen (1992) suggests that the activities in each phase should be applied in an iterative fashion, but he does not mention that iteration would be needed between the main phases. He considers application of guidelines as one of the activities in the design phase of the lifecycle. There is some evidence that this view is problematic. Smith (1988) notes that many guidelines actually imply the need for a careful task analysis to determine design requirements. Since the task analysis is clearly something that has to be considered before the design, this suggests that guidelines should be reviewed already before the design, not during design, as Nielsen (1992) suggests.

By studying carefully the guidelines in Chapters 3.2.1 and 3.2.2, one finds that at least guidelines Be12, Be17, M1, and G2 imply actions that should be done before starting the actual design of the interface. For example, guidelines M1 and G2 both call for a careful task analysis in order to allow the designer to make sure that the tasks that are the most important for the users will be easy to carry out. Similarly, Be12 calls for the system to address the topics in the order users expect topics to be addressed. This order can be found only by studying users' current behavior, which should be done together with other user studies before designing anything.

By considering the guidelines only during the design phase, the designer takes a risk that some of the guidelines imply a certain specific user study that could have been done with a small effort in conjunction with other studies in the pre-design phase. Making one of the studies later will probably be more expensive and time-consuming. Still, one should not forget that the most important use of guidelines is in designing the prototypes. Both the comments from Smith (1988) and the analysis of the guidelines presented in this thesis seem to suggest that application of guidelines should be considered as a meta-method in Nielsen's lifecycle. With meta-methods Nielsen (1992, 1993) means methods that are applicable in most of the phases and which keep the activities on the right track. Since the role of guidelines was one of the questions that this thesis studies, this issue has been analyzed further during our

design process. Our experiences from applying guidelines in the design of the multimodal TV-guide will be discussed in Chapter 6.

4. Refined guidelines for interfaces with speech and WAP

As discussed in Chapter 3.2.3, some of the guidelines presented in Chapter 3.2 overlap and the style and language are in need of harmonization. During the design process, we developed a more coherent list of guidelines, which we utilized in different phases of the design process. This process resembles the translation of guidelines into design rules, which was mentioned in Chapter 3.2. The difference is that our intention was not to create rules specific for this application. Our intention was to keep the list generic enough to help in future projects. Our list might form a starting point for building a style guide but, in the current form, the list cannot be considered as a style guide either.

Since we had concluded that guidelines are useful already in the pre-design phase, we started gathering the guidelines early in the design process, but we refined the set during the process. In the following, the list of guidelines we developed is introduced. First, the voice-related guidelines are presented and next the WAP-related guidelines. Last, the guidelines that we have developed ourselves by analyzing properties of voice and WAP are described. The guidelines are numbered so they can be referred to in later chapters. They are classified by the area they cover to help the designer to find the appropriate guidelines for the problem he or she is trying to solve.

4.1 Voice-related guidelines

Most of the guidelines were directly derived from one of the guidelines described in Chapter 3.2. Some of them have been combined with another guideline and the language of the guidelines has been slightly harmonized. Refined set of voice-related guidelines are listed in Table 4.1.

Since violations against Be5 and Be6 are not likely to happen unless there is a malfunction in the system, these two guidelines were not used in our design process. For this reason, there is no equivalent for Be5 and Be6 in Table 4.1 either. If a guideline in Table 4.1 is combined from two original guidelines, there is always a reference to both guidelines in the third column of the table. Guidelines 4 and 14 resemble each other a lot so they could have been combined to one guideline. Since Bensen et al. (1998) wanted to present these as two different guidelines, we decided to do the same. The guidelines highlight the two problems related to long prompts. Users will probably become inattentive, if prompts are very long. They might also be encouraged to use unnecessary long phrases themselves, which are difficult for the system to understand. Guidelines 6, 14, 18, and 31 have been slightly expanded to make their purpose more apparent. A small addition is also made to guideline 8 because multimodal interfaces might provide options for browsing that make summarization unnecessary. All the guidelines from Kamm and Walker (1997) and the first guideline from Mané et al. (1996) were re-written in an imperative mood to be consistent with other guidelines.

Table 4.1 Voice-related guidelines. The guidelines are classified to four groups: prompt design (PD), error recovery (ER), anticipating user input (AUI), and miscellaneous voice guidelines (MV). Abbreviations in the fourth column refer to the guidelines in Chapter 3.2.1. ('KW' stands for the guidelines by Kamm and Walker, 'Be' is for Bernsen et al., and 'M' for Mané et al.)

No.	Guideline	Class	Derived from
1	Provide feedback on each piece of information provided by the user. Inform that the user's request was understood and the system is acting to fulfill the request.	PD	KW3, Be3
2	Keep syntax and semantics consistent across all prompts in the system.	PD	M10, Be10
3	Provide sufficient task domain knowledge and inference.	PD	Be20
4	Keep prompts brief to encourage the user to be brief.	PD	M11, Be11
5	Supply confirmation messages frequently, especially when the cost or likelihood of a recognition error is high.	PD	M4
6	Summarize the commitments users have made during the dialog to make sure that key information exchanged has been correctly understood.	PD	Be2
7	Do not list all voice commands in the first prompt of a menu, but offer them to the user if he or she seems to have problems knowing what to say.	PD	KW1
8	Provide summarization when there is a lot of information to present to the user and options for scanning and browsing are cumbersome.	PD	KW6
9	Assume the user does not know the active vocabulary and guide him towards responses that maximize clarity.	PD	M2
10	Guide users toward natural in-vocabulary responses.	PD	M8
11	Do not give too many options at once.	PD	M9
12	Avoid prompts that are too similar.	PD	M12
13	Make your contribution as informative as is required (for the current purposes of the exchange).	PD	Be1
14	Do not provide too much information on a single system turn.	PD	Be4
15	Be relevant, i.e. be appropriate to the immediate needs at each stage of the transaction.	PD	Be7
16	Avoid obscurity of expression.	PD	Be8
17	Avoid ambiguity.	PD	Be9
18	Address the task-relevant topics in an order the user expects them to be addressed.	PD	Be12

19	Inform the users of important non-normal characteristics which they should take into account in order to behave cooperatively in spoken interaction. Ensure the feasibility of what is required of them.	PD	Be13
20	Provide clear and comprehensible communication of what the system can and cannot do.	PD	Be14
21	Provide clear and sufficient instructions to users on how to interact with the system.	PD	Be15
22	Make clarifying questions to the user if the user does not give sufficient information or if his or her input is ambiguous.	ER	KW5, Be24
23	Assume errors are the fault of the recognizer, not the user.	ER	M5
24	Design graceful recovery when the recognizer makes an error.	ER	M6
25	Enable repair or clarification meta-communication in case of communication failure.	ER	Be21
26	Initiate repair meta-communication if system understanding has failed.	ER	Be22
27	Initiate clarification meta-communication in case of inconsistent user input.	ER	Be23
28	Provide consistency across features in a multi-featured application i.e. use a subset of the vocabulary that is always available and that serves the same function regardless of where the user is.	AUI	KW2
29	Take partners' relevant background knowledge into account.	AUI	Be16
30	Take into account possible (and possibly erroneous) user inferences by analogy from related task domains.	AUI	Be17
31	Take into account legitimate partner expectations when defining the requirements of the system.	AUI	Be19
32	Allow for the user not knowing the answer to a question or not understanding a question.	AUI	M3
33	Make sure that typical scenarios are painless.	MV	M1
34	Assume a frequent user will have a rapid learning curve.	MV	M7
35	Make sure that the system has a low latency and that it allows interruption of its prompts when possible, so the natural flow of the dialog is preserved.	MV	KW4
36	Separate whenever possible between the needs of novice and expert users (user-adaptive interaction).	MV	Be18

Several authors provide instructions for user interface design that they do not formulate as guidelines but rather as informal hints. Some of them we considered so valuable that we decided to create a few more guidelines based on them.

The most important choice the designer of a spoken language system has to make is the selection of dialog strategies the system uses in conversations with the user. If the

system can handle only a small vocabulary and the designer wants to minimize errors, the designer can use directive prompts that tell the user the exact words they can say. More implicit prompts demand a larger vocabulary and a more flexible grammar because they do not guide users to use pre-defined phrases. Implicit prompts, however, are more conversational and allow more natural interaction, although first-time users might have more problems with them. To utilize the best of both strategies the designer can first present an implicit prompt to the user, like "You have heard all messages, please give an instruction." If the user has problems coming up with an instruction, the system can next provide a more explicit prompt, like "The accepted speech commands are replay, delete, and turn off." (Yankelovich 1996, Walker et al. 1998)

Both Yankelovich (1996) and Mané et al. (1996) suggest that, in multimodal systems which combine speech and visual modalities, the visual modality should be designed to give hints for users of what they can say. It is clear that using explicit prompts listing all voice commands right away would lead to very tiresome interaction. Since we are also able to use the visual modality to help the users know what they can say, the implicit prompts should not be as difficult for users as they are in voice-only systems. These ideas led us to reformulate guideline KW1 into stricter guideline 7 in Table 4.1. We wanted to have very consistent prompts so we made this guideline stricter than the other guidelines. Based on the previous discussion, we also formulated a new guideline "Consider using visual output to let the users know what they can say."

As described above, the visual modality can be used to complement the weaknesses of the speech modality. Speech can be used in a similar way to compensate for the disadvantages the direct manipulation interfaces have. The size of the display limits the number of objects a system may show at a time. Hence, in a multimodal system, the user may benefit from the ability to use speech to refer to actions and objects that are not visible (Cohen & Oviatt 1994). Due to this, Mané et al. (1996) suggest that the designer should allow the users to specify non-visible actions and objects with voice. We formulated this into a guideline "Consider allowing users to use voice input to specify actions and objects that are not visible in the visual output."

As described in Chapter 3.2.1, speech is temporal, which means that the user has to concentrate on prompts or otherwise the information will be lost. This led Kamm (1994) to recommend that users should be given a way to request repetition of information whenever they need it. The guideline we derived from this is "Allow users request repetition of a prompt whenever they want." Kamm (1994) also notes that, if the system has difficulties to decide what was the user's utterance, the system should not ask the user to repeat what he or she just said. This utterance has a history of not being handled successfully, so Kamm (1994) suggests that it is better if the system asks the user to confirm the most likely recognition result. In a situation where the system cannot make a reliable guess, the user might be asked to paraphrase his or her input (Mané et al. 1996). Our guideline states this in the following way "Do not make the users repeat their utterances, unless it is absolutely necessary. If the system can make a good guess of the utterance, the system should ask if the guess is right. If it cannot, the system should encourage the users to paraphrase his or her input."

The last guideline we formulated is "Make sure that the synthesizer pronounces all the words properly." This was inspired by Yankelovich's (1994) workshop paper, where she explains that her research group collected considerable amount of calendar data to be able to make their voice controlled calendar application to pronounce the most frequently used abbreviations correctly.

Table 4.2 Additional voice-related guidelines. These guidelines were inspired by the literature but they were formulated into guidelines by us. The discussion of origins of these guidelines can be found above. The guidelines are classified to four groups: combining modalities (CM), prompt design (PD), anticipating user input (AUI), and miscellaneous voice guidelines (MV).

No.	Guideline	Class
37	Consider using visual output to let the users know what they can say.	CM
38	Consider allowing users to use voice input to specify actions and objects that are not visible in the visual output.	CM
39	Allow users request repetition of a prompt whenever they want.	AUI
40	Do not make the users repeat their utterances, unless it is absolutely necessary. If the system can make a good guess of the utterance, the system should ask if the guess is right. If it cannot, the system should encourage the users to paraphrase his or her input.	PD
41	Make sure that the synthesizer pronounces all the words properly.	MV

4.2 WAP-related guidelines

In addition to voice-related guidelines, we also collected a set of WAP-related guidelines. They are listed in Table 4.3.

WAP guidelines did not overlap, so only slight clarifications were needed to guidelines 51 and 54. A small spelling error was corrected in guideline 48. Like discussed in Chapter 3.2.3, S11 and S12 are questionable, so those were not used in our design process and, hence, they are excluded from the table.

Table 4.3 WAP-related guidelines. The guidelines are classified to four groups: hierarchy design (HD), output design (OD), input design (ID), and miscellaneous WAP guidelines (MW). Abbreviations in the third column refer to the guidelines in Chapter 3.2.2. ('G' stands for the guidelines by Grimstad et al., 'Bu' is for Buchanan et al., and 'S' for Schmidt et al.)

No.	Guideline	Class	Derived from
42	Design WAP applications more structured than web applications.	HD	G1
43	Trim the page navigation down to a minimum; use simple hierarchies, which are similar to the phone menus that users are already familiar with.	HD	Bu2
44	Develop phone-based WAP services that provide direct, simple access to focused valuable content. Usable and useful WAP services on phones will be the ones that give the user key, summarized information with very few keystrokes or text entry.	HD	Bu1
45	Reduce the amount of vertical scrolling to a minimum by simplifying the text you wish to display (avoid wordy messages; go for action oriented keywords).	OD	Bu3
46	Assess the screen size and quality of the target devices with text and graphics.	OD	S8
47	Reduce the output by customizing to the user's needs.	OD	S9
48	Design information chunks that are seen at once on the screen. Larger text blocks (more than 20 words) within one card should be structured.	OD	S10
49	Reduce the number of keystrokes you expect the user to do. You can do this by simplifying navigation and by replacing text input with other types of interaction method (e.g. list selection).	ID	Bu4
50	Use numbers for input whenever possible.	ID	S2
51	Use common abbreviations in input like country codes.	ID	S3
52	Keep the input mechanism in mind if letters are used e.g. prefer first letter on key.	ID	S4
53	Offer choices (e.g. numbers, list boxes, radio buttons, link lists) or default values when applicable.	ID	S5
54	Provide labels to soft buttons where possible.	ID	S6
55	Use standard conventions on buttons (e.g. back).	ID	S7
56	Make sure that the user is able to solve the tasks they have. To realize this, one normally needs to put lot of effort into investigating which tasks are the most important for the intended users.	MW	G2
57	Combine theoretical and empirical evaluation to provide further insights.	MW	Bu5

58	Identify the benefits for bringing an application to the WAP platform. Supporting these benefits must be the primary goal of the application development.	MW	S1
----	---	----	----

4.3 Guidelines based on modality analysis

Because of the lack of guidelines specific to combination of speech recognition and WAP, we decided to analyze strengths and weaknesses of speech modality and the modalities WAP provides. With this, we wanted to find guidance for using speech and WAP to support each other. Since properties of modalities differ, for some user groups and in some usage contexts, one modality may be more suitable than another for e.g. presenting certain type of information.

If we examine properties of speech and WAP, we will find that they share some weaknesses. Arbitrary text input is not easy with either of them. Voice recognition is, at least currently, based on the fact that the computer knows all the possible words the user might say any given time. This allows the system to compare the user's utterance to all the words it expects the user to say at that point and find the word that resembles closest the user's utterance (Schmandt 1993). In a user interface of, for instance, a voice-enabled search engine, the user could say almost anything, so the system would have to compare the user's utterance to every word belonging to the language used. The task would be so difficult that the system would be very slow and it would make many mistakes. Therefore, to allow the user to input arbitrary words with voice, the system has to ask the user to spell the words letter by letter, which is very cumbersome for the user. WAP applications can allow users to type arbitrary words with the buttons but usually WAP-enabled devices have keypads with about 10 keys for inputting more than 20 letters. This means that more than one letter is assigned to one key and the user has to press each key several times to express which letter he or she wants.¹³ It appears that both voice and WAP are quite cumbersome in inputting arbitrary words (like in a user interface of a search engine) but WAP seems a little faster and more convenient of the two.

Another weakness that both modalities share is that they are not suitable for presenting very long texts. The display of a WAP device can be so small that less than ten words fit on the screen at a time. This makes reading of long articles very annoying. The situation is even worse with the speech modality. Since speech is temporal and people's working memory is capable of containing only limited amount of information at a time, users often forget the beginning of a long sentence before the system reaches the end of the sentence (see e.g. Eysenck & Keane 1995 and

¹³ To make the text input task less cumbersome for users, the mobile phone industry has introduced predictive capabilities to mobile phones, which often allows the phone to guess the right word even if the user has pressed each key only once. This technique is based on knowing in advance what words user might type so, in the case of arbitrary input, this technique faces the same problems that voice recognition does. Information about predictive text input in mobile phones can be found from <http://www.t9.com/> [checked December 6, 2002].

Sinkkonen et al. 2002). Human speakers help listener by dividing long sentences to smaller chunks by controlling the rhythm and the tone of their speech (Eysenck & Keane 1995). Speech synthesizers are not yet capable of using these cues as fluently as humans are, which makes the problem more severe in human-computer interaction than it is in human-human interaction.

Some conclusions can be drawn from this. Neither speech nor WAP is suitable for long texts or even for long sentences. The need to avoid long texts results in more structured interaction where information is divided into smaller chunks, which can make complex applications quite awkward. WAP is suitable for slightly longer texts than voice because the user can scroll back to check the beginning of a text if he or she forgets it. This suggests that, if the system offers many options for the user at a time, WAP will probably be more suitable than speech for presenting the list of options. In this way, WAP can be used to alleviate the most severe limitation of speech as a user interface component.

Speech and WAP are better suited to some user groups than others. For children, speech might be more suitable than WAP since young children are not able to read. For elderly, both speech and WAP may be difficult because they might have problems with hearing as well as poor eyesight. People who speak with a strong accent will probably run into a lot of recognition errors if the system is not tuned to their style of speaking. On the other hand, people with repetitive stress injury in their arm might not like buttons. The variety of usage situations where mobile applications may be used also creates the same kind of diversity. Sometimes users may be in a public place where they might not want to use voice and some other times they may be carrying something and, hence, be unable to use buttons. In a multimodal system, one important decision the designer must make concerns what actions the designer allows the user to carry out with each modality. It seems that, in many cases, it is wise not to make any decision on this beforehand and let users decide which modality they want to use to carry out certain action.

Some of the points made in this chapter are already included in the guidelines presented in the previous chapters, but we decided to formulate three more guidelines from those aspects that earlier guidelines do not take into account. They are listed in Table 4.4.

Table 4.4 Guidelines based on modality analysis. All guidelines belong to the same group: combining modalities (CM).

No.	Guideline	Class
59	If you cannot avoid input of arbitrary text, use WAP rather than voice.	CM
60	If you have to present many options or large amount of information to the user, use WAP rather than voice.	CM
61	Offer as many actions as possible with every modality and let users choose what modality they want to use at any time.	CM

5. Iterative guideline-supported design

In this chapter, the iterative design process of our multimodal TV-guide is described. The motivation for designing the application was two-fold. We wanted to demonstrate the possibilities of multimodal interfaces and the chosen architecture. We also wanted to gather experience from designing multimodal interfaces with a user-centered process. Especially, our focus was on the role of guidelines in the process.

Our design process was strongly inspired by Nielsen's usability engineering lifecycle presented in Chapter 3.1.2. Before starting the design of the interface, we conducted two studies:

- Diary study
- Natural dialog study

These studies were made to let us know our users. TV program information is typically looked briefly in the morning and then a few times in the evening. By visiting people's homes and observing their TV program information use for the duration of, say, one day, we would have been able to record only couple of TV program information related activities. Since we wanted information about a larger number of activities with reasonable effort, we conducted a diary study where participants wrote down their activities related to TV program information during one week. The natural dialog study provided us with additional information about the vocabulary people use in discussing TV programs, which is very important in building speech recognition applications. During the design, our activities were:

- Parallel design
- Prototyping
- Iterative design
- Heuristic evaluation
- Two usability studies

These activities are straight from Nielsen's usability engineering lifecycle, although our scarce resources forced us to use quite limited versions of most of the methods. This thesis will not describe the last activities in the process because the author did not participate in the project after the second usability test. Due to the tight schedule, we had to select only those methods from the lifecycle that we anticipated would have the greatest impact. Some of the methods were self-evidently unnecessary, like coordinated design of the total interface and collecting feedback from field use. No training or user documentation was produced for the application, so the concept of total interface has little significance in our case. Since the application was created for demonstration purposes only, there was not any field use that would have provided us with feedback. We anticipated the competitive analysis and participatory design to be less cost-effective than the other activities, so we concentrated our effort on the activities we predicted to be more important. Our prediction of the most valuable

activities is in line with the results of Nielsen's (1992) survey presented in the end of Chapter 3.1.2.¹⁴ We considered setting usability goals but, since the number of users in each test was quite small, the results of the tests measuring the performance of the system against the goals would not have been reliable. Applying guidelines was done during every design activity, so they are not described as a separate activity.

In designing speech systems, it is common to carry out so-called Wizard of Oz studies. In these studies, a human operator simulates the speech recognition system and selects the prompts manually according to the interaction logic defined beforehand. By recording and analyzing conversations where users interact with such a system, it is possible to collect words to the vocabulary and find the flaws in the interaction model. The Wizard of Oz method can be seen as a form of prototyping and empirical testing. Wizard of Oz studies are quite laborious to conduct, so for a system with a simple interaction logic, one might want to start the prototyping straight with a computer system. (Bernsen et al. 1998)

Because the interaction logic of our system was relatively simple and because we had good prototyping tools available, we decided not to use the Wizard of Oz technique. Each design activity is described in detail in Chapters 5.1 - 5.6. Iterative design and prototyping are not considered as separate activities, but rather each prototype is described either as a result of the study, like in parallel design, or as a subject of the study, like in empirical evaluations.

The tight schedule we had to maintain hindered the amount of documentation we were able to maintain. We created a draft user interface specification during the parallel design activity, but later the prototypes themselves worked as a kind of user interface specification. When we concentrated on finding short and consistent prompts, we created a document that listed all the prompts of the prototype on one page, so they could easily be compared. We also created some technical documents about the database schema and communication methods of the software components, but they did not include information related to user interface design. Discussion about the effects of the inadequate documentation follows in Chapter 6.

We did not define the intended target group officially because our purpose was not to create a product that would be sold to customers. Rather we wanted to create an application that would demonstrate possibilities of multimodal interfaces and that would be presented to public only in conferences and workshops. To help the design process, we nevertheless wanted to have some idea of the people who might want to use the kind of system we were building. We hypothesized that 20-35 year old people living in cities and having limited spare time might already have WAP-enabled phones. We expected them to be willing to acquire interesting information during a bus trip or a similar short break, during which they might not have other information sources available. For the sake of the design process, this group of people was considered as our intended target group, although no studies were made to confirm our assumption.

¹⁴ In the survey, iterative design, task analysis, and empirical testing were rated more important than participatory design.

In addition to the multimodal TV-guide we designed, another TV-guide was built in the project. It was a unimodal voice-only application that worked over a regular telephone line. Our pre-design studies were aimed to provide useful information for the design of both applications but later studies concentrated exclusively on the multimodal TV-guide. The applications shared some software components, so in some cases we had to keep the voice-only application in mind also in the later phases of the study. This thesis describes the work done for the multimodal application, but the voice-only TV-guide will be mentioned when it has had an effect on the design of the multimodal application.

We planned to build both English and Finnish versions of the TV-guide, which affected some design activities where we had to decide which language we should use in the study. Unfortunately, the implementation of the Finnish speech recognition trailed somewhat, so only the English system is described in this thesis.

Size of the team that took part in designing and implementing the multimodal application was four people. The team consisted of a project manager, a programmer, and two usability specialists, although the roles were very flexible. All of us contributed to the design of the user interface and, on the other hand, debugged the system. The author of this thesis took a role of a designer or a usability expert, depending on the design activity. The author participated in every activity, as did the project manager. The programmer changed after the first usability test and three usability experts alternated in the role of the second usability specialist.

General description of the application can be found in Chapter 1.3. The architecture and chosen technologies implied some restrictions on the functionality of the system. We used Voice eXtensible Markup Language¹⁵ (VoiceXML) version 1.0 to describe the voice part of the application. It does not include very flexible support for handling natural language utterances, but it suits well for describing menus with reasonable amount of options. The visual user interface was described using Wireless Markup Language¹⁶ (WML) version 1.1. It enabled us to create menus that resemble web pages, although it does not allow as flexible formatting as HTML.

The architecture developed in the CATCH-2004 project consists of a VoiceXML browser, a WML browser, and a central coordinating entity that is responsible for the synchronization of the behavior of the different browsers. The architecture would allow taking advantage of more than the two browsers mentioned above, but the application presented in this thesis uses only VoiceXML and WML browsers. The

¹⁵ VoiceXML is a language for representing spoken human-computer dialogs. It has been specified by VoiceXML Forum and the specification has been delivered to W3C as a basis for their forthcoming dialog description language. The specification of VoiceXML 1.0 can be found from <http://www.voicexml.org/specs/VoiceXML-100.pdf> [checked December 6, 2002].

¹⁶ WML is one element of WAP. It is intended for use in describing the content and the user interface for services using narrow-band devices like cellular phones. The specification of WML 1.1 can be found from <http://www1.wapforum.org/tech/documents/SPEC-WML-19990616.pdf> [checked December 6, 2002].

coordinating component gets the content from the web server in a single document, divides it into modality-specific parts, and sends them to each component browser (Kleindienst et al. 2002). When the user highlights a link from one of the browsers, special synchronization messages are sent between the coordinating component and the browsers to keep the content synchronized between the different browsers (Kleindienst et al. 2002). IBM Websphere Voice Server 2.0 acted as our VoiceXML browser and IBM ViaVoice Millennium edition as a speech recognition engine. For the second usability test, we upgraded the IBM ViaVoice to version 9.0. Our WML browser was Nokia WAP Toolkit 2.0. Both browsers were augmented with some code that connected them to the other software components we used. The application used speaker-independent speech recognizer, so users did not have to train the system to recognize their voice but, on the other hand, the active vocabulary had to be kept small at every menu. The architecture was designed to support the fusion of modalities, at least in the sense of coordinated simultaneous input from several modalities, but the status of implementation of the software components did not yet allow this (Kleindienst et al. 2002).

5.1 Diary study

In a diary study, participants are asked to record those daily activities that are expected to be relevant to the project. They can be asked to write their diaries as free-form descriptions of their activities or as more structured descriptions that they write on a preprinted log form. Diaries are usually kept for one or two weeks because a longer time would require too much effort from the participants. (Rieman 1993, Nielsen 1995)

Diaries provide information about how users currently behave in carrying out tasks that interest the project (Nielsen 1995). The information is more concrete and detailed than information gathered with just interviews. In an interview about past events people usually try to abstract and summarize, which reduces the usefulness of the data (Beyer & Holtzblatt 1998). On the other hand, interviews can provide more in-depth information about people's activities than diaries. For this reason, the diary study is usually supplemented with a post-study interview of each participant (Nielsen 1995).

5.1.1 Research subject

Five persons took part in the study. All participants were females. Two of the participants were 10 years old children and others were aged between 20 and 34. The adults belonged to the intended target group of the application. The two children were included because we expected that they would have highly different TV-habits than adults have. We hoped that considering their habits would provide us with some ideas that we might not have come up with if we had considered strictly our intended target group.

5.1.2 Method

Our research group in this study consisted of two persons. We planned, conducted and analyzed the study together. We created a preliminary version of the diary form the participants should fill every evening and we gave it to two pilot persons, who filled the form concerning their previous day. Since the comments from the pilot persons were positive and they seemed to understand all the questions in the form, we did only some minor stylization to the form before we gave it to the participants.

The form was written in Finnish because all the participants were native Finnish speakers. The diary consisted of seven pages, one for each day of the week. Each page was composed of instructions and a table, where participants could fill information concerning each program they had watched during the day. They filled answers to the following questions.

- What was the name of the program?
- Where they had looked for the information related to that program (if they had looked for any information)?
- What information they had looked for (e.g. starting time, name etc.)?
- Did they watch the whole program?
- Did they do anything else besides watching that particular program?

In addition to answering to the questions concerning each program, they wrote down if they had looked for any TV program information that did not relate to any program they eventually watched.

We suspected that a diary form that would have concentrated solely on questions about the TV program information usage would have encouraged participants to describe their television watching as more schedule-oriented than it actually is. Hence, we included questions about what they did during watching the programs etc. Some discussion about successfulness of this approach and alternatives to it will follow in the end of this chapter.

We delivered the diary forms to the participants and explained what we wanted them to do. We asked the participants to fill one page of the diary each day, preferably program by program during the evening. We encouraged them to fill the form informally, so if they did not know the program they had watched they could leave that information out etc. They were told that they could call us if they had any problems concerning the study. We also planned to interview each participant about his or her diary, but we were not able to do that because other work items took our time. After the participants had filled the diaries, we collected and analyzed them. Since the number of participants was quite small, we did not use any formal methods in analyzing the diaries.

5.1.3 Results and design suggestions

Results and design suggestions are collected to Table 5.1.

Table 5.1 Results from diary study.

No.	Result	Suggestion
1	The program name, the starting time, the ending time, and the length were the most frequently checked properties of programs.	All the mentioned properties should be visible when feasible.
2	One person twice looked for information after the program ended because she wanted to know what was the program she just had watched.	There should also be a way to check information about programs that have been just shown.
3	The three tasks that participants most often carried out were following. <ul style="list-style-type: none">• They browsed through the evening's programs to see if anything interesting would be on.• They checked some detail about a particular program (e.g. the starting time of the program, the name of the episode).• They searched for programs of a certain type (e.g. something interesting had happened and the participant wanted to know if there still would be any news on).	There should be a good support for casual browsing and a way to search programs by type or name.
4	All the adult users also read longer reviews about programs.	Longer reviews about the programs should also be included, if feasible.

5.1.4 Discussion

The suggestion about including longer reviews to the system is not feasible with the current choice of modalities. If some other modalities are available later, the reviews should be added, but the current modalities are not suitable for presenting long texts, as discussed in Chapter 4.3.

Since the number of participants was only five, the results are not as reliable as they would be with a larger number of participants. The suggestions made in this study should be checked against the results of other pre-design studies and conflicting suggestions should be examined more carefully. There is a discussion and comparison of the pre-design study results in the end of Chapter 5.2. Suggestion 2 is made on the basis of one participant's diary. For this reason, this suggestion should not be given too much significance, though it would not be wise to reject the result altogether. Although this study does not tell how common it is to check information about past programs, it does suggest that some users might be interested in checking information about programs that have just ended.

A diary study may encourage participants to try to describe their activities to be more logical than they actually are. In this study, there is a risk that participants may have

exaggerated the amount of planning related to deciding what program to watch. Partly this effect is due to the fact that, while participating the study, people are more conscious of their acts than they normally are. In addition, people might be tempted to act the way they think researchers hope them to act. To avoid this, we tried not to emphasize too much the program information aspects of the study in designing the diary forms. We for example added some questions that concerned more the actual watching of programs rather than finding the program information. This is hardly the most sophisticated method of avoiding this effect and it provides us with very few ways to find out how strong the effect was.

Another alternative would have been to study the participants' morning routines by asking them to fill a diary of their activities in the morning. In this option, we would have hoped that people also read newspapers in the morning and check the TV programs when they make their daily plans. This way we would not have guided users too much to search program information, but this approach would have had some downsides as well. Not all people decide their day based on TV programs, so many of the diaries would not have been related to our application. Actually, the modalities we had available were not suitable for making daily plans, but rather for checking a couple of details of one or two programs. The information in this less directed diary study would have been very useful for sketching product concept ideas. However, in our case the user-centered methods did not cover the product concept design phase, so we decided to use a more direct approach.

It is quite likely that our study provides biased view of the amount of program information that people use, but this result was not the one we were interested. We wanted to know what properties of programs people check regularly from newspapers and that information is difficult to get with less direct methods. Probably this information is also less affected by the directness of the diary form questions than e.g. the frequency of program information use.

Our application is best suited for checking TV program information while sitting in a bus or a café or anywhere else where people do not have their regular TV-guides available. Since the diaries depicted people's use of regular TV-guides, like newspapers, results must be considered with some reservations. The most questionable result is the one concerning the users' need to check information about past programs. Both times a participant wanted to check a detail of a past program, it happened when the program had ended and she was still sitting in front of the TV. In this situation, a newspaper or teletext is probably a much easier source of the information than a mobile phone. Due to this, the feature was never implemented.¹⁷

The study proved to be useful. We got some suggestions of the most valuable properties of programs and we could see what tasks are the most important for users of our future application. We were also able to derive some tacit knowledge of our

¹⁷ Another reason to leave this feature out was a resistance among the designers of the voice-only application. Since voice-only and multimodal application used same database for TV program information, implementing the option to offer information about past programs would have resulted in a fair amount of extra work for them.

future users' TV related habits, which we could take advantage of when designing the first drafts of the user interface. The children's diaries did not provide as much ideas as we had hoped for because, according to their diaries, they did not check much program information. A pre-printed log form may not be a good tool to gather information from children, so actually they might have used more program information than their diaries suggest.

Our original plan of interviewing each participant regarding her diary would have provided us with more elaborate information about each participant's week. It is very likely that the children would also have been able to offer us more information in a discussion than by filling pre-printed log forms. Unfortunately, our time frame for this study was too short and interviews had to be left out. We should also have sought men to take part in the study. It is not very likely that the kind of results we looked for would be gender-specific, but still both men and women should have been included.

In this study, we took advantage of guidelines 33 and 56, which suggest putting a lot of effort into investigating which tasks are the most important for the intended users.

5.2 Natural dialog study

In a natural dialog study, the intention is to construct a setting for human-human interaction that approximates the interaction between the user and the system that will be built in the future. The setting is constructed to stimulate a discussion between participants about subjects that the voice-recognition system should be able to handle when the system will be ready. The discussion between the participants is recorded and analyzed. An example of such a study is one made by Sun Microsystems, where they asked the participants to use a regular mouse-controlled graphics editor, but allowed them to move mouse only on the white drawing area provided by the application. If the participants wanted to access any tools or menu items, they had to ask verbally another person sitting next to them to select the tool or the menu item for them. This approximates somewhat a multimodal graphics editor that the user can control with a mouse and voice commands. Natural dialog studies are useful in designing speech recognition applications and that naturally includes any multimodal applications which utilize speech as one modality. (Yankelovich 1997)

For some applications, there is an exact equivalent in the regular human-human conversations. One example is a ticket reservation, which means that a developer of a speech-recognizing ticket reservation system can study real-life ticket reservation telephone conversations between customers and travel agents. For other applications, the designer must create a simulated situation where participants discuss the intended topics, as in the study of the graphical editor described above.

Studying human-human interaction helps in defining the vocabulary of the application, designing the appropriate wording for prompts, and identifying the probable syntax of frequent user utterances (Kamm 1994). From the structure of the human-human conversations, the designer gets hints about the order the user expects the topics to come up during the interaction (Bernsen et al. 1998). Natural dialog

studies are also used in refining the requirements and the functionality of the application (Yankelovich 1997). Still, one has to bear in mind that users interact quite differently with machines than with humans, so a designer of a speech recognition application should not try to model any human-human dialog precisely (Kamm 1994).

5.2.1 Research subject

Four people participated in the study. Two participants took part at a time and we chose them so that they knew each other well beforehand. All participants were females and they were aged between 20 and 30. All persons belonged to the intended target group of the application. All participants spoke Finnish as their native language.

5.2.2 Method

Our research group in this study consisted of two researchers. The work was distributed relatively evenly in every phase of the study. We grouped participants into two pairs and met each pair separately. We gave the participants written instructions which included scenarios that described the situation we wanted them to pretend to be in. We asked the participants to act like they would meet their friend in the evening and we asked them to discuss on the telephone about a movie they would like to watch together. Only one of the pair had a newspaper in front of her so she had to describe to the other one, what would be on the TV in the evening. The other one took part in the conversation without seeing the newspaper in front of her.

Our intention was to create a situation where the participants would have a conversation that would resemble a discussion a user might have with the application we were going to build. The person with the newspaper would be talking about things that a computer must be able to speak and the person without the paper would be in a similar situation than a user of our future system would be. The situation resembles much more using a voice-only system than a multimodal system since there is no visual information of which to take advantage. This study was planned keeping in mind both the multimodal application and the voice-only application that were built in the project, although it might have slightly reduced the usefulness of the results in the design of the multimodal application. Some discussion follows in the end of this chapter on how we would have conducted the study if we had had only the multimodal application to consider. We planned to implement English and Finnish versions of both applications and we chose to conduct this study in Finnish with native Finnish speakers. We anticipated that we would be able to collect some Finnish vocabulary and information about the syntax of frequently occurring user utterances, which our future Finnish system should be made to recognize. We expected that we would also get some results that we could generalize to all versions of the system, regardless of the language, e.g. information about the requirements and the needed functionality as well as the expected order of topics.

5.2.3 Results and design suggestions

Results and design suggestions are collected to Table 5.2.

Table 5.2 Results from natural dialog study.

No.	Result	Suggestion
1	We collected information about the vocabulary, phrases, and the style of speaking.	This information should be utilized when designing a Finnish version of the application in the future.
2	The participants frequently rounded times to the nearest quarter of an hour (e.g. it lasts about one and a half hours, it starts half past five).	The application should allow rounded time expressions in the voice commands.
3	The participants mentioned frequently the starting time, the ending time, and the name of the movie. Also actors were discussed a lot.	The mentioned properties of programs should be shown when feasible.
4	One of the participants marked programs they planned to watch with a pen to the newspaper.	There should be a way in the application to mark interesting programs or get an SMS that lists the programs the user wants to watch.
5	The topics were discussed in a very informal order. The overall structure was that participants asked if there would be anything interesting on and, if there was something, they wanted to know more details about the interesting program.	The similar order should be followed in the application.

5.2.4 Discussion

The number of participants in both the diary study and the natural dialog study was quite small, which reduced the reliability of both studies. A sensible interpretation of these studies is to prioritize the results that both studies confirm, but also keep the other results in mind when designing the system. Both studies showed that the name, the starting time, the ending time, and the duration are important properties of TV programs, but only the natural dialog study suggested actors to be significant. The conversation in this study concerned deciding about a movie, so certain movie-specific aspects, like actors, are probably over-emphasized in the natural dialog study. Since it is quite probable that occurrence of actors is due to the research setting of this study, the other properties should be given higher priority in the user interface. Other results of the diary study and the natural dialog study did not overlap or conflict.

Our tight schedule did not allow us to make the study in both languages, although it would have been useful. We decided that Finnish was more important because fewer speech recognition applications exist in Finnish. This meant that we were more on

our own in designing the Finnish application. Unfortunately, the implementation of the Finnish speech recognition trailed somewhat, so only the English system is described in this thesis and this study did not provide us with information about the English vocabulary.

The fact that people talk differently with machines than with humans should be kept in mind when interpreting the results of this study. At least the vocabulary and the style of language should be taken with some reservations. The effect may actually be stronger in this study than in an average natural dialog study because checking TV programs is a far more casual task than for instance a ticket reservation. The fact that the participants knew each other well may also have made the discussion more relaxed than any discussion with a computer would be. Since it is not common to discuss television programs in formal situations, this effect was difficult to avoid.

Despite the uncertainties in interpreting the results of the study, the natural dialog study proved to be useful. We were able to refine the functionality and the requirements of the application with suggestions 2, 3, and 4. We also got some vocabulary and some probable user utterances that we can use in the design of the Finnish system. The study provided us with some information about the order the users expect the topics to come up in the interaction, although it seems that people do not discuss television programs in a structured way. In this respect, the TV program information retrieval differs considerably from many other areas where voice recognition systems have been built previously, like the ticket reservation.

If we had been able to plan the study without having to keep in mind the voice-only application, we could have tried to create a situation where the participants would have interacted more multimodally. However, there would have been several problems in creating such a situation. We could have allowed the participants to discuss the subject next to each other in front of a newspaper, so both participants could have used gestures and the visual modality in addition to speech in the discussion about television programs. The properties of a newspaper nonetheless differ a lot from the properties of a mobile phone, which probably would have made the results useless in our design process. TV programs in the newspapers are normally printed on pages that are at least fifty times larger than a screen of a regular mobile phone. Since the small screen might be the most important property of a mobile phone that has to be taken into account when designing WAP applications, the results of such a study would not have helped much the design of our user interface. Our application cannot take advantage of gestures, so that part of the study would not have been very useful either. Another option we might have tried would have been to print the TV program information on small pieces of paper and give those to the participants instead of a newspaper. This would have approximated better the use of a mobile phone in searching for information, but our design of the pieces of paper would have affected the results a lot. Since we would have had to decide beforehand what program-related information would have been displayed simultaneously in the study, we could not have got any reliable results concerning the important properties of programs or order of the topics. The study would have resembled a lot a user test with a paper prototype, which would have been more useful in a later phase of the process. It actually seems that the voice-only situation

was not a bad option for this study, even from the viewpoint of the multimodal application.

We took advantage of guidelines 33 and 56, which both highlight the need for putting some effort into investigating which tasks are the most important for the intended users. We also utilized guideline 18, which presumes that the designer should study the order in which the users expect the topics to come up in the interaction.

5.3 Parallel design

Basics of a parallel design method and the motivation to use it are described in Chapter 3.1.2.

5.3.1 Research subject

Aim of this activity was to create a first sketch of the user interface. Hence, the research subject is the user interface of a multimodal speech-enabled TV-guide.

5.3.2 Method

Our design team consisted of two designers. One of us created several preliminary ideas of the possible structure and the contents of the menus, which the other designer commented and refined. Then the first one took the preliminary designs again to make some comments and refinements of his own, which he then passed to the other designer, who continued iterating the ideas. During this process, the ideas were just sketches on a paper, as Figure 5.1 depicts. We combined the best features from several such design ideas and formed the first version of the user interface draft. Last phase consisted of adding the missing details to the draft and creating an electronic version of the user interface document, so it could be more easily delivered to other project members.

5.3.3 Results

A slightly simplified outline of the structure of the design is presented in Figure 5.2.



Figure 5.1 Excerpt from notebook containing two draft design ideas.

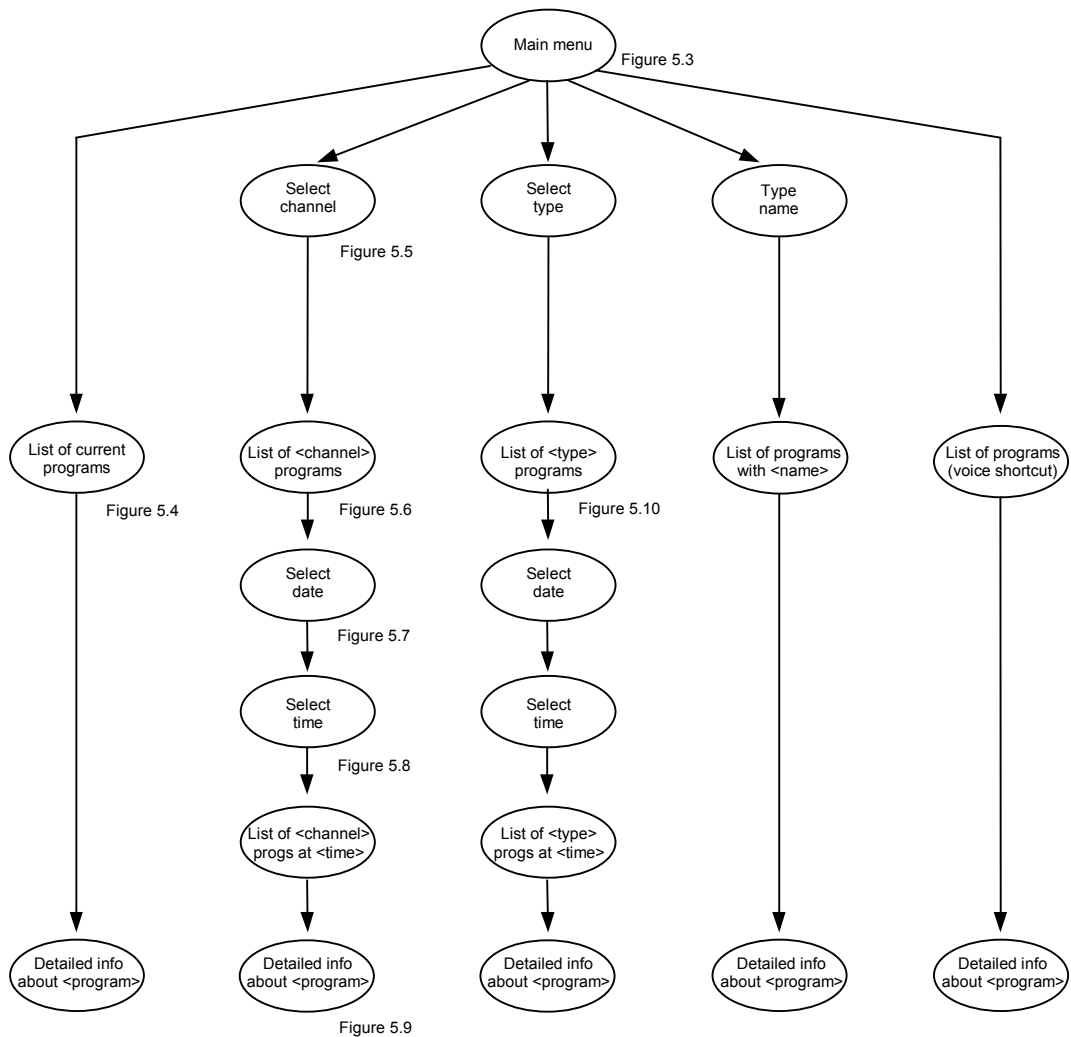
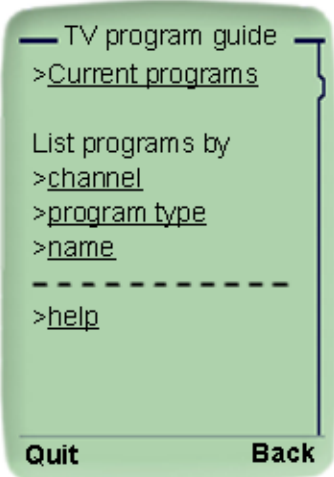


Figure 5.2 Simplified outline of structure

Detailed designs of the most interesting menus are presented in the following pages. The menus are presented in a continuous way, although the WAP Toolkit, which was used in presenting the menus, uses a simulated screen that allows displaying only three and half lines of text simultaneously (see Figure 5.11). From fourth line on, the text is hidden unless the user uses buttons and scrolls down.

The main menu of the TV-guide:

U: "Current programs"



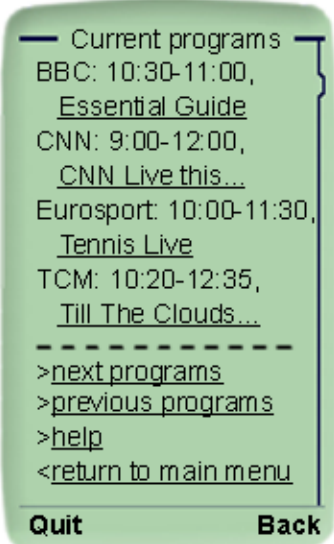
C: "Welcome to TV guide! You can get current programs or list TV programs by channel, type or name?"

↓ U: (long silence)

C: "You have following options: help, current programs, list by channel, list by type, list by name."

Figure 5.3 Main menu

All the links can be selected with either buttons or with a voice command. The valid voice commands are the underlined words (i.e. links) as well as the commands the system suggests in prompts. 'U:' means a possible user utterance and 'C:' means a computer prompt. If the user selects *current programs*, the following list will be shown:



C: "Current programs are now on the screen. You can for example get next programs or return to main menu."

↓ U: (long silence)

C: "You have following options: help, return to main menu, next programs and previous programs."

Figure 5.4 List of current programs

If the user selects *channel* from the main menu (Figure 5.3), the following menu will be shown:

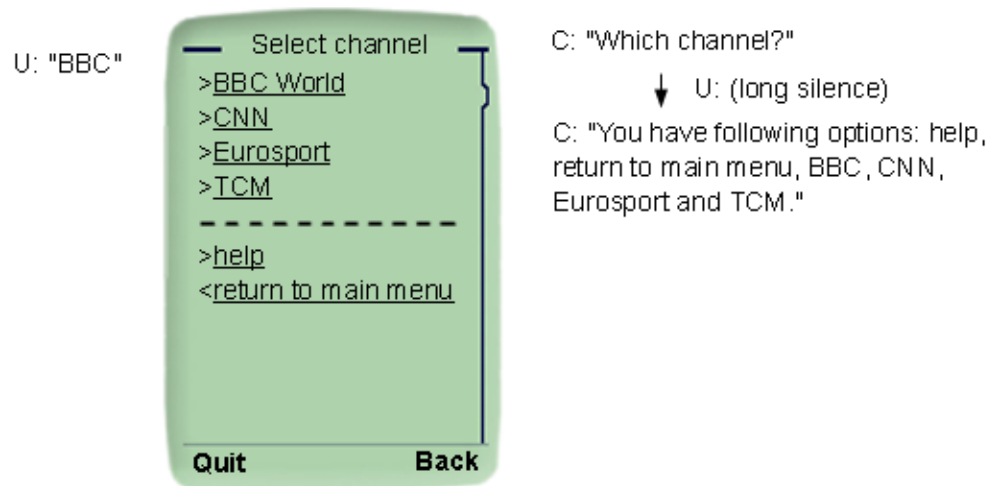


Figure 5.5 'Select channel' menu

If the user selects *BBC World*, the following list will be shown:

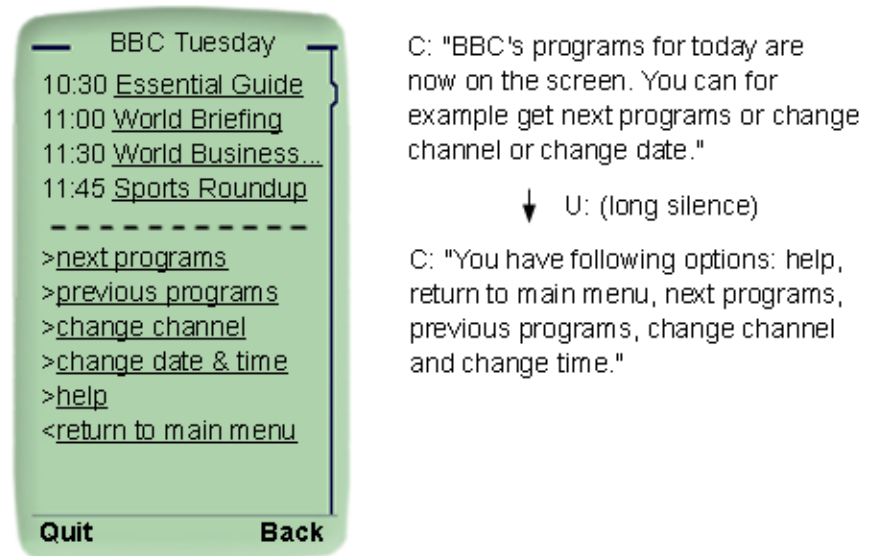



Figure 5.6 List of BBC programs on Tuesday morning

If the user selects *change channel*, a menu similar to Figure 5.5 will be displayed and, if the user selects one of the channels from the menu, a program list similar to Figure 5.6 will be displayed. The illustration of the structure in Figure 5.2 is somewhat simplified because it does not show that the user can change the channel as well as browse programs with links to the next and previous programs.

If the user selects *change date & time*, the following menu will be shown:

U: "Wednesday" or
U: "Tomorrow"

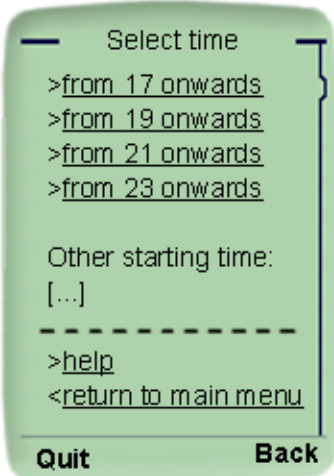


C: "Which day of the week?"
↓ U: (long silence)
C: "You can say any day of week, for example Wednesday."

Figure 5.7 'Select date' menu

If the user selects *Wednesday*, the following menu will be shown:

U: "5 PM"



C: "From what time onwards?"
↓ U: (long silence)
C: "You can say any time of day"

Figure 5.8 'Select time' menu

If the user selects *from 17 onwards*, a program list similar to Figure 5.6 will be shown. The user can also use shortcuts, i.e. in the main menu as well as in any program list (like Figure 5.6) the user can say for example, "TCM Wednesday at five," and bypass the 'select channel', 'select date' and 'select time' menus. These shortcuts are available only with voice. The illustration of the structure in Figure 5.2 is simplified in the sense that the picture does not highlight that, in addition to the main menu, the voice shortcuts are available in every program list.

If the user selects the name of any program in any of the program lists, a description similar to the following will be shown:

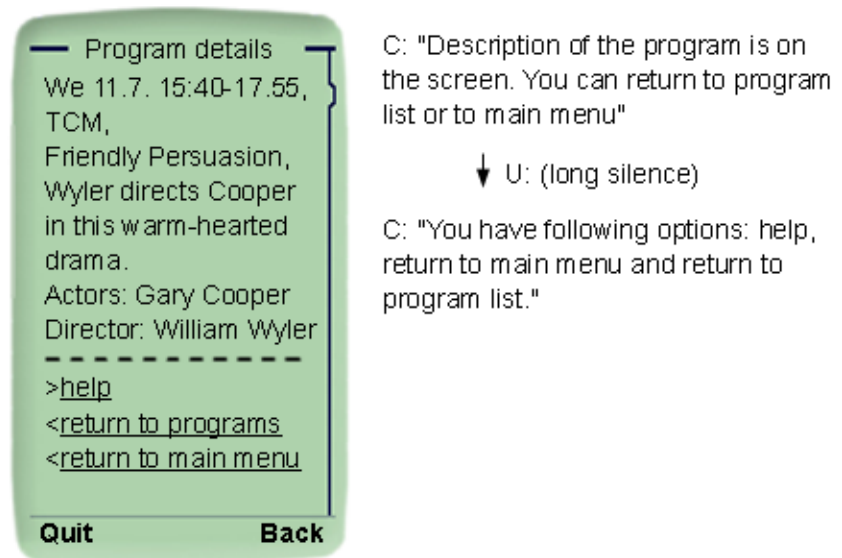


Figure 5.9 'Program details' screen

If the user selects *type* from the main menu (Figure 5.3), a menu quite similar to Figure 5.5 will be shown. Instead of channels, the user can choose from program types, like movies, news, and sports. If the user selects news from the menu, the following list will be shown:

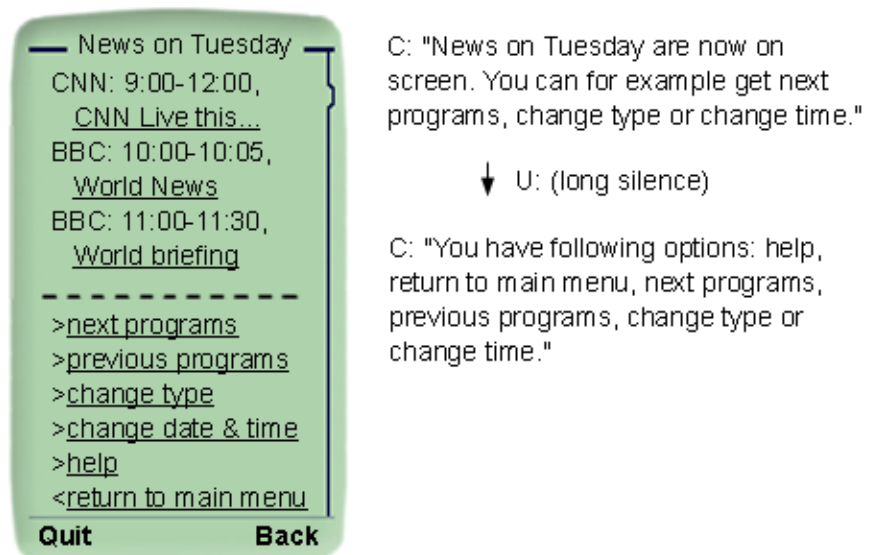


Figure 5.10 List of news programs on Tuesday

If the user selects *name* from the main menu (Figure 5.3), a form is shown where the user can type the name of the program he or she is interested in. After the user has

typed the name, a program list is shown, which resembles a lot Figure 5.10, except that the dates of all programs are shown in addition to the other properties.

5.3.4 Discussion

Contrary to Nielsen's suggestions about the parallel design method described in Chapter 3.1.2, we had our designers working in a team and not individually. This was due to the limited time the other designer could spend in working for this project, but unfortunately it did reduce the diversity of preliminary designs that we were able to generate. The use of several individually working designers would have provided us with more divergent views of the problem. Nonetheless, taking into account the resources we had, the results were satisfying. We were able to generate several suggestions and discuss the benefits and the drawbacks of the alternative approaches before concentrating on the details of the interface.

We interpreted the results of the diary study and the natural dialog study in such a way that the names of the programs, the starting times, and the ending times should be shown on every program list. We also decided that the system should include information about the actors, but it would be offered to the users only when they choose to check the details of a certain program. Since the users can calculate the length from the starting and ending times, we decided to show only the starting and ending times in the program lists. This is also a typical convention in the newspapers. Although the calculation may be cumbersome to the users, the screen of a phone is so small that we wanted to minimize the information displayed in the program lists. The need to know the channel did not occur as the result of either pre-design study, but it is so obvious that we also included information about the channel in all lists.

The three strategies found in result 3 of the diary study were supported with following features.

- There are links to the next and previous programs in every program list, which should support easy casual browsing.
- The user can search for details of a particular program using several approaches. For instance, if the user knows the name of the program, he or she can request a list of programs with that name.
- There is a possibility to search programs by type.

Longer reviews were not included, like already discussed in Chapter 5.1. The user interface does not support marking programs and getting an SMS containing information about selected programs. We considered this feature too laborious to implement in this project, although result 4 of the natural dialog study suggested it would be useful. The user interface allows rounded time expressions, at least in this phase. Feasibility of this feature has to be checked when we do have more experience with the size of the vocabulary the speech recognizer can handle. The order of topics in our user interface plan is what result 5 of the natural dialog study suggests. The user can first list programs by certain criteria and then request for more details about some particular program.

Since this is the activity where guidelines should be most useful, some highlights of the guidelines that had the strongest impact on the design are presented next. We made our best to formulate all the prompts consistently and understandably, as guidelines 2 and 16 suggest. We also tried to keep the prompts brief, like guidelines 4 and 14 instruct, although the prompts in the program lists may still be too long. The number of options available to the user was also kept reasonable, as guideline 11 suggests. If the system does not understand what the user has said, it is designed to start simple repair meta-communication, like guideline 26 suggests. Some voice commands are designed to work in every menu of the application, like help and back (guideline 28). The back-functionality helps the users out of situations where they do not know the answer to some question (guideline 32). As guideline 30 implies, we mimicked conventional TV-guides on newspapers where we found it possible. The prompts are designed to be relatively implicit and only, if it seems that the user does not know what to say, the system lists the options explicitly, like guideline 7 suggests. The system allows the user to interrupt the prompts, as guideline 35 requires. The users can use the underlined words on the screen as voice commands, so the visual output is utilized to let users know what they can say (guideline 37). The voice commands are not limited to the ones shown on the screen since the user can use voice shortcuts to provide directly the channel and the time he is interested in, bypassing the menus (guideline 38).

Guideline 41 states that we should make sure that the speech synthesizer pronounces all the prompts correctly. This might be a challenge in our TV-guide application because, for instance, some program names and actor names may be written in French or German. Partly because of this, we designed the prompts so that the system does not have to pronounce program names or actors at all.¹⁸ This also helped us to keep the prompts quite brief. Guidelines 43, 44, 45, and 49 all call for a design that provides the most essential information with minimal scrolling and keystrokes needed from the user. We wanted to follow closely this principle. The system also uses link lists for input, as guideline 53 suggests, provides labels for soft buttons, as guideline 54 suggests, and uses one of the buttons for back, as guideline 55 implies. For arbitrary text input in querying programs with certain name, the system uses only buttons, like guideline 59 states. Our decision to use the visual modality to present the longer lists was based on guideline 60. Apart from the name query and voice shortcuts, the system offers every functionality with both voice and WAP, as guideline 61 suggests.

We naturally tried to design the layout as clear and pleasant as we could, but we were restricted by the limits of a text-only interface. In the program lists, we decided to show only those properties of programs that are not obvious from the context. Since

¹⁸ One reason for the decision to build a TV-guide was the multilingual nature of the content. We wanted to see what considerations would be needed for such applications. Our finding was that the tools we had available for the multimodal application did not allow changing the language of the synthesizer or the recognizer in the middle of a sentence. On the other hand, the tools that the designers of the voice-only TV-guide application had, allowed them to build a system that was able to switch between Finnish and English. This provided the voice-only application the ability to pronounce the names of most programs and actors.

in the list of programs by type (Figure 5.10) the weekday is mentioned in the heading, there is no need to repeat the information about the day in each list item. There is a similar situation in the list of current programs (Figure 5.4) because it is obvious which day the listed programs are shown. In the list of programs by channel (Figure 5.6), channel information and ending times are also omitted because the first one is mentioned in the heading and the second one can be deduced from the starting time of the next program. The list of programs by name may include programs from several days and channels, so none of the properties could be left out from this list. Since only the name and the starting time had to be listed in the list of programs by channel (Figure 5.6), we decided to use only one line for each program. If a program has a long name, this means that only the beginning of the name fits on the screen. Our idea was that reduced scrolling would complement the lack of consistency in this case. The advantages of this approach are questionable and it later proved not to be even technically feasible.

We considered carefully if we should have included an option to search programs by date and time. The resulting list would have contained programs from every channel at a certain time. We feared that the users would have had difficulties to interpret the list since few users would have been used to see programs listed this way. We decided not to include the list in this phase and discuss the matter again during later phases.

5.4 Heuristic evaluation

In a heuristic evaluation, a small group of evaluators examines the user interface of an application and judges its compliance with the heuristics. The output from the method is a list of usability problems annotated with references to those usability principles that were violated. Because the results are based partly on evaluators' opinions, it is worthwhile to have from three to five evaluators taking part in the study. (Nielsen 1993)

The heuristics Nielsen has suggested for use in heuristic evaluation are listed on page 14. Basics of prototyping and the motivation to use it are described in Chapter 3.1.2.

5.4.1 Research subject

The application we evaluated was the first prototype of the multimodal TV-guide application. It combined features of a vertical and a horizontal prototype in the sense that it did not support all the functionalities that we had designed during the parallel design activity and it was also not connected to a real database containing program information. The menus and the lists that the application produced were static, i.e. the contents remained the same from one session to another. Of course, the final version of the application was planned to contain up-to-date information about TV programs.

Many details of the prototype were implemented differently than we had designed during the parallel design activity. This was due to a number of reasons, which are discussed later in this chapter. Some changes were intentional, namely the reduction

of features. Noteworthy functionalities that we decided to leave out from this prototype were voice shortcuts, listing programs by name, and the ability to browse the program lists freely. The last included links to the next and previous programs as well as ability to change the channel and the time in every program list. All these were due to the missing connection to the database. The creation of menus that would have mimicked these features would have been extremely laborious without the ability to use dynamic menus and the connection to the database.

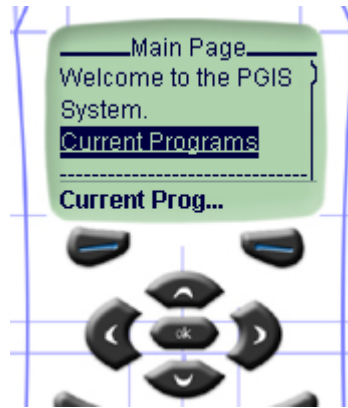


Figure 5.11 Detail of screen capture from Nokia WAP Toolkit.

The structure of the first prototype was following:

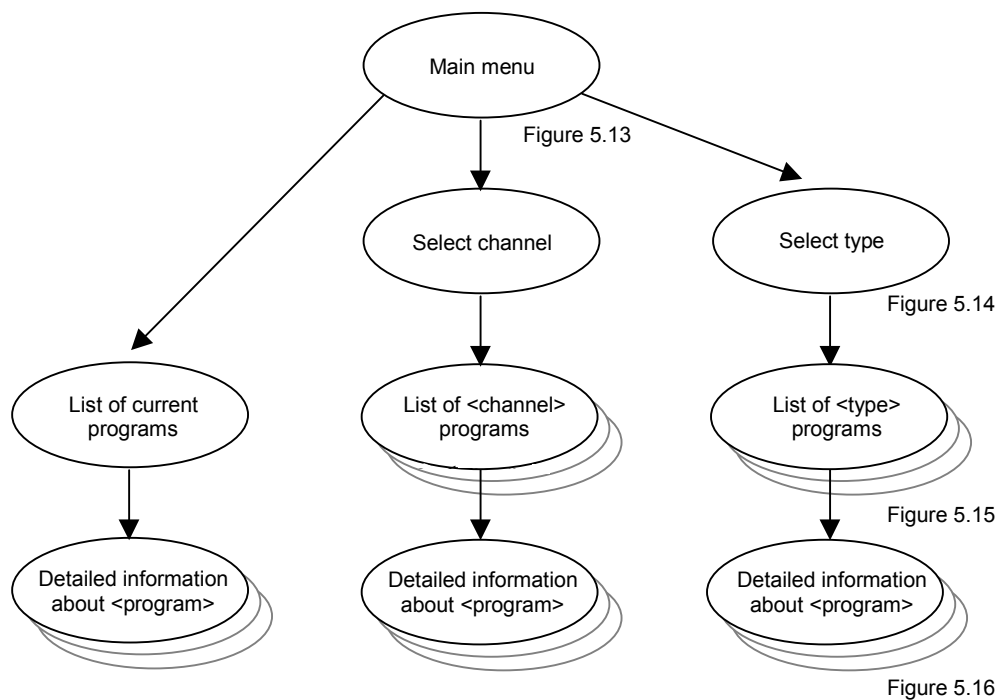


Figure 5.12 Structure of first prototype

The following pictures describe some of the menus of the first prototype in detail. The second and the third prototype will be described in later chapters. The changes in the menus can be easily followed since the same menus of every prototype are always described. The visual user interface is depicted on the left, the prompts are

listed in the middle, and on the right there is a list of the voice commands the users can use in that particular menu. Pictures of the visual menus are combined from several screen captures, so the user actually sees less than four lines of each menu without scrolling. The size difference between actual screen and descriptions of the menus is evident by comparing Figure 5.11 and Figure 5.13.

The main menu of the first prototype:

	<p>Spoken prompts</p> <p>When user enters this page:</p> <ul style="list-style-type: none"> • "Welcome to the Program Guide Information System! You can use this system to search for TV program information. You can list programs by channel, program type or current programs in all channels. You can say current programs or channel or type or repeat." <p>If user is quiet for 15 seconds:</p> <ul style="list-style-type: none"> • "I did not hear you." <p>If user invokes "not understood" -message:</p> <ul style="list-style-type: none"> • "I did not understand you." <p>When user selects link with buttons:</p> <ul style="list-style-type: none"> • "You can get programs by <link>." 	<p>Voice commands user can use</p> <ul style="list-style-type: none"> • "Current programs" • "Channel" • "Type" • "Repeat"
--	---	---

Figure 5.13 Main menu

If the user selects *type*, the following menu will be shown:

	<p>Spoken prompts</p> <p>When user enters this page:</p> <ul style="list-style-type: none"> • Please select the type you are interested in. <p>If user is silent for 15 seconds:</p> <ul style="list-style-type: none"> • "I did not hear you." <p>If user invokes "not understood" -message or asks help:</p> <ul style="list-style-type: none"> • "I did not understand you." <p>When user selects link with buttons or with voice:</p> <ul style="list-style-type: none"> • "You have selected <link>." 	<p>Voice commands user can use</p> <ul style="list-style-type: none"> • "Music" • "Movie" • "News" • "Repeat" • "Back"
--	---	--

Figure 5.14 'Select type' menu

If the user selects *news*, the following menu will be shown:



Spoken prompts

- When user enters this page:
- "The available programs are: World News from 15 hours to 15:05 hours, People In The News from 20:30 hours to 21 hours, Sports Tonight from 23 hours to 0 hours. Please select People In The News or World News or Sports Tonight or back."

If user is silent for 15 seconds:

- "I did not hear you."

If user invokes "not understood" -message or asks help:

- "I did not understand you."

When user selects link with buttons or with voice:

- "You have selected <link>."

Voice commands user can use

- "World News"
- "People In The News"
- "Sports Tonight"
- "Repeat"
- "Back"

Figure 5.15 List of news programs

If the user selects *People in the News*, the following menu will be shown:



Spoken prompts

- When user enters this page:
- "Program People In The News starts at 9:30 hours and ends at 10:00 hours. The channel of broadcast is CNN. Performers in the program are John Smith. The program is classified into type News and subtype political. Restrictions: suitable for everybody. Description: Profiling the most influential newsmakers from all walks of life. Please select back to search again."

If user is silent for 15 seconds:

- "I did not hear you."

If user invokes "not understood" -message:

- "I did not understand you."

When user selects link with buttons or with voice:

- "You have selected <link>."

Voice commands user can use

- "Repeat"
- "Back"

Figure 5.16 Details of program People in the News

If the user selects *channel* from the main menu (Figure 5.13), a menu similar to Figure 5.14 will be shown, except that the menu contains channels instead of program types. If the user selects any channel from the list, he or she will get a program list similar to Figure 5.15. By selecting *current programs* from the main menu, the user will get a menu quite similar to Figure 5.15, but the list also includes the name of the channel where each program will be shown.

5.4.2 Method

Our evaluation team consisted of two evaluators who separately inspected the user interface of the prototype. Evaluators tried to find violations against Nielsen's heuristics and against guidelines we had collected in Chapter 4. After evaluating the prototype, we combined and discussed our findings.

5.4.3 Results and design suggestions

In this chapter, a description and some discussion of each usability problem we found in the heuristic evaluation are presented. The problems and the suggestions are summarized in Table 5.3.

Number: 1

Description: When the user highlights an item, speech output is generated, like "You've selected..." or "List programs by..."

Reasoning: The speech output is not necessary because the visual feedback is quite clear and the speech feedback interrupts the users if they try to concentrate on reading.

Violates: Nielsen's heuristic "Aesthetic and minimalist design"

Suggestion: The speech feedback caused by highlighting an item should be removed.

Number: 2

Description: When the user tries to list e.g. news programs, only starting and ending times (no indication of the date or the weekday), and the name of program is displayed on the screen (see Figure 5.15).

Reasoning: Since programs in this list may be from any channel, the channel should also be included. Either the weekday or the date should be added because the TV-guide may contain music programs from several days.¹⁹

Violates: Guideline 44

Suggestion: The channel and the weekday should be added to those lists where they are not self-evident. Abbreviations of the weekdays should be enough e.g. 'Th' for 'Thursday'.

Number: 3

Description: A large part of the 'program details' screen is used by titles, like 'Time', 'Channel', 'Type', and 'Subtype' (see Figure 5.16). The title 'Time' is also included in each program list.

Reasoning: The titles do not carry any extra information since all the items are obvious without the titles and they take a lot of space that could be used for more valuable information. For instance, a text '21:00-22:50, TCM, romantic movie' is clear and does not need any titles.

¹⁹ This has been implemented differently than in the parallel design. In the parallel design phase this list was intended to contain programs only from one day, which would have been mentioned in the heading, so in that case the weekday would not have been needed.

Violates: Nielsen's heuristics "Match between system and the real world" (the real world being here TV-guides in the newspapers) and "Aesthetic and minimalist design" as well as guideline 48

Suggestion: The titles should be left out from the 'program details' menu and the lists.

Number: 4

Description: The 'program details' screen contains information like 'Restrictions: Suitable for everybody' and 'Performer: John Smith'.

Reasoning: The 'program details' screen is rather long even without these fields. The text 'Performer: John Smith' means actually that there is no information about performers in the database, which could be conveyed to the users by leaving the whole text out. 'Suitable for everybody' is not really a restriction, rather it means that there is no restriction. It is typical that a restriction is mentioned in TV guides only when there is a restriction.

Violates: Nielsen's heuristics "Match between system and the real world" (the real world being here TV-guides in the newspapers) and "Aesthetic and minimalist design" as well as guideline 48

Suggestion: 'Restrictions' and 'Performer' fields should be left out if they have default values like 'Restrictions: Suitable for everybody' or 'Performer: John Smith'.

Number: 5

Description: One of the soft buttons is used to select links. There is also an 'ok' button that the user can use for the same purpose. To get back to the previous page, the user has to scroll to the end of the screen and select a 'back' link.

Reasoning: Soft buttons are buttons that can be associated with shortcut commands. Since there are only two such buttons available, neither one of them should not be wasted on something that can be achieved by clicking another button of the phone.

Violates: Nielsen's heuristics "Consistency and standards" and "User control and freedom" as well as guidelines 49 and 55

Suggestion: Soft buttons should not be used for selecting links, but rather to offer an easier way to use 'back' functionality.

Number: 6

Description: The text 'Welcome to the PGIS system' takes space at the main menu and hides the other options (see Figure 5.13). The acronym 'PGIS' is not familiar to the users.

Reasoning: Acronyms that are unfamiliar to the users do not serve any purpose. The main menu should be designed very carefully to include only valuable content, so the users will not be distracted by auxiliary comments.

Violates: Nielsen's heuristics "Match between system and the real world", "Aesthetic and minimalist design" and guideline 48

Suggestion: The 'PGIS' acronym should not be used in the user interface. Since the text 'Welcome to the PGIS system' is not absolutely necessary, it should be left out altogether.

Number: 7

Description: The users will have to click the 'back' link 3-4 times if they want to go to the main menu from the program details page.

Reasoning: After finding one interesting program, the users might want to search for another program. It is very likely that they start their new search from the main menu, but there is no efficient way to go back to the main menu.

Violates: Nielsen's heuristic "User control and freedom" and guideline 49

Suggestion: A link to the main menu should be added to every card.

Number: 8

Description: It can be seen from the log files of the speech recognizer that quite often the recognizer is unsure what the user has said, although its best guess is actually correct. In such situations the system says, "I did not understand you."

Reasoning: Repeated messages stating that the system cannot understand the user will soon annoy the user. Everything should be done to avoid at least some of these messages.

Violates: Guideline 40

Suggestion: If the recognition engine reports low confidence on a specific utterance, the system should ask for a confirmation, like "Did you say Current Programs?" rather than reject the utterance totally and say, "I did not understand you."

Number: 9

Description: The system reads the contents of the program lists and the program details aloud, like "Program People in the news starts at 9:30 hours and ends at 10 hours. The channel of broadcast is CNN..."

Reasoning: The speech output is quite slow, so the user almost always has read the information from the screen before the computer. Long speeches will also irritate the user if he or she tries to concentrate on reading.

Violates: Nielsen's heuristic "Aesthetic and minimalist design", as well as guidelines 4 and 14

Suggestion: The speech output should be reduced to a bare minimum. The system should just say, "Programs for BBC," or "Details of UK Top 20," and not read the program list or details aloud. The system should take advantage of the main strengths of the visual screen: fast output and the fact that the user is able to easily re-check previous information.

Number: 10

Description: In the 'select type' screen, the first letter of word 'news' is not a capital letter, but first letters of all the other choices are (see Figure 5.14).

Reasoning: The users will be distracted if menus are formatted inconsistently.

Violates: Nielsen's heuristic "Consistency and standards"

Suggestion: All the first letters of the entries in the lists should be written with capital letters.

Descriptions of the problems found and design suggestions are summarized in Table 5.3.

Table 5.3 Results of heuristic evaluation. More elaborate discussion of each problem is available above.

No.	Description of the problem	Suggestion
1	When the user highlights an item, a speech output is generated, like "You've selected..." or "List programs by..."	The speech feedback caused by highlighting an item should be removed.
2	When the user tries to list e.g. music programs, only starting and ending times (no indication of the date or the weekday), and the name of the program is displayed on the screen.	The channel and the weekday should be added to those lists where they are not self-evident. Abbreviations of the weekdays should be enough e.g. 'Th' for 'Thursday'.
3	A large part of the 'program details' screen is used by titles, like 'Time', 'Channel', 'Type', and 'Subtype' (see Figure 5.16). The title 'Time' is also included in each program list.	The titles should be left out from the 'program details' menu and lists.
4	The 'program details' screen contains information, like 'Restrictions: Suitable for everybody' and 'Performer: John Smith.'	'Restrictions' and 'Performer' fields should be left out if they have default values like 'Restrictions: Suitable for everybody' or 'Performer: John Smith.'
5	One of the soft buttons is used to select links. There is also an 'ok' button that the user can use for the same purpose. To get back to the previous page, the user has to scroll to the end of the screen and select the 'back' link.	Soft buttons should not be used for selecting links, but rather to offer easier way to use the 'back' functionality.
6	The text 'Welcome to the PGIS system' takes space at the main menu and hides the other options. The acronym 'PGIS' is not familiar to the users.	'PGIS' acronym should not be used in the user interface. Since the text 'Welcome to PGIS system' is not absolutely necessary, it should be left out altogether.
7	The users will have to click 'back' 3-4 times if they want to go to the main menu from the program details page.	A link to the main menu should be added to every card.
8	It can be seen from the log files of the speech recognizer that quite often the recognizer is unsure what the user has said, although its best guess is actually correct. In such situations the system says, "I did not understand you."	If the recognition engine reports low confidence on a specific utterance, the system should ask for a confirmation, like "Did you say Current Programs?" rather than reject the utterance totally and say, "I did not understand you."

9	The system reads the contents of the program lists and the program details aloud, like "Program People in the news starts at 9:30 hours and ends at 10 hours. The channel of broadcast is CNN..."	The speech output should be reduced to a bare minimum. The system should just say, "Programs for BBC," or "Details of UK Top 20," and not read the program list or details aloud. The system should take advantage of the main strengths of the visual screen: fast output and the fact that the user is able to easily re-check previous information.
10	In the 'select type' screen, the first letter of word 'news' is not a capital letter, but first letters of all the other choices are.	All the first letters of the entries in the lists should be written with capital letters.

5.4.4 Discussion

Many details of the prototype were implemented differently than we had designed during the parallel design activity. For example, the 'quit' feature was not implemented because it was not technically possible. Neither was it technically feasible to use only one line for program names in the list of programs by channel. Many details were also added, like the titles in the 'program details' screen and the welcome message in the main menu. Many of these changes are listed in the previous pages because they violated Nielsen's heuristics and our guidelines.

Several factors contributed to these changes. A programmer who had not taken part in the parallel design activity did the actual implementation of the prototype. He was working at a different location than the designers, so the communication occurred mainly through e-mails. The motivation for the design decisions made during parallel design activity was neither discussed between the members in the team nor documented. The person who implemented the application made his own decisions about some aspects of the interface because the design rationale of the previous work was not available. He was also forced to decide some of the details, like some prompts because the parallel design document did not include all the details it should have included. In case we could have avoided these inconsistencies, we could have concentrated on the usability problems our parallel design had. Now it was more important to discuss the problems in this prototype since we wanted to develop a prototype that we would be able to test with users.

We were able to use only two evaluators, which undoubtedly affected the reliability of the results. It is likely that we were not able to find as many usability problems as we could have found with four or more evaluators. The evaluators' personal opinions may also have affected the study a bit more than would have been the case with a larger group of evaluators. Despite from the restrictions caused by our limited resources, the heuristic evaluation proved to be very useful. We were able to create ten valuable suggestions with very limited effort. This turned out to be also a good point for our development team to discuss the motivations behind the design decisions made in the parallel design activity. This discussion should of course have occurred already before the prototype was made, but it was delayed.

We found violations against seven guidelines and four usability heuristics, which can be seen in the description of the results in the pages 57-59. Most of the problems we found simultaneously violated at least one guideline and one heuristic. This raises a question about the usefulness of the guidelines since many of the problems would have been found with the heuristics only. Our finding was that the guidelines were more concrete, so using them required smaller amount of interpretation and was affected less by the evaluators' personal opinions. On the other hand, we had collected more than 60 guidelines whereas Nielsen lists only 10 heuristics, which naturally made it cognitively more feasible to use the heuristics in the evaluation.

5.5 First usability test

In a usability test, real users use the system to carry out tasks that the designers of the test have asked them to complete. The users should be a comprehensive representation of the intended users of the system. The tasks that the users carry out during the test should also be as representative as possible of the typical tasks people will conduct with the final system. It is common that the users are asked to think aloud during the test. This provides information about how the users interpret each interface item. Often the usability tests are videotaped, so the problems that come up during the test can be thoroughly analyzed. After the test, the user is interviewed and he or she can comment on the system freely. The most important outcome of a usability test is a list of usability problems in the interface and suggestions to improve it. (Nielsen 1993)

Increasing the number of test users who participate the study increases the reliability of the study. Unfortunately, using a large number of participants also results in expensive tests that may take a lot of time to analyze. It is typical that testing with more than 5 users does not provide considerable amount of new information, so in most cases usability tests should be run with about 5 users. (Nielsen 2000)


When conducting usability tests for speech recognition systems, one important aspect has to be taken into account. If the test users are asked to think aloud during the test, the system will interpret the user's verbalized thoughts as spoken commands and it tries to carry out them. This will confuse the user and affect the results of the test. To get information about how the users interpret the prompts, the experimenter can record the interaction and play it back to the user after the test. Before playing the tape, the speech recognition system is turned off. This way the user can comment on the interface and provide information about how he or she understood each prompt. The experimenter may also make some questions after each test task to get some idea of how the user perceives the system. If this method is used, it is important that the experimenter is careful not to influence the user too much with the questions.

5.5.1 Research subject

The application that was tested in this study was the second prototype of the multimodal TV guide. In many respects, it was similar to the first prototype described in Chapter 5.4. The menus and the lists were static because the connection to the database was not available. This meant that the program information had to be invented and added manually to the application. The structure of the prototype was exactly the same as in the first prototype, depicted in Figure 5.12. The system ran on a desktop PC that was controlled with a mouse. The users were wearing a headset with a microphone.

The following pictures describe some of the menus in detail. The previous version of each menu was described in Chapter 5.4.1, so the implemented changes are evident if the descriptions are compared. The changes will also be highlighted in the text. Most changes are due to the results of the heuristic evaluation, but some details have been altered for another reasons.


The main menu of the second prototype:



Spoken prompts	Voice commands user can use
<p>When user enters this page:</p> <ul style="list-style-type: none"> • "Welcome to TV-guide! You can use this guide with voice commands or with buttons on the phone. What would you like to do?" <p>If user is quiet for 15 seconds:</p> <ul style="list-style-type: none"> • "Sorry, but you did not make your choice yet. Possible voice commands are underlined on the screen." <p>If user invokes "not understood" - message:</p> <ul style="list-style-type: none"> • "I did not understand you." 	<ul style="list-style-type: none"> • "Current programs" • "Channel" • "Type" • "Repeat"

Figure 5.17 Main menu

The text 'Welcome to PGIS system' has been removed from the main menu (due to result 6 of the heuristic evaluation). There is no speech feedback when the user selects a menu item (due to result 1 of the heuristic evaluation). If the user selects *type*, the following menu will be shown:



Spoken prompts	Voice commands user can use
<p>When user enters this page:</p> <ul style="list-style-type: none"> • "Which type?" <p>If user is silent for 15 seconds:</p> <ul style="list-style-type: none"> • "Sorry, but you did not make your choice yet. Please select from the list of possible types visible on your screen." <p>If user invokes "not understood" - message or asks help:</p> <ul style="list-style-type: none"> • "I did not understand you." 	<ul style="list-style-type: none"> • "Music" • "Movie" • "News" • "Repeat" • "Main page" • "Back"

Figure 5.18 'Select type' menu

The first letter of 'News' is now written with capital letters (due to result 10 of the heuristic evaluation). The headline suggests now what the user should do in the menu, like was planned in the parallel design activity. In addition, the initial prompt has been changed to the shorter one suggested in the parallel design activity. If the user selects *News*, the following menu will be shown:

	<p>Spoken prompts</p> <p>When user enters this page:</p> <ul style="list-style-type: none"> • "Programs are now on your screen." <p>If user is silent for 15 seconds:</p> <ul style="list-style-type: none"> • "Sorry, but you did not make your choice yet. Possible voice commands are underlined on the screen." <p>If user invokes "not understood" - message or asks help:</p> <ul style="list-style-type: none"> • "I did not understand you." 	<p>Voice commands user can use</p> <ul style="list-style-type: none"> • "World News" • "People In The News" • "Sports Tonight" • "Repeat" • "Main page" • "Back"
--	--	---

Figure 5.19 List of news programs

Information about the weekday is added to each item in the list (due to result 2 of the heuristic evaluation). The text 'Time:' is also removed from each list item (due to result 3 of the heuristic evaluation). Times are now presented in the 12-hour format, which should be more common in English. The system does not read the contents of the program lists aloud anymore (due to result 9 of the heuristic evaluation). In addition, the system tries to guide the user more in case the user cannot decide what to do. If the user selects *People in the News*, the following menu will be shown:

	<p>Spoken prompts</p> <p>When user enters this page:</p> <ul style="list-style-type: none"> • "The details of the program <program name> are now visible on your screen" <p>If user is silent for 15 seconds:</p> <ul style="list-style-type: none"> • "Sorry, but you did not make your choice yet. Please select back or main page to search again." <p>If user invokes "not understood" - message:</p> <ul style="list-style-type: none"> • "I did not understand you." 	<p>Voice commands user can use</p> <ul style="list-style-type: none"> • "Repeat" • "Main page" • "Back"
--	--	---

Figure 5.20 Details of program People in the News

The second prototype still includes a list of current programs, which resembles the Figure 5.19. It also allows the user to list programs by channel. This feature is offered to the user with menus quite similar to Figure 5.18 and Figure 5.19.

5.5.2 Method

A very light version of the usability test method was used. One person planned, conducted, and analyzed the study by himself. Only two test users were used and the users carried out only two test tasks. In addition, the interview after the test was quite light. The whole process had to be finished in two days, so it was not possible to have more users. The test users were both men and they spoke Finnish as their native language. They belonged to the intended target group. The tasks were written in Finnish and the users were interviewed in Finnish, but they used the application in English. The test tasks are available in Appendix A. The test users had no prior experience with a voice recognition system, but both had tried a few WAP applications.

5.5.3 Results and design suggestions

Results and design suggestions are collected to Table 5.4.

5.5.4 Discussion

Suggestion 5 of the heuristic evaluation stating that soft buttons should not be used for selecting links, was not technically possible to implement. When the user interface was inspected again in this phase, we found that the label for the button that was used for selecting links changed depending on the link (see Figure 5.19 and Figure 5.20). This is a violation against Nielsen's heuristic "Consistency and standards", but it was not noticed in the heuristic evaluation because we did not plan to use the button for links at all. It was caught now, so it has been included in the results of the usability study (result 5), although it was not found in the actual usability test. Suggestion 8 of the heuristic evaluation advised to use a confirmation in the case of low confidence on the user's utterance. The tools we had available did not allow this, so this feature was not implemented.

Using only two users in the test is clearly against the suggestions of a reasonable number of test users. We had to be able to analyze the videotapes and write the report of the study in one day, so we could not use more users. There is an obvious danger in making changes to the system based on the opinions of one user. It is possible that the user's preferences differ much from the general opinion. This can be detected only by using several users and checking if other users run into the same kind of problems or comment on the same aspects of the interface. It is also a fact that a usability test with two users can discover only a portion of the usability problems in the interface. Nielsen (2000) suggests that, in a typical case, two users find only half of the problems. In addition, the prioritization of the suggestions was done by only one person. This can be a risk, as discussed in the beginning of this chapter. Despite the problem of too few test users, we wanted to run a light usability test before the application was shown to the public in an international conference. We concluded that the risks involved in presenting an application to the public without any testing with real users would be greater than the ones involved in usability tests with too few

Table 5.4 Results of first usability test

No.	Result	Suggestion	Severity
1	One user understood the main prompt in the main menu so that he needs to choose either voice or buttons.	The prompt should be changed to "Welcome to TV-guide! You can use this guide with voice commands and with buttons on the phone." ("Or" is changed to "and".) Additionally, "What would you like to do?" should not be played when the user is at this menu first time.	High
2	The users did not always state the voice commands exactly as they were written on the screen. For example, instead of "The Hook" the users tried to say, "Hook," and, "Select the Hook."	As long as the vocabulary of the application remains reasonably small, select / view / list / show prefixes should be allowed. Likewise, the articles should be optional i.e. "The Hook" and "Hook" should work identically.	Medium
3	One user commented that he did not like the tone of the prompt "Sorry, but you didn't ... please select..."	It should be changed to "When you are ready to make your choice..."	Medium
4	One user hoped to get more support from the system, when he is in troubles. One user also tried to use longer commands that were not supported.	A new prompt should be added. It should be played after the user has invoked several "not understood" messages. The style of the prompt should be more directive: "You can say..." The same prompt should be used when the user asks help. However, one should not list the names of the programs in the prompt because then the list would become too exhaustive for the user. A more general expression should be used instead: "You can say a name of any program listed on your screen..."	Medium
5	A new look at the interface reveals that the label for the left soft button changes when the user selects different links. This violates Nielsen's heuristic "Consistency and standards".	Label for the left soft button should be always 'Select'.	Low

users. In addition, we had already decided that we would conduct a more elaborate user test later when a larger portion of the user interface is implemented.

In cases where only limited resources are available, Nielsen (1993, 1994b) suggests that one could abandon the videotaping and rely on the notes made during the test. If we had used this method, we would have saved some time that we could have used for testing the system with more users. We decided to try this idea by videotaping the tests but making notes at the same time. If the notes had seemed elaborate enough, we would not have needed the tapes. After the tests with two users, we did not feel that the notes would be enough. This is probably due to two reasons. Thinking aloud was not possible during the test because the speech recognition engine would have confused the verbalized thoughts and speech commands. This meant that the interaction had to be analyzed more closely to see what the users tried to do and how the system reacted. We also wanted to gather some vocabulary for the application, which resulted in a great deal of notes.

The users were using a speech recognition system for the first time, which always makes them a bit uncertain, especially if they have to use a foreign language to control the system. The presence of the experimenter seemed to distract the users a bit in the beginning and they hesitated to use the voice commands. To avoid this effect, we could have let the user use the system alone. The experimenter could have made notes in a room next to the usability laboratory monitoring the test through a semi-transparent window. We decided to have the experimenter in the same room with the user because we expected the user to need help with the system. The positioning of the microphone affected the accuracy of the speech recognition considerably. We suspected that the positioning would be too difficult without the help of the experimenter. The experimenter was also able to guide the users to use a bit louder voice and he could ask the users' first reactions of the system immediately after each task was done. If the system had worked more robustly, it might have been wise to let the users use the system alone. When people have to use a speech recognition system with a foreign language, it is common that many recognition errors occur. This often makes users feel a bit embarrassed because they may think that they do not speak the foreign language well enough.

The usability test provided us with information about the vocabulary that the users might want to use with the system. We got a few comments concerning the prompts as well and saw that with a couple of modifications we would be able to present our prototype to public, without fearing that it would be totally incomprehensible for the audience. With more resources available, we could have used more test users and showed the videotapes of the interaction to each user. With the aid of the tapes, the users could have commented on the application more elaborately. We were not able to build a mobile speech-recognizing prototype. The system ran on a desktop PC that presented a picture of a mobile phone (see Figure 5.11) and the test users had to click the buttons with a mouse. This obviously made the tests slightly artificial since the usage situation was quite different from the normal usage context of a WAP device. On the other hand, we wanted feedback to help us in refining our prototype before presenting it to a conference audience and our tests were able to provide that. Guideline 7 helped us in drafting suggestion 4.

5.6 Second usability test

The basics of the usability test method are described in the beginning of Chapter 5.5.

5.6.1 Research subject

The application tested in the second usability test was a third prototype of the multimodal TV-guide. The application was still not connected to the database but some dynamism was added to the contents of the lists. The prototype was not organized into a strict tree-structure anymore because some shortcuts were implemented. Without the connection to the database, it was not possible to include tens of TV programs in the prototype, but dynamically created program lists were mimicked with a couple of alternative static program lists. The starting times of the programs in these lists changed dynamically to allow listing of programs at any time of the day. With this semi-dynamism, we wanted to test some of the features that we considered would be important for the final application, although real dynamically created menus and lists had to wait until the connection to the database could be implemented. The most important of these features were the voice shortcuts, i.e. the ability to bypass the menus and get any program list with a single utterance. The new features required rewriting the program code of the prototype, which resulted in some changes that were not intended. Some details that had been implemented in the previous prototypes were now unintentionally left out. The system ran in a slim tablet-shaped PC that was controlled with a touch screen and a pen. The users were wearing a headset with a microphone. The test setup is pictured in Figure 5.21.



Figure 5.21 User using multimodal TV-guide with pen and headset

A slightly simplified version of the structure of the third prototype is presented in Figure 5.22. Voice shortcuts were available in every menu, not only in the main menu as the figure presents it.

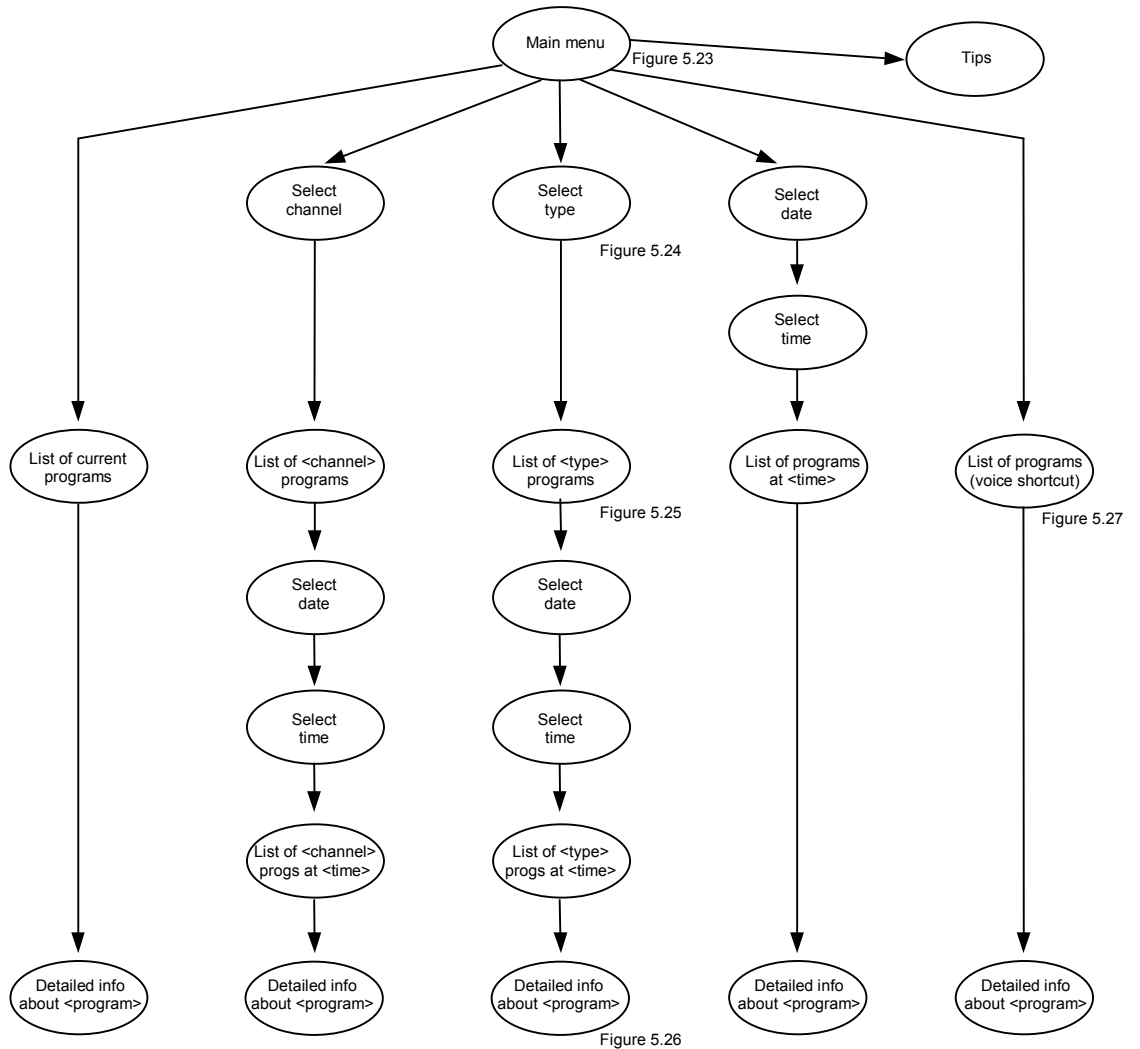
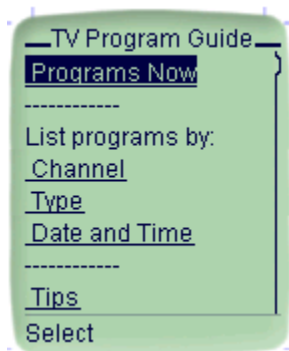


Figure 5.22 Simplified illustration of structure of third prototype

Some of the menus and lists are presented in detail in pages 70 - 72. The changes to the previous prototype are highlighted in the text.

The main menu of the third prototype:



Spoken prompts

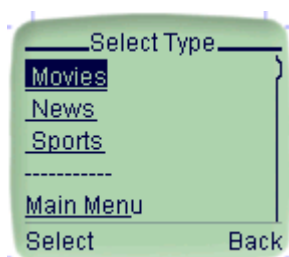
- First visit at this page:
- "Welcome to TV-guide! You can use this guide with voice commands and with buttons on the phone."
- Subsequent visits (not to be played at first visit):
- "What would you like to do next?"
- If user is quiet for 15 seconds:
- "When you are ready to make your choice, you can say any word underlined on your screen."
- Subsequent times user is quiet for 15 seconds:
- "You can speak any of the underlined words. Say 'tips' to hear, how you can find programs more quickly."
- If user invokes "not understood" - message or asks help:
- "Sorry, I did not understand. Try any word underlined on the screen or say 'tips' to hear more."

Voice commands user can use

- "Programs now"
- "Channel"
- "Type"
- "Date and time"
- "Tips"
- "Repeat"
- "Help"

Figure 5.23 Main menu

The label for the left soft button is 'Select' on every menu of the TV-guide (due to result 5 of the first usability test). The initial prompt does not suggest anymore that the user could use only either buttons or voice (due to result 1 of the first usability test). The prompt that is played when the system has not understood what the user has said guides the user to use the underlined words on the screen (due to result 4 of the first usability test). In addition, the prompt that is played after the user has been quiet for 15 seconds is now qualified by a friendlier tone (due to result 3 of the first usability test). There is also an alternative version of the prompt, so the user does not have to listen the same prompt repeatedly. If the user selects *type*, the following menu will be shown:



Spoken prompts

- When user enters this page:
- "What program type do you want?"
- If user is silent for 15 seconds:
- "When you are ready to make your choice, you can say any word underlined on your screen."
- Subsequent times user is quiet for 15 seconds:
- "You can speak any of the underlined words."
- If user invokes "not understood" - message or asks help:
- "Sorry, I did not understand. Try any word underlined on screen."

Voice commands user can use

- "Movies"
- "News"
- "Sports"
- "Repeat"
- "Help"
- "Main menu"
- "Back"

Figure 5.24 'Select type' menu

In the previous versions of the prototype, a text 'Main Page' was used as a link to the main menu. Now it is changed to 'Main Menu.'

If the user selects *News*, the following menu will be shown:



Spoken prompts

- When user enters this page:
- "Programs are now on your screen."
- If user is silent for 15 seconds:
- "When you are ready to make your choice, you can say any word underlined on your screen."
- If user invokes "not understood" - message or asks help:
- "Sorry, I did not understand. You can say the name of any program listed on your screen to get more information about it. You can also say: repeat, back or main menu."

Voice commands user can use

- "World News"
- "Newshour"
- "News Tonight"
- "Next"
- "Next programs"
- "Previous"
- "Previous programs"
- "Date and time"
- "Repeat"
- "Help"
- "Main menu"
- "Back"

Figure 5.25 List of news programs

It was planned that the user could browse the program lists further by selecting *Next programs*, but this feature was not implemented yet. If the user selects *Date and time*, he or she can list news programs that will be shown some other time. Unfortunately, the ending times of programs are not presented in this list anymore. This was one of the details that were not implemented because of the too tight schedule to create the prototype. If the user selects *World News*, the following menu will be shown:



Spoken prompts

- When user enters this page:
- "The program details are now on your screen"
- If user is silent for 15 seconds:
- "When you are ready, you can say repeat, back or main menu."
- If user invokes "not understood" - message or asks help:
- "Sorry, I did not understand. You can say repeat, back or main menu."

Voice commands user can use

- "Repeat"
- "Help"
- "Main menu"
- "Back"

Figure 5.26 Details of program World News

It was also planned that the user could browse the details of programs in a list by selecting *Next on the list*, but the feature was not implemented in this phase. A new feature in this prototype is the ability to get any program list with a single utterance.

If the user says, "News Wednesday at six," the following list will be shown:

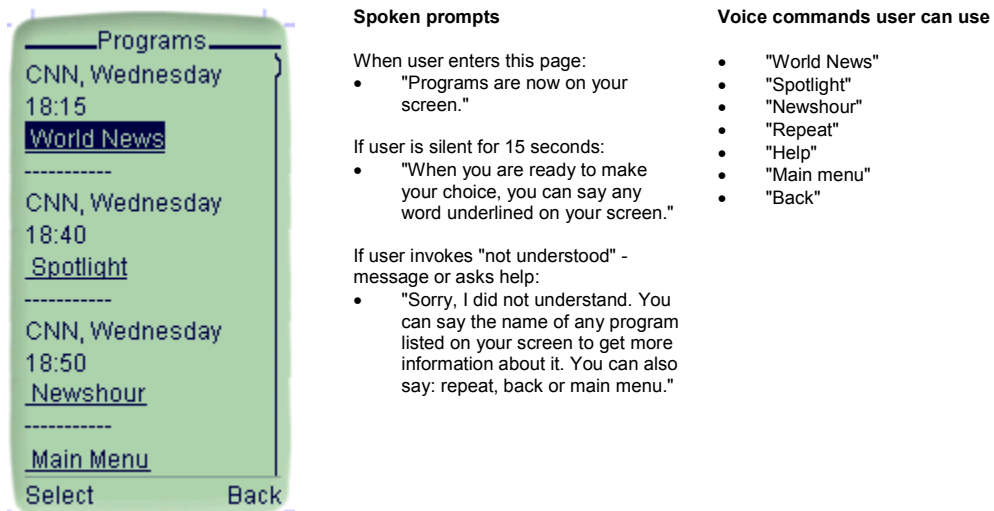


Figure 5.27 List of news programs on Wednesday at six o'clock

In the main menu, there is a link to the tips that explain how these voice shortcuts can be used. In one utterance, the user can specify channel, weekday, and time, like "CNN Wednesday at six," or program type, weekday, and time, for example "News Tuesday at nine." The user may also leave out any of the slots and say for example, "CNN at six." If the weekday is left out, the system interprets that the user wants those programs that will be shown on that particular day. If the time is left out, it is substituted with the current time and, if no channel or type is given, the system provides programs of any type that are on any channel. The application uses the 24-hour format because of its unambiguity.²⁰ A program list generated by a voice shortcut does not contain links to the next or previous programs. It does not allow changing the date and the time either. This behavior was not intended, but it was not noticed before the test took place. If the user selects *Date and time* from the main menu, he can specify the weekday as well as the time and get a list of programs at any channel on that particular time. The menus that allow listing programs by channel are similar to Figure 5.24 and Figure 5.25.

5.6.2 Method

During this activity, we conducted a full-blown usability test, as opposed to the light tests done in the previous round. Two persons planned, conducted, and analyzed the study. Five test users were used and the users carried out six test tasks. Two of the test users were women and three were men. Three of the test users were native English speakers and two spoke Finnish as their native language. The tasks were

²⁰ The time format was changed back and forth during the design process because both options had their benefits and drawbacks. The 24-hour format was finally considered better because it seemed easier for the speech recognition.

written in English and users were interviewed in their native language. The test tasks and the outline of the interview are described in Appendices B and C. Both the test and the interview were videotaped. Two users had tested some voice recognition application before, but none had tried any WAP application. They belonged to the intended target group of the application. The first user acted as our pilot user and a small modification to the contents of the tips was made based on the pilot test.

5.6.3 Results and design suggestions

Results and design suggestions are summarized in Table 5.5.

Table 5.5 Results of second usability study

No.	Result	Suggestion	Severity
1	The way the system responds to brief voice shortcuts (e.g. the voice shortcut that contains only a channel or a type) seems not to be the way the users expect it to respond. The users seem to expect that they can refine their queries one term at a time. If they say first, "BBC," and then a little later, "Wednesday," they expect to get BBC programs on Wednesday. Instead, the system interprets that the user wants any programs on Wednesday.	Implementing the behavior that the users seem to expect would be technically very difficult. Instead, a clarification dialog might help here. If the user says, "BBC," the system could ask, "BBC programs on which day?" and, "Which time on Wednesday?"	High
2	The lists that are created with voice shortcuts or by selecting <i>Date and time</i> from the main menu do not contain links to the next or previous programs. They also do not include link to change the date and the time. Links are also absent from the lists that are created when the user selects <i>Date and time</i> from the end of some program list. Because of this, the users seem to conclude that there will be no more programs on that channel.	Links to the next programs, the previous programs, and programs on another date and time should be added to every list, as they already are in some lists.	High

3	<p>The users seem surprised that when they use the voice shortcut "BBC on Wednesday" the system presents a list of BBC programs on Wednesday exactly 24 hours from this moment (assuming today is Tuesday). For instance, if the current time is 2 p.m., the list of programs on Wednesday also begins from the program that starts at 2 p.m. If the user is querying programs on another day, it is not very likely that the time he or she uses the system would have anything to do with the time he plans to watch TV.</p>	<p>Suggestion 1 of this study would also take care of this problem. Another option would be that the system will present prime time programs if the user specifies date but not time. In case the user is asking programs on that particular day, the current time is a reasonable guess of the time the user is interested in.</p>	Low
4	<p>Voice shortcuts are available from every menu in the application. Hence, the user can use voice shortcuts in menus like select date, select time, select channel, and select type. This sometimes caused the user to be thrown to a quite different list of programs than he or she had expected. The system, at least currently, does not provide enough feedback for the users so that they would be able to detect that the system understood e.g. "Sunday" as "CNN on Sunday."</p>	<p>Since speech recognition is prone to errors, the voice shortcuts should be disabled in the menus that concern selecting something.</p>	Medium
5	<p>The users used several time expressions that the voice shortcuts were not able to handle. Very often the users tried to say e.g., "CNN Sunday at 10 o'clock," and couple of times, "CNN Sunday at 10 p.m." One user also tried, "CNN Sunday evening." In addition, quite often the users used "today" or "tomorrow" instead of the weekday. After some tries with the above mentioned time expressions, some users concluded that the system is not able to handle times at all.</p>	<p>To allow some flexibility in the time expressions, "o'clock", "a.m.", and "p.m." should be allowed. All the time expressions that are listed in 'select time' menu (i.e. afternoon, evening etc.) should also be allowed because all the entries in the rest of the selection menus are allowed in the shortcuts. In addition, "today" and "tomorrow" should be added to the vocabulary because they proved to be quite popular.</p>	Medium

6	At least one user tried to use a direct time expression in the 'select time' menu, such as "seven p.m." The system accepts only expressions like "afternoon" and "evening".	If it is possible to include expressions like "seven," "seven o'clock," "seven a.m.," and "seven p.m." to the vocabulary of this menu, the users will have more direct control of what programs they want to list.	Low
7	In the end of some program lists, there was an option for the user to get a similar list from another day and hour by clicking <i>Date and time</i> . This option was not apparent to all users.	The link will be more obvious if it is changed to e.g. select date & time or change the date.	Low
8	The system reacts so slowly to buttons that the users pressed keys several times unnecessarily.	Some immediate feedback should be provided for pressing buttons. A short sound might be useful here.	Medium
9	Some users ran into a lot of voice recognition problems, so the system kept repeating "Sorry, I didn't understand. You can say..." message. Sometimes the reason for the recognition error was that the microphone was not properly positioned and sometimes the user's utterance was not in the vocabulary of the system. In addition, some users mentioned that they got tired of hearing the options repeatedly.	An incremental prompt should be added. In case of the first recognition error, it should tell what options are and, in case of successive errors, it should give hints about how the user should talk to the system. The users might need guidance e.g. to speak with normal pace and not to pause between words. Prompts might also hint that the user is able to interrupt the system any time.	High
10	One user tried if part of the name of the program would be enough to select it. He commented later that he would have liked that feature. Another user tried to say, "Show tips." The previous usability test done to this system suggested that the users might also say, "Select tips," or "List tips."	As long as the vocabulary of the application remains reasonably small, select / view / list / show prefixes should be allowed. Likewise, the articles should be optional i.e. "The Hook" and "Hook" should work identically.	Medium

5.6.4 Discussion

Our implementation of the voice shortcuts proved to be difficult for the test users. The main source of the difficulty is probably the restricted format that the users have to use in the commands. The channel, the date, and the time have to be provided in the correct order and no extra words are allowed in the utterances. Since these cannot

be regarded as natural language utterances, the users have to learn the format before they can use the shortcuts. Visual aid, in the form of examples, is available in the 'tips' menu but, if the user wants to use the shortcuts in any other menu, he or she needs to use them without any support from the system. With the modifications that were described in suggestions 1, 3, and 5, the voice shortcuts could be made easier to use. In addition, one has to remember that they are mainly intended for the experienced users and our test users had only about ten minutes experience with the system when they were asked to test the shortcuts. Nonetheless, since none of our users was really comfortable with our implementation of the shortcuts, we should conduct one more usability test with the system after the suggested modifications are done. If shortcuts remain too difficult to use, even the experienced users will probably avoid using them.

Suggestions 2, 5, 6, and 10 all add words to the vocabulary, which reduces the recognition accuracy of the system. Because the system already had some problems in recognizing what the users said, one has to be very careful in adding new words to the vocabulary of the system. New tests are necessary before knowing the feasibility of these suggestions.

Some of the intended features were not implemented due to the lack of time and some inconsistencies were not noticed before the tests. Although there were links to the next programs as well as the previous programs in the program lists, clicking these links resulted in a message that explained that this feature was not implemented yet. When the users found this out they seemed slightly distracted but were able to continue the tasks without problems. The links to the next programs, the previous programs, and changing the date were unintentionally left out from some of the program lists. This seemed to cause some additional confusion because many users interpreted that the missing link to the next programs meant that there were no more programs on that channel.

We continued having the experimenter with the test user during the tests because we anticipated that the users would need help with positioning the microphone. We also expected that another person in the same room would not distract native English speakers too much. Our experience from this test confirmed this assumption. The test users tended to repeat in their utterances some expressions that were written in the test tasks. For example, time expressions were often repeated exactly as in the written tasks. In a real field use, the users would have used more varying time expressions, so the test results concerning time expressions and the vocabulary are not as reliable as they could be. In the previous test, the system ran on a desktop PC that was controlled with a mouse. For this test, we had acquired a tablet PC with a touch screen. Using the system with a touch screen and a pen resembles more the usage style of the final product since the user can press the buttons on the screen directly and not with a mouse. Of course, the prototype still was quite far from an actual portable device, so mobility-related aspects could not be tested.

This test provided us with many valuable suggestions on how to improve the prototype. We could have got more comments if we had played the videotapes to the users after the test, but we chose not to do that. We wanted to restrict the amount of material we had to analyze and going through the interaction with the user would have at least doubled the amount of data produced. Guideline 1 helped in drafting

suggestion 8, guideline 7 supported suggestion 9, and guideline 22 was used in compiling suggestion 1.

6. Discussion

6.1 Special considerations in design process

Multimodal interfaces are a fairly incoherent group of interfaces. Because different modalities have different characteristics, the special considerations related to designing multimodal interfaces depend heavily on the chosen combination of modalities. However, one consideration might be valuable in designing any multimodal interface.

In the literature, a designer finds guidance for very few modality combinations. Even if the design team has already decided what modalities the application should support, it is useful to analyze each of the modalities to find their strengths and weaknesses. If one modality is bad for presenting some type of information, there might be another modality that presents the same information well. Such an analysis provides informal information about the tasks for which each modality is best suited. If wanted, this information may be put into the form of guidelines, which may be more straightforward to use in the design process.

Bernsen (2002) has noted that a particular modality is not simply good or bad for presenting a certain type of information. He reminds that, in addition to the type of the task, the intended user groups, the work environment as well as some human learning and cognitive properties have to be considered. Therefore, our analysis of speech modality and the modalities provided by WAP considered these aspects as well. The analysis is described in Chapter 4.3. We focused only on the type of the task, the user groups, and the usage situations to keep the complexity of the analysis within reasonable bounds. Based on the analysis, three guidelines were formed that took into account some aspects that the guidelines found in the literature had not considered. These guidelines helped us to make decisions about what to present with each modality and what we should not even try to present with the modalities we had available. These decisions had a profound effect on the design of the application. For instance, on the basis of the guidelines, we decided to offer most actions with all the available modalities, but dedicate the input of arbitrary text and the output of descriptions of programs exclusively to WAP.

The discussion about the modality analysis concerns a situation where the design team already has selected what modalities the application will support. This selection might already be based on a similar analysis but, if such an analysis were carried out exhaustively, properties of all possible modality combinations the current technology is able to offer would have to be analyzed. Since the task would be infeasible, other methods must be utilized in selecting the modalities. Bernsen (2002) suggests that the designer might rely on those combinations that are tested and found good in some case studies, but the designer has to keep in mind that the results may not be transferable if the user group or the usage situation differs considerably. Some computer-based decision support tools might also be crafted, although the complexity of the problem space makes the task very difficult (Bernsen 2002). In our

case, the decision about the modalities was made already before the project started and it was mainly based on the interests of the participating companies.

Although the above discussion applies to the design of multimodal interfaces in general, the focus of this thesis is on the user-centered design of speech-recognizing multimodal interfaces. Speech differs considerably from the visual modalities that are predominant in the traditional graphical user interfaces. Due to these differences, several considerations are needed in the user-centered design of interfaces that rely on speech. For the sake of concreteness, the following discussion uses Nielsen's (1992, 1993) usability engineering lifecycle as an example of a user-centered design process model because that was used as the basis of our design process.

Traditionally, in the beginning of the design process, "know the user" activity concentrates on users' tasks and needs as well as their previous experience with computers. If speech recognition is used in the interface, the designer also needs to know how the potential users speak about the task domain. Natural dialog studies provide information about the vocabulary and the type of utterances the system must be able to recognize. They also help in designing prompts and give hints about the order in which the users expect topics to appear. We carried out a natural dialog study that provided us with some vocabulary, style of speaking, and order of topics. The usefulness of these results was slightly reduced because, due to the limited resources, we could not conduct the study with two languages. Since both Finnish and English versions of the application were designed in the project, it would have been useful to conduct the natural dialog study with both languages.

In applying guidelines, the ones specific to the exact modality combination used would be most valuable. Since such guidelines do not exist for most of the combinations, the designer has to settle for guidelines specific to each modality, which are more common in the literature. For speech, a lot of guidelines are available in the research literature. The designer may use the literature to draw some guidelines of his own for any modality combination if relevant observations are available in the literature. The modality analysis discussed above may also provide some guidelines that the designer may want to use in the absence of more established guidance. We collected a set of guidelines specific to speech interfaces and a similar set concerning WAP services. We also derived some guidelines based on less formal suggestions in the literature. After the modality analysis, we added a few more to help us in combining the modalities. We had not designed for speech or WAP before, so guidelines were highly useful for us. Both speech and WAP differ a lot from conventional interfaces and the designer must consider many aspects that are specific to them. Without guidelines, it would take many trials and errors before the resulting interface would be satisfying. Of course, guidelines do not automatically lead to good interfaces. Some user involvement, like prototyping and usability testing, is needed. The role of guidelines in the design process is discussed separately later.

Prototyping is potentially a more difficult task with multimodal systems. There should not be any reason why low-fidelity paper prototypes would be impossible, but often a lot of effort will be needed in creating such. If the prototype must simulate many recognition-based input modes, several human assistants might be needed to act in the roles of different recognizers. Such prototypes are very common in the development of speech interfaces and they are used to collect words to the

vocabulary for the application (Bernsen et al. 1998). These studies are usually called Wizard of Oz studies. We made some early plans of a paper prototype of our system, but we came to the conclusion that we were lucky in selecting such modalities and tools that allowed us to create electronic prototypes with almost the same effort as paper prototypes. Our prototypes were described using VoiceXML and WML markup languages, which were also the basis of the final application. The prototypes only simulated dynamism and did not provide all the features the final system did, but all of our prototypes used computerized speech recognition.

The speech modality creates many considerations in the course of the usability testing activity. Typically, the users are asked to think aloud during a usability test. This is not possible in testing speech interfaces because the speech recognition system would not know which of the utterances it should consider as a speech command and which it should ignore as a thinking-aloud utterance. Instead, the interaction between the user and the computer can be recorded and played back to the user after the test. With the help of the tapes, the user can comment on his or her thoughts in the different phases of the study. A substitute that provides less information but also demands fewer resources is to ask the user some questions from between the test tasks. We employed the latter approach because we did not have enough time to analyze the amount of data that the first approach would have created. Compared to our experience from previous usability tests, where we had played the tapes to the users after the test, drawing conclusions about the usability tests described in this thesis was more difficult. Nevertheless, we were able to draft many valuable suggestions for enhancing the interface. If the resource constraints are tight, it is possible to get reasonable results with the approach used in this thesis, but if resources allow, it will pay off to allow the users to comment on the interface after the test with the aid of tapes.

Recording the interaction during a usability test is even more important in testing speech interfaces than in testing graphical user interfaces. The tapes help in collecting words to the vocabulary and they can be used in enhancing the recognition accuracy of the system. A person conducting usability tests on a speech interface has to make a decision about staying with the user during the test or leaving the user alone while he or she is performing the test tasks. First-time users of speech recognition systems may feel uncomfortable if the experimenter openly observes their use of the system. On the other hand, if the experimenter stays with the user, he or she can help the user if the user has too severe problems with the tasks. We used the latter approach in both our tests because the system was very sensitive to the position of the microphone and we anticipated that the users would need help in aligning the microphone. The users who had to use a language other than their own did seem slightly uncomfortable about the situation, but native speakers of English seemed quite relaxed through the study. The test users repeated in their utterances some expressions that were used in the written test tasks, which would not have happened in real use. This reduced the reliability of the results concerning the vocabulary. Bernsen et al. (1998) noticed a similar effect in their Wizard of Oz studies. They found that changing a written time expression to the drawn face of a clock reduced this effect considerably.

6.2 Role of guidelines

The experiences from our design process supported the outcome of the analysis in Chapter 3.3, which stated that applying the guidelines should not be considered as a separate activity in the usability engineering lifecycle. Rather, it should be seen as a meta-method that focuses work in almost every design activity. Although this is in line with Smith's (1988) opinion, it contradicts somewhat Nielsen's (1992, 1993) view. Nielsen (1992, 1993) presents the application of guidelines as a part of the design phase of the usability engineering lifecycle and he does not suggest that iteration would be needed between the pre-design and the design phases. In our process, each design activity benefited from the guidelines and, for example, the studies in the pre-design phase might not have provided as valuable information without the guidelines as they provided now.

6.3 Relevant guidelines for speech and WAP

Speech interface guidelines and WAP guidelines are widely available. Some examples can be found in Chapters 3.2.1 and 3.2.2. During the design process, we removed the overlapping guidelines and formed a more coherent list of guidelines. To complement some deficiencies in the lists we had found in the literature, we added a few guidelines of our own that were either inspired by informal suggestions found in the literature or the modality analysis discussed above. The list of the guidelines is available in Chapter 4. We applied the guidelines during the design process and felt that the list was comprehensive enough to help us in designing our application. Some more guidelines concerning the interplay of speech and WAP might have been valuable.

Our list contained more than 60 guidelines, which was too many to be practical. It was quite difficult to keep the guidelines in mind while designing the system or evaluating it. By further simplifying the list or by structuring it more, we might have made our design task easier. At least the guidelines concerning prompt design should be divided into smaller groups. We utilized approximately half of our guidelines and most of them during the parallel design activity. This does not imply that the other half of the guidelines, which were not used in this design process, would be useless because, in the design of a different application, different aspects would have been emphasized. For example, in the design of a more natural dialogue, more sophisticated error recovery techniques would have been more important and corresponding guidelines would have been used more extensively. Previous work by Tetzlaff and Schwartz (1991) suggests that guidelines will be easier to apply if they are illustrated with visual examples. For speech interface guidelines, visual examples are difficult to construct, but WAP guidelines might have benefited from illustrations.

6.4 Design process

The user-centered approach we chose for our design process was successful. The user studies in the pre-design phase provided a basis that we could build on and later studies brought concrete data about how the users were able to use our prototypes.

Although our design process was based on Nielsen's (1992, 1993) usability engineering lifecycle, we had to make some modifications to it. We left out some activities because we did not have resources to follow every detail of the model. We also made some changes to the model due to the speech in the interface, as already discussed above.

With our limited resources, we tried to make quite many studies and ended up making the studies with a small number of participants and limited possibilities in analyzing the studies. By concentrating our efforts on fewer studies, we might have got more out of each study. More resources for the pre-design phase would have allowed us to interview the diary study participants and analyze more dialogs in the natural dialog study. With more resources in the design phase, we might have used more designers working separately in parallel design and more evaluators in the heuristic evaluation. In addition, we might have acquired more users for the first usability study and we could have gathered more comments from the users in the second usability study by playing tapes of the interaction to each user. All these would have been valuable, but concentrating more on some studies would have meant that we would have had to leave some of the studies out. Since all of the studies provided important information, we would also have lost something if we had concentrated our efforts on fewer studies. The large number of studies provided versatile results, which would not have been possible with fewer studies.

If we had had more resources, we might also have put more effort into documentation. Had the rationale of all design decisions been documented, we might have been able to avoid many of the misunderstandings between different members of the design team. Now it took some iteration with the prototypes before the design team had common understanding of all the details concerning the application.

6.5 Successfulness of our research

We were able to acquire experience from the user-centered design of speech-recognizing multimodal interfaces and special considerations related to it. We also collected guidelines that we can use in future projects and built a demonstration of a multimodal system. Many improvements could have been done to the guidelines, but because we wanted to concentrate more on the process as a whole, we did not go very deeply into refining the guidelines. Most of the guidelines concern only either speech or WAP. The list might be more useful if the guidelines were synthesized to cover similar tasks regardless of the modality. However, it may prove to be impossible to synthesize guidelines because of the distinct features of speech recognition. The system we built could have used multimodality more extensively, but the tools we had did not allow advanced features, like coordinated input from several modalities.

7. Conclusions

7.1 Research questions

The results of this research indicate some special considerations that may be needed in the user-centered design of speech-recognizing multimodal interfaces. The nature of the research was constructive, so one has to remember that the results were mainly based on experiences from one design process. Therefore, scientifically valid generalizations are out of the scope of this thesis.

Our experiences indicate that the designer of a multimodal system should analyze the strengths and the weaknesses of each modality the system will support. In the analysis, the designer should consider how well each of the modalities is able to present different types of information for each of the user groups in several possible usage situations. The results of the analysis will provide guidance for the designer in important decisions concerning how to divide the functionalities of the application between the available modalities.

If the system utilizes speech recognition, the designer has to know how the potential users speak about the task domain. A natural dialog study provides information about the vocabulary as well as the type and the style of utterances. Additionally, it may provide hints for designing prompts and for deciding the order of topics. Prototypes that simulate recognition-based modalities, like speech, may require human assistants that act in the role of recognizers. Such prototypes of speech recognition systems are often called Wizard of Oz studies and they are used to refine the vocabulary of the application.

A designer planning a usability test for a speech-recognizing system has several aspects to consider. The users cannot be asked to think aloud during the study, which means that another method must be used for collecting comments about the system. If a research team has resources to analyze the resulting data, the interaction between the user and the system may be recorded and played back to the user after the test. Hearing the recording helps the user to remember his or her thoughts during the test and he or she can explain them to the experimenter. Research teams with fewer resources may consider asking the users some questions between the test tasks. Recording the usability tests is important because the tests provide refinements for the vocabulary of the application, so the recordings can be used to enhance the recognition accuracy of the system. If the experimenter stays with the user during the test tasks, some users may feel uncomfortable about speaking to the computer, especially if they have to use a language other than their own. The experimenter should stay with the user if it is likely that the user will need some help with the system, otherwise he or she should leave the user alone during the test tasks. In their utterances, the users tend to repeat the expressions that are used in the written test tasks. To avoid this, visual equivalents of important expressions might be tried instead.

Guidelines provide suggestions that may help the designer in almost any phase of user-centered design. For example, some guidelines imply a need for pre-design

studies, so guidelines should be considered already before any pre-design studies are made. The most relevant guidelines that are available for an interface that utilizes speech and WAP are the ones that concern either speech interfaces or WAP interfaces. Additionally, a few informal suggestions exist for multimodal interfaces that combine speech with some visual modality. Guidelines for the specific combination of speech and WAP are not available. We collected a list of modality-specific guidelines we found in the literature. Because we did not feel that they would be enough, we formed some guidelines of our own for the combination of speech and WAP based on suggestions in the literature and the modality analysis. The resulting list is available in Chapter 4.

7.2 Reliability and limitations of results

The results about special considerations were based on a study of the related literature and experiences gathered from designing a multimodal TV-guide. A study of one design process is not sufficient to verify any of the results, rather it just indicates that these findings deserve more attention. The results concerning natural dialog studies and usability testing are limited to speech interfaces, but are not necessarily limited to multimodal interfaces. Findings about prototypes concern any application that takes advantage of recognition-based modalities, whereas modality analysis may prove to be useful in the design of any multimodal interface.

Both the study of the literature and the experiences from our design process suggested the conclusions about the role of guidelines. They are not limited to the design of speech interfaces or multimodal interfaces, instead they should apply to any user-centered design process. The list of relevant guidelines for interfaces that combine speech and WAP was created on the basis of the literature and our modality analysis. There would have been several options for the sources of the guidelines, which means that another person collecting a similar list might have chosen a slightly different set of guidelines. Most likely, the list would still have dealt with the same issues even if the actual guidelines had been from different sources. Guidelines 1 - 36 and 39 - 41 are specific to speech interfaces, guidelines 42 - 58 are specific to WAP services, guidelines 59 - 61 are specific to combination of speech and WAP, whereas guidelines 37 and 38 are specific to combination of speech and any visual modality. Our list might also help a designer trying to combine some other modalities with either speech or WAP. The guidelines are neither specific to VoiceXML, which we used to describe the voice menus, nor are they specific to the browsers we used to present the menus. Capabilities of the tools we used set the limits of the areas where we looked for guidelines, although the guidelines were not specific to them.

There are some concerns about the reliability of the results of the individual studies because the small number of participants might have affected the diary study, the natural dialog study, and the first usability study. To increase the reliability of pre-design studies, their results were reviewed together before making any design decisions. Naturally, most of the studies are highly specific to this particular application. The only exceptions might be the diary and the natural dialog studies because they contain some observations about people's use of TV program related information and the way they talk about it.

7.3 Future research

The design process of the multimodal TV-guide should be continued based on the suggestions in the second usability study. A longer lasting user study should be conducted after a robust portable version of the system is developed. The study should provide information about how well people learn to use the voice shortcuts when they have sufficient time to become acquainted with them.

To confirm the indications about the special considerations needed in the user-centered design of multimodal interfaces, one should analyze several similar design processes and check if they indicate the same considerations. The effect of visual expressions in usability study tasks should be tested. The work with the guidelines should also be continued to make the most out of them in future projects. The number of guidelines should be reduced, they should be structured further and visual examples should be added, where possible. After that, the list should be tested again. Its capability to cover most of the possible problems should be evaluated by classifying a large set of interaction problems with systems that combine speech and WAP. If the list of guidelines is able to cover the most frequent problems, its applicability should be tested. The list should be given to a group of designers, who should be asked to create test designs with the aid of the guidelines. The comments from the designers and the usability of the resulting designs should then be analyzed.

To help the designer of a multimodal application, the problem of combining modalities should be further investigated. The designer needs to know the circumstances where simultaneous use of several modalities makes the interaction more difficult for the user rather than more efficient. Extensive tests need to be made before established guidance is available about this subject.

References

- Bernsen, N. O. (2002). Multimodality in Language and Speech Systems - From Theory to Design Support Tool. In Granström, B., House, D. & Karlsson, I. (ed.). 2002. *Multimodality in Language and Speech Systems*. (Text, Speech and Language Technology, Vol. 19). Dordrecht: Kluwer Academic Publishers. Available at <http://www.nis.sdu.dk/~nob/publications/MILASS-CHAP-3.10.pdf> [checked December 6, 2002].
- Bernsen, N. O., Dybkjær, H. & Dybkjær, L. (1998). *Designing Interactive Speech Systems*. London: Springer-Verlag.
- Bernsen, N. O. & Dybkjær, L. (2001). Combining Multi-Party Speech and Text Exchanges over the Internet. In *Proceedings of 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*. Aalborg, September 3-7
- Bernsen, N. O. & Luz, S. (1999). SMALTO: Advising Interface Designers on the Use of Speech in Multimodal Systems. In Ostermann, J., Ray Liu, K. J., Sørensen, J. A., Deprettere, E. and Kleijn, W. B. (ed.). 1999. *Proceedings of IEEE Workshop on Multimedia Signal Processing*. Piscataway, NJ: IEEE. Available at <http://www.nis.sdu.dk/~nob/publications/IEEE99-F.pdf> [checked December 6, 2002].
- Bernsen, N. O. & Verjans, S. (1997). From Task Domain to Human-Computer Interface, Exploring an Information Mapping Methodology. In Lee, J. (ed.). 1997. *Intelligence and Multimodality in Multimedia Interfaces: Research and Applications*. Menlo Park, CA: AAAI Press. Available at <http://www.nis.sdu.dk/~nob/publications/IMAP-29.3.pdf> [checked December 6, 2002].
- Beyer, H. & Holtzblatt, K. (1998). *Contextual Design: Defining Customer-Centered Systems*. San Francisco, CA: Morgan Kaufmann Publishers.
- Boyce, S. & Gorin, A. (1996). User Interface Issues for Natural Spoken Dialog Systems. In *Proceedings of International Symposium on Spoken Dialog (ISSD '96)*. Philadelphia, PA. October.
- Buchanan, G., Farrant, S., Jones, M., Thimbleby, H, Marsden, G. & Pazzani, M. (2001). Improving Mobile Internet Usability. In *Proceedings of International World Wide Web Conference (WWW10)*. Hong Kong. May 1-5. Available at <http://www10.org/cdrom/papers/pdf/p230.pdf> [checked December 6, 2002].
- Cohen, P. & Oviatt, S. L. (1994). The Role of Voice in Human-Machine Communication. In Roe, D. & Wilpon, J. (ed.). 1994. *Voice Communication between Humans and Machines*. Washington, DC: National Academy Press. 34-75.
- Eysenck, M. W. & Keane, M. T. (1995). *Cognitive Psychology: A Student's Handbook* (3rd edition). Hove: Lawrence Erlbaum Associates.

- Gould, J. D. & Lewis, C. (1985). Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM*. Vol. 28, No. 3.
- Grimstad, T., Stegavik, H. & Saastad, E. (2000). *User Interface Design Guidelines for WAP Applications, Version 1.4*. Fornebu: Telenor.
- ISO 13407. (1999). *Human-centred design processes for interactive systems*. Genève: International Organization for Standardization.
- Kamm, C. (1994). User Interfaces for Voice Applications. In Roe, D. & Wilpon, J. (ed.). 1994. *Voice Communication between Humans and Machines*. Washington, DC: National Academy Press. 422-442.
- Kamm, C. & Walker, M. A. (1997). Design and Evaluation of Spoken Dialog Systems. In *Proceedings of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*. Santa Barbara, CA. December 14-17.
- Kleindienst, J., Seredi, L., Kapanen, P. & Bergman, J. (2002). CATCH-2004 Multi-modal Browser: Overview Description with Usability Analysis. In *Proceedings of IEEE International Conference on Multimodal Interfaces (ICMI '02)*. Pittsburgh, PA. October 14-16.
- Mané, A., Boyce, S., Karis, D. & Yankelovich, N. (1996). Designing the User Interface for Speech Recognition Applications. *SIGCHI Bulletin*. Vol. 28, No. 4. Available at <http://www.acm.org/sigchi/bulletin/1996.4/boyce.html> [checked December 6, 2002].
- Molich, R. & Nielsen, J. (1990) Improving a human-computer dialogue. *Communications of the ACM*. Vol. 33, No. 3.
- Murray, J., Schell, D. & Willis, C. (1997). User Centered Design in Action: Developing an Intelligent Agent Application. In *Proceedings of the 15th Annual ACM SIGDOC Conference on Computer Documentation*, Salt Lake City, UT, October 19-22.
- Nielsen, J. (1992). The Usability Engineering Life Cycle. *IEEE Computer*. Vol. 25, No. 3. 12-22.
- Nielsen, J. (1993). *Usability Engineering*. Boston, MA: Academic Press.
- Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*. Boston, MA. April 24-28.
- Nielsen, J. (1994b). Guerilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. In Bias, R. G. & Mayhew, D. J. (ed.). *Cost-Justifying Usability*. 1994. San Diego, CA: Academic Press. 245-272. Available at http://www.useit.com/papers/guerrilla_hci.html [checked December 6, 2002].
- Nielsen, J. (1994c). Heuristic Evaluation. In Nielsen, J. & Mack, R. (ed.). 1994. *Usability Inspection Methods*. New York, NY: John Wiley & Sons.

- Nielsen, J. (1995). Scenarios in discount usability engineering. In Carroll, J. (ed.). 1995. *Scenario-Based Design, Envisioning Work and Technology in System Development*. New York, NY: John Wiley & Sons. 59-83.
- Nielsen, J. (2000). Why You Only Need to Test with 5 Users. *Jakob Nielsen's Alertbox*. March 19, 2000. Available at <http://www.useit.com/alertbox/20000319.html> [checked December 6, 2002].
- Nigay, L. & Coutaz, J. (1993). A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion. In *Proceedings of ACM SIGCHI & IFIP TC.13 Conference on Human Factors in Computing Systems (INTERCHI '93)*. Amsterdam. April 24-29.
- Oviatt, S. L. & Olsen, E. (1994). Integration Themes in Multimodal Human-Computer Interaction. In *Proceedings of International Conference on Spoken Language Processing (ICSLP '94)*. Yokohama. September 18-22.
- Oviatt, S. L. (1999a). Mutual Disambiguation of Recognition Errors in a Multimodal Architecture. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. Pittsburgh, PA. May 15-20.
- Oviatt, S. L. (1999b). Ten Myths of Multimodal Interaction. *Communications of the ACM*. Vol. 42, No. 11. Available at <http://www.cse.ogi.edu/CHCC/Papers/sharonPaper/Myths/myths.html> [checked December 6, 2002].
- Oviatt, S. L., Cohen, P. (2000). Multimodal Interfaces That Process What Comes Naturally. *Communications of the ACM*. Vol. 43, No. 3.
- Oviatt, S. L., Cohen, P. R., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. & Ferro, D. (2000). Designing the User Interface for Multimodal Speech and Pen-based Gesture Applications: State-of-the-Art Systems and Future Research Directions. *Human-Computer Interaction*. Vol. 15, No. 4. Available at <http://www.cse.ogi.edu/CHCC/Publications/hci2000/hci.htm> [checked December 6, 2002].
- Raisamo, R. (1999). *Multimodal Human-Computer Interaction: A Constructive and Empirical Study*. Academic Dissertation. Tampere: University of Tampere. Department of Computer Science. Available at <http://acta.uta.fi/pdf/951-44-4702-6.pdf> [checked December 6, 2002].
- Reber, A. S. (1985). *The Penguin Dictionary of Psychology*. Harmondsworth: Penguin Books.
- Rieman, J. (1993). The Diary Study: A Workplace-Oriented Research Tool to Guide Laboratory Efforts. In *Proceedings of ACM SIGCHI & IFIP TC.13 Conference on Human Factors in Computing Systems (INTERCHI '93)*. Amsterdam. April 24-29.
- Rosenzweig, E. (1996). Design Guidelines for Software Products: A Common Look and Feel or a Fantasy? *ACM Interactions*. Vol. 3, No. 5.

Schmandt, C. (1993). *Voice Communication with Computers*. New York, NY: Van Nostrand Reinhold.

Schmidt, A., Schröder, H. & Frick, O. (2000). WAP - Designing for Small User Interfaces. In *Extended Abstracts of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2000)*. Hague. April 1-6.

Schomaker, L., Nijtmans, J., Camurri, A., Lavagetto, F., Morasso, P., Benôit, C., Guiard-Marigny, T., Le Goff, B., Robert-Ribes, J., Adjoudani, A., Defée, I., Münch, S., Hartung, K. & Blauert, J. (1995). *A Taxonomy of Multimodal Interaction in the Human Information Processing System*. A Report of the ESPRIT project 8579 MIAMI. Available at <http://www.cogsci.kun.nl/~miami/taxonomy/taxonomy.html> [checked December 6, 2002].

Shneiderman, B. (1987). *Designing the User Interface*. Reading, MA: Addison-Wesley Publishing.

Sinkkonen, I. (1996). Yrityskohtaiset tyylioppaat. In Kalimo, A. (ed.). 1996. *Graafisen käyttöliittymän suunnittelu, Opas ohjelmistojen käytettävyyteen*. Espoo: Suomen ATK-kustannus.

Sinkkonen, I., Kuoppala, H., Parkkinen, J. & Vastamäki, R. (2002). *Käytettävyyden psykologia*. Helsinki: IT-Press.

Smith, S. L. (1988). Standards Versus Guidelines for Designing User Interface Software. In Helander, M. (ed.). 1988. *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier Science Publishers. 877-889.

Smith, S. L. & Mosier, J. N. (1986). *Guidelines for Designing User Interface Software*. Bedford, MA: MITRE. ESD-TR-86-278. Available at <ftp://ftp.cis.ohio-state.edu/pub/hci/Guidelines/> [checked December 6, 2002].

Tetzlaff, L. & Schwartz, D. R. (1991). The Use of Guidelines in Interface Design. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*. New Orleans, LA. April 27-May 2.

Walker, M. A., Fromer, J., Di Fabbrizio, G., Mestel, C. & Hindle, D. (1998). What Can I Say?: Evaluating a Spoken Language Interface to Email. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '98)*. Los Angeles, CA. April 18-23.

Yankelovich, N. (1994). *SpeechActs & The Design of Speech Interfaces*. Presented at CHI 94 Workshop on the Future of Speech and Audio in the Interface. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '94). Boston, MA. April 24-28. Available at <http://research.sun.com/speech/publications/chi94/CHI94Workshop.ps> [checked December 6, 2002].

Yankelovich, N. (1996). How Do Users Know What to Say? *ACM Interactions*. Vol. 3, No. 6. Available at <http://research.sun.com/speech/publications/acm-interactions-1996/Interactions.html> [checked December 6, 2002].

Yankelovich, N. (1997). *Using Natural Dialogs as the Basis for Speech Interface Design*. Available at <http://research.sun.com/speech/publications/mit-1998/MITPressChapter.v3.html> [checked December 6, 2002]. Submitted to MIT Press as a chapter for the upcoming book Luperfoy, S. (ed.). *Automated Spoken Dialog Systems*.

Appendices

A Tasks used in first usability test

The tasks were given to the users in Finnish, but they are translated to English for this thesis.

Original tasks in Finnish:

1. Haluaisit katsoa BBC World -kanavan uutiset joten tarkista, milloin ne alkavat.
2. Muistelet, että illalla tulisi elokuva nimeltä The Hook. Etsi keitä näyttelijöitä siinä esiintyy. Käytä ensisijaisesti puhetta sovelluksessa etenemiseen.

English translation of the tasks:

1. You would like to watch news from the BBC World, so check when the program begins.
2. You remember that tonight there will be a movie called The Hook on the TV. Find out who will be the main actors of the movie. Use the system mainly with voice commands.

B Tasks used in second usability test

The first users took part in the test on Friday, the next users on following Monday, and the last user on Wednesday. Task 3 was slightly modified for each day so that those users that participated the test on Friday looked for the Olympics on Sunday and those that participated test on Monday queried for the Olympics on Wednesday etc.

1. You have just arrived home from work and opened the TV. There seems to be an interesting movie on BBC. You would like to know when it has started and also check who are the main actors of the movie.
2. Check what will be on BBC after the movie you are now watching.
3. You haven't had much time to watch the Olympic games yet, but on [Sunday / Wednesday / Friday] it seems that you don't have anything better to do. Check, whether they'll be showing the Olympic games then around seven p.m.
4. You have promised to pick up your friend at the airport tomorrow and you'll have to leave at 6.40 p.m. Before that you would like to watch the news. Check, at what time there would be news before you'll have to leave.
5. Now, on the main page there is a link to tips. Check what kind of tips there are and perform the previous task according to the directions mentioned.
6. Your friend is coming to visit you tonight at 10. Using a voice shortcut, check whether there would be a nice movie on TV at that time.

C Outline of interview in second usability test

The interview was semi-structured, i.e. the order and the exact formulation of the questions varied. Some subjects were explored further if the user's answers indicated interesting issues. The test began with a short briefing that was followed with some questions. The briefing concerned practicalities of the test and background information about the application. The questions were the following.

- Have you used any speech-recognizing device earlier?
- How much do you use mobile phone?
- Have you used any WAP service earlier?
- How much have you used them?
- What kind of WAP services?
- Could you tell approximately how old are you?
- What kind of job tasks you have?

After the questions, the user was shown how to scroll the screen of the mobile phone and how soft buttons are used. Between some of the test tasks, the user was asked following questions.

- Did the system understand you the way you meant it?
- Was it clear how the system understood you?
- Did you get the answer you hoped to get?
- Was there something that surprised you?

After the user had completed all test tasks, the user was asked some more questions.

- How did you feel about discussing with the system?
- Was it difficult to understand what the system spoke?
- Did the system understand you well enough?
- How natural it felt to use the system?
- How natural it felt to switch between speaking to the system and pressing the buttons?
- Would you rather speak to the system or press the buttons if both were available?
- Would you use this kind of service if you had one available?
- In what kind of situation?
- How good approximation of using an actual mobile phone you felt that the pressing the buttons on the screen with a pen was?