The MUST guide to Paris

Implementation and expert evaluation of a multimodal tourist guide to Paris

L. Almeida[1], I. Amdal[2], N. Beires[1], M. Boualem[3], L. Boves[4], E. den Os[5], P. Filoche[3], R. Gomes[1], J. E. Knudsen[2], K. Kvale[2], J. Rugelbak[2], C. Tallec[3], Narada Warakagoda[2]

[1]*Portugal Telecom Inovação,* [2]*Telenor R&D,* [3]*France Telecom R&D,* [4]*University of Nijmegen,* [5]*Max Planck Institute for Psycholinguistics*

Abstract: In this paper we present the implementation and expert evaluation of a speech centric multimodal demonstrator that has been developed in the EURESCOM[1] MUST project (MUltimodal, multilingual information Services for small mobile Terminals). The demonstrator is a tourist guide for Paris. The paper focuses on the technical implementation and interface design of the demonstrator. Based on GALAXY Communicator software, Telenor and Portugal Telecom were able to build transparent, modular, and stable versions of the demonstrator in a relatively short time. User Interface experts at Telenor and Portugal Telecom evaluated the demonstrator in two phases. In phase one they explored the interface. Phase two was a Cognitive Walkthrough with predefined tasks and action sequences. It appeared that it was not obvious that the interface was multimodal, and in particular that it was possible to tap and talk simultaneously. However, some experts discovered simultaneous multimodal interaction after a while, and we observed a very steep learning curve. The experts foresee problems for naïve users, if no special attention is paid to the introduction phase. The implications of the expert evaluation for the planned user tests are discussed at the end of the paper.

Key words: speech centric multimodal application, simultaneous coordinated multimodality, expert evaluation

[1] EURESCOM, the European Institute for Research and Strategic Studies in Telecommunications, is the leading company for collaborative R&D in telecommunications in Europe. Founded in 1991, EURESCOM provides comprehensive collaborative research management services to network operators, service providers, suppliers and vendors

1. INTRODUCTION

For Telecom Operators, due to the large investments made, it is essential to invoke the widest possible use of their future UMTS services. To be successful, these new services must offer more or better functionality than existing alternatives, and they must have simple and natural interfaces. Especially the latter requirement is difficult to fulfil with the interaction capabilities of the small lightweight mobile terminals. The usability problems of small terminals might be solved by means of multimodal interfaces that combine speech, text and pen at the input side, and text, graphics, and speech at the output. However, the combination of multiple input and output modes in a single session appears to pose completely new technological and human factors problems of its own. Therefore, the Research departments of three Telecom Operators (Telenor, Portugal Telecom, and France Télécom) collaborate with two academic institutes (University of Nijmegen and the Max Planck Institute of Psycholinguistics) in a two year EURESCOM project, called MUST - *MUltimodal, multilingual information Services for small mobile Terminals* - that has two main aims:

(1) To obtain knowledge about the issues involved in the implementation of a simultaneous coordinated multimodal application for a small terminal. Simultaneous coordinated multimodal interaction is the term used by W3C[2] for the most advanced form of multimodal interaction, where all available input devices are active simultaneously, and their actions are interpreted in context.

(2) To obtain information about user behaviour in a purpose built multimodal application that implements simultaneous coordinated interaction.

This paper presents the functionality and implementation experience of the first version of the demonstrator. In addition it presents the results of the first step of the user evaluation. User interface experts of Telenor and Portugal Telecom tested the first version of the demonstrator using the cognitive walkthrough method.

2. THE DEMONSTRATOR

Investigating user behaviour and preferences in multimodal interaction requires a combination of theoretical, engineering and behavioural approaches. Services that involve navigation and selection on the basis of a

[2] http://www.w3.org

map have proven to be good candidates for user testing of multiple input and output modes. In e.g. (Oviatt et al. (1997)), multimodal interaction was observed most frequently during spatial location commands. An electronic tourist guide is a commercially interesting example of civil map-based applications. Thus, we decided to use such a service as the platform for the implementation and user testing in MUST (Boves & den Os (2002), EURESCOM (2002)).

2.1 The functionality of the demonstrator

The MUST tourist guide for Paris combines speech and pen at the input side, and text, graphics, and speech at the output side. The service is the equivalent of a printed tourist organised around detailed maps of small sections that function as a navigation and orientation aid. Compared to a printed guide, leafing through the electronic guide should be easier and more rewarding, since the user can determine what information is shown on the screen of an online version. Moreover, up-to-date dynamic information can be provided.

The tourist guide is organized in the form of small sections of the town around "Points of Interests"(POI's), such as the Eiffel tower, the Museum of the Louvre, etc. These POI's are the major entry points for navigation. When the user selects one of the POI's, a detailed map of the surroundings of that object is displayed on the screen (cf. Fig. 1).

Map sections may contain additional objects that might be of interest to the visitor. By pointing at these objects on the screen they are made the topic of the conversation, allowing the user to ask questions about these objects, for example "What is this building?" or "What are the opening hours?". The user can also ask general questions about the section of the city, such as "What restaurants are there in this neighbourhood?", The information returned by the system is rendered as text, graphics (maps, and pictures of hotels and restaurants), and text-to-speech synthesis.

Users are allowed to ask questions about POI's for which the answer is not in the database of the service (e.g., 'Who is the architect of this building?'). Answers to these questions are passed to a multilingual Question/Answering (Q/A) system (developed by France Télécom) that tries to find the answers on the Internet.

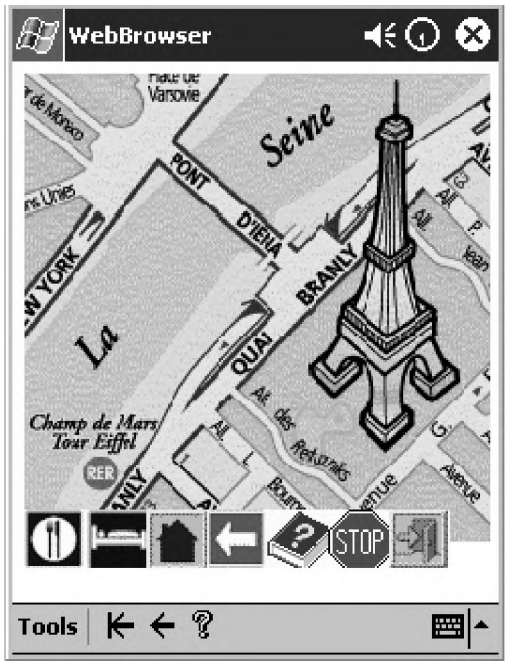

Figure 1. Graphical User Interface of the MUST guide to Paris.

2.2 The architecture of the demonstrator

MUST set out to investigate implementation issues related to coordinated simultaneous multimodal input, i.e., *all* parallel inputs must be interpreted in combination. This is implemented as late fusion of the information from all channels. The overall architecture of the MUST demonstrator is shown in Figure 2.

2.2.1 The application server

The application server comprises six main autonomous modules that communicate with each other via a Hub.

The Hub has been built on the GALAXY Communicator Software, a public domain reference version of DARPA Communicator maintained by

MITRE[3]. GALAXY ties individual components (e.g., ASR, TTS, Dialogue & Context Manager, etc.) together by providing extensive facilities for passing messages between the components. In the MUST demonstrator the Hub is script based.

Messages that are passed between Galaxy modules are based on the key-value pair (attribute-value pair or name plus a value) format. This message format was found to be sufficient for dealing with relatively simple operations like connection set-up, synchronization, and disconnection etc. However, some operations, such as database lookup and GUI display requests, involve more complex data structures. This necessitated an extension to the message format, which was provided by defining an XML based mark-up language MxML - MUST extensible Mark-up Language. The complex data structures are represented by MxML strings and embedded in the basic key-value pair Galaxy messages. In this way we can combine the message passing mechanism provided by Galaxy with the flexibility and power of XML.

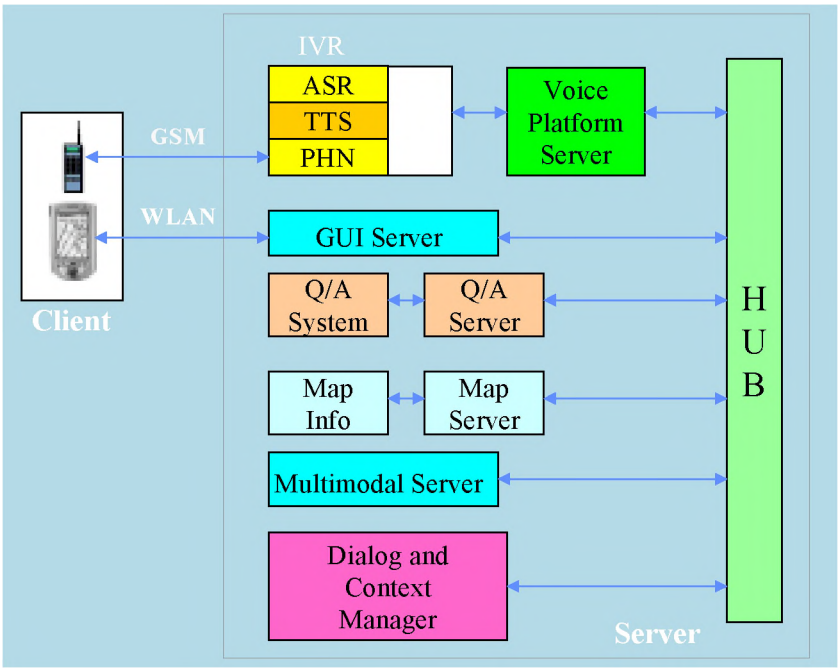

Figure 2. Overall architecture of the MUST tourist guide to Paris. Acronyms are explained in the text.

This modular architecture offers a high degree of flexibility. For example, we have used two different voice servers. This could be done with

[3] http://fofoca.mitre.org

minimal effort. The modular architecture also supports multilinguality, by allowing to separate language dependent and language independent parts of the individual modules. In this way we could quickly adapt the system to a different language, by plugging-in the language dependent components. The server modules are written in Java or in C++.

2.2.1.1 Multimodal server

This server is responsible for multimodal integration. The temporal relationship between speech and graphical input channels is handled by considering all input information received within a pre-defined time window. This information is packed in a single message and passed on to the dialog manager as a first step in the late fusion process. At this stage the message may contain contradictory elements and the interpretation of the combined contents is left to the dialog manager. The duration of the time window is a variable parameter that can be adjusted according to the dialog state.

The multimodal server also performs fission. The message from the dialog manager is broken down into two messages. One of the messages contains only speech and is sent to the voice server; the other one, containing only graphics, is forwarded to the GUI server.

2.2.1.2 Voice Server

Two different versions of the voice server were developed:
The first is based on the InoVox IVR platform from Portugal Telecom Inovação. Philips SpeechPearl2000 is used for automatic speech recognition and the L&H TTS engine for speech synthesis. These components support both English and Portuguese. In addition, the system integrates the France Télécom TTS engine to support French and English.

The other version of the voice server is based on the TABULIB telephony platform of Telenor R&D. This voice server also uses Philips SpeechPearl2000. The voice server contains a generic interface to Microsoft SAPI 4.0, and any TTS engine that supports this standard can be used. In the MUST demonstrator we used the Microsoft TTS engine for English and Telenor's own engine Talsmann® for Norwegian.

An important feature of the messages exchanged by the voice server is that they are asynchronous. Thus, the module that has sent a message to the voice server does not wait for an answer or an acknowledgement, but it proceeds with its next operation. A potential drawback of asynchronous messages is tat it may affect the stability and reliability of a system. However in our case, we found that asynchronous messages did not affect the reliability.

We could have implemented ASR, TTS and the telephony module (PHN in Fig. 2) as separate Galaxy servers. However, lumping them together in a single Galaxy server avoids the need to send large amounts of (speech) data via TCP/IP connections, thereby improving the response time of the system. Moreover, a voice server is a typical component of a conventional (commercial grade) voice only dialog system. Therefore, it is much easier to use this component 'as-is'. To incorporate the existing voice servers in the Galaxy based architecture, we only needed to implement a "wrapper" that sits between the Hub and the existing servers. This wrapper is responsible for processing the Galaxy messages and invoking the appropriate voice operation (TTS or ASR).

2.2.1.3 Question Answering server

When during the interaction process with the MUST tourist service the user issues a spoken request to the service the module that handles speech recognises what has been said and sends a message with the corresponding semantic representation, through Multimodal Server, to the Dialogue Manager. The message is then parsed and interpreted by the dialogue manager that checks whether the requested information belongs to the service domain. If the information cannot be found in the application database, the dialogue manager redirects the request to Question Answering (QA) server and notifies the user that information was not available on service's domain, but that it will try to find it nevertheless. The dialogue manager will not be stuck until an answer is received from the QA host. The user can proceed interacting with the service and he/she will be notified by the Dialogue Manager when the response to the out-of-domain question arrives.

The QA system searches for the answer in the Internet. It is obviously inappropriate to try and render complete documents on the iPAQ screen, and leave it to the user to detect the answer to the question. Therefore, the QA system analyses the documents that it retrieves in detail, to extract a number of answers, each of which is assigned a score for the probability that it is correct. The answers are such that they can be formulated in a short phrase or sentence. If the QA system is not able to find an answer, it will respond with the message that it failed to find the requested information.

The QA server is physically located in one single site at the premises of France Télécom R&D, due to its complexity. However, the functionality of the QA system can be accessed through the Web, since it is implemented as a Web Service. The Web Service approach implies the use of SOAP formatted messages over HTTP for the communication between the server and the applications that access its functionality through the Internet. The implementation of this communication mechanism directly in the Dialogue Manager would result in additional complexity to the module without any advantage

in terms of service performance. So it has been decided to create an independent module, named QA Proxy Server, to provide and handle the communication mechanism between the MUST informative service and the remote QA host. It receives a message from the dialogue manager with the question issued by the user, formats the request in SOAP XML encoding and sends it to the server using the HTTP protocol. The QA server runs a listener that accepts the incoming SOAP calls, reads the information from the XML SOAP packets, and maps them to its own processing logic. The proxy QA server parses the response packet in SOAP XML encoding and extracts the answer according to its own internal logic, which is the answer with the highest score. Then the proxy constructs a message with the answer and sends it back to the Dialogue Manager.

.

2.2.1.4 Dialog Manager Server

The Dialog & Context Manager module is written in Java. It consists of four main components, implemented as classes, viz. (1) Context Manager, (2) User model, (3) System response generator, and (4) XML processor.

The *Context Manager* is the heart of the module. It is a finite state machine that contains four main states, START, POI, GOF and FAC.

START: The dialog is yet to start

POI: User has selected a point of interest (POI)

GOF: User has selected a group of facilities (GOF)

FAC: User has selected one particular facility such as a restaurant.

The state machine approach with only a few states was possible because of the hierarchical nature of the application. The application consists of several POIs, each of which in turn consists of GOFs. Finally, each GOF comprises a set of facilities. When the user generates an event, a state transition can occur. A state transition is defined by the tuple (S_t, I_t), where S_t is the current state and I_t is the current user input. Each state transition has a well- defined end state S_{t+1} and an output O_t.

The *User Model* is an array of concepts whose length is set to a predefined value. The concept table is filled using the values output by the speech recogniser and the GUI client that lie within a predefined time window. During the filling operation input ambiguities were solved, in this way comleting the late fusion. Once filled, the concept table defines the current input I_t. If the values in the concept table are $I_t(1)$, $I_t(2)$,.....$I_t(n)$, then the N-tuple $(I_t(1), I_t(2), I_t(n))$ is the current input I_t. The number of different inputs can be prohibitively large, even if the length of the concept table (M) and the number of values a given concept can take (K) are moderate. In our

case we have reduced the number of inputs by employing a many-to-one mapping from the original input space to a new smaller sized input space.

The *System response generator* is responsible for the generating O_t. It is essentially a mapping from space formed by the tuples (S_t, I_t). It looks at the current state S_t and the input I_t, and generates an output O_t that contains both speech and graphic contents. The output can contain pre-stored strings, parameters extracted from the input itself, and data obtained from the back-end map database or the QA system. Speech output is generated by concatenating components appropriately. Graphical output is generated as an XML string.

The *XML processor* performs the XML operations. Since it is difficult to generate complex XML string through concatenations, we maintain a DOM (Document Object Model) tree that always represents the current graphical output. This is generated from the previous DOM through tree operations such as deletions and insertions. The XML processor is based on the open source XALAN[4].

2.3 The client

The client part of the demonstrator consists of two major modules, one for handling the graphics and another for the speech. Graphics is implemented on a Compaq iPAQ running Windows CE, which is connected to the server via a 802.11b WLAN connection. Speech is handled by a mobile phone. The test users in the evaluation will not notice this "two terminal" solution, since the phone is hidden and the interface is transparent. Only the headset (microphone and earphones) with Bluetooth connection will be visible for the user.

2.3.1 GUI Client

The GUI Client transfers the GUI signals (tap and graphical information) back and forth between the client and server. The GUI is based on the Pocket Internet Explorer web browser. The use of ActiveX controls in the web browser gives a powerful interface that supports a variety of GUI components, such as gif image display, hotspots, push buttons, select lists, and text fields. The input to the web browser (from the GUI Server) is an HTML file. The GUI is defined and controlled by the use of Microsoft JScript inside the HTML body. This allows the application server to define the appearance of the GUI, and therefore no software update on the iPAQ is necessary.

[4] http://www.apache.org/xalan

The feedback from the Dialog Manager is an XML body that reflects what to be displayed on the GUI Client. The GUI Server retrieves the content of the XML body, and wraps this into an HTML format to be forwarded to the GUI Client. The HTML file is actually stored on an HTTP (Web) server, and fetched by the GUI Client that is just an advanced web browser. We use XSLT[5] to transform the XML body to the HTML file. Using style sheets the appearance of the GUI display can be easily be altered in services where the GUI format is dependent of the dialog context, or the users profile.

2.4 The user interface

One important feature for the user interface is the "Tap While Talk" functionality. When the pen is used shortly before, during or shortly after speech, these two input actions are integrated into one combined action. For example "Show hotels here" while tapping at Notre Dame. When tapping occurs more than approximately one second before, or after the speech, the actions are considered to be serial and independent.

The overall interaction strategy is user controlled, very much in accordance with what is usual in graphical user interfaces. This implies that the speech recogniser must always be open to capture input. Obviously, this complicates signal processing and speech recognition. However, it is difficult to imagine an alternative for a continuously active ASR without changing the interaction strategy. Users can revert to sequential operation by leaving enough time between speech and pen actions.

The output is mainly presented in the form of text (e.g."the entrance fee amounts 3 euro") and graphics (maps and pictures of hotels and restaurants). The text output appears in a text box at the upper side of the screen.

To help the user keep track of the system status, the system will always respond to an input. In most cases the response is graphical. For example, when a Point Of Interest (POI) has been selected, the system will respond by showing the corresponding map. If the system senses input, but does not know what to do with it (e.g. if audio input was detected, but ASR was not able to recognise the input with sufficiently high confidence), it provides a prompt saying that the system did not understand the utterance.

The graphical part of the user interface consists of two types of maps: an overview map showing all POIs, and more detailed maps with a POI in the centre. The dialogue/interaction management is designed such that the interaction starts without a focus for the dialogue. Thus, the first action that a

[5] http://www.w3.org/TR/xslt

user must take is to select a POI. Selecting an object automatically makes it the focus of the dialogue: all deictic pronouns, requests etc. now refer to the selected object. Selection can be accomplished in two ways: by speaking and by pointing (or by both). Irrespective of the selection mode, the application responds by showing the section map that contains the POI. A selected object is marked by a red frame surrounding it, as a graphical response to the selection action. All additional selectable objects on a map are indicated by green frames. When the user has selected a POI, several groups of facilities (GOF) such as hotels and restaurants can be shown as objects on the maps. This can be accomplished through speech (by asking a question such as 'What hotels are there in this neighbourhood?'), or by tapping on one of the 'facility' buttons that appear at the bottom of the screen, just below each section map. Fig. 1 shows the buttons that were present in the first version of the GUI. Two buttons are related to the functionality of the service (hotels and restaurants), and three buttons are related to navigation: a help button, a home button, and a button that can force the application back to the previous state of the dialogue (a kind of error recovery). 'Help' was context independent in the first version of the demonstrator; the only help that was provided was a short statement saying that speech and pen can be one by one or combined to interact with the application.

Speech input allows to make shortcuts. For example, at the top navigation level (where the overview map with POIs is on the screen) the user can ask questions such as 'What hotels are there near the Notre Dame?'. That request will result in the detailed map of the Notre Dame, with the locations of hotels indicated as selectable objects. However, until one of the hotels is selected, the Notre Dame will be considered as the topic of the dialogue. In this context selection by means of the pen is easiest.

3. THE EXPERT EVALUATION

Usability experts were involved in the first phase of the evaluation of this speech centric multimodal demonstrator. The aim of the expert evaluation was to identify general usability problems with this type of multimodal interfaces and to identify specific usability problems of the MUST tourist guide. The next phase will be a usability evaluation with naïve users.

3.1 Method of expert evaluation

For the expert evaluation we used the Cognitive Walkthrough technique (Lewis & Wharton, 1997). This technique is suitable for evaluating proce-

dural dialogues. Cognitive Walkthrough uses a set of predefined tasks (action sequences) as the starting point of the evaluation. The technique can be used with minimum training by the experts and relies on their previous knowledge of usability requirements. Cognitive Walkthrough is an inexpensive way of identifying obvious problems in the user interface offering more effective naïve user testing.

Seven User Interface experts from Telenor and five from Portugal Telecom participated in the expert evaluation. All test sessions were videotaped, to capture the moment when and the location where the experts tapped on the screen and recordings of what they said.

The expert evaluation procedure consisted of five steps:

(1) Introduction (10 minutes); the MUST project was presented and the aim of the evaluation was explained. It was stressed that the demonstrator had been built to test simultaneous multimodal interaction.
(2) Exploratory phase (10 minutes); the expert explored the prototype and commented on any apparent usability (or other) issues.
(3) Cognitive Walkthrough introduction (15 minutes); the technique was explained, including the questions the expert had to answer after performing each step in an action sequence. These questions are:
    a. Will a naive user try to achieve the right effect?
    b. Will the user notice that the action is available?
    c. Will the user associate the action with the desired effect?
    d. If the action is performed, will the user see that progress has been made towards the goal?
(4) Cognitive Walkthrough evaluation (1 hour); the experts performed the Cognitive Walkthrough technique for three pre-defined action sequences, each consisting of 4 to 6 steps. For each step a goal was defined, e.g. 'check opening hours for the Eiffel Tower', as well as an action, 'say "what are the opening hours?" while tapping the Eiffel Tower'.
(5) Debrief session (25 minutes); the experts discussed their written and other comments with the experimenter.

3.2 Main results of the expert evaluation

Since only twelve experts participated in this evaluation, results should be interpreted very carefully. There were great similarities between the remarks and observations of the Portuguese and Norwegian experts. The most remarkable and clear observations will be discussed here.

During the exploratory phase, most experts started to use the two input modalities one by one, and some of them never tried to use them simultaneously. After a while five of the twelve experts started to use pen and speech simultaneously.

Timing between speech and pointing has been studied in other experiments (Oviatt et al (1997)) and (Gustafson et al (2000)). There are, however, several important differences between the tasks in our experiment and in the studies of Oviatt and Gustafson, so the results cannot be directly compared. In the experiments of Oviatt and Gustafson the interaction style was mainly sequential, and the users tended to use the pen to draw or to point, and to speak one to four seconds after the pen signal. Our experiment focused on simultaneous use of pen and speech, where the users typically tapped at the end or shortly after the utterance. This was especially the case when the utterances ended with deictic expressions like 'here' or 'there'. If no deictic expressions were present, tapping often occurred somewhat earlier. Timing relations between speech and pointing will be investigated in more detail in the user evaluation experiment that is now being designed

The results from the exploratory phase indicate that the Usability Experts in this study, who happened to be frequent PC and PDA users, are accustomed to use a single modality (pen or mouse) to select objects or using menus to narrow down the search space. Even if they are told that it is possible to use speech and pen simultaneously, they will have to go through a learning process to get accustomed to the new simultaneous coordinated multimodal interaction style. But once they have discovered and experienced it, the learning curve appears to be quite steep.

It was not intuitive and obvious that the interface was multimodal, and in particular that the two modalities could be used simultaneously. This indicates that for the naïve user evaluation we should pay much attention to the introduction phase where we explain the service and the interface to the user.

During the Cognitive Walkthrough many usability issues came to light. They can be divided into interaction style issues and issues that are specific for the MUST tourist guide. The MUST guide specific issues related to buttons, feedback, prompts, the way of highlighting selected objects, and the location of the POIs on the screen. From the comments by the experts it was clear that much more attention should be paid to the graphical interface and the design of the buttons. Most of the problems can be solved rather easily. Based on the comments of the experts a second version of the demonstrator will be built that will be used for the user tests.

The main problem that was observed by the experts related to the interaction style, was that almost all experts agreed that without some initial train-

ing and instruction, users would probably not use the simultaneous multimodal interaction style. With the present lack of multimodal applications for the general public. there is a need to introduce the capabilities of simultaneous coordinated interaction explicitly before customers start using the new products. According to the experts a short video or animation would be suitable for this purpose. Once the users are aware of the multimodal capabilities of the system, they should be able to associate the actions and desired effects with a minimal cognitive effort.

3.3 Design of the user evaluation

Based on the comments by the experts a new version of the MUST tourist guide will be implemented that will try to solve all the MUST demonstrator specific usability problems.

For the naïve user evaluation we have decided to focus on the introduction phase, since this is a very crucial phase according to the experts. There will be three introduction versions:

1. Introduction video, explaining the service and interaction style, especially emphasising deictic expressions like 'there', 'this', etc. Deictic expressions are hypothesised to trigger simultaneous multimodal interaction.
2. Introduction video with less emphasis on deictic expressions.
3. Textual introduction on the screen; this short introduction was also present in version 1. The experts indicated that this was not enough to explain the interaction style. This version will be used as a reference.

The user testing will take place at Telenor, Portugal Telecom and France Télécom, and each version will be tested by at least 5 subjects. The user evaluation will take place in September 2002.

4. REFERENCES

Boves, L, Den Os, E. (Eds.) (2002) *Multimodal services – a MUST for UMTS.* http://www.eurescom.de/public/projectresults/P1100-series/P1104-D1.asp

EURESCOM (2002) *Multimodal and Multilingual Services for Small Mobile Terminals.* Heidelberg, EURESCOM Brochure Series.

Gustafson, J. et al. (2000) ADAPT – a multimodal conversational dialogue system in an apartment domain. *International Conference on Spoken Language Processing* (pp II:134-137). Beijing.

Lewis, C. and Wharton, C. (1997) Cognitive Walkthroughs. In Helander, M., Landauer, T.K., and Prabhu, P.V. (Eds.) *Handbook of Human Computer Interaction* (pp. 717-732). Amsterdam: North-Holland Elsevier Science Publishers

Oviatt, S., DeAngelli, A., Kuhn, K. (1997), Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction. *Computer Human Interaction.* Atlanta, Georgia.