Multimodal Dialogue System with Multitouch Input

Vassiliki F. Kouloumenta

Department of Electronic and Computer Engineering

Technical University of Crete

Thesis committee:
Alexandros Potamianos, Supervisor
Vasileios Digalakis
Aikaterini Mania

A thesis submitted in partial fulfillment for the Diploma degree of

Electronic and Computer Engineering

Chania, September 2009

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Σχεδιασμός και Υλοποίηση Πολυτροπικής Διεπαφής, που συνδιάζει είσοδο Φωνής και Πολυαφής

Βασιλική Φ. Κουλουμέντα

Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών

Χανιά, Σεπτέμβριος 2009

This work is dedicated to my parents, that made my choices come true by supporting me every step of the way.

Acknowledgements

This work took almost one and half year to complete, and a lot of things happened in my life during that time, things that made working on this thesis harder than I expected. I would like to take some time here and thank the people that made completion of this work possible despite of all that happened.

I'd like to thank all my colleagues and friends in TU Crete, who made these years in Chania what is possibly the best years in my life so far. Their help during my studies was invaluable to me, and I will always be grateful for it.

Of course it goes without saying that tremendous help was provided by my supervisor professor Potamianos. From the undergraduate courses all the way to the diploma thesis, he was always open to new ideas, providing his knowledge and experience to make what we had in mind possible. Also I would like to thank earnestly my supervisor assistant Perrakakis, for his priceless assistance during this project.

Most importantly, I would like to thank my family, my parents and brother, for their help, support, guidance and love that made my life and my choices clear, easier, and most importantly: possible.

Finally, I would like to thank the person that made these past years truly the best years of my life; my friend Vassilis. Thank you for bearing with me during my good times and my bad times, I don't know if I could have made it without you!

Abstract

In this thesis we propose a multimodal travel reservation dialogue system with touch input. Whereas traditional applications use only one modality for interaction, we suggest a different approach without that convention, and using more than one modalities we try to eliminate the unimodal systems' disadvantages. In our application we try to benefit from the advantages, that multimodality offers. The main idea behind this application is to combine the graphical user interface with speech, in order to achieve high accuracy and performance in speech recognition, and speed in task completion. So users can use either GUI or speech in order to complete a certain task.

The system we propose is based on the client-server architecture, and separates completely the task and the interface. While the main system's functionality is implemented in task manager.

Also in this work we examine how users benefit from multimodal systems according to different criteria, such as efficiency, satisfaction and functionality, in contrast with unimodal ones, by testing various user evaluation scenarios. Our goal in this thesis is to understand by the evaluation process which factors affect the modality selection by users, and which system the users prefer most. Finally it would be interesting if we measure the performance among the different modes.

Abstract (in Greek)

Στην διπλωματική αυτή προτείνουμε μια πολυτροπική εφαρμογή για κρατήσεις πτήσεων, που συνδιάζει είσοδο φωνής και αφής. Μέχρι σήμερα οι παραδοσιακές εφαρμογές χρησιμοποιούν ένα μόνο μέσο για αλληλεπίδραση με το σύστημα, ενώ εμείς προτείνουμε μία διαφορετική προσέγγιση, χωρίς την παραπάνω σύμβαση. Χρησιμοποιώντας πάνω από ένα μέσα για αλληλεπίδραση προσπαθούμε να εξαλείψουμε τα προβλήματα που εισάγουν τα μονοτροπικά συστήματα. Σε αυτή την εφαρμογή προσπαθούμε να ωφεληθούμε από τα πλεονεκτήματα, που προσφέρουν τα πολυτροπικά συστήματα. Η βασική ιδεά πίσω από αυτή την εφαρμογή είναι ο συνδιασμός γραφικής διεπαφής και φωνής, ώστε να επιτύχουμε υψηλή ακρίβεια και απόδοση σχετικά με την αναγνώριση φωνής, και ταχύτητα στον τερματισμό μίας εργασίας. Έτσι λοιπόν ο χρήστης μπορεί να χρησιμοποιήσει είτε την γραφική διεπαφή, είτε την φωνή ώστε να ολοκληρώσει μία εργασία.

Το σύστημα που προτείνουμε βασίζεται στην αρχιτεκτονική του client-server και διαχωρίζει πλήρως το κομμάτι της εργασίας από την διεπαφή. Το μεγαλύτερο κομμάτι της λειτουργικότητας το αναλαμβάνει ο διαχειριστής εργασίας.

Επίσης σε αυτή την εργασία μέσω πειραματικής διαδικασίας εξετάζουμε, πώς οι χρήστες μπορούν να ωφεληθούν από τα πολυτροπικά συστήματα, σχετικά με διάφορα κριτήρια, όπως αποδοτικότητα, ικανοποίηση και λειτουργικότητα, σε αντίθεση με τα μονοτροπικά συστήματα, εξετάζοντας ποικίλα σενάρια. Στόχος μας είναι να μελετήσουμε, μέσα από την διαδικασία της αξιολόγησης ποιοί παράγωντες επηρεάζουν την επιλογή του κάθε χρήστη και ποιό σύστημα οι χρήστες φαίνεται να προτιμούν. Τέλος είναι σημαντικό να μετρήσουμε την απόδοση μεταξύ διαφορετικών συστημάτων.

Contents

List of Figures

Chapter 1

Introduction

Human communication is easy and efficient due to the fact that, humans when communicate use several modalities, such as speech, gestures, gaze etc. Humans, in their everyday face - to - face communication, use several means in order to express emotions, opinion etc. For example a face expression, a gaze, or a movement may change the whole meaning of a sentence. That's why human communication is so alive and direct.

Also, disabled people consist a significant part of our society, that unfortunately can not have access in many services. For example people with acoustic and speech disabilities, confront great difficulty to use speech recognition systems. On the other hand people suffering of kinetic disabilities, find it difficult to manipulate graphical interfaces. As a fact, the existence of more than one modalities could facilitate disabled people whenever they want to interact with a system. As an example deaf and dump people could use touch rather than speech, while handicapped people could use speech rather than touch.

Furthermore, in many cases people are too busy with a job and they can not use either their hands or their eyes in order to interact with a graphical interface, for example in driving hands and eyes are engaged, but user can speak their commands. In such cases, the existence of more than one modalities could facilitate users, and especially in driving it could also avert accidents, that are caused because driver was busy with something else than driving.

As a fact, the humans' ability to communicate under all circumstances and the existence of people with disabilities, in combination with the emergence of powerful mobile devices, such as iPhone, trigger research interest about new, more efficient ways of interaction, that involve combination of more than one modalities. Such examples of modalities include speech, pen, gestures, head and body motions.

1.1 Related Work

Until nowadays, previous work in reference with multimodal interfaces, was centered on desktop and PDA systems. Examples of such efforts include the QuickSet multimodal system, which has been developed in conjunction with several map based applications [4, 5]. QuickSet enables user to create and position entities on a map by using speech, pointing gestures with a pen and / or direct manipulation. Another multimodal, map based system is the MATCH system, which provide a mobile multimodal speech / pen interface to restaurant and subway information for the New York City [6]. Those two systems referred previously, are based on the "Put - that - there" prototype, introduced by Bolt [7]. More specifically, in Bolt's system, user can create and position simple entities onto a large screen, placed in front of him. The user can use speech and pointing gestures, in order to create and move objects on the screen. For example, user can point to a spot on the screen and say "create a blue rectangle there", and a blue rectangle will appear at the specific spot. Last but not least is the "Multimodal Flight reservation" application, implemented for both desktop and PDA devices [2, 3, 8]

1.2 Our Application

This project introduces a multimodal travel reservation dialogue system, implemented for the iPhone. The main idea is that the user can interact with the system by using either speech or gesture input, but also a combination of those two modalities. The selection of those two modalities is based on the perception that voice user interfaces (VUI) are complementary to those of graphic user interfaces (GUI) and vice-versa [3, 5]. Survey indicates that, GUI interfaces have low error rates and offer easy error correction, while VUI interfaces can be faster for list selection, when the list contains more than fifteen items. As a result the combination of those two modalities can offer great advantages in multimodal systems' functionality, as user has the opportunity to choose the most appropriate input modality for each context.

For the purposes of this thesis, two unimodal and three multimodal modes were implemented. The unimodal modes that were created are: 1) the "GUI only", where user has to use only the graphical interface, in order to complete a certain task and 2) the "Open Mike - Speech Input", which provides speech as the default input modality, while the output blends speech and GUI.

As far as the multimodal modes are concerned, first is "Click to Talk", in which GUI is the default input modality, while for speech input user has to click onto the speech button, that lays below GUI. Second comes the "Open Mike", which provides speech as the default input modality, while for GUI interaction users have to press onto GUI. And last but not least is "Modality Selection", which is a mix of the previous two modalities ("Open Mike" and "Click to Talk"). The system proposes to users at each turn, to use a certain modality, according to the attribute size. For example if the list attribute is large, the system proposes to the user to use the "Open Mike", otherwise it proposes the "Click to Talk". Of course, the user can use whichever modality they consider to be the best at each turn or even they can use a fusion of the two modalities, in order to complete a certain task.

attribute name	attribute size
hotelname	250
city	135
airline	93
date	22
car type	15
car rental	10
time	9

Table 1.1: Attribute Size

By the evaluation of the three systems ("GUI only", "Open Mike - Speech Input" and "Modality Selection") we can extract really important conclusions about the factors, that affect the modality selection by the user and also about the efficiency among the different systems.

This application was built in collaboration with Manolis Perakakis. I undertook to design and implement the graphical user interface, as well the evaluation process, while M.Perakakis, implemented the voice user interface and a part of the backend.

1.3 Thesis Outline

The remainder of this thesis is structured as follows. After this introductory section, section 2 presents the unimodal and multimodal systems' fundamentals and the benefits the latter ones have against the former ones. Section 3 discusses about the iPhone travel reservation application and the system design. Section 4 describes the evaluation methodology and presents the evaluation results. Finally we conclude the thesis with section 5.

Chapter 2

Unimodal and Multimodal systems

2.1 Introduction

As mobile devices are used for different tasks by different users, there is need to provide efficiency in interaction, so each user can interact by using whichever mode or combination of modes they want. By this way communication between the user and the device is more natural and efficient, and also the interface itself is more easy to be learned and used.

Until nowadays most of the application systems provided one single modality for manipulation, but the fact that in face to face human communication more than one modalities take part, gave to scientists in computer science the signal for more research in the direction of creating new systems, that are going to provide a set of modalities. Consequently users can combine different modalities in order to achieve a more effective communication with a device. By effectiveness in general we mean, ease in use and learn of the whole system, but also velocity in termination of a specific task. Modality, in general is the way of interaction with a system.

Another social sector that multimodality is going to accommodate is that of handicapped people. Conventional means that are provided by most applications, incommode interaction for disabled people. For example blind people can not interact with graphical interfaces but they can easily communicate with a voice recognition system. Or deaf and dump people could use touch rather than speech. So it is easy to understand that the existence of more than one modalities makes technology more accessible to these people.

In order to understand better the advantages of multimodal systems against unimodal ones, a reference to both systems' fundamentals is needed.

2.2 Interaction in Unimodal Systems

As the word unimodal indicates, unimodal systems provide a single modality for interaction. Consequently the user is obligated to use the one and only modality, provided by the application, in order to complete a specific task. That has its negatives, as some means of interaction in some cases are too slow, display poor accuracy or high error rates, making the interaction a hard task. For example graphical user interfaces are time - consuming when users have to choose from a large option list, and speech recognition systems display poor accuracy and high error rates when users are in a noisy environment.

Also the existence of one modality has impact on disable people, as it makes the access in some services impossible. As it was mentioned before people with kinetic disabilities find the interaction with graphical user interfaces a hard task, while people with phonetic disabilities they can not interact with speech recognition systems.

Furthermore, unimodality makes interaction difficult in cases where people are engaged with an other task, and can not manipulate an application. A palpable example of such a case is driving, where hands and eyes are engaged and can not interact with graphical user interfaces.

By experience we enumerate currently available modalities provided for interaction. The most popular modalities, that are provided by computer interfaces for manipulation, are mouse and keyboard. Also pointing and gestures are widespread in touch sensitive devices, such as touch - pads, touch - screens etc. Moreover speech becomes very popular as it approaches the human communication, making the interaction between user and device more natural. Other less popular modalities are lip movements, that assist speech recognition both in human communication (for example deaf and dumb people can understand what others say by “reading” their lips) and in human - computer interaction by increasing speech recognition accuracy in noisy environments. Also eye gaze can be used as an alternative pointing device in case of physically handicapped people who can not use their hands or in cases that hands are engaged with another task. To specify in our application, two modalities are used: Graphical User Interface and speech.

2.2.1 Gestures

The first one is Graphical User Interface (GUI) modality, that provides pointing gestures as a mean of interaction. In general, people employ gestures in their everyday communication with other people, something that makes the human communication easy, natural and effective. For example we can indicate agreement or disagreement by a head movement, or we can point at objects by using our forefinger. As far as gestures (that are provided for interface manipulation) are concerned, we can classify them in three general categories [9]. First category includes pointing gestures, that refer to the usage of a pointing device on a touch sensitive surface. Pointing is typically sampled by using either pointing devices such as pen, or pointing devices emulated a touch sensitive display such as a finger. It would be interesting if we persist for a while to pen (or stylus) devices.

As pen is useful for mobile devices such as wireless tablet PCs, PDAs or GPS receivers, is an essential instrument used in interaction with touch sensitive surfaces. In general, pen refers to usage of any product allowing for mobile communication. A device such as pen, generally is used as pointing device to press upon a graphics tablet or a touchscreen. For example user can point at a specific object by employing a single tap or a double tap or even by making shapes on the touch sensitive surface.

Second category includes 2D gestures, that refer to movements on a flat surface. 2D gestures can by further classified in two subcategories. First one uses a pointing device, for example a pen or a finger as a mean of interaction. Given one or more strokes or a sequence of movements user can complete a certain task. For example touch sensitive mobile phones provide the users with the opportunity to use pen and hand writing, in order to compose a message.

Second one, namely multitouch, denotes a set of interaction techniques, which allow users to control graphical applications with several fingers. Multitouch consists of a touch sensitive display, such as a touch screen or a touch pad, as well as a software that recognizes multiple simultaneous touch points. This effect can be achieved by several means, including heat, finger pressure, high capture rate cameras, optic capture, microphones, shadow capture etc. For example in iPhone users can zoom in and zoom out photos by applying a diagonal movement on the photo by using two fingers. Multitouch technology is not something new, as it dates back to 1982, when the first multitouch display was developed in University of Toronto by Nimish Mehta. After that, research in multitouch displays continued and more multitouch products have been developed since then by several companies such as Bell, Microsoft, Apple and Fingerworks.

Last category involves 3D gestures, which refer to movements of fingers, hand or head in three dimensional space. 3D gestures are part of human communication. For example a facial expression can indicate happiness, sadness, abomination etc. Or we can use thumbs up to indicate agreement or thumbs down to indicate disagreement. Consequently it is very seductive to apply 3D gestures as a way of interaction with computational systems, as they approach human communication. Also 3D gestures can enhance speech recognition accuracy in noisy environments, where speech recognizers have poor performance. Another reason that 3D gestures are practical, is that deaf and dumb people use sign language as a mean of communication, so 3D gestures can replace in some occasions speech or graphical interfaces modalities.

Now that we have categorized gestures in three general categories it is important to refer in gesture recognition techniques [9]. In general, gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. As we mentioned before, gestures can originate from bodily motion or state but commonly originate from our face or hands. As a result gesture recognition can be considered as a way for devices, from computers to mobile phones, to begin understand body language, by building a bridge between humans and machines. So it is easy to understand why gesture recognition constitutes a challenge for researchers. Especially, while pointing gestures do not require recognition beyond identifying which displayed object user wants to refer to, recognizing 2D or 3D gestures is a typical pattern recognition problem.

As far as 2D gestures are concerned, there are three main approaches to gesture recognition. First come the hand coded recognizers. Hand coded gesture recognizers make the resulting system difficult to create, maintain and modify, so they rarely are used. Second approach involves template based gesture recognizers, which compare the input pattern with prototypical templates and choose the best matched template. Each gesture is characterized as a class of shapes and represented as one or more templates. Particularly, input is compared to each template by first transforming the gesture to match the templates as closely as possible, and then computing the difference using the mean square error measure. After transformation, the template that indicates the lowest difference below a certain threshold is considered to be the best match. Input label can be labeled as unknown if all differences scores exceed the threshold. Last but not least approach is the feature based approach, where features can be extracted from the stream of input coordinates. After feature extraction, a pattern classification algorithm is applied to assign the gesture to one of predetermined gesture categories. We have to mention that applying smoothing and filtering during feature extraction improves recognition accuracy.

As far as 3D gestures are concerned, there are two main approaches to recognize body motion. First one, captures gesture movements by using dedicated input devices, such as sensing gloves or position trackers, and afterwards applies pattern classification techniques. For example raw data from a sensing glove runs through two layers of abstraction before it is passed on to a gesture parser that integrates gesture information. The second approach uses computer vision algorithms, observing the user with one or more cameras and applying computer vision algorithms to segment and classify the image data. The first task is to locate the active user who is performing gestures. Secondary, user's arms and hands have to be located. And finally, the hand gestures have to be classified. The recognition in the latter approach is less robust rather than in the former one, but the latter gives points to former one as it does not need any intrusive devices.

2.2.2 Speech

Speech is the vocalization form of human communication. It is based upon the syntactic combination of lexicals and names, that are drawn from very large vocabularies. Each spoken word is created out of the phonetic combination of a limited set of vowel and consonant speech sound units. But since speech is a part of human voice (all that sounds made by vocal cords for talking, singing, screaming, laughing etc), it also displays pitch and tone. Meaning that when we talk our voice fluctuate, in order to show emotions and attitude.

The second modality that is provided is the Speech modality. Users can interact with the application by using spoken words and utterances. As a result, communication between the user and the application becomes more natural, as it approaches the face to face human communication. Also people with disabilities are another part of the population that benefit from using speech as a mean of interaction. It is especially useful for people, who have difficulties with or are unable to use their hands, from repetitive stress injuries to involved disabilities that require an alternative input for support with accessing an application system.

Spoken dialogue systems consist of several components, each with a certain functionality [1]:

2.2.2.1 Speech Recognition

By speech recognition, in computer science and telecommunications, we generally mean the conversion of an input speech utterance, consisting of a sequence of acoustic - phonetic parameters, into a string of words.

Different speaking styles are known to have a large impact on the accuracy of an automatic speech recognizer (ASR). In fact ASRs convert spoken words to machine readable input, for example to key presses, using the binary code for a string of character codes. Common speech recognition systems can be classified as discrete speech recognition systems where the speaker has to separate each word by a pause, something that makes it easier for the system to recognize the enunciated word. Also we can classify speech recognition systems as continuous speech recognition systems, where speech is more difficult to recognize, as the beginning of a word is not easily distinguished from the end of the previous word. Another problem frequently encountered in automatic speech recognition, is background noise. Clicks of tongue, pauses or other grunts, that accompany speech make its recognition more difficult. Consequently, ASRs work well in lab conditions but their performance drops dramatically in everyday environments, such as offices and public places. In such environments ASRs have to recognize these background signals and displace them. Also low quality microphones or poor recording environments can dilute the audio signal, reducing, by this way, speech recognition performance. But speech recognition does not focus only to what a user said but also to how they said it, meaning that people when talk their voice fluctuate in order to show their attitude. As a result, voice quality and intonation are important features of speech, as they indicate the voice tone and the emotions [9].

2.2.2.2 Natural Language Understanding

As natural language we define each language that is used for communication, either is spoken or written, and it is distinguished from constructed languages, such as computer programming language.

Natural language understanding is the analysis of a string of words with the aim of producing a meaning representation for the recognized utterance that can be used by the dialogue manager. This function involves syntactic analysis, in order to determine the structure of the recognized string (for example how the words group together), and semantic analysis, in order to determine the meaning of that string. In general the Natural Language Understanding is determined on the one hand by the nature of the input signal as received from the speech recognizer, and on the other hand by the input type required by the dialogue manager. The understanding of natural languages reveals much about not only how language works (in terms of syntax, semantics, phonetics, phonology etc) but also about how human mind and human brain process language [1].

2.2.2.3 Dialogue Manager

The dialogue manager involves the control between the system and the user, including the coordination of the other components. In general the main function of a dialogue manager is to control the dialogue flow. This involves a number of tasks. First, determining whether sufficient information has been elicited by the user, in order to enable communication with the application. This function requires dealing with information that the system recognizes as false or incomplete, and using confirmation strategies to verify that the input recognized by the system was indeed what was intended by the user. Secondly, communication with the application must be achieved and finally the processed information must presented to the user [1].

2.2.2.4 Communication with external System

Generally dialogue systems require some form of communication with an outside source such as a database, expert system, or other computer or device application, in order to retrieve the information requested during the course of the dialogue [1].

2.2.2.5 Response generation

Assuming that the requested information has been retrieved from the external source, the response generation component now has to construct the message that is to be sent to the speech output component to be spoken to the user [1].

2.2.2.6 Speech output

Speech output involves the translation of the message constructed by the response generation component into spoken form. In the simplest cases prerecorded canned speech or Text - to - Speech may be used [1].

Referring to system's output, an important technology concerning speech output, is text - to - speech (TTS). A TTS system converts normal language text into speech. In the simplest cases prerecorded canned speech may be used. This method works well when the messages to be output are constant, but synthetic speech is required when the text is variable and unpredictable, when large amounts of information have to be processed and selections spoken out, and when consistency of voice is required. In such cases TTS synthesis is needed.

TTS synthesis can be seen as two a two stage process involving text analysis and speech generation. The text analysis has to do with the analysis of the input text, that results in linguistic representation, that can be used by the speech generation stage to produce synthetic speech from the linguistic representation. By linguistic representation we mean, the representation of an utterance, that uses symbols to represent linguistic information about the utterance, such as information about phonetics, phonology, morphology, syntax or semantics. The second stage, which is often referred to as phoneme - to - speech conversion, involves the generation of a prosodic description, followed by speech generation that produces the final speech waveform. The analysis stage of text - to - speech synthesis comprises four tasks. First is text segmentation and normalization, meaning that the text has to be segmented into units such as paragraphs and sentences and normalized into a form that can be spoken (for example interpretation of dates, times etc). The second task, namely morphological analysis, is used to deal with the problem of storing pronunciations of large numbers of words that they vary morphologically one another. Also tagging is required to determine the parts of speech and to permit a limited synthetic analysis. And last but not least, modeling continuous speech effects is concerned with achieving natural sounding speech when the words are spoken in a continuous sequence [1].

The way those components communicate in a dialogue system, is shown in Figure 2.1.

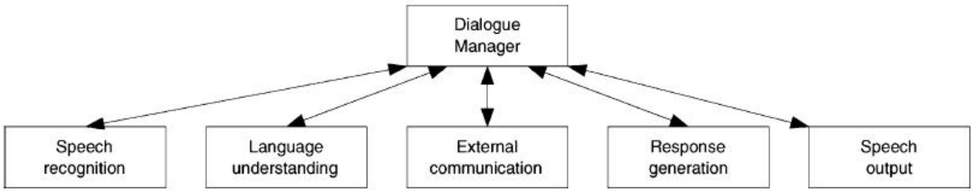


Figure 2.1: An architecture for spoken dialogue systems [1]

2.3 Interaction in Multimodal Systems

On the other hand, in multimodal systems two or more modalities are combined in order a more efficient interaction to be achieved. So the user is free to select from the provided modalities whichever considered to be the best, in order to complete a certain task.

During multimodal communication, we speak, employ face expressions, gesture and move in a powerful flow of communication. This gave the signal for importing people's natural behavior as the center of human-machine interaction. As a result multimodal interfaces are developed in an effort to employ our highly skilled and coordinated communicative behavior in order to control system interfaces. Voice and body movements once captured by sensors such as cameras and microphones, become the ultimate multimodal input modalities [4]. But why multimodal interfaces display such a great interest?

The most important reason for developing multimodal interfaces is their potential to offer accessibility of computing to different and nonspecialist users. Since there can be large differences in people's abilities and preferences in usage of different modes in communication, multimodal interfaces increase the accessibility for users of different ages, skill levels, native languages, cognitive styles, even sensory and kinetic deterioration. This results to the fact that multimodal interfaces offer users the potential to select how they are going to interact with the application. As an example blind people or people with kinetic problems may prefer speech as input modality, while deaf and dumb people or people with strong accent may prefer pen as a mean of manipulation. Moreover multimodal interfaces allow the alternance between input modalities, meaning that probable damage to any individual modality during large periods of use can be prevented [4].

Multimodal interfaces that provide modalities like speech and pen facilitate the systems adaptability. Any individual modality may be ideal for some tasks and environment conditions, but inappropriate in others. For example speech works well in laboratory conditions and in cases that we have to select among many items, but presents low performance in public places or in cases that we have to select among less than fifteen items. As a result multimodal interfaces permit users to switch between modalities as needed during the continuously changing conditions of mobile use.

A second reason for developing multimodal interfaces is that they improve the performance stability and robustness of recognition based systems [4]. By experience multimodal systems offer flexible interfaces, in which people can use input modalities in a more efficient way, so that errors can be avoided. In order to eliminate errors in a multimodal system, the input modalities have to provide complementary functionality, meaning that users can complete a certain task by using whichever mode or combination of modes is considered to be the best. Thanks to this attribute, multimodal systems can also prevent ambiguity between the two input signals. In order to explain that, we are going to use an example presented in Oviatt - Cohen paper [4]. If user says "ditches" but speech recognizer confirms "ditch" as its best guess, then parallel recognition of several graphic marks in pen input can correct accuracy and stability in performance.

2.3.1 Multimodal Systems' Architecture

After that short report to multimodal systems' fundamentals, is time to center on the way multimodal systems work. A lot of research last years has been done about multimodal architectures and many proposals have been done relatively to that issue. In this thesis we are going to center on the most popular multimodal architectural approaches, with emphasis to client - server architecture as this is the one that our system is based on.

Common characteristic of all multimodal architectures is the effort to reduce the uncertainty that recognition based technologies import to the system and to increase robustness in order to support human communication patterns and performance. One general approach is to built a system with at least two sources of input signals that can be fused. Multimodal systems that provide input modalities with similar time scales, such as speech and lip movements, are structured so that are based on machine learning techniques. Such architectures, namely feature level architectures, use multiple hidden Markov models (HMMs) or neural network techniques [4]. Another architectural approach involves semantic level fusion, meaning that performs the combination of multimodal input at the meaning level. This architectural approach considered to be appropriate for systems that provide as input modalities speech and gestures. The most important attribute in such a multimodal architecture is that input signals is not essential to occur simultaneously, so they can be recognized independently [4]. Systems that apply semantic fusion are based on Bolt's "Put that there" prototype [7]. An example of such an architecture constitutes the QuickSet multimodal architecture [4, 5]. A more comprehensive report to the QuickSet system is going to take place in Section 3.

2.3.1.1 Client - Server Architecture

Client - server architectural approach seems to be the most widespread in multimodal systems, as it provides distribution in computational work - load when multiple modalities are processed. By this way maintenance of a system, that consists of more than one recognizers, becomes an easy task. Client - server computing is a distributed application architecture that partitions tasks or work - loads between server and client. Server is a high performance host which shares its resources with clients. On the contrary client does not share any of its resources, but requests a service function by the server. Client therefore initiate communication sessions with the server which "listen to" incoming requests. Client - server architecture offers great advantages to multimodal systems, as it provides ease of maintenance (for example it is possible to replace, repair or upgrade a server while its clients remain both unaware and unaffected by that change). Also all data are stored on the server, something that generally have far greater security controls than most clients. Servers can better control access and resources, in order to guarantee that only those clients with the appropriate permissions may access and change data. Since data storage is centralized, updates are far easier, than in other architectures. In general client - server technology is designed to ensure security, friendliness of user interface and ease of use.

Our system architecture is identical to the one described in [2]. Based on the client - server architecture described in previous paragraph, our system clearly separates the task and the interface (Figure 2.2). The main system's functionality is implemented in task manager. The main tasks the task manager performs are the information (information goal and related attribute - value pairs) retrieval from the user, the creation of database queries using the information supplied by the user and finally the presentation of query results in order the user to navigate through the query results. Also task manager performs some secondary tasks. For instance, task manager allows the user to update system's beliefs via correction and clarification actions. Also task manager undertake to resolve possible ambiguities in input values, inform user about updates and updates dynamically the interface after possible modifications. On the other hand, interface constitutes the graphical environment, that offer the user the opportunity to manipulate objects by using actions corresponding to the physical world, such as gestures and speech.

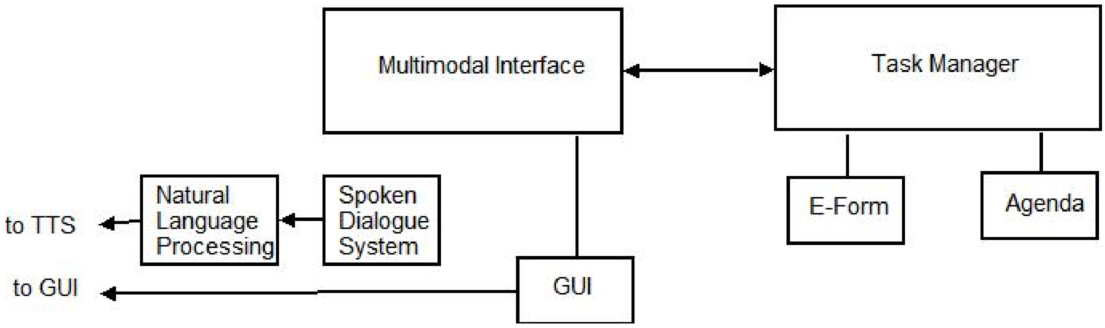


Figure 2.2: Architecture of the multimodal dialogue system [2]

To specify in our application's multimodality, three different multimodal modes, based on the client - server architecture, have been implemented, combining two modalities (speech and gestures). The first is "Click - to - Talk", in which GUI is the default input modality, and for speech user has to tap onto speech button. The second is "Open - Mike - Speech Input", where speech is the default input modality, while the output is multimodal (blends audio and visual (GUI) output). And last but not least is the "Modality Selection" mode, which is a fusion of the previous two modalities, meaning that each user has the opportunity to combine the two providing modalities, in the way they consider to be the best, in order to end a specific task.

2.4 Benefits of Multimodal Systems against Unimodal ones

Now that we have described unimodal and multimodal interaction, let us explain why multimodal excel in virtues against unimodal one [9].

Multimodal interfaces can benefit from modality synergy on both the input and output sides of the system. Using different modalities to provide complementary information can facilitate interaction. For example deictic references to graphic objects are easier to express by pointing gestures rather than speech. Also pointing gestures, as a modality offer low error rates and are easy to correct. On the other hand commands are easier to speak than to choose from a large menu. Survey indicates that speech is faster for list selection, when the list contains more than fifteen items.

Also multimodal interfaces provide freedom of choice. Users may differ in their modality preferences and therefore it is important to be able to choose among different modalities. As we mentioned before, different users may have different needs. For example disabled people (handicapped or blind people) can not use the conventional input devices, such as pen, mouse or keyboard.

At last, offering multiple ways of interaction with a system is more natural for users, as it is closer to human - human communication. When humans communicate, they interpret a mix of audio and visual signals (speech, facial expressions, gestures, gaze etc). That characteristic makes human-human communication efficient. And that characteristic gave the signal for further research in multimodal technology field.

Chapter 3

System Design

3.1 Introduction

Now that we understood why multimodal systems give points to unimodal ones, it is high time to introduce our application.

Mobile interfaces need to allow the user and the system to adapt their choice of communication modalities to user preferences, task and physical and social environments. In this chapter we describe a multimodal travel reservation (flight, hotel and car reservation) application, implemented for the iPhone device. Our application provides a multimodal pen / speech interface, so that user can use pointing gestures onto iPhone's touch screen and / or speech in order to complete a travel reservation task.

3.1.1 Related Work

Before we continue with our application's functionality specification, it is important to make a brief reference to previous work in multimodal interfaces.

Previous work in multimodal interfaces was centered on computer and PDA devices. Multimodal "Flight Reservasion"" application, implemented for both desktop and PDA devices is an example of such an efford [2, 3, 8]

QuickSet [4, 5] is another example of such an effort. QuickSet is an agent - based multimodal system that runs on personal computers ranging from hand held to wall sized ones. The basic system supports map based applications. User interacts with the application by creating and positioning entities on a map with speech, pen - based gestures and / or direct manipulation. To interact with the QuickSet, the user touches the screen to engage the microphone while speaking and drawing.

Another example of a multimodal interface is the MATCH (Multimodal Access To City Help) [6] architecture. MATCH is another map based application for PDAs. MATCH provides a mobile multimodal speech pen interface to restaurant and subway information for New York City. In brief, users interact with a graphical displaying restaurant listings and a dynamic map showing locations and street information. They are free to provide input using speech by drawing on the display with a pen or by using combination of both modalities. Both systems (QuickSet and MATCH) are inspired by Bolt's “Put that there” prototype [7].

More specifically, in Bolt's system, the user sits on a chair, which is supplied with two joysticks one at each arm, which are sensitive to pressure and direction. Nearby each joystick there is a touch pad. By using the joysticks and the touch pads, the user can navigate through the data information (maps, e - books, videos etc), which is projected in a large screen placed in front of him. The user also can create and position simple entities onto the screen. The user can use speech and pointing gestures, in order to create and move objects on the screen. For example, user can point to a spot on the screen and say “create a blue rectangle there”, and a blue rectangle will appear at the specific point. Also user can say “Put that there”, and the object the user pointed at, will move to the spot that the user's hand indicated on the screen.

3.1.2 IPhone based Multimodal System

Our system benefits from the combination of the graphical user interfaces (GUI) and the voice user interfaces (VUI). Previous research has indicated that when GUI and VUI are combined to create a multimodal system, they provide high complementarity [3]. Since GUI interfaces offer easy error correction and have low error rates and speech is not error free, their combination offers high speech accuracy and performance. On the other hand, since speech is faster for list selection, when list contains more than fifteen items, those two modalities' fusion can offer high speed in a reservation task's termination. Users can interact with the system through the multimodal user interface, which is generated automatically from the application ontology and the interface specification described in [2].

3.2 Graphical User Interface

A graphical user interface (GUI) allows people to interact with electronic devices, by providing graphical icons and visual indicators. Users' actions in GUI interfaces are usually performed through direct manipulation of graphical elements. By direct manipulation we mean the interaction style, which involves continuous representation of objects of interest, and rapid, reversible, incremental actions and feedback. The intention is to allow users manipulate objects presented to them, using actions that correspond to the physical world.

In our application, GUI interface consists of three main views, one for flight reservation namely "Flight", one for hotel reservation namely "Hotel" and one for car reservation namely "Car". Each view is separated in three main parts: two tab bars at the top of the screen, a table view that contains a navigation bar with the view's title and name - attribute value pairs (according to the necessary information each reservation type needs) and a speech button at the bottom of the screen that provides access in speech modality (Figure 3.2). Speech button takes three different colours (Figure 3.1):

1. Gray, means that speech button is inactive
2. Red, means that Voice Activity Detection is active
3. Yellow, means that speech button is active and user can start speaking

So in GuiOnly the speech button is gray as it is inactive. In Click to Talk the speech button is gray but if users want to talk they have to click on it, wait until the button's colour turn into ren and then they can start to speak. Finally, in Open Mike - Speech Input the speech button is yellow as is active, and when user starts to speek it turns into red.

Selected attribute fields (for example "Departure City", "Airlines" or "Hotel Name" etc) are implemented as detailed views, that contain all possible values

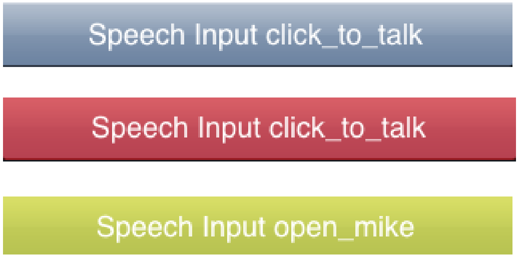


Figure 3.1: Speech Button Colors

(Figure 3.3).So, if users want to fill an attribute value, they have to tap onto the indicator lays on the right side of each attribute. When user taps the indicator, a detailed view pops out animated with all possible choices.

After a choice is done, our system returns automatically to the parent view and fills the attribute's value, that lays below the corresponding attribute name, with the selected value, as (Figure 3.4) depicts. The backend undertakes to update the parent view, meaning that the backend retrieves the information by the user, and then updates the view.

The transition from one reservation state to another one (for example from "Flight" reservation to "Hotel" reservation) can occur in two ways. In the first way, the transition takes place smoothly through the system's dialogue flow. Otherwise, flight, hotel and car reservation are accessible as tab bar items at the top of the screen and user can tap on to the item they want to interact with.

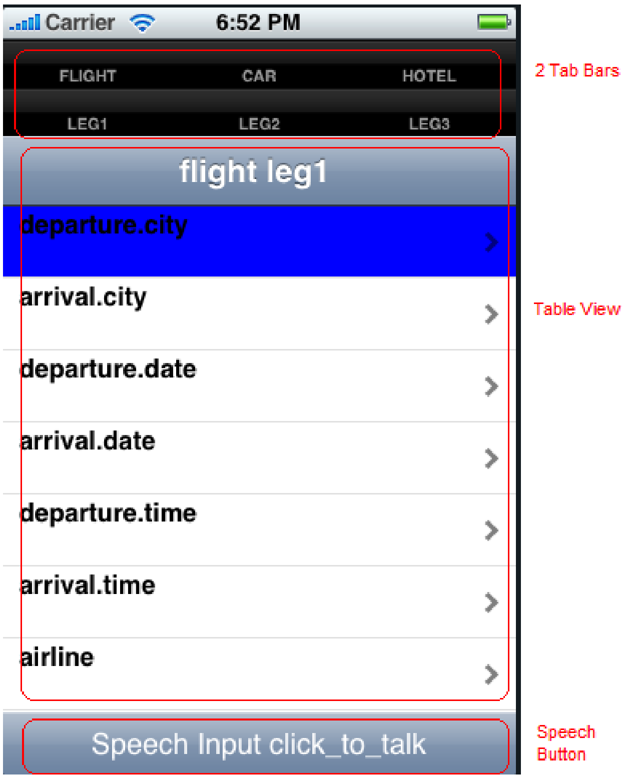


Figure 3.2: Flight Leg1 View

Users have the opportunity to choose if their trip will be one way, two way or three way. One way trip means that the user goes from city A to city B. Two way trip or round trip means that the user goes from city A to city B and returns to city A. Finally three way trip means that the user goes from city A to city B, then to city C and finally returns to city A.

Consequently, since a trip can be one, two or three way, our system gives the user the opportunity to select for example if they are going to have a one way trip or a round trip etc. Navigation to different flight legs is implemented with tab bar items too, at the top of the screen above the reservation items. Consequently, the user can either tap the flight leg they prefer or follow the dialogue flow, as it is provided by our system. Here we have to mention that flight leg 1 state has to be filled before we continue to another reservation or flight leg states.

Also users have the opportunity at any time in the interaction, to specify, correct or modify the attribute values. But also our system asks the user to

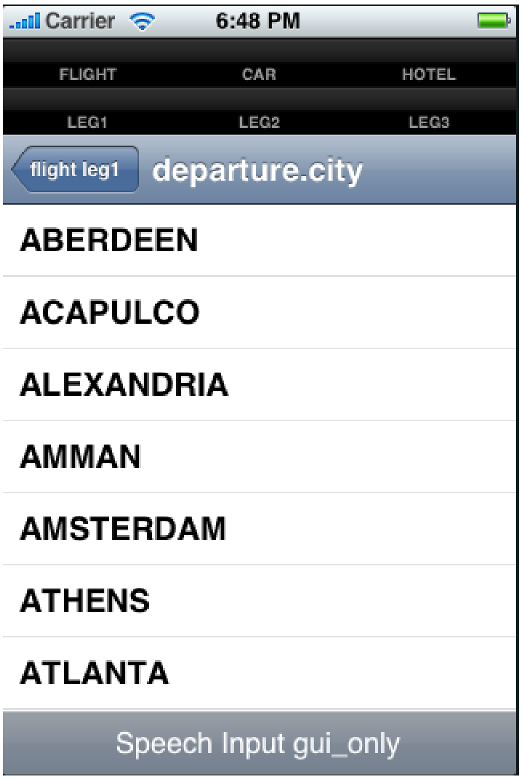
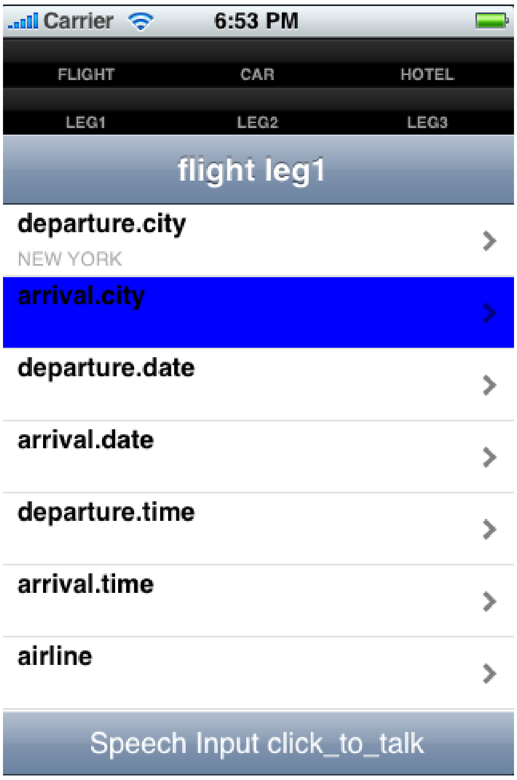


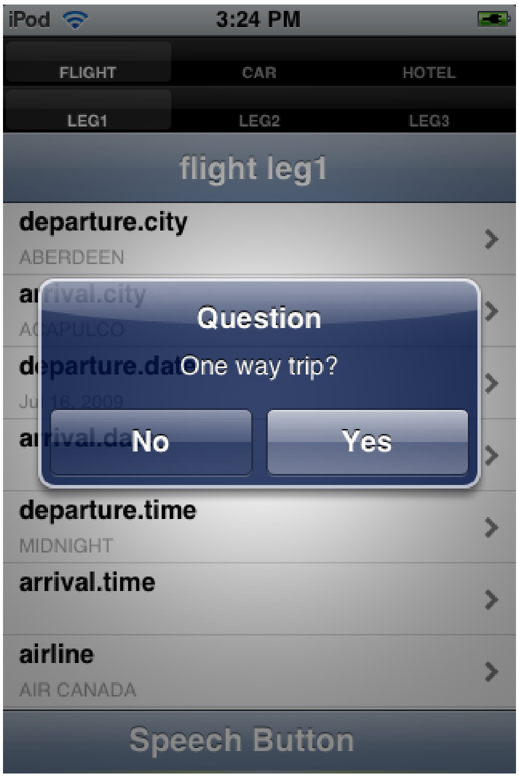
Figure 3.3: Departure City Detailed View



departure.city
NEW YORK
arrival.city
departure.date
arrival.date
departure.time
arrival.time
airline

Figure 3.4: Flight Leg1 view - Value Filled

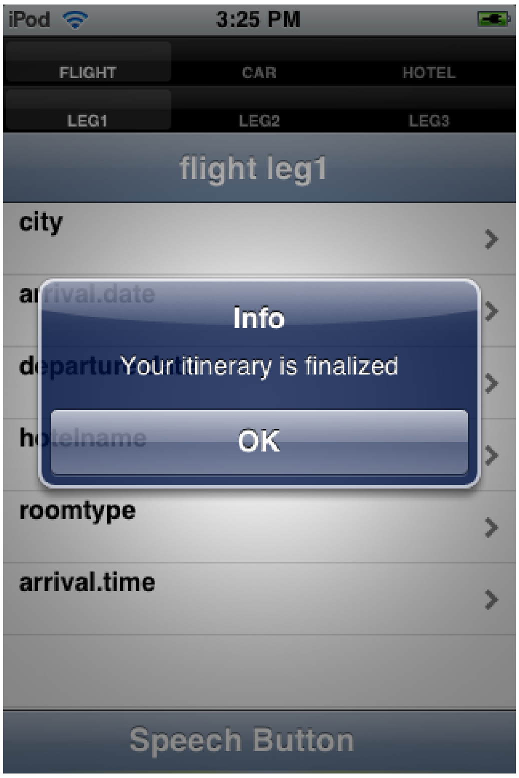
confirm their selection after all attribute values have been filled. By this way, our system tries to prevent possible errors, by giving the user the opportunity to check again their selections if they are not certain about them. Once the attribute values are verified by the user, the system implements a dialogue interface namely alert view (Figure 3.5), asking them if they want to continue further. For example when flight reservation view has been filled and the user has confirmed their choices, an alert view pop ups asking “Would you like to make a hotel reservation?”. Moreover, flight, car and hotel reservation views have some common attributes. So, in order to avoid a scenario, where user selects different values for the same attribute in two different reservation views (for example in “Flight” the user selects New York as the arrival city, while in “Car” selects Los Angeles as a value for the same attribute), our system undertakes to fill automatically the common attribute values and do not allow user to modify them, so possible ambiguities



departure.city
ABERDEEN
arrival.city
ACAPULCO
departure.date
Jul 16, 2009
arrival.date
departure.time
MIDNIGHT
arrival.time
airline
AIR CANADA

Question
One way trip?

Figure 3.5: Alert View



city
arrival.date
departure
hotelname
roomtype
arrival.time

Info
Your itinerary is finalized

Figure 3.6: Confirmation Alert View

can be avoided (Figure 3.6).

3.3 Voice User Interface

A voice user interface (VUI) makes human interaction with an application system possible through a voice / speech platform, in order to initiate an automated service or process. VUI is the interface to any speech application, and provides easy manipulation as users have only to speak out their command. Some years ago, the control of an application by simply talking seemed to be a science fiction scenario. However with the advantages in technology, VUIs become all the more widespread and people are taking advantage of the value that these hands - free and eyes - free interfaces provide in many situations. For example when user's eyes and hands are engaged with an other task.

The VUI, however, often proves to be a major challenge. People have very little patience for a “machine that does not understand”. Therefore, VUIs need to be perfect, otherwise users are going to reject them. Designing a good VUI, requires knowledge in computer science and in human - factors psychology, as well as knowhow. As far as human psychology is concerned, the closer the VUI matches to the user’s mental model of the task, the easier it will be to be used with little or no training, resulting in both efficiency and user satisfaction. The characteristics of the target audience are very important. For example, a system designed for the general public should emphasize ease of use and provide a lot of help and guidance for first time callers. In contrast, a system designed for a small group of power users, should focus more on productivity and less on help and guidance.

Also, a VUI is important to provide high quality of Voice Activity Detection (VAD). By VAD we mean that technique which is used in speech processing, wherein the presence or absence of human speech is detected in regions of audio (which may also contain noise). A VAD operating in a mobile phone must be able to detect speech in the presence of a range of many diverse types of acoustic background noise. In our application a highly flexible audio platform was designed and implemented, which provides both Voice Activity Detection and barge - in, meaning that an extension can break into a conversation between two extensions or an extension and an outside line (for example user can speak over system prompts). The audio platform interfaces with Bell Labs recognizer and the FreeTTS synthesizer through network sockets [3]. The VUI interface is identical to that one described in [2]. The spoken dialogue manager promotes mixed - initiative system - user interaction, meaning that all types of user requests and user input are allowed any time in the dialogue flow. System prompts are focused and try to capture specific information from the user, (for example an attribute value). Confirmation is also used to confirm the values of an attribute. Confirmation is used in all cases through the interaction.

In general our system provides great advantages, as it is a highly mobile system that currently enables mobile users to make travel reservations through iPhone. Also it provides flexible multimodal input. Users interact with a graphical interface either by using speech, or pen, or even by using simultaneous multimodal combinations of the two modes. Our application also provides flexible multimodal output including both speech and visual feedback.

3.4 Mockup

Before an application development starts, a mockup design is essential. Mockup is a full - size model of a device or design, used for demonstration or evaluation of a design. In general mockups are a way of prototyping user interfaces on paper, in order to look like the real thing, but will not do useful work beyond what users see. But before we represent our mockup it is important to put out some of the most important principles (as these are described in “GUI Bloopers” book) [10] that we abide in our graphical user interface.

3.4.1 Focus on the users and their tasks, not the technology

First principle: “Focus on the users and their tasks, not the technology”. Users constitute an important factor in human interface design. As far as users are concerned, it is essential to produce a profile that describes the typical intended user of the planned product or service, or perhaps a range of intended users. The profile should include information such as job, education, age, how familiar they are with the device our application is designed for and physical or social characteristics. The creation of such a profile, gives the developers the appropriate information about what they are aiming at. As far as the task is concerned, it is important for the designers to decide what the intended task domain will be, meaning that before starting to design or implement anything, developers need to learn as much as they can about exactly how the intended users do the tasks that the software is supposed to support. The goal of a task analysis is to develop a thorough understanding of the activities the product is intended to support. In general, to produce an effective product, developers must understand the context in which it will be used. When designers of a software application do not consider the context in which the application will be used, a common result is that the users will find the application awkward. To specify in our application; as it shall be a travel reservation application there are not many restrictions as far as users' profile is concerned. The only restriction is that users must be familiar with the iPhone device and the widgets iPhone applications provide, as that same widgets are used in our application too.

3.4.2 Consider functionality first and presentation later

Second principle yields: "Consider functionality first and presentation later". In many cases GUI developers and even user interface designers, immediately begin trying to decide how the interface will look. Starting by worrying about appearances is a great mistake in user interface design, as it results in products that lack functionality. By "consider functionality first" we mean that software developers should consider the purpose, structure and function of the user interface, before considering the presentation (the interface appearance) of the user interface. But before sketching displays or hacking code, developers should focus their efforts on answering some questions first. For example, what concepts will the product expose to users, meaning that either concepts can be new or users can recognize the concepts from the task domain. Our application concept can be presented as extension of familiar iPhone applications concepts. Also developers must decide what options, choices, settings and controls will the application provide.

This does not concern how to represent controls (for example radio buttons, type - in fields, menus etc.) but their function purpose and role in the product. It is about what options the software provides and what the possible values of those options are. After developer makes their decisions, they are ready to develop a conceptual model. A conceptual model is the model of a product that the designers want users to understand. Once a conceptual model has been crafted, it should be possible to write scenarios depicting how people can use the application. For example the scenario which is presented in our mockup includes a two way trip with a car and hotel reservation. It is important to clarify that second principle does not mean "Get the functionality designed and implemented first, and worry about the user interface later". This approach will definitely lead to unsuccessful software.

3.4.3 Conform to the users' view of the task

Principle 3 says: "Conform to the users' view of the task". Software should be designed from the users' point of view, meaning that developers should perform a task analysis before begin to design. If task analysis does not take place, it is possible to get led to extraneous ("unnatural") activities. "Unnatural activities" is a term that was imported by Jeff Johnson, in order to describe superfluous steps users have to perform in order to get to their goal. Software that requires such actions, seems amateurish and arbitrary to users because such actions are difficult to learn and easy to forget. In order to understand better what unnatural activity means, we are going to use an example from the chess. Moving a piece in a chess game requires indicating, firstly which piece is to be moved and secondly where is to be moved. Any other action besides the selection of the piece to be moved and the specification of its destination, is unnatural. Also, users are not interesting in how the software works, just in complete their work. Therefore, details of the software's interval workings should remain interval, out of sight and mind of the users. The user interface should represent only the objects and the actions, and not information about technology and implementation.

3.4.4 Do not complicate the users' task

Another very important principle is "Do not complicate the users' task", meaning that common tasks should be easy. In any domain, users will have goals ranging from common to rare ones. Consequently, software should be designed to recognize this range. If a user's goal is predictable and common, the user should not have to do or specify much in order to get it. On the other hand, it is fine if unusual tasks require more specification. Additionally, although people are good in multitasking, this ability is limited to stuff they already know how to do, meaning that working out solutions to new problems is one activity that human mind cannot multitask effectively. People have plenty of their own problems to solve and goals to achieve in the domain of their work. That is why they use products and services, to solve those problems and achieve their goals. They do not want to be distracted from those problems and goals by extra ones imposed by the same products or services. Therefore, different products and services should be designed to provide users all the appropriate means, in order to let them focus their attention on their problems and goals, without being distracted. Consequently this principle is very important in designing. Software that is full of inconsistencies, even minor ones, forces users to keep thinking about it, thereby distracting from the attention they can devote to the task at hand. So when beginning a design, we should develop a conceptual model and perform an object / action analysis first. Another important fact to consider when designing an interactive product is that people make mistakes. A product that is risky to use is one that makes it easy for users to make mistakes or makes it costly to correct them. A low risk situation, in which people do not have to worry about mistakes encourages exploration and hence fosters learning.

3.4.5 Other important factors

To specify in GUI designing we have to mention some important factors that developers should have in mind, when they begin to design. Firstly, as in their everyday life, so in their interaction with an application, people need to know where they are. People use environmental clues to see where they are, therefore our interface must provide such clues. Application windows / views identify themselves with a title in their navigation bar, as we show in section 3.1 our application provides titles for each reservation type and trip leg. So it is important applications to title all windows / views, including dialogue boxes. Secondly, different title must be provided for different windows / views. Sometimes windows / views may have the exact same title, something that can mislead users about where they are. In order to avoid something like that every window / view should have a unique title. Moreover, a software application is designed to support certain user goals and the user interface should guide users toward those goals. So it is essential interfaces not to distract users from their tasks, but help them finish them. Also it is important for each control to be used a consistent name in order the system to be easy in learning and the user to navigate through the application without any difficulties. Either way, Caroline Jarrett, an authority on GUI and forms design said: “Same name, same thing; Different name, different thing”.

Once we have GUI controls that are appropriate for our software application and we have labeled them well, we have to decide on the presentation details, such as layout, color, text font. Usually software developers have not yet learned to develop and follow strict standards for layout and graphic design and to pay much attention to detail. Graphic design and layout blooper definitely diminish software's perceived quality. Poor graphic design and layout can decrease users' ability and motivation to absorb whatever information or content the software offers, and make the product look amateurish and untrustworthy.

Most of the common graphic design and layout bloopers concern the layout of the information and controls and the placement of the windows / views on the display. Many software developers assume that if information is displayed, users will see it. This is not exactly correct, as people miss information constantly. Our perceptual system filters out more than it lets in, meaning that we ignore most of what is going on around us and focus our attention on what is important. This feature help us to function in this rapidly changing world. Consequently, good design focuses users' attention on what is important by taking advantage of how human perception works, but unfortunately, many applications provide unimportant details that draw user's attention away from the important information. We can enumerate, by experience, some cases of such an erroneous graphic design. For example, some software applications display important information in such a small size that in might as well not be there. Also, often important information appear in out - of - the - way locations, where many users won't notice them.

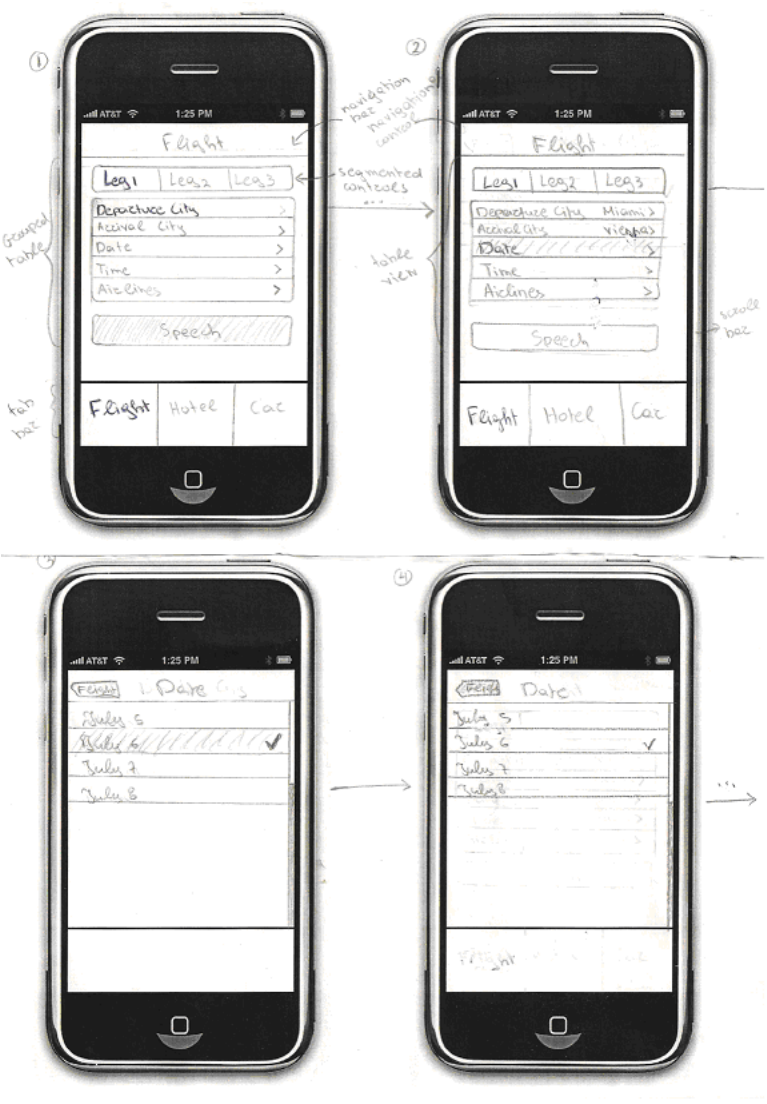
In order to avoid such a blooper, it is important to organize information displays into "pieces", "sub - pieces" etc, so that user can quickly spot them. That lets users to find the important information faster than if they had to scan or read everything on the screen. Also, the on - screen information must be large enough in order to seize users' attention. Moreover, placing information closer to the center of the viewer's filed (peripheral vision is poor) improves its visibility and legibility. The use of color to highlight, important information, is needed in order to draw users' attention. If information is absolutely critical, an effective technique, that can make messages nearly impossible to ignore, is to use dialogue boxes and pop ups (in our application alert views). Error messages, confirmation messages and warnings can be displayed in dialogue boxes that pop out in the screen. In fact dialogue boxes are modal, blocking users from doing anything else with the application until they acknowledge or dismiss the dialogue box. But we have to be careful with dialogue boxes, as sometimes provide no way other than a direction users do not want to go. This can happen if there is not, for example, a "Cancel" button or if all choices are not provided.

Using the above techniques and following the principles that were previously described we designed our system. Our system design lays below:

3.4.6 The Mockup Design

Here we see a part from the mockup design. In the mockup design we present the alternance between the iPhone screens, according to a certain scenario. Here for example the first and the second screens depict the designed Flight view; the first one depicts the flight view as it is initially while the second one depicts the same view after some interaction. In the third and the fourth screens we see how users can interact with the detailed views. The rest mockup design is in appendix A.

For example the user wants to interact with the attribute date. The mockup depicts the way that users will interact with the different attributes in the real application. Users will have to press onto the attribute they want to interact with (e.g Date), and then they have to make their choice among the different list items (Detailed View).


①
②
③
④
1:25 PM
Flight
Leg1
Leg2
Leg3
Departure City
Arrival City
Date
Time
Airlines
Speech
Hotel
Car
Miami
Vienna
navigation bar
navigation control
segmented controls
grouped table
table view
tab bar
scroll bar
Date
July 5
July 6
July 7
July 8

Chapter 4

Evaluation

4.1 Introduction

In general multimodal systems are particularly challenging to evaluate and since the field is still new, there are few commonly accepted practices and standards.

4.1.1 Types of evaluation

Depending on what an evaluation wants to examine, we can distinguish three evaluation types [9].

4.1.1.1 Adequacy evaluation

Adequacy evaluations centered on the system"s fitness for a purpose, meaning that this type of evaluation tries to extract information about how well the system meets the requirements and at what cost. The requirements are mainly determined by user needs. So, user needs must be defined first.

This approach involves more than one users completing one or more tasks. Task, user and environment characteristics must match those for which the system is being designed. Data on how user and system behave are collected while the user performs experimental tasks.

4.1.1.2 Diagnostic evaluation

Diagnostic evaluation obtain a profile of system performance. This requires the specification of an appropriate test suit. This approach involves an expert using the system in a more or less structured way, to determine whether the system matches the predefined criteria and guidelines. This type of evaluation yields the subjective evaluator"s judgment on the system"s conformity to general human factors, principles and guidelines.

4.1.1.3 Performance evaluation

Performance evaluation measures system performance in specific areas. This type of evaluation is only meaningful if a well - defined baseline performance exists, meaning a previous version of the system, or a different technology that supports the same functionality.

This approach involves a designer or evaluator calculating the match between the task or user model, and the system specification. This generates quantitative values for the learn - ability or the usability of the evaluated system. This evaluation type needs neither user - computer interaction nor a system prototype.

4.1.1.4 Why multimodal systems" evaluation is challenging

As it has already been mentioned, evaluation of multimodal systems is challenging [9]. But why?

The point of multimodal systems lies in the combination of more than one, different modalities. Since multimodal interaction is by nature application specific, there are currently no benchmark databases for multimodal applications. However, the available benchmarks for the component technologies are useful in evaluating the performance of the components of a multimodal application.

Moreover, multimodal interaction is hard to record under normalized, easily reproduceable conditions. Multimodal interactions depend on user"s behavior and current hardware / software. The evaluation criteria are frequently unclear, in part since qualitative aspects play a significant role. This is in contrast to component recognition technologies, where accuracy is a commonly accepted criterion. So the lack of commonly accepted evaluation criteria makes it difficult to compare across different evaluations of multimodal systems

And last but not least, the evaluation of qualitative aspects is difficult, since user studies are very costly and user self - reports are unreliable because they are subjective.

4.2 Evaluation Methodology

This work is based on a previous work [3, 8].

For the purposes of this thesis three modes were evaluated, two unimodal namely "GUI only" and "OpenMike - Speech input" and one multimodal namely "Modality Selection". As its name indicates, "GUI only" mode provides only the GUI modality for interaction. The "OpenMike - speech input" mode provides speech modality as input and GUI accompany with speech as output, thus this mode will help us to investigate the audio / visual feedback effect. And last but not least the "Modality Selection" mode gives the user the opportunity to combine the two modalities (gestures and speech) in a way considered to be the most efficient.

Our goal is to investigate input modality usage from the user point of view and to understand efficiency considerations and user biases in input mode selection. Our goal is not only to compare the efficency among different systems, but also measure the various factors that affect the efficiency and modality selection by users [3].

4.2.1 Evaluation Setting

These three systems were evaluated on five travel reservation scenarios of high enough complexity: one one - way trips, three round - trip and one three - way trip and hotel/car reservations, as Table 4.1 presents. Thus, we can collect enough data, without the need for many users. Table 4.2 shows the type of the five scenarios. The five scenarios are:

- From Las - Vegas to Miami on August 25th in the morning with Northwest airlines.
- From Orlando to Boston on August 28th in the morning with Quantas airlines. Return on August 29th in the evening.
- From Miami to Vienna on August 26th in the morning with United airlines. Return on august 27th in the evening. Reserved hotel is Four Seasons
- From Tucson to Phoenix on August 26th in the morning with Southwest airlines. Return on August 28th, anytime. Car rental of a Station Wagon type car from Budget.
- From Tucson to Orlando on August 26th in the morning with TWA airlines. Next flight to Phoenix on August 29th, anytime. Return to Tucson on August 31st in the evening

Scenario ID	leg1	leg2	leg3	hotel	car
1	x				
2	x	x			
3	x	x		x	
4	x	x			x
5	x	x	x		

Table 4.1: Evaluation Scenarios [3]

Evaluation took place in a quiet office environment with the server software (spoken dialogue system, speech platform, task manager) running on a host desktop computer and the client software (graphical user interface) running on the iPhone.

Initially each user is given a sort introductory document which explains the system functionality, with emphasis on the modes that are going to be evaluated. Then to familiarize users with the system, each user is asked to complete a demo task using all three modes. After that sort process, each user is ready to complete the five scenarios, using all three modes. Systems were evaluated in random order

attribute name	scenario usage					
	1	2	3	4	5	total
hotel name	0	0	1	0	0	1
city	2	3	3	3	3	14
airline	1	1	1	1	1	5
date	1	2	2	2	3	10
car type	0	0	0	1	0	1
car rental	0	0	0	1	0	1
time	1	2	2	2	3	10

Table 4.2: Attribute usage for the five travel reservation scenarios [3]

and logs for each session were saved for later processing by our analysis software (Matalab scripts). Upon completion of each run user is asked to fill in a short questionnaire, according to their own subjective point of view.

4.2.2 Objective Evaluation

Interface evaluation of multimodal dialogue systems is a fairly complex task and different metrics may be used to evaluate different aspects of such systems. Such metrics, namely objective metrics are [3]:

4.2.2.1 Modality Selection and input modality overrides

We measure the usage of each input modality as a function of number of turns, and duration of turns attributed to each modality. Modality usage is also measured as a function of context, for example attribute for which input is expected.

Another measure, is the number of input modality overrides, for example the number of turns that users preferred to use a mode other than the one proposed by the multimodal system. Low numbers of overrides reveals that the multimodal mode matches user”s modality preferences or that the modality selection process is system - initiated for the user. High number of overrides, on the other hand, reveals a mismatch to user”s modality preferences and a power - user that takes the modality selection initiative.

4.2.2.2 Turn duration, inactivity and interaction times

Duration statistics at the turn and task level are important factors, since efficiency is defined as being inversely proportional to task duration. Inactivity time, refers to the idle time interval starting at the beginning of each turn, until the moment the user actually interacts with the system using either GUI or Speech input. Particularly in GUI input, inactivity time is specified as the time between the start turn and the moment the user clicks onto GUI. For the case of speech input, inactivity time is specified as the time between the start turn and the moment of a voice activity detection event. During this time period, user has to understand system"s response and state and then plan his own response. The response, in general, includes entering the system"s requested information, using a preferred modality at each turn. By interaction time, we define the time interval between the moment that the first event occurs and the moment that systems recognizes that user made a selection. For GUI input, interaction time is specified as the time between the first touch event and the moment that user makes their selection. For speech input, interaction time is the time betwen the first VAD event and the moment that ASR result becomes available. Figure 4.1 depictes inactivity and interaction times.

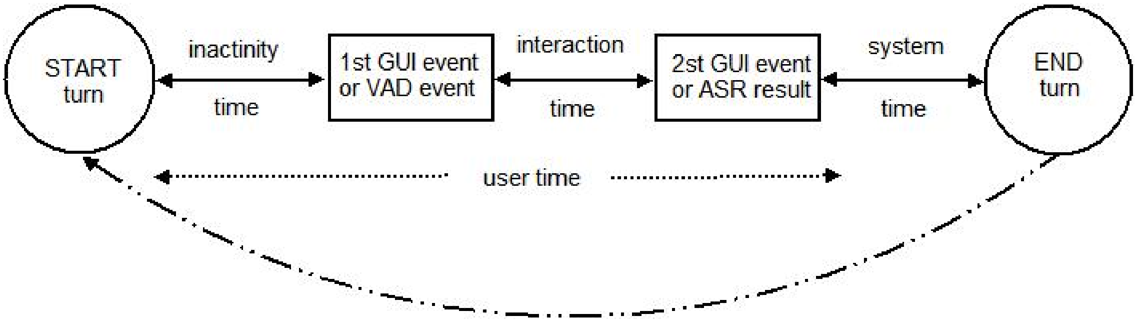


Figure 4.1: User inactivity time, user interaction time and system time [3]

4.2.2.3 Context Statistics

Context statistics refer to objective measures regarding attributes such as city, hotel name, airline, date, car type, car company. Thus, it is expected , that in “Modality Selection” mode the default modality is chosen according to the number of available values for each attribute (for example if the list contains more than fifteen items it is expected that Open Mike with Speech input will be the default modality of interaction).

4.2.2.4 User Statistics

User variability is another important issue that has to be investigated. The relative efficiency of each modality is different for each user, due to the fact that each user has different experience with each modality and different capabilities. Also users have different modality preferences, which largely affects the modality selection and performance of course. Meaning that some users were familiar with multimodal interfaces but some others were not, some users were quick in using the GUI for large lists but some others were not, etc. Finally, another factor that affects the efficiency is the different speech recognition accuracy, meaning that voice quility and accent play a significant role in speech recognition.

4.2.3 Subjective Evaluation

In subjective evaluation we are generally concerned about user’s point of view about our system. Thus, measures such us efficiency, usability and user satisfaction are mainly interest the evaluators. Efficiency, in general, is centered on the time required to perform a certain task (for example: Was the time enough to perform a task?), while satisfaction is mainly concern the system preference (for example: Is the user satisfied enough, in order to use this system again?). Furthermore, usability has to do with criteria such as the amount of interaction (for example, how many actions needs a user to do, in order to complete a task), the ease of search (for example: Is it easy enough for the user to browse through the application?), the ease of learning to use and finally the ease of use. In general it is critical to offer to users a system, that will encourage them to use it again.

The main idea is to ask the users questions about the overall system and their impression. So after the objective evaluation, a short interview was conducted by the evaluators. The users were asked to score each system according to the following evaluation criteria:

- Overall Impression

    1. Overall impression
    2. Graphical User Interface
    3. Voice User Interface
    4. System's functionality

- System's Efficiency

    1. Task comletion speed
    2. Response time

- User's Satisfaction

    1. The system worked the way I expected
    2. I would like to use this system again

- System's Usability

    1. Overall impression
    2. Graphical user interface
    3. Voice user interface
    4. System's functionality
    5. Ease of learning to use
    6. Ease of use
    7. How easy was to get the information
    8. How easy was to understand what to do
    9. I knew what to do/say at each turn

4.3 Evaluation Results

4.3.1 Objective Evaluation Statistics

4.3.1.1 Modality Performance Results

Figure 4.2 show the user times for the three evaluated systems.

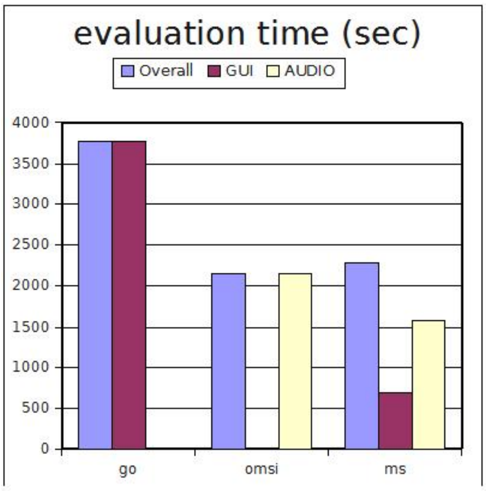


Figure 4.2: Overall usage time per system

We can denote that "GUI only" is the slowest mode, as users had to search in large lists until find their preference, a really time - consuming process. While, the "Open Mike - Speech Input" mode seems to be the the fastest, closely followed by the "Modality Selection".

In Figure 4.3 is depicted the number of turns for the three evaluated systems.

Here we can see that the number of turns in "Open Mike - Speech input" exceeds the other modes' turns. This is attributed to the fact that in OMSI the only way to correct someone a false option, is to use speech. So in many cases users had to repeat (3 or 4 times) what they said until the system understand correctly the user's choice. Also, it is clearly comprehensible by Figure 4.4, that users in "Modality Selection", use efficiently enough both modalities. The GUI input usage is 32 percent, while the speech input is 68 percent, which was expected as large attributes (departure city, arrival city and airline)are more than small ones (date and time).

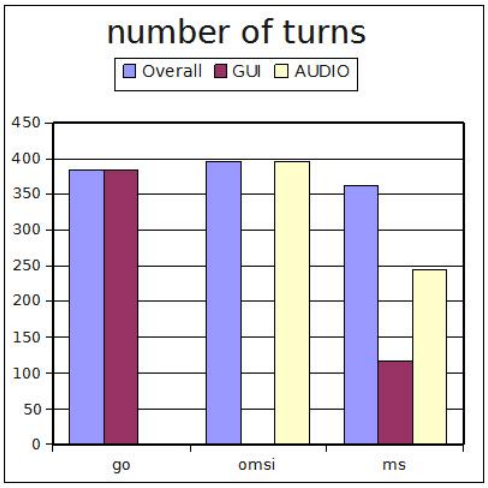


Figure 4.3: Overall number of turns per system

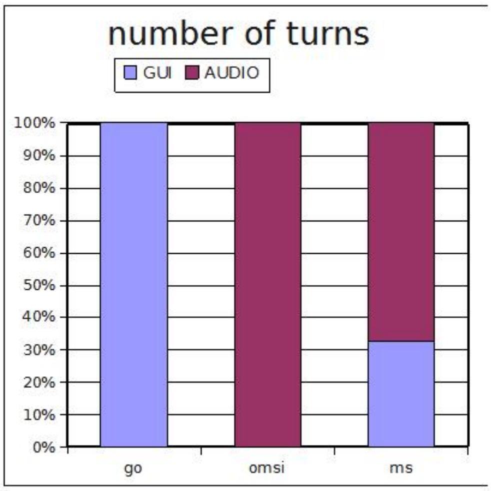


Figure 4.4: Percantage number of turns per system

It is remarkable that turn duration in “Modality Selection” mode is higher than in “Open Mike - Speech input” mode, as Figure 4.5 shows. That happens because in “Open Mike - Speech input” mode speech is anyway the active input modality, so users can speak their preferences without any delay. While, in “Modality Selection”, when interaction comes up with small attributes GUI is the default input modality, and if users want to speak their choice, they have to press the speech button and after a while they can speak. This process, import a small delay, which is detectable by the system.

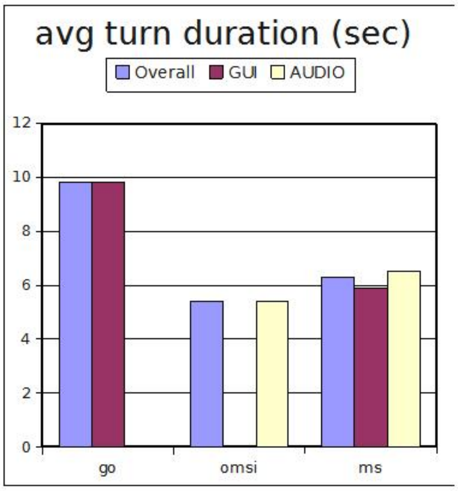


Figure 4.5: Average Turn duration per system

4.3.1.2 User Statistics Results

Table 4.3 and Figure 4.6 show the speech recognition accuracy per user. In order to measure the speech recognition accuracy, we calculated the Word Error Rate (WER) per user, which is a common metric of the performance of a speech recognition. We calculated the WER by first aligning the recognized word sequence with the reference (spoken) word sequence using dynamic string alignment. Word error rate can then be computed as:

$$WER = \frac{I + D + S}{N}$$

Where,

- I is the number of insertions,
- D is the number of deletions,
- S is the number of substitutions,
- N is the number of words the reference.

The Word Recognition Rate (WRR) or speech recognition accuracy is defined as the percent of words recognized correctly over the total number of words said by the user:

$$WRR = 1 - WER = \frac{N - I - D - S}{N} = \frac{H - I}{N}$$

Where,

$$H = N - D - S$$

is the number of the correctly recognized words.

User	WRR
User1	92.1
User2	88.3
User3	44.6
User4	53.8
User5	78.8
User6	83.6
User7	69.9
User8	50.8
User9	65.1

Table 4.3: Speech Recognition Accuracy per User

As we can see the WRR ranges between 44.6 percent and 92.1 percent. The WRR presents such a large range because users differ in voice characteristics, such as accent, voice tone, intonation etc. As we mentioned in chapter 2 subsection 2.2.2.1 a problem that frequently encountered in automatic speech recognition is background noise, such as clicks of tongue, pauses or other grunts, that accompany speech and make its recognition more difficult. Also voice quality and intonation are important features of speech, and thay can also affect the ASR's accuracy and the performance.

For example users who had bad accent, stressed the wrong syllable or they had high - pitched voice, there was difficult for the ASR to recognize them. As a result their accuracy rate is too low.

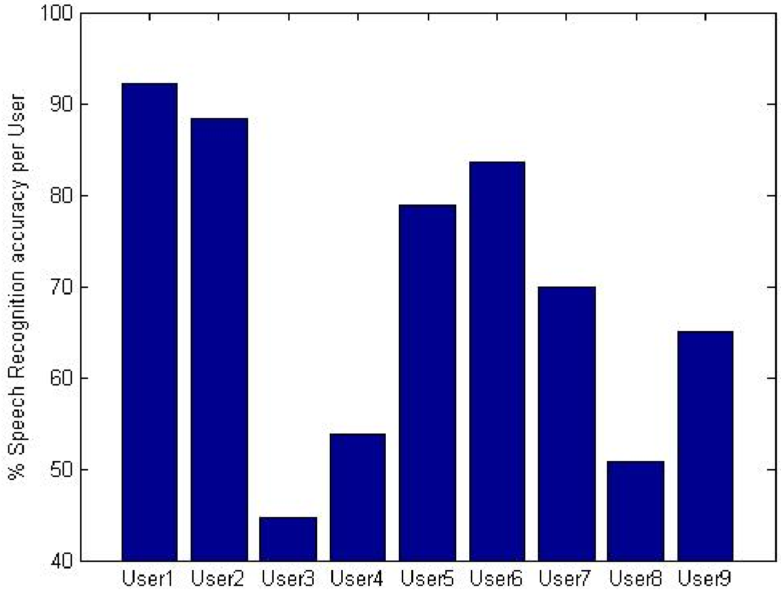


Figure 4.6: Word Recognition Rate per User

Figure 4.7 shows the task duration for each user, for the three evaluated systems. Task duration per user is further broken into GUI and speech input. The figure yields that performance of "GUI only" mode highly varies between users (user 5 is two times slower than user 2). Also, for all users "Modality Selection" is more efficient compared to the "GUI only", but only for users 3, 4, 7, 8 and 9 "Modality Selection" seems to be more efficient than "Open Mike - speech input" mode. The latter, is attributed to the fact that some users had poor speech accuracy than others, so for them the "Open Mike - Speech Input" was time - consuming.

Figures 4.8 and 4.9 show the number of turns and the average turn duration for each user. Here it is remarkable that users with high speech accuracy (user 1 and user 2) they rarely use the GUI modality in "Modality Selection" mode, as they prefered to say all the attribute values at once. It is also important to denote that, users with poor speech accuracy (user 3, 4, 5, 7, and 9) they use GUI and speech input (in "Modality Selection") just as often. Also, users that have poor accuracy in speech recognition, they waste a large number of turns in

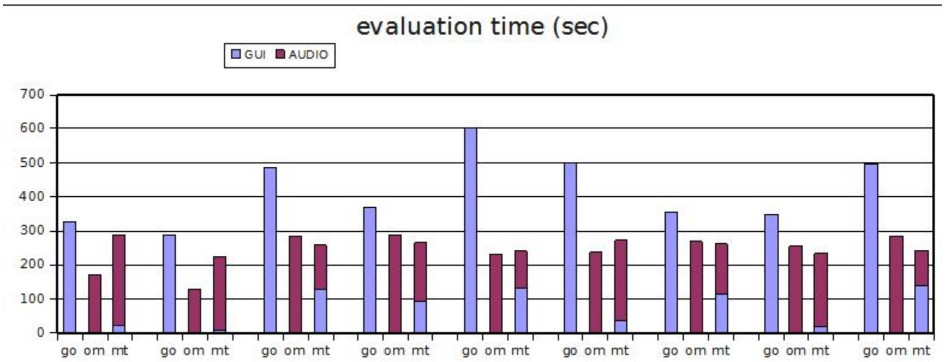


Figure 4.7: Task Duration per user

"Open Mike - Speech input" mode, sometimes larger than that in "Gui only".

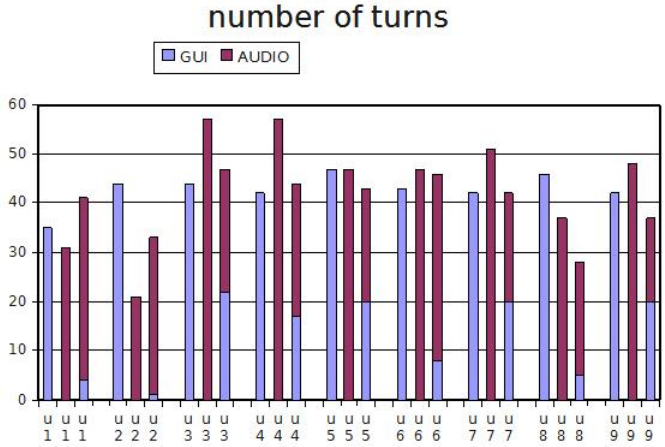


Figure 4.8: Number of turns per user and per system

For GUI input turns, the average turn duration is between 13 and 6.5 secs, while for speech input, the average turn duration is between 5 and 7 secs. As far as the "Modality Selection", is concerned, the average turn duration for all users is around 6 secs.

4.3.1.3 Context Statistcs Results

Table 4.4 and Figure 4.10 show the speech recognition accuracy per context.

As we can see the WRR range between 52.4 percent and 97.1 percent. The date has the poorest Word Recognition Accuracy, something that is expected, as

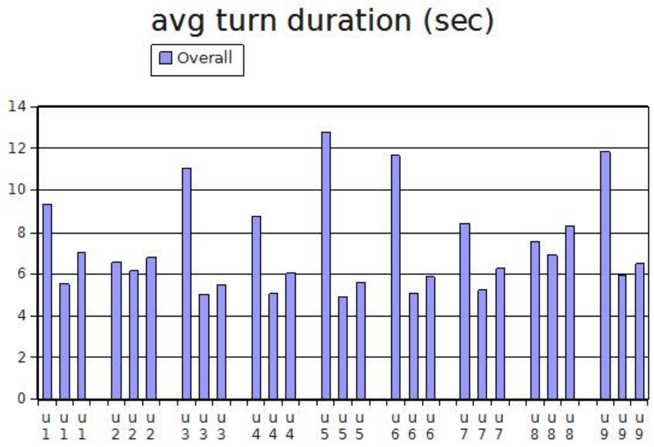


Figure 4.9: Average turn duration per user and per system

Context	WRR
City	75.8
Date	52.4
Time	97.1
Airline	77.8

Table 4.4: Speech Recognition Accuracy per Context

most users confront great difficulties in date recognition.

Figure 4.11 shows the speech recognition accuracy for the "Open mike - speech input" mode and the "Modality Selection" mode.

We denote that, the "Modality Selection" mode exceeds in speech accuracy (80 percent), in contrast with "Open Mike - speech input" mode, which presents low accuracy (60 percent). This is attributed to the fact that in OMSI the only way to interact someone with the application is speech. So users are obligated to use speech, even when the system has problem to understand what they said. On the contrary, in "Modality Selsction", when users denote that the system does not understand, they have the alternative to use the GUI modality.

Figures 4.12, 4.13 shows the total evaluation time per context. As it is expected, for the large attributes, such as city and airline, "GUI only" significantly differs from the other two systems.

Also it is remarkable that in large attributes such as city and airlines, users

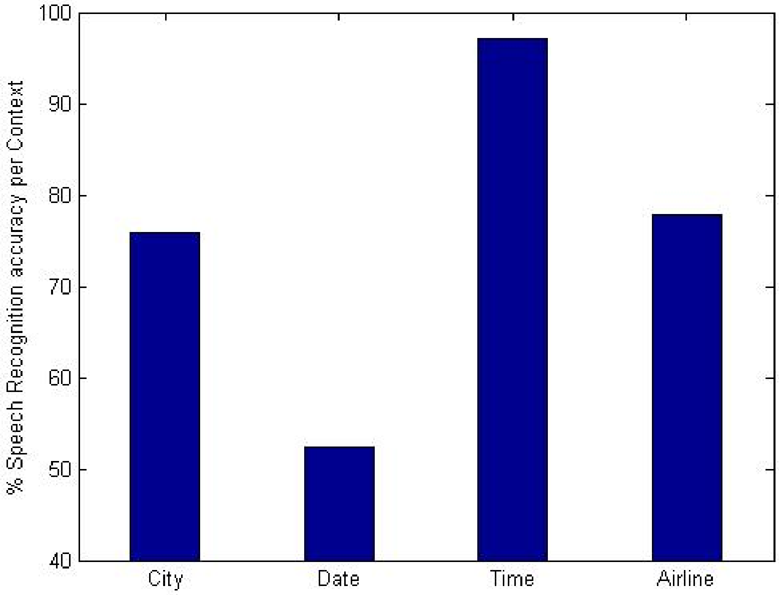


Figure 4.10: Word Recognition Rate per Context

prefered not to use the GUI input, for the “Modality Selection”, but only speech, while in small attributes users prefered most the GUI input (60 - 65 percent), rather than speech input (35 - 40 percent) (Figure 4.14). Speech bias is comprehensive, if we consider the table 4.4 and the Figure 4.10, where large attributes (city, airline) seem to have high enough Word Recognition Rates, so users prefered to interact with speech rather than GUI, even in cases where the system had problem to understand what user said. On the other hand, in interaction with small attributes such as time and date, users prefered mostly the GUI modality. This is expected as far as the date attribute is concerned, as the system had great problem to understand the date user said. But also, is is easier to interact with small lists by using GUI rather than speech.

Figure 4.15 show the percentage number of default input modality overrides for the ”Modality Selection” mode.

As it is expected for the two large attributes (city and airline) users prefered the suggested speech input modality so there are no input overrides, while for the small attributes (date and time) the modality overrides are 40 and 35 percent

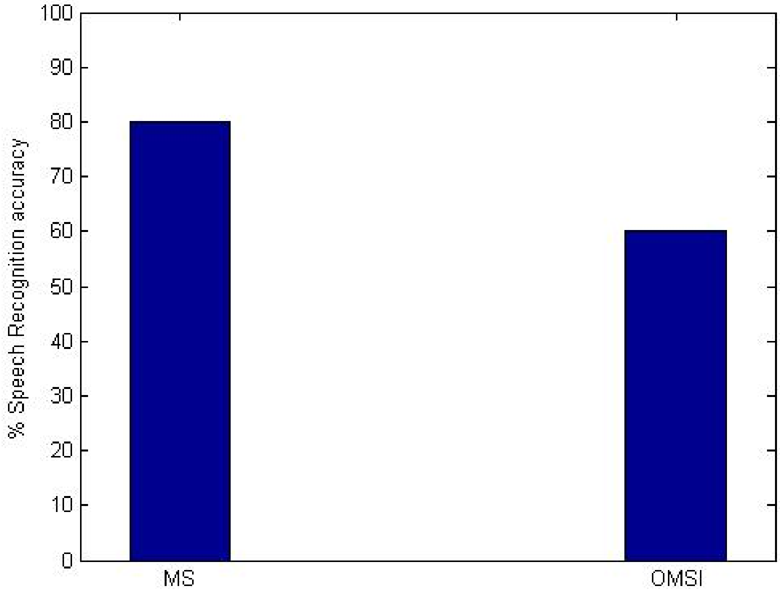


Figure 4.11: Word Recognition Rate for MS and OMSI modes

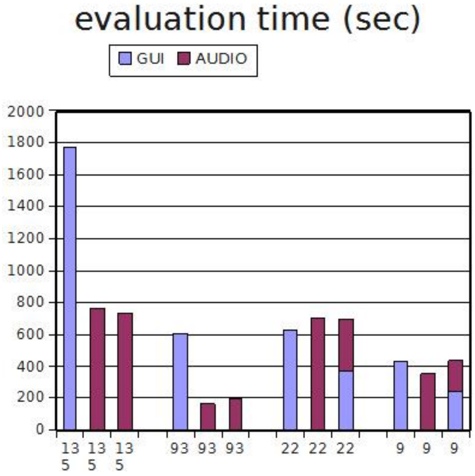


Figure 4.12: Total Evaluation Time per context and per system

each.

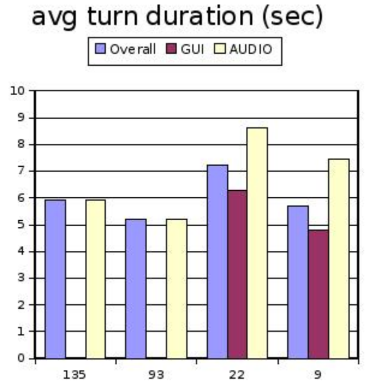


Figure 4.13: Average turn duration per context

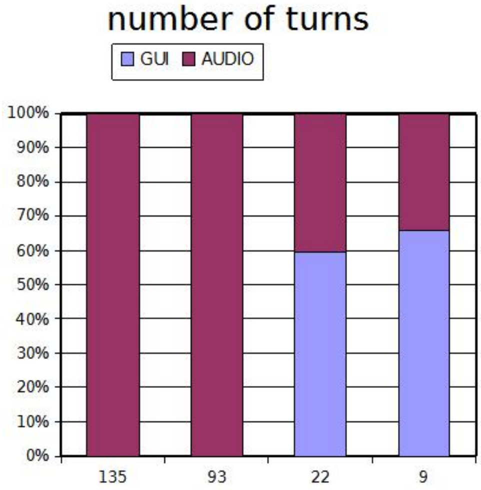


Figure 4.14: Percentage number of turns per context and per input modality

4.3.2 Subjective Evaluation Statistics

In Table 4.5 and Figure 4.16, is shown the subjective evaluation scores, that were supplied by the users after the exit interview.

Results show that the three systems insignificantly differ. Nevertheless,users seem to prefer better the GO system, due to the fact that they feel more familiar with grafical interfaces, although it is difficult to manipulate large lists through them. Many users mentioned that it would be very helpfull if in large lists there was a surch bar or something like that, in order to minimize the number range among the options.

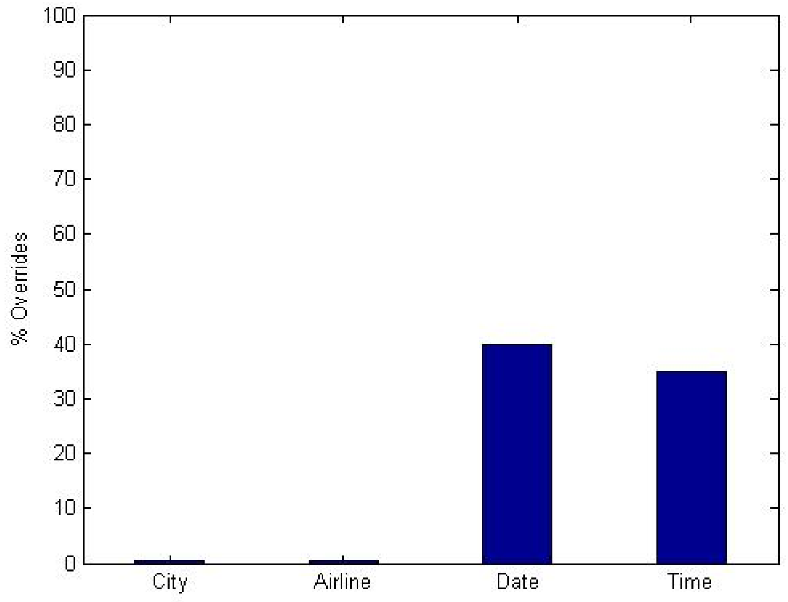


Figure 4.15: Percentage input modality overrides for the MS mode

User	GO	OMSI	MS
user1	9	8.9	5.1
user2	8.1	8.5	8.3
user3	9.5	7.3	8.8
user4	9.7	8.1	7.4
user5	8.3	7.7	10
user6	9.4	9.1	8
user7	8.5	8.5	8.9
user8	8.8	5.7	7.8
user9	9.4	8.2	8.8
Average	8.97	8	8.12

Table 4.5: Subjective Evaluation Results

Second in users' preference comes the MS system. Although, MS seems to be the most efficient system as far as velocity and task completion time are concerned, many users found it difficult to detect the alternance between the

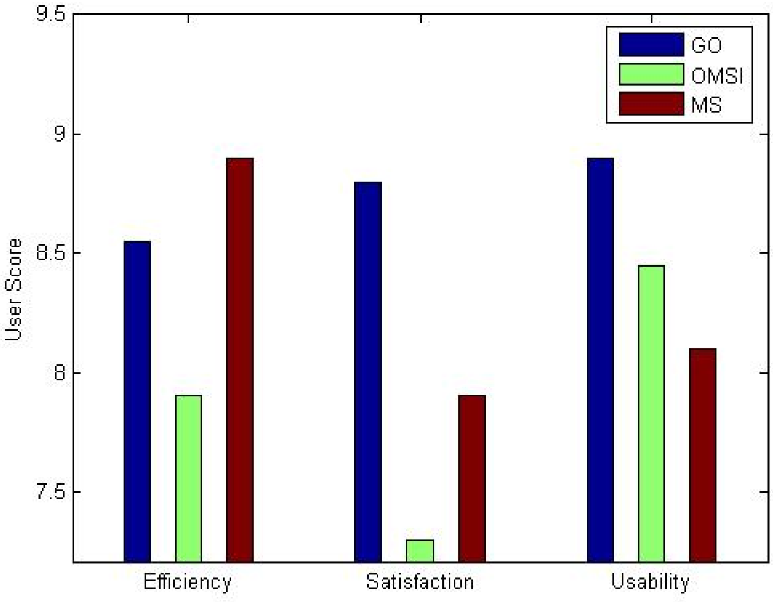


Figure 4.16: User Evaluation Results per system and per criterion

two modes. Most users drifted by the system's speech output and as the speech button was not in the field of their view, they used speech although the mode had turned into Click to Talk. In MS system most users mentioned that it would be better if there was a bleep to inform them that modality changed.

Finally, last in users's preference comes the OMSI system, because most users had problems with speech recognition. So they found it the most time - consuming and the less efficient system.

Also we conducted ANOVA analysis for the three different systems. A within subjects ANOVA shows that the effect of GO ($p < 0.001$), OMSI ($p < 0.001$) and MS ($p < 0.001$), are highly significant. So the three evaluated systems, as far as performance is concerned, are significantly differ, as Figure 4.17 shows.

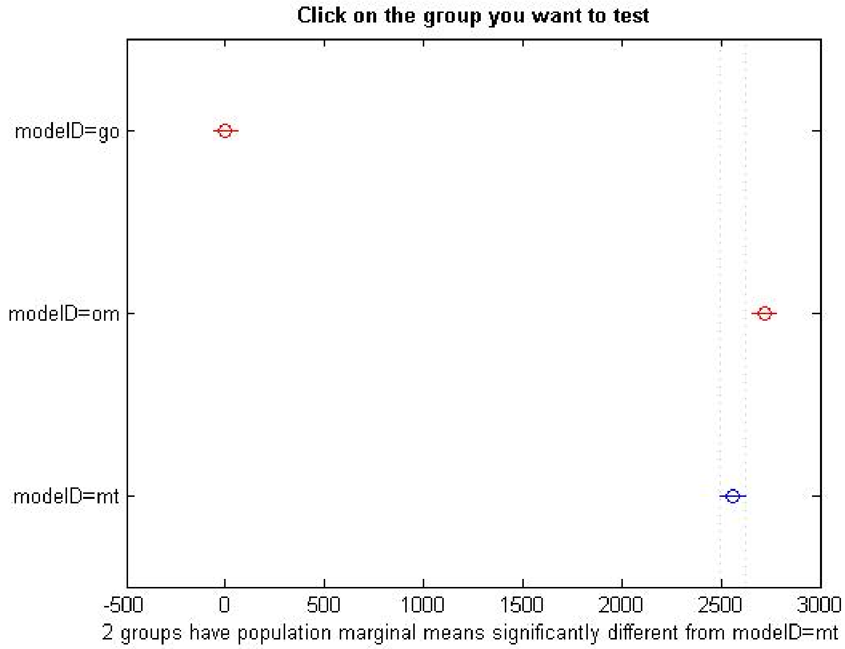


Figure 4.17: ANOVA analysis

Chapter 5

Discussion, Conclusions and Future Work

5.1 Importance of results and implications

We have shown that the multimodal system is indeed more efficient than the other two systems. “Modality Selection” outperform the other two modes.

What is noteworthy, is that users tend to use GUI input more often when it is the active input modality in “Modality Selection”, while in large attributes, shuch as city and airline, users use only speech, even in cases where they want to correct an option. An explanation for this is that users in general, try to avoid interaction with large lists when the speech accuracy is high.

Also important is the fact that, speech errors affect the input modality selection, meaning that in attributes where Word Error Rate is high enough, users prefered mostly the GUI input modality, rather than speech. But in general, users tend to use the most efficient mode at each turn.

Moreover, the subjective results show that users prefered mostly the “GUI only” mode, while the lowest in users preferences is the “Open Mike with Speech input”. The “Modality Selection” mode, was low rated since the users found it a little bit awkward, meaning that users could not denote easily the alternance between the different modes. “GUI only”, was high rated because users were more familiar with graphical interfaces, and felt certainty, in contrast with the other two modes.

Finally, if we compare the current results with those came up by previous studies, we could say that in this application users found the large attributes manipulation with GUI a time consuming task, so they avoided to use it in large attributes selection. While in previous systems, users sometimes prefered to use the pen input for large lists selection, as they could scroll at once in the area of their interest. So the GUI manipulation for large lists in the current system seems to be less easier compared to that in previous systems.

5.2 Future Work

There are a few aspects of this work that can be extended or improved. In terms of implementation, the way the application works is still a bit primitive. There is a lot of room for work in terms of interfacing with the end user. Emotion detection and the replacement of the GUI by maps are probably the next steps in this work.

The idea is that user will interact with maps instead of GUI forms, and the system will have the capability to alternate among different modalities, based on user's emotions. If, for example, user feels stress because system does not understand, the system should detect the stress and immediately should change modality.

Finally it would be interesting evaluating the unimodal and multimodal systems for various levels of task complexity and interface efficiency (for example, different speech recognition error rates). Through this experiments, we could export important conclusions about modality usage, unimodal and multimodal interface efficiency, and the relationship between efficiency, user satisfaction and input moudality usage.

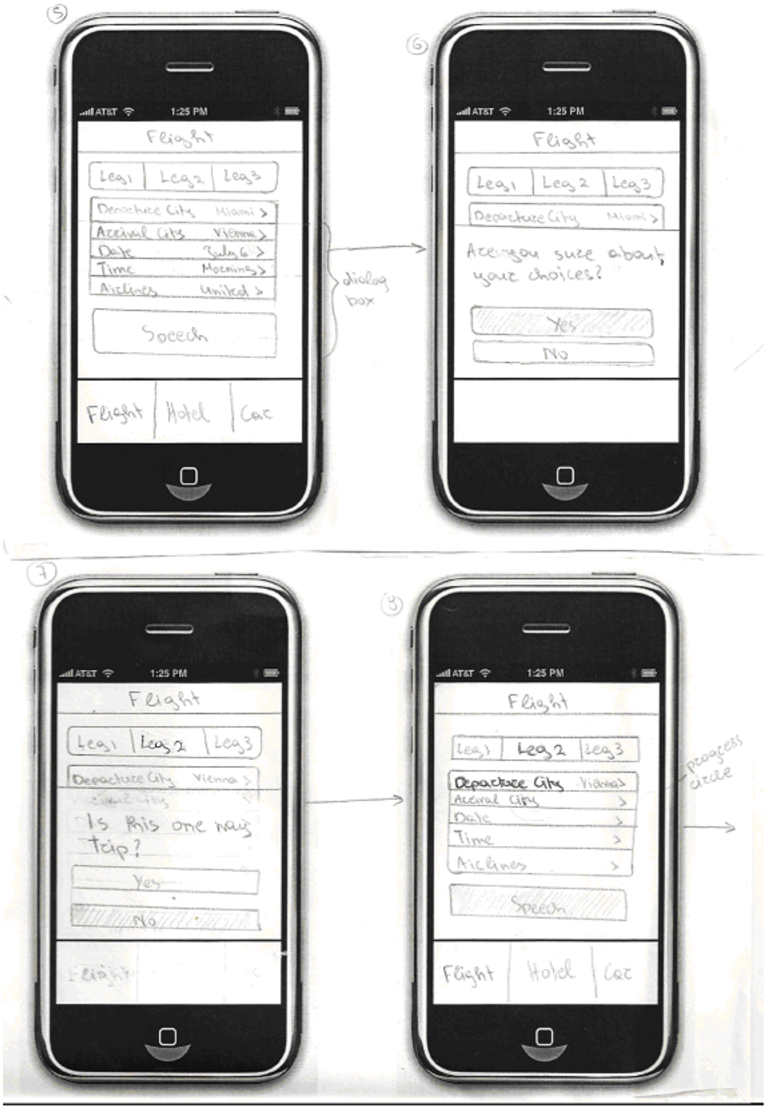
5.3 Conclusions

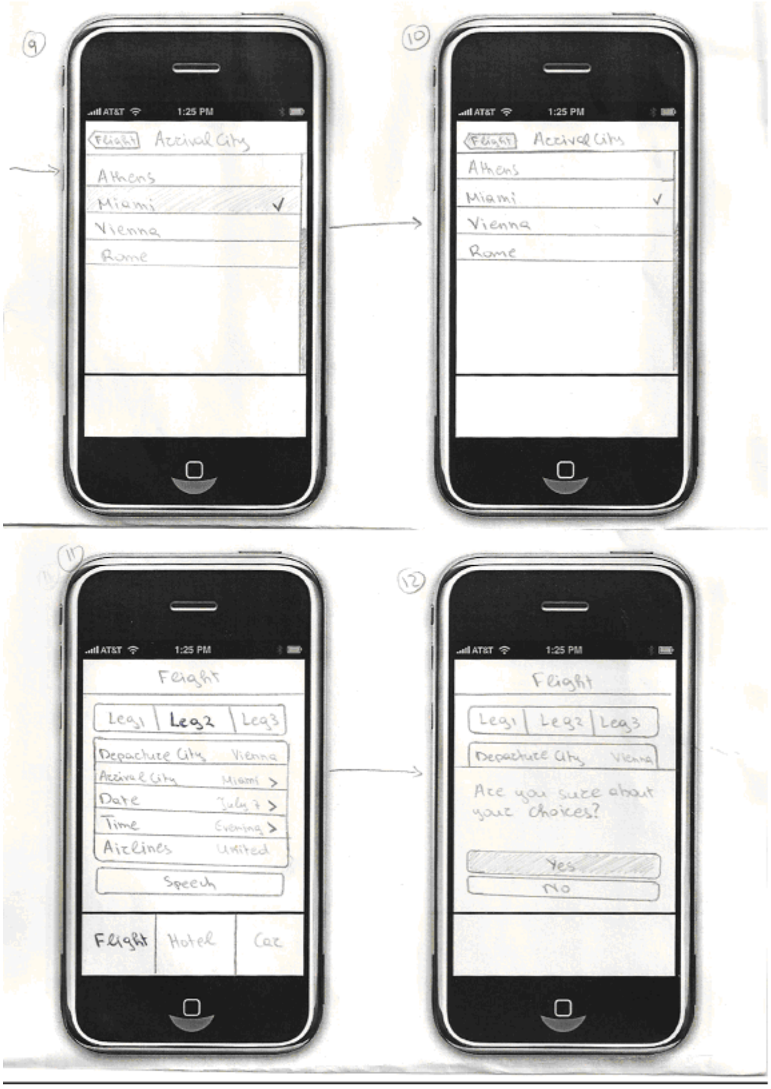
Obviously the results were produced in sort of laboratory conditions, and not in real world conditions. However, we believe that as this work presents a relatively new approach in interface interaction, since the results are positive, it should provide some incentive for further research in this direction.

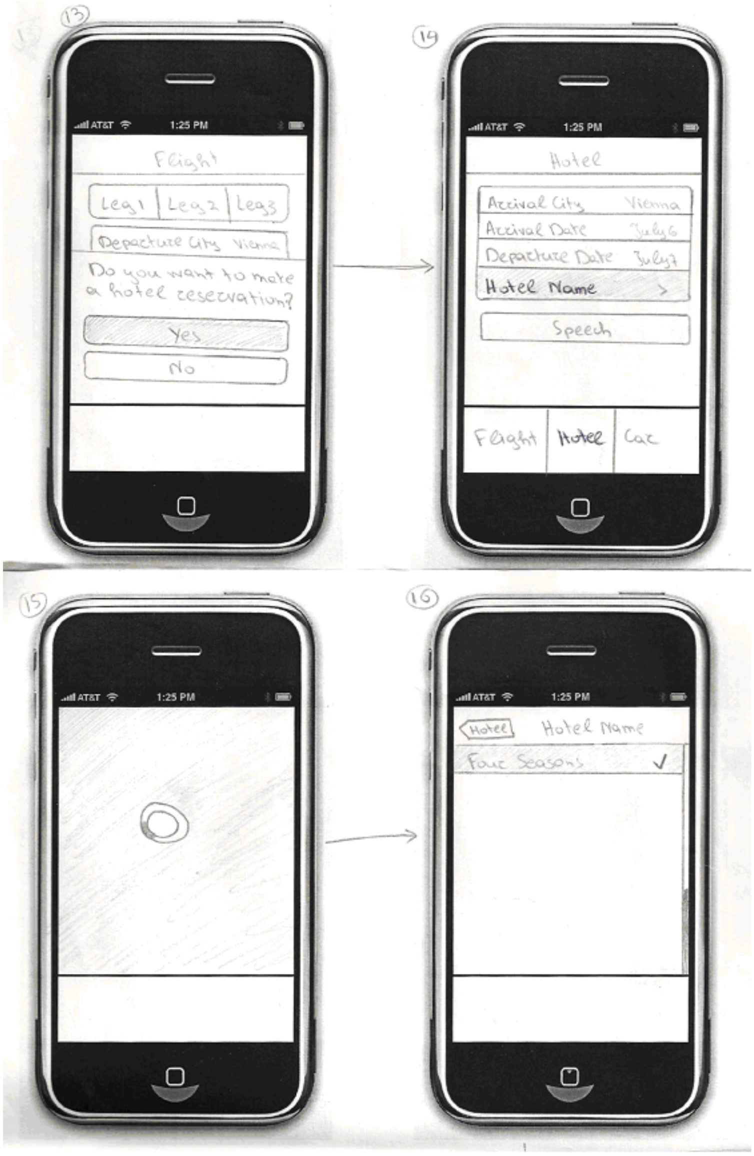
Having achieved a positive result in this research, we have proven that multimodality contributes in velocity, completion time and systems performance, in contrast with the conventional systems. Hopefully this will provide the necessary motive for more research in this area.

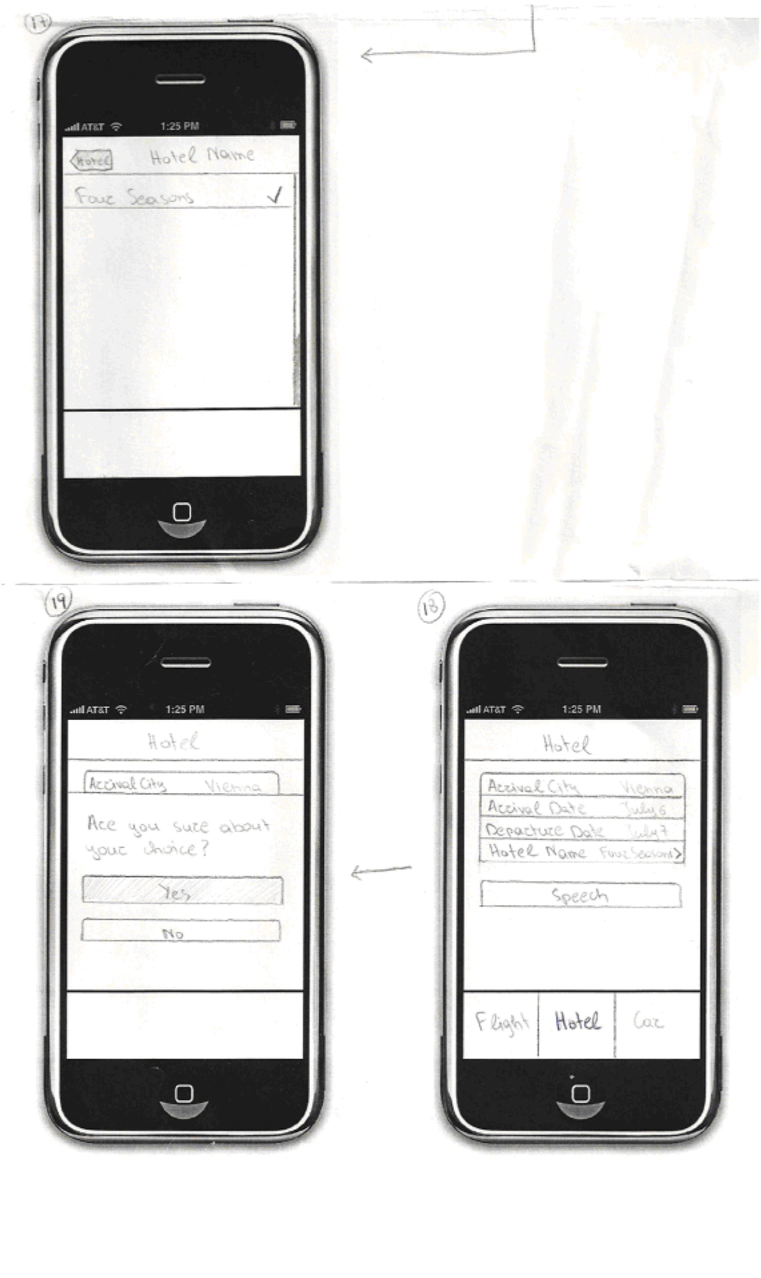
Appendix A

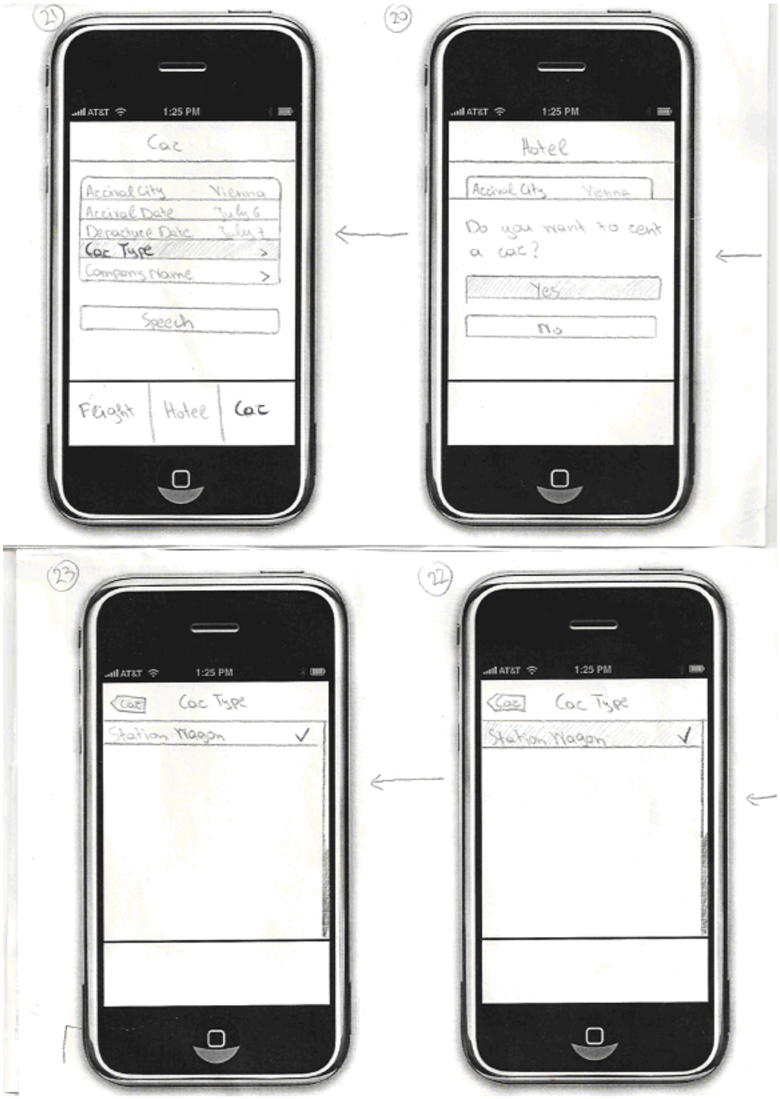
Mockup Design

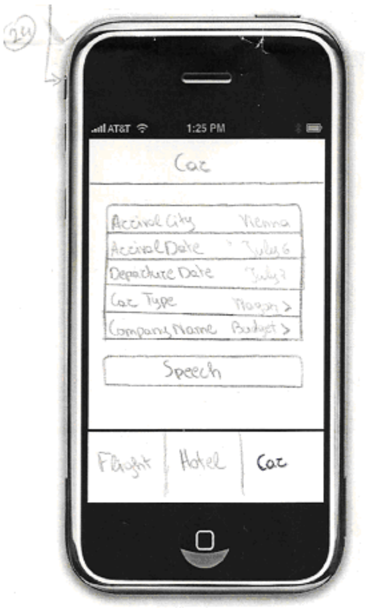

5
AT&T 1:25 PM
Flight
Leg1 | Leg2 | Leg3
Departure City Miami >
Arrival City Vienna >
Date July 6 >
Time Morning >
Airlines United >
Speech
Flight | Hotel | Car
dialog box
6
AT&T 1:25 PM
Flight
Leg1 | Leg 2 | Leg3
Departure City Miami >
Are you sure about your choices?
Yes
No
7
AT&T 1:25 PM
Flight
Leg1 | Leg 2 | Leg3
Departure City Vienna >
Arrival City
Is this one way trip?
Yes
No
Flight
8
AT&T 1:25 PM
Flight
Leg1 | Leg 2 | Leg3
Departure City Vienna >
Arrival City >
Date >
Time >
Airlines >
Speech
Flight | Hotel | Car
progress circle


9
10
11
12
AT&T 1:25 PM
Flight
Arrival City
Athens
Miami ✓
Vienna
Rome
Flight
Leg1
Leg2
Leg3
Departure City Vienna
Arrival City Miami >
Date July 7 >
Time Evening >
Airlines United
Speech
Flight
Hotel
Car
Are you sure about your choices?
Yes
No


13
AT&T 1:25 PM
Flight
Leg1
Leg2
Leg3
Departure City Vienna
Do you want to make a hotel reservation?
Yes
No
14
AT&T 1:25 PM
Hotel
Arrival City Vienna
Arrival Date July 6
Departure Date July 7
Hotel Name >
Speech
Flight
Hotel
Car
15
AT&T 1:25 PM
16
AT&T 1:25 PM
Hotel
Hotel Name
Four Seasons ✓


17
AT&T 1:25 PM
Hotel
Hotel Name
Four Seasons ✓
19
AT&T 1:25 PM
Hotel
Arrival City Vienna
Are you sure about your choice?
Yes
No
18
AT&T 1:25 PM
Hotel
Arrival City Vienna
Arrival Date July 6
Departure Date July 7
Hotel Name Four Seasons >
Speech
Flight
Hotel
Car


21
AT&T 1:25 PM
Car
Arrival City Vienna
Arrival Date July 6
Departure Date July 7
Car Type >
Company Name >
Speech
Flight
Hotel
Car
20
AT&T 1:25 PM
Hotel
Arrival City Vienna
Do you want to rent a car?
Yes
No
23
AT&T 1:25 PM
Car
Car Type
Station Wagon ✓
22
AT&T 1:25 PM
Car
Car Type
Station Wagon ✓



Arrival City	Vienna
Arrival Date	July 6
Departure Date	July 7
Car Type	Wagon >
Company Name	Budget >

Appendix A

Implementation Details

A.1 iPhone

An iPhone touch interface is not a traditional desktop interface, though it has a codebase closely to Mac OS X. Also it is not a traditional mobile interface though iPhone is a mobile device.

The iPhone is an internet - connected, multimedia smart phone designed and marketed by Apple. Because its minimal hardware interface lacks a physical keyboard, the multi touch screen renders a virtual keyboard when necessary. The iPhone"s interface is based around the home screen, a graphical list of all the available applications. Almost all input is given through the touch screen, witch understands complex gestures using multi touch. The iPhone"s interaction techniques enable the user to move the content up or down by a touch - drag motion of the finger. For example scrolling through a long list or menu is achieved by sliding a finger over the display from bottom to top, or vice versa to go back. In this way, the interface simulates the physics of a real object. Other visual effects include the vertically sliding keyboard and bookmarks menu, and widgets that turn around to allow settings to be configured on the other side. Menu bars are found at the top and bottom of the screen when necessary. In menu hierarchies, a "back" button in the top - left corner of the screen displays the name of the parent folder.

The iPhone introduces innovative mobile platforms that are a joy to program. The iPhone runs a first class of OS X with a rich and varied SDK that enables developers to design, implement, and realize wide rage of applications [11].

A.1.1 Xcode Environment

Xcode is the most important tool in the iPhone development arsenal. It provides a comprehensive project development and management environment, complete with source editing, comprehensive documentation, and a graphical debugger. Xcode is built around several open source GNU tools, namely gcc compiler and dgb debugger.

The main application of the suite is the integrated development environment (IDE), also named Xcode. The Xcode suite also includes most of Apple's developer documentation, and Interface Builder, an application used to construct graphical interfaces. Interface Builder provides a rapid prototyping tool that enables developers to lay out user interfaces graphically and link to those prebuilt interfaces from the Xcode source code. With Interface Builder, developers can draw out their interface using visual design tools and then connect those on screen elements to objects and method calls in their application [12, 13].

A.1.2 Simulator

The iPhone Simulator runs on the Macintosh and enables developers to create and test applications on their desktop. They can do this without connecting to an actual iPhone . The Simulator offers the same API used on the iPhone and provides a preview of how the concept designs will look. When working with the Simulator, Xcode compiles Intel x86 code that runs natively on the Macintosh rather than ARM - based code used on the iPhone.

A.1.3 Objective - C

As iPhone code is normally written in Objective-C so our application was implemented by using the same programming language. Objective - C is an object oriented programming language that constitutes superset of ANSI C, and introduces Smaltalk - style messaging in C [14, 15].

References

[1] M.F.McTear, “Spoken Dialogue Technology: Enabling the Conversational User Interface,” *ACM Computing surveys*, vol. 34(1), 2002. xiii, 10, 11, 12, 13

[2] A.Potamianos, E.Fosler - Lussier, E.Ammicht and M.Perakakis, “Information Seeking Spoken Dialogue Systems. Part 2: Multimodal Dialogue ,” *IEEE Transactions on Multimedia*, April 2007. xiii, 2, 16, 17, 19, 21, 26

[3] M.Perakakis and A.Potamianos, “A Study in Efficiency and Modality Usage in Multimodal Form Filling Systems,” *Audio, Speech, and Language Processing, IEEE Transactions*, vol. 16, 2008. xiii, 2, 19, 20, 26, 37, 38, 39, 40

[4] Sharon Oviatt and Philip Cohen, “Multimodal Interfaces that process: What comes Naturally,” vol. 43, 2000. 2, 14, 15, 16, 19

[5] P.Cohen, M.Johnston, D. McGee, S.Oviatt, J.Pittman, I.Smith, L.Chen and J.Clow, “QuickSet: Multimodal Interaction for Distributed Applications,” *In Proceedings of the fifth International Multimedia Conference (Multimedia ’97): Seatle, WA, November*, pp. 31–40, 1997. 2, 16, 19

[6] M.Johnston, S.Bangalore, G.Vasireddy, A.Stent, P.Ehlen, M.Walker, S.Whittaker and P.Mallor, “MATCH: An Architecture for Multimodal Dialogue Systems,” *Proceedings of the 40th Annual Meeting on Association for Computational Linguistis*, 2002. 2, 20

[7] Richard A. Bolt, “”Put-that-there”: Voice and Gesture at the Graphics Interface,” *Computer Graphics*, 1980. 2, 16, 20

[8] M.Perakakis, M. Toutoudakis and A.Potamianos, "Blending Speech and visual input in multimodal dialogue systems," *Spoken Language Technology Workshop, 2006, IEEE*, 2006. 2, 19, 37

[9] C.Benoit,J.C.Martin,C.Pelachaud,L.Schomaker and B.Suhm, "Audio-visual and Multimodal Speech Systems," 2000. 7, 8, 11, 18, 35, 36

[10] J. Johnson, *Gui Blooper Do's and Dont's for software developers and web designers*. 340 Pine Street, Sixth Floor, San Francisco: Morgan Kaufmann Publisers, 2000. 27

[11] Erica Sadun, *The iPhone Developer's Cookbook: Building Applications with the iPhone SDK*. Addison - Wesley, 2008. 70

[12] Apple, "Xcode." http://developer.apple.com/technology/Xcode.html. 70

[13] B.Altenberg, A.Clarke and P.Mougin, *Become an Xcoder*. 2008. 70

[14] Apple, *The Objective-C 2.0 Programming Language*. 2008. 70

[15] Apple, "UI Class API." http://developer.apple.com/iphone/library/navigation/Frameworks/CocoaTouch/UIKit/index.html. 70