Discussion Board System with modality variation: From Multi-modality to User Freedom

June Miyazaki

Tampere University
Computer Science
M.Sc. Programme
July 2002

Tampere University
Department of Computer Science
By: June Miyazaki
Discussion Board System with modality variation: From multi-modality to user freedom
M.Sc. Programme in User Interface Development Thesis, 55 pages, 6 reference pages
July 2002

Abstract

This thesis discuss about modality variability of common resource usability. The purpose of discussion is that gives a user freedom to access resource through different devices and sensory spaces. A user has a right to choose communication modality with the system. The current discussion is about how the users choose their modality situation to situation through socio-social force and design implementation according to proposed model. Speech interface design (SID) was considered as a main user interface to implement an independent-sequential modality in mobile telephony. The discussion board system - voice BBS was based on distribution model, to use VoiceXML technology.

In addition, the thesis examines the usability in terms of Human Computer Interaction with telephony and multi-modality. Overall point is that the users can choose the different modality to access the same resource as independent-sequential modality, rather than combined-parallel modality.

Keywords: speech user interface (SUI), multi-modality, voice mail, mobile telephony, distribution model

Index

1. Introduction

There has been discussion about human-computer interaction through many perspectives in the past, like multi-modality, multimedia, and speech interface. However, all those studies leave out the freedom of the user to choose the way to communicate the system or application at different situation to situation (it includes place, time and circumstances). Some of studies are focused on the natural human interaction, like speech interface.

Speech carries the great possibility as a way for human-computer interaction. Speech is natural; indeed, the huge majority of humans are already fluent in using it for communication. Technology already exists for reliable process and response to basic human speech and it is currently being used as commercial interface applications such as dictation systems (e.g. IBM ViaVoice, NEC SmartVoice) [http://www.amuseplus.com/smartvoice/].

The characteristic of human speech carries many underlying meaning the words such as prosody, a string of words based on their order, spoken contextual and situation. This would be required heavy duty for the system to interpret human semantic communication, yet not so accurate. To overcome those issues, the system defined some of simple structure to process user utterance, such as barge-in or word spotting technique. This way gives the limitations of speech recognition and language processing, the interface should also convey to the user the fact that their conversational system is just a tool to retrieve the information, in order to discourage the user from high expectation for intelligence to the system and exceeding functional capacities. This paper presents the work on designing one of aspects of solving mentioned above problems in speech interface design through simple keyword technique. The goal of the system is that to have the user and the machine compromise their capacities. This study explores situation when the user uses simple command utterance to interact the system rather than un-strict natural dialog (which is heavy duty for the system, but light for the user).

Another part of the discussion is devoted to natural human computer interaction from the viewpoint of multi-modal interaction. Though graphical user interface (GUI) has dramatically improved human-computer communication, it is still required to be trained a user to be familiar with the system. GUI relies on sizeable screen, keyboard, and mouse device. The ambiguity of spoken language and the memory burden of using speech as output modality on the user prevent it becoming the choice of the trend interface. It is considered that the multi-modality is a normal interaction model for human-human communication and it is dramatically enhancing the usability of speech interface system by adding GUI. Unfortunately, GUI

conventions have not transfer into speech step by step. Human being doesn't speak about terminology using the same vocabulary that existing in the graphical interface, even if this application is open on the screen in front of the user [Clark, 1994]. Therefore, an effective speech interface should be designed for real conversation based on natural dialog studies. That leads to human-computer dialog which allows a user to specify information so that a user more likely accept a system that exhibits cooperative behavior.

GUI and speech technology move to ideal human computer interaction. It has investigated a wide range of information services to make generalizations and draw conclusions for develop some voice application guidelines. The desired speech recognition technology might seem impossible to empirically investigate other important aspects of user acceptance and satisfaction. It cannot be wait the technology to catch up. Speed and the amount of time-spent waiting were important to users. They did not want to wait a long time while data was being retrieved.

Delays in voice applications cause more frustration than in GUI applications. If the system provided no feedback telling users that it was retrieving data, whereas with a GUI application, the user may get feedback by looking at the display. Users require constant interaction in voice applications because they have no other method of control or feedback.

With a voice-only interface, exceptionally speech feedback can be used. Different cues can be used to tell a user when they should speak, when the system has or has not understood their requests, when it is fetching information (if long delays), and so on. Earcons and environmental sounds intuitively perceived are an effective way of providing audio feedback to users.

In the previous discussion, how users feel a lack of control when waiting for system responses and the burden that voice output places on cognitive load. The same effect is felt when response granularity, the level of detail in system responses, is coarse. The alternate goal for speech interface is to provide the user as precise a response as possible, giving the feeling of greater control and revealing information progressively in order to reduce the demands on users' memory. The next section discusses background of speech user interfaces (SUI) and speech interface design (SID) through multi-modality perspective. An application of SUI is considered in the context of voice mail system.

2. Speech User Interfaces

Voice mail applications, multi-modal application, multi-media system leaves out simple mono modality. In the past, those studies were carried out for development of speech interface / voice mail technology.

Here we would like to discuss some features, benefits and lacks, of the four-email reading systems, they are: ELVIS (email voice interactive system) [Walker et al., 1998], MiPad [Huang, 2000], SpeechActs [Yankelovich *et al*, 1994] and MailCall [Marx *et al.*, 1996]. However, all of them are prototype applications for research purpose. This section discusses each feature and system.

2.1 MiPad

MiPad is an application prototype of Dr. Who, a research project in the Speech Technology Group of Microsoft Research and Microsoft Speech.Net Group [http://www.research.microsoft.com/srg/drwho.asp]. It offers a conversational, multi-modal interface to Personal Information Manager including calendar, contact-list, and e-mail. The application has a *Tap and Talk* interface that allows users to effectively interact with a PDA device. Dr. Who is a Microsoft's research project aiming at creating a speech-centric multi-modal interaction framework, which serves as the foundation for the .NET natural user interface. MiPad is the application prototype that demonstrates compelling user advantages for wireless Personal Digital Assistant (PDA) devices. MiPad fully integrates continuous speech recognition (CSR) and spoken language understanding (SLU) to enable users to accomplish many common tasks using a multi-modal interface and wireless technologies. It tries to solve the problem of pecking with tiny styluses or typing on minuscule keyboards in today's PDA.

MiPad is unlike a cellular phone, avoids speech-only interaction. It incorporates a built-in microphone that is activated whenever a field is selected. As a user taps the screen or exploits a built-in roller to navigate, the tapping action narrows the number of possible instructions for spoken understanding. MiPad currently runs on a Windows CE Pocket PC with a Windows 2000 machine where speech recognition is performed. The Dr. Who CSR engine uses a unified CFG and n-gram language model. The Dr. Who SLU engine is based on a robust chart parser and a plan-based dialog manager. Spoken language has the potential to provide a consistent and unified interaction model across these three classes, albeit for these different application scenarios, you still need to apply different user interface (UI) design principles. MiPad is one of Dr. Who's applications that address the mobile interaction scenario. It is a

wireless PDA that enables users to accomplish many common tasks using a multi-modal spoken language interface (speech + pen + display) and wireless-data technologies. This section describes MiPad's design, implementation work in progress, and some of preliminary user studies in comparison to the existing pen-based PDA interface that had held at Microsoft Speech.Net Group in 2000. Several functions of MiPad are still in the designing stage, including its hardware design. MiPad tries to solve the problem of pecking with tiny styluses or typing on minuscule keyboards in today's PDAs (personal digital assistance). It also avoids the problem of being a cellular telephone that depends on speech-only interaction. It has a built-in microphone that activates whenever a visual field is selected. MiPad is designed to support a variety of tasks such as E-mail, voice-mail, and Web browsing, cellular phone.

This collection of functions unifies the various devices that people carry around today into a single, comprehensive communication tool. While the entire functionality of MiPad can be accessed by pen alone, it can also be accessed by speech and pen combined. The user can dictate to a field by holding the pen down in it. The pen simultaneously acts to focus where the recognized text goes, and acts as a push-to-talk control. As a user taps the screen or uses a built-in roller to navigate, the tapping action narrows the number of possible instructions for spoken language processing.

MiPad's hardware prototype is based on Compaq's iPaq. It is configured with client-server architecture. The client is based on Microsoft Windows CE that contains only signal processing and UI logic modules. The wireless local area network (LAN), which is currently used to simulate wireless 3G, connects the client to a Windows 2000 Server where CSR and SLU are performed. The bandwidth requirement between the signal the signal processing module and CSR engine is about 2.5-4.8kbps. MiPad applications communicate via our dialog manager to both the CSR and SLU engines for coordinated context-sensitive *Tap and Talk* interaction

The client is based on a Windows CE iPAQ, and the server is based on a Windows 2000 server. The client-server communication is currently based on the wireless LAN.

The present pen-based methods for getting text into a PDA (Graffiti, Jot, soft keyboard) are barriers to broad market acceptance. As an input modality, speech is generally not as precise as mouse or pen to perform position-related operations. Speech interaction can be adversely affected by the ambient noise. When privacy is of concern, speech is also disadvantageous since others can overhear the conversation. Despite these disadvantages, speech communication is not only natural but also provides a powerful complementary modality to

enhance the pen-based interface. Because of these unique features, it needs to leverage the strengths and overcome the technology limitations that are associated with the speech modality. Pen and speech can be complementary and they can be used very effectively for handheld devices. You can tap to activate microphone and select appropriate context for speech recognition. The advantage of pen is typically the weakness of speech and vice versa. This implies that user interface performance and acceptance could increase by combining both. Thus, visible, limited, and simple actions can be enhanced by non-visible, unlimited, and complex actions.

Since a language model for speech is recognition, it can be used the same knowledge source to reduce the error rate of the soft keyboard when it is used instead of speech recognition. It models the position of the stylus tap as a continuous variable, allowing the user to tap either in the intended key, or perhaps nearby in an adjacent key. By combining this position model with a language model, error rates can be reduced. In their preliminary user study, the average user made half as many errors on the fuzzy soft keyboard, and almost all users preferred the fuzzy soft keyboard. It is its ultimate goal to make sure that Dr. Who technologies add value to their customers. It is necessary to have a rigorous evaluation to measure the usability of the MiPad prototype. The major concerns are *"Is the task completion time much better?"* and "*Is it easier to get the job done?"* For their preliminary user study, it is set out to assess the performance of the current version of MiPad (with PIM features only) in terms of task-completion time (for both CSR and SLU), text throughput (CSR only), and user satisfaction. The focal question of this study is whether the *Tap and Talk* user interface can provide added value to the existing PDA user interface. *Is the task completion time much better?* 20 computer-savvy users tested the partially implemented MiPad prototype. These people had no experience with PDAs or speech-recognition software. The tasks they evaluated include creating a new email, checking calendar, and creating a new appointment. Task order was randomized. It alternated tasks for different user groups using either pen-only or *Tap and Talk* interfaces. The text throughput is calculated during e-mail paragraph transcription tasks.

Compared to using the pen-only user interface, it was observed that the *Tap and Talk* interface is about 50% faster transcribing email documents. For the overall command and control operations such as scheduling appointments, the *Tap and Talk* interface is about 33% faster than the existing pen-only interface. Error correction for the *Tap and Talk* interface remains as one of the most unsatisfactory features. In their user study, calendar access time using the *Tap and Talk* methods is about the same as pen-only methods, which suggests

that simple actions are very suitable for pen-based interaction. *Is it easier to get the job done?* Most users it tested stated that they preferred using the *Tap and Talk* interface. The preferences are consistent with the task completion times. Indeed, most users' comments concerning preference were based on ease of use and time to complete the task.

MiPad is a work in progress for it to develop a consistent Dr. Who interaction model and Dr. Who engine technologies for three broad classes of applications. A number of discussed features are yet to be fully implemented and tested. Their currently tested features include PIM functions only. Despite their incomplete implementation, it was observed that speech and pen have the potential to significantly improve user experience in its preliminary user study. Thanks to the multi-modal interaction, MiPad also offers a far more compelling user experience than standard telephony interaction.

The success of MiPad depends on spoken language technology and always-on wireless connection. With upcoming 3G wireless deployments in sight 3, the critical challenge for MiPad remains the accuracy and efficiency of its spoken language systems since likely MiPad may be used in the noise interruption circumstance without using a close-talk microphone, and the server also needs to support a large number of MiPad clients.

2.2 MailCall

MailCall [Marx *et al.*, 1996] is a telephone-based messaging system, which employs speech recognition for input and speech synthesis for output. It was developed on a Sun Sparcstation 20 under both SunOS 4.1.3 and Solaris, using the DAGGER speech recognizer from Texas Instruments and DECtalk for text-to-speech synthesis. Call control is facilitated by XTL, ISDN software from Sun Microsystems.

Unified voice/text message retrieval, MailCall retrieves incoming messages and places them in categories depending on their importance. The user can ask the sender, subject, arrival time, or recipients of any message. Audio attachments are processed and played as sound files, and email notification sent by a homegrown voice mail system acts as a pointer to the original voice message.

Messaging is "unified" in that there is no differentiation by media; the user might have two email messages and one voice message from the same person, and they would be grouped together. *Sending messages,* The user can send a voice message in reply to any message or to anyone in the Rolodex. If the recipient is a local voice mail subscriber, it will be placed in the appropriate mailbox; if not, then it is encoded available formats include Sun, NextMail,

MIME, and uuencode-and sent as electronic mail. (Dictating replies to be sent, as text is not feasible with current speech recognition.)

Voice Dialing, Instead of sending a voice message, the user may elect to place a call instead. If the person's phone number is available in the Rolodex, MailCall uses it-and if there is both a home and work number, MailCall prompts the user to choose one or the other. If someone's phone number cannot be found, the user is prompted to enter it.

2.2.1 User Interface Design

Retrieving messages over the phone is more cumbersome than with a GUI-based mail reader. With a visual interface, the user can immediately see what messages are available and access the desired one directly via point and click. In a non-visual environment, however, a system must list the messages serially, and since speech is serial and slow, care must be taken not to overburden the user with long lists of choices. Organizing the information space by breaking down a long list of messages into several shorter lists is a first step. Once these smaller, more manageable lists are formed, the system must quickly present them so that the user can choose what to read first. And once the user is informed of available options, the system must provide simple, natural methods of picking a particular message out of the list. A first step towards effective message management in a Non-visual environment is prioritizing and categorizing messages. Like many other mail readers, MailCall filters incoming messages based on a user profile, which consists of a set of rules for placing messages into categories. Although rule-based filtering is powerful, writing rules to keep up with dynamic user's interests can require significant effort on the part of the user. Capturing dynamic user interests either by requiring the user to write filtering rules or attempting to infer priorities from past behavior ignores a wealth of information in the user's work environment. The user's calendar, for instance, keeps track of timely appointments, and a record of outgoing email suggests people who might be important. MailCall exploits these various information sources via a background process called CLUES, which scans various databases and automatically generates rules to be used for filtering.

CLUES can detect when someone returns a call by correlating the user's record of outgoing phone calls-created when the user dials using one of a number of desktop dialing utilities-with the Caller ID number of voice mail. Our voice mail system sends the user email with the Caller ID of the incoming message. MailCall's categorization breaks up a long list of messages into several smaller, related lists, one of those being the messages identified as important by CLUES. Once the messages have been sorted into various

categories, the user needs a way to navigate among categories. Although messages may be filtered in order of interest, categories can nonetheless serve as navigational landmarks, which assist in keeping context and returning to already-covered ground. The MailCall user can jump from category to category in nonlinear fashion, saying, “Go to my personal messages” or “go back to my important messages.”

Categorization of messages helps to segment the information space, but when there are many messages within a single category, the user once again is faced with the challenge of finding important messages in a long list. Creating more and more categories merely shifts the burden from navigating among messages to navigating among categories; rather, the user must have an effective method of navigating within a category-or, more generally, of finding one’s way through a large number of messages.

Efficiently summarizing the information space is the second step toward effective non-visual messaging. With a GUI-based mail reader, the user is treated to a visual summary of messages and may point and click on items of interest. This works because a list of the message headers quickly summarizes the set and affords rapid selection of individual messages. These are difficult to achieve aurally, however, due to the slow, non-persistent nature of speech. Whereas the eyes can visually scan a list of several dozen messages in a matter of seconds, the ear may take several minutes to do the same; further, the caller must rely on short-term memory in order to recall the items listed whereas the screen serves as a persistent reminder of one’s choices. Although the latter summary does not list the subject of each message, it is more quickly conveyed and easier to remember. By grouping messages from a single sender, it avoids mentioning each message individually; instead providing a summary of what is available.

In addition, MailCall attempts not to overburden the user with information. When reading the list, for instance, it does not say the exact number of messages but rather a “fuzzy quantification” of the number. Now that the user can hear a summary of available messages, it is practical to support random access to individual messages. Random access refers to the act of nonlinear information access-i.e., something other than the neighboring items in a list. The chart delineates four general modes of random access.

By *location-based* random access it mean that the navigator is picking out a certain item by virtue of its position or placement in a list-i.e., “Read message 10.” Location-based random access may either be *absolute* (as in the preceding example), when the user has a specific message in mind, or *relativ*e, when one moves by a certain offset: e.g., “skip ahead five messages.” (It may be noted

that sequential navigation is a form of relative location-based navigation where the increment is one.) Location-based random access does impose an additional cognitive burden on the user, who must remember the numbering of a certain message in order to access it.

With *content-based* random access the user may reference an item by one of its inherent attributes, be it the sender, subject, date, etc. For instance, the user may say, “Read me the message from John Linn.” Thus the user need not recall the numbering scheme. Like location-based navigation, both relative and absolute modes exist. Relative content-based access associated with following “threads,” multiple messages on the same subject. *Absolute content-based navigation* is the contribution of MailCall, allowing the user to pick the interesting message(s) from an efficient summary without having to remember details of position.

It is practical to support absolute content-based navigation thanks to recent advances in speech recognition. Normally a speech recognizer has a static, precompiled vocabulary, which cannot be changed at runtime. This makes it impractical for the speech recognizer to know about new messages, which arrive constantly. Recently, however, a dynamic vocabulary-updating feature added to the Dagger speech recognizer enables us to add the names at runtime. When the user enters a category, MailCall adds the names of the email senders in that category to the recognizer’s vocabulary. Thus the user may ask for a message from among those listed in a summary. One may also ask if there are messages from anyone listed in the Rolodex, or from whom one has recently sent a message or called (as determined by CLUES). Supporting absolute content-based random access in MailCall with Dagger dynamic vocabulary updating is a positive example of technology influencing design. Absolute content-based random access brings MailCall closer in line with the experience one expects from a graphical mail reader.

MailCall is non-visual interaction approaches the usability of visual systems through a combination of message categorization, presentation, and random access. MailCall monitors conversational context in order to improve feedback, error-correction, and help. Studies suggest that its non-visual approach to handling messages is especially effective when the user has a large number of messages. To evaluate the effectiveness of MailCall, a user study was conducted. The goal was not only to determine how usable the system was for a novice, but also how useful it would prove as a tool for mobile messaging. Since their goal was not only to evaluate ease of learning but likelihood of continued use, it had conducted a long-term user study. The five-week study involved four novice (yet technically savvy) users with varying experience

using speech recognition. In order to gauge the learning curve, minimal instruction was given except upon request. Sessions were not recorded or monitored due to privacy concerns surrounding personal messages, so the results described below are based chiefly on user reports. The experiences of the two system designers using MailCall over a period of three months were also considered.

Feedback from novices centered mainly on the process of learning the system, though as users became more familiar with the system, it also commented on the utility of MailCall's non-visual presentation. Seasoned users offered more comments on navigation as well as the limits of MailCall in various acoustic contexts.

Bootstrapping: As described above, their approach was to provide a conversational interface supported by a help system. All novice users experienced difficulty with recognition errors, but those who used the help facility found it could sustain a conversation in many cases. A participant very familiar with speech systems found the combination of error handling and help especially useful: I have never heard such a robust system before. I like all the help it gives. I said something and it didn't understand, so it gave suggestions on what to say. I really liked this.

Other participants were less enthusiastic, though nearly all reported that their MailCall sessions became more successful with experience.

Navigation, users cited absolute content-based navigation as a highlight of MailCall. One beginning user said, "I like being able to check if there are messages from people in my Rolodex [just by asking]."

For sequential navigation, however, speech was more a bane than a boon. The time necessary to say "next" and then wait for the recognizer to respond can be far greater than just pushing a touch-tone, especially when the recognizer may misunderstand. Indeed, several used touch-tone equivalents for "next" and "previous." And since some participants in the study received few messages, they were content to step through them one by one. These results suggest that MailCall is most useful to people with high message traffic, whereas those with a low volume of messages may be content to simply step through the list with touch-tones, avoiding recognition errors.

2.2.2 Usability study

The results of the user study suggested several areas where MailCall could improve, particularly for novice users. Some changes have already been made, though others will require more significant redesign of the system.

First, more explanation for beginners is required. Supporting conversational prompts with help appears to be a useful method of communicating system capabilities to novices.

The experience with four novice users, however, suggests that its prompts and help were not explicit enough. As a step in iterative design, we lengthened several prompts including those at the beginning of a session and raised the level of detail given during help; a fifth novice user who joined the study after these changes had been made was able to log on, navigate, and send messages on his very first try without major difficulties. This suggests that prompts for beginners should err on the side of lengthy exposition.

Second, more flexible specification of names is necessary. Specifying names continues to be an elusive problem. MailCall should allow the user to refer to someone using as few items as necessary to uniquely specify them. Doing so would involve two additions to MailCall: a "nickname generator" which creates a list of acceptable alternatives for a given name.

Third, it is mode vs. modeless interaction. If MailCall is to be usable in weak acoustic contexts (like the cellular phone) for people with a large Rolodex, its interaction may need to become more modal. It intentionally designed MailCall to be modeless so that users would not have to switch back and forth among applications, but as the number of people in the Rolodex grows, it may become necessary to define a new "rolodex" application.

Telephone-based messaging systems can approach their visual counterparts in usability and usefulness if users can quickly access the messages they want. Through a combination of message organization, presentation, and navigation, MailCall offers interaction more similar to that of a visual messaging system than previously available.

Consideration of context helps to meet user expectations of error-handling and feedback, though beginning users may require more assistance than was anticipated. Results suggest, however, that a large-vocabulary conversational system like MailCall can be both usable and useful for mobile messaging.

2.3 SpeechActs

SpeechActs [Yankelovich *et al*, 1994] is a prototype test-bed for developing spoken natural language applications. In developing SpeechActs, its primary goal was to enable software developers without special expertise in speech or natural language to create effective conversational speech applications-that is, applications with which users can speak naturally, as if they were conversing with a personal assistant.

The SpeechActs applications was wanted to work with one another without requiring that each have specific knowledge of other applications running in the same suite. For example, if someone talks about “Tom Jones” in one application and then mentions “Tom” later in the conversation while in another application, that second application should know that the user means Tom Jones and not some other Tom. A discourse management component is necessary to embody the information that allows such a natural conversational flow. The current suite of SpeechActs telephone-based applications targets business travelers, letting them read electronic mail, look up calendar entries, retrieve stock quotes, set up notifications, hear national weather forecasts, ask for time around the world, and convert currency amounts. The dialogue below captures the flavor of a SpeechActs conversation. In this example, a business traveler has telephoned SpeechActs and entered his name and password.

Because technology changes so rapidly, it also did not recommend tying developers to specific speech recognizers or synthesizers. It was wanted them to be able to use these speech technologies as plug-in components. These constraints-integrated conversational applications, no specialized language expertise, and technology independence-led them to a minimalist, modular approach to grammar development, discourse management, and natural language understanding. This approach contrasts with those taken by other researchers working on spoken-dialogue systems.

It believes it has achieved a degree of conversational naturalness similar to that of the outstanding Air Traffic Information Systems dialogues; they have done so with simpler natural language techniques. At the same time, SpeechActs applications are unique in its level of speech technology independence. Currently, SpeechActs supports a handful of speech recognizers: BBN’s Hark, 4 Texas Instruments’ Dagger, 5 and Nuance Communications’ recognizers 6 (derived from SRI’s Decipher).

These recognizers are all continuous-they accept normally spoken speech with no artificial pauses between words-and speaker-independent-they require no training by individual users. For output, the framework pro-vides text-to-speech support for Centigram’s TruVoice and AT&T’s TrueTalk. The system’s architecture makes it straightforward to add new recognizers and synthesizers to the existing set. Like several other research systems, SpeechActs supports multiple, integrated applications. The framework comprises an audio server, the Swiftus natural language processor, a discourse manager, a text-to-speech manager, and a set of grammar-building tools. These pieces work in conjunction with third party speech components and the

components supplied by the application developer. In this article, it is placed Swiftus, the discourse manager, and the grammar tools in context.

The audio server presents raw, digitized audio (via a telephone or microphone) to a speech recognizer. When the speech recognizer decides that the user has completed an utterance, it sends a list of recognized words to Swiftus.

The speech recognizer recognizes only those words contained in the relevant lexicon-a specialized database of annotated vocabulary words.

Swiftus parses the word list, using a grammar written by the developer, to produce a set of feature-value pairs. These pairs encode the semantic content of the utterance that is relevant to the underlying application.

Developers had carried out the usability test. It had been done by formative evaluation study design.

There had been fourteen users participating in the study. The first two participants were pilot subjects. After the first pilot, they redesigned the study, solved major usability problems, and fixed software bugs. After the pilots, nine users, all from their target population of traveling professionals, were divided into three groups of three. Each group had two males and one female. An additional three participants were, unconventionally, members of the software development team. They served as a control group. As expert SpeechActs users, the developers provided a means of factoring *out* the interface in order to evaluate the performance of the speech recognizer.

After testing each group of target users, they altered the interface and used the next group to validate their changes.

Some major design changes were postponed until the end of the study. These will be tested in the next phase of the project when they plan to conduct a longer-term field study to measure the usefulness of SpeechActs as users adapt to it over time. During the study, each participant was led into a room fashioned like a hotel room and seated at a table with a telephone.

They were asked to complete a set of 22 tasks, taking approximately 20 minutes, and then participate in a follow-up interview. The tasks were designed to help evaluate each of the four SpeechActs applications, as well as their interoperation, in a real-life situation. To complete the tasks, participants had to read and reply to electronic mail, check calendar entries for themselves and others, look up a stock quote, and retrieve a weather forecast.

Instead of giving explicit directions, it embedded the tasks in the mail messages. Thus the single, simple directive “answer all new messages that require a response” led to the participants executing most of the tasks desired. For example, one of the messages read as follows: “I understand you have

access to weather information around the country. If it's not too much trouble, could you tell me how warm it is going to be in Pittsburgh tomorrow?" The participant had to switch from the mail application to the weather application, retrieve the forecast, return to the mail application, and prepare a reply.

Although the instructions for completing the task were brief, participants were provided with a "quick reference card" with sample commands. For example, under the heading "Mail" was phrases such as "read me the first message," "let me hear it," "next message," "skip that one," "scan the headers," and "go to message seven." In addition, keypad commands were listed for stopping speech synthesizer output and turning the recognizer on and off.

In the study, their main aim was not to collect quantitative data; however, the statistics they gathered did suggest several trends. As hoped, they noticed a marked, consistent decrease in both the number of utterances and the amount of time required to complete the tasks from one design cycle to the next, suggesting that the redesigns had some effect. On average, the first group of users took 74 utterances and 18.5 minutes to complete the tasks compared to the third group, which took only 62 utterances and 15 minutes (Table 1).

Participants	**Utterances**	**Time (minutes)**
Group1	74	18.67
Group2	63	16.33
Group3	62	15.00
Developers	43	12.33

Table 1. Average number of utterances and time to complete tasks.

(http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/ny_bdy.htm)

At the start of the SpeechActs project, they were aware that the state of the art in speech recognition technology was not adequate for the conversational applications they were building.

One of their research questions was to determine if certain types of interface design strategies might increase users' success with the recognizer. Unfortunately, none of their redesigns seemed to have an impact on recognition rates-the number of utterances that resulted in the system performing the correct action. They remained consistent among the groups, with the developers showing about a 10% better rate than the first-time users. More significant than the design was the individual; for instance, female participants, on average, had only 52% of their utterances interpreted correctly compared to 68.5% for males. Even with these low recognition rates, the

participants were able to complete most of the 22 tasks. Males averaged 20 completed tasks compared to 17 for females (Table 2).

Participants	Recognition Rates	Tasks Completed
Female	52%	17
Male	68.5%	20
Developers	75.3%	22

Table 2. Average recognition rates and number of tasks completed.

[http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/ny_bdy.htm]

They found that recognition rates were a poor indicator of satisfaction. Some of the participants with the highest error rates gave the most glowing reviews. It is their conclusion that error rates correlate only loosely with satisfaction.

Users bring many and varying expectations to a conversation and their satisfaction will depend on how well the system fulfills those expectations.

In addition, expectations other than recognition performance colored users' opinions. Some participants were expert at using Sun's voice mail system with its touchtone sequences that can be rapidly issued. These users were quick to point out the slow pace of SpeechActs; almost without exception they pointed out that a short sequence of key presses could execute a command that took several seconds or longer with SpeechActs.

Overall, participants liked the concept behind SpeechActs and eagerly awaited improvements. Barriers still remain, however, before a system like SpeechActs can be made widely available. They have concluded that adhering to the principles of conversation does, in fact, make for a more usable interface.

2.4 ELVIS

ELVIS [Walker et al., 1998] is spoken dialogue system that allows an access to email by talking to an agent named "Elvis" (EmaiL Voice Interactive System). ELVIS was developed through a combination of empirical studies and automatic optimization techniques such as reinforcement learning and performance modeling. These systems were built using a general-purpose platform developed at AT&T, combining a speaker-independent hidden Markov model speech recognizer, a text-to-speech synthesizer, a telephone interface, and modules for specifying data-access functions and dialogue strategies. It has been used for experiments on automatic adaptation in dialogue, using both reinforcement learning and automatic identification of problematic situations in dialogue. ELVIS was also a vehicle for the

development of the PARADISE evaluation framework, and for developing predictive models of user satisfaction. It has also been used to compare dialogue strategies for mixed-initiative vs. system-initiative dialogue, and for evaluating the effectiveness of tutorial dialogues. In ELVIS, it is the electronic mail spool of the user.

In order to determine the basic application requirements for email access by telephone, it conducted a Wizard of Oz study. The Wizard simulated an email agent interacting with six users who were instructed to access their email over the phone at least twice over a four-hour period. In order to acquire a basic task model for email access over the phone, the Wizard was not restricted in any way, and users were free to use any strategy to access their mail. The study resulted in 15 dialogs, consisting of approximately 1200 utterances, which were transcribed and analyzed for key email access functions.

The email access functions was ranged into general categories based on the underlying application, as well as language-based requirements, such as the ability to use referring expressions to refer to messages in context (as *them, it, tha*t), or by their properties such as the sender or the subject of the message [Walker et al., 1998].

From this exploratory study it concluded that the email agent should minimally support: (1) reading the body of a message and the header information; (2) summarization of the contents of an email folder by content-related attributes, like sender or subject; (3) access to individual messages by content fields such as sender and subject; (4) requests for cancellation and repetition by the user and for clarifying help from the system [Walker et al., 1998].

It implemented both the system-initiative and the mixed initiative versions of the email agent within a general-purpose platform for voice dialog agents, which combines ASR, text-to-speech (TTS), a phone interface, an email access application module, and modules for specifying the dialog manager and the application grammars. The email application demands several advanced capabilities from these component technologies. First, ASR must support barge-in, so that the user can interrupt the agent when it is reading a long email message. Second, the agent must use TTS due to the dynamic and unpredictable nature of email messages; prerecorded prompts are not sufficient for email access. Third, the grammar module must support dynamic grammar loading because the ASR vocabulary must change to support selection of email messages by content fields such as sender and subject.

There is a report [Walker et al., 1997] that it describes experimental results comparing a mixed-initiative to a system-initiative dialog strategy in the context of a personal voice email agent.

It presents the results of an experiment in which users perform a series of tasks by interacting with an email agent using one of the dialog strategies. It also describes how its experimental results can be framed in the PARADISE [Walker et al., 1997] framework for evaluating dialog agents. The goal was to compare performance differences between the mixed-initiative strategy and the system-initiative strategy, when the task is held constant, over a sequence of three equivalent tasks in which the users might be expected to learn and adapt to the system. The mixed-initiative strategy might result in lower ASR performance, which could potentially reduce the benefits of user initiative.

In addition, its assumed that users might have more trouble knowing what they could say to the mixed-initiative agent, but that they would improve their knowledge over the sequence of tasks. Thus, the system-initiative agent might be superior for the first task, but that the mixed initiative agent would have better performance by the third task.

The experimental design [Walker et al., 1997] consisted of three factors: strategy, task, and subject. Effects that are significant as a function of strategy indicate differences between the two strategies. Effects that are significant as a function of task are potential indicators of learning. Effects that are significant by subject may indicate problems individual subjects may have with the system, or may reflect differences in subjects' attitude to the use of spoken dialog interfaces. They discuss each of these factors in turn.

For example, the most commonly played prompt for MI was "*You can access messages using values from the sender or the subject field." If you need to know a list of senders or subjects, say 'List senders', or 'List subjects'. If you want to exit the current folder, say 'I'm done here'. [Walker et al., 1997]*

In terms of user satisfaction measures, there were no differences in the Task Ease measure as a function of strategy; users did not think it was easier to find relevant messages using the SI agent than the MI agent, even on the first day.

Users' perceptions of whether Elvis is sluggish to respond (System Response) also did not vary as a function of strategy, probably because the response delays were due to the application module, which is identical for both strategies [Walker et al., 1998].

2.5 A pass to the future Speech Interface Design (SID)

Though all of four applications are not used in practical, they give some of the perspective for speech application development in the future. MiPad is the most user and practical use focused project among them. Because other three are emphasized research purpose in order to improve speech application development. However, those studies show us that speech application has a lot of obstacle to over come both hardware and interface design technology in order to accept by users. According to their studies we could say that hardware development and user interface design/system design come close and work together to overall speech application performance. Though it has not discussed about error handling in this paper, this area needs to work with hardware side (e.g. recognition error, etc.), especially. Based on previous investigations, I have come up to my new multi-modal interaction system idea that will be presented in the next chapter.

3. Motivation and design goal

My system design was tried to focuses on the usability and usage of new computer technology such as interactive systems that support the combination different input media such as voice, gesture and video in the first place. However, I decide to change to focus on independent-sequential input modality to deal with SID (speech interface design) to keep system design simple and realistic implementation. Coming to that point, there is the reason to discuss to combined -parallel multi-modal interaction design was not suitable to my project. Though there is a high potential for systems allowing the use of combined input and output media but our knowledge for designing, building, and evaluating such systems is still primitive. My primary goal is to clarify and structure such knowledge from the system perspective.

3.1 Multimodality interface

The multi-modality in interface design has been quested many HCI (Human Computer Interaction) researchers. Using two channel gesture and speech are commonly integrated [Bos et at., 1994] as well as gesture and gaze [Koons, 1993]. These modes reflect the natural multi-modality of human communication (visual /auditive) and (visual/visual) . In contrast, Buxton and colleagues have focused on multi-modality utilizing both hands as input [Buxton and Myers, 1986].

Nigay and Coutaz [Nigay and Coutaz, 1993] have described a design space model for multi-modality characterized by three dimensions. Two of these dimensions are 1) the presence or lack of fusion between modalities (combined or independent) and 2) the temporal use of modalities (sequential or parallel). Combinations of those two dimensions provide a useful framework for characterizing four styles of multi-modal interaction: alternate (combined / sequential), synergistic (combined / parallel), exclusive (independent / sequential), concurrent (independent / parallel).

The most of previous works have been categorized either combined-sequential or combined-parallel multi-modality that have been assumed the system is stable setting environment (e.g. desktop) or specific device (e.g. PDA). The Mipad is a PDA in terms of device and combined-parallel multi-modality.

The project goal includes that the task must be completed in mono-modality in order to meet the one of project goal that it provides the modality choice of freedom to the user seamlessly between device and modality. If the environmental setting is fixed, it is difficult to use the system at any situation.

3.1.1 Multi-modal design in HCI

According to definition in physiology of senses, there are five categories of modalities: visual (eyes), auditive (ears), tactile (skin), olfactory (nose), gustatory (tongue) and vestibular (organ of equilibrium) [Silbernagel, 1979]. However, thee perception-channels, visual, auditive and tactile have been most popular modality among in these days' systems. Those are defined as follows: visual: concerned with, used in seeing (comp. against optical), auditive: related to the sense of hearing (comp. against acoustical), tactile: experienced by the sense of touch [Charwat, 1992]. In terms of SDI, sense of hearing is most concern. Whenever more than two of these modalities are involved, it means multimodality. In this sense, every human-computer interaction has to be considered as multimodal. Because the user looks at the monitor, types in some commands or moves the mouse (or some other device) and clicks at certain positions, hears the reaction (beeps, key clicks, etc.) and so on. Therefore, in our understanding of multimodality is restricted to those interactions which comprise more than one modality on either the input (i.e., perception) or the output (i.e., control) side of the loop and the use of more than one device on either side. Thus, the combination of, e.g., visual, auditive, and tactile feedback which is experienced by typing on a keyboard is explicitly excluded. The combination of visual and auditive output produced by the monitor and a loudspeaker when an error occurred is a 'real' multimodal event. In this sense, speech interface itself is multimodal interaction model because speech would be the result of either input or outcome of user interaction.

To deal with today's computer system users' demand for interfaces that are easy to use and learn. The research in intelligent human-machine interfaces has become more important in the last few years. Therefore, to bridge the gap between the user and the machine through a mediating system which translates the user's input into commands for the machine and vice versa should be important in system design. McNeill [McNeil, 1992] proposes the concept of "growth points" that represent the semantic content of an utterance from which gestures and speech develop in close relation. He suggests a temporal displacement of approximately one or two seconds between two successive semantically units. Similarly, Ballard [Ballard, 1997] presents an organization of human computation into temporal bands of 10 seconds for complex tasks, 2 seconds for simple tasks, 300 ms for physical acts, etc. Different tasks and acts - like moving the eyes or saying a sentence: show a tightly constrained execution time.

On the other hand, the speech is depending on the user's pacing of recording speed and is not controllable by the listener. Therefore, "…the listener cannot scan or skip sections of the recording in the same manner as visually scanning printed text, nor can the listener slow down difficult-to-understand portions of the recording" [Portnoff, 1978].

3.2 Speech in Multi-modality

The new network communication technology creates new communication styles including multi-hypermedia like video conferencing, video broadcast, audio broadcast (radio), web-cast, telephone, telephone conferencing, web pages, chat, email, bulletin board and newsgroups, and cost performance based on bandwidth: the greater the bandwidth the greater the cost [Edwards *et al*, 2001]. However, it doesn't have specific correlation between effectiveness and cost.

The orthodox way of communication style is categorized two extremes that have occupied both axis and none: meet a person face to face and snail mail. Compare with GUI (Graphical user interface) application, the SUI (Speech user interface) application deal with temporal interaction factors to design a system works well.

3.2.1 Mobile phone as handy device

User mobility and ubiquity are two key features for the information technology infrastructure. The needs for mobile phone in the social network increase where people cultivate the relationships outside the physical place and time. [Brown *et al.,* 2001]. The family who lives in a same house but has

different every day life could be the classical user. For example, the parents work day time and high school kids go out at evenings. Since they have different time slot activities, it is difficult to inform and exchange their daily schedule to update on every day bases. The mobile phone network helps to make up this kind of communication barrier in a casual manner by just dialing up at almost any time and place. It enhances the element of social activity and connection of social relationships.

3.2.2 Virtual Present

The objects in the universe interacts each other based on time and place axis. It means that people are engaged in boundary dedication in both time and physical attendance in order to meet. However, the new network communication technology tools like email, SMS (short message service), and cellular phone make possible seamless boundaries in terms of time and place.

Those technologies create new phenomenon "virtual present," the separation of time and place to allow people and engage in the social interaction behavior model. Therefore the people's interaction doesn't necessary to correspond with physical appearance of time and place. As long as the core purpose of activity has been set, spatiotemporal factor of time and place are able to handle by network. For example, calling in the friend gathering covers place barrier issues by focus on time axis. Synchronization of time axis of the event (friend gathering) allow the user physical absence of the event. The user is able to participate the activity physically remote. The virtual presence, that physical appearance can be different from the event of the physical place, generates new interaction style that frees people from imposingly keeping to schedules that are necessary in the regulation of society. This relationship can be described as two new terminologies like *immediate* and *deferred*. Immediate, means that a user is interacting with a server or another user with certain maximum delay bound. Deferred means that a user is interacting with another user or a user is interacting with server without maximum delay requirements.

Although these fundamental behavioral conflicts affect users and nonusers alike, shows a difference in their degree of tolerance of public mobile-phone use. Attitudes are tempered by firsthand experience with the technology. In the Palen's study, the new users found that attitudes about public mobile phone use change dramatically from strong disdain to a much higher degree of acceptance [Palen, 2000].

3.3 Type of multi-modal designs

All previous discussion leads to an idea that satisfies both limitations of current technology and user needs. It fulfills the situational usability. The current systems have lack of modality selection. It gives an abstraction, parallelism and fusion in the design space [Nigay *et al.*, 1993]. For example, the MiPad application is a synergistic multi-modal system, combines graphical user interfaces (GUI) and pen input.

Focusing on graphics sub-tasks and categorization according to the sub-tasks a device make possible to perform at a higher level of abstraction. [Foley *et al.*, 1984].

3.3.1 Design controls

The design space of the system is located at this higher level of abstraction; it deals with tasks at the granularity of commands. It address the issues of how command is specified using the different available modalities and how a command is built from raw data. The data is received from a particular device may be processed at multiple levels of abstraction. For instance, speech input may be recorded as a trigger, or described as a sequence of phonemes, or interpreted as a meaningful parsed sentence. In the output, data may be produced from symbolic abstract data or from a lower level of abstraction without any computational detection of meaning. For example, a voice message may be synthesized from an abstract representation of meaning, from pre-stored text or may simply be replayed from a previous recording. The important point of the system design is that data is represented and processed at multiple levels of abstraction. This process makes possible the extraction of meaning from symbolic abstract representations. In order to simplify the presentation, it is considered only two values along the axis of abstraction: “Meaning” and “No meaning.” A multi-modal system belongs to “Meaning” [Nigay *et al.*, 1993].

As proposed by Nigay, “Use of modalities” indicates the temporal availability of multi-modalities. It matters the absence or presence of parallelism at the user interface. Absence of parallelism is referred to as “Sequential use.” “Parallel use” is the presence of parallelism. The characteristic of sequential use allows the users to use the modalities on after another whereas “Parallel use,” allows the users to engage in multiple modalities simultaneously.

3.3.2 Fusion in design

According to Nigay [Nigay *et al.*, 1993], the fusion in the modality deal with the possible combination of the different data. A data type is related to a

particular modality. The absence of fusion is called as “Independent”. The presence of fusion is called as “Combined”.

The fusion might be performed with or without knowledge about the meaning of the data exchanged. For instance, synchronization of audio and email text data as supported in the MiPad platform, is a temporal fusion, which does not involve any knowledge of meaning. The MiPad [*Huang et al.*, 2000] platform incorporates a built-in microphone that activates whenever a field is selected. As a user taps the screen or uses a built-in roller to navigate, the tapping action narrows the number of possible instructions for spoken language processing. It is based on the concepts of strands, which correspond to audio, or text data, of ropes, which are combinations of strands, and a logical time system that allows several strands and ropes to be played synchronously. This example of fusion results in an interpretation at a high level of abstraction in terms of the task domain by meaning of mixed modalities to build on input or output operation.

3.3.3 Design and human interaction factors

Since the structure of the system behaviour is predefined by the designer, it is able to handle concatenated speech dialog through predefined vocabulary and conversation pattern. The unexpected data like text data input is able to handle through formant speech rather by reading out than dialog. For the purpose of initial welcome message, it provides the security of the how the system is able to handle the user interaction. If the user feels comfortable to the welcome message, the user develops the perception of the system. If this interaction is placed to the Psychologist Piaget’s cognitive development of the sensorimoter period how the human develop the perception to the world in order to be acquired the language in terms of breaking down the human interaction during first contact of the world, infant competency, there are some consideration point that can adapt and modify for human-computer interaction. Piaget described the six stages of the cognitive development during the sensorimoteor period [Piaget’s, 1967]. This period is important for the language development.

The stages one is that do little more than exercise the reflexes with they were born. To notice that the born is replaced as first contact of the human and the system whenever we discussed about Piaget’s cognitive development theory in this paper same as baby. During this period the user develop a foundation of the cognitive structures through the activities. For example, the user tries out many commands or utterances to see how the system understands and reacts his / her speech without logical order and thinking.

The stage two is that is first habits like appearance of the *primary circular reactions.* The user repeats some body action. For example, the users throw their arms when the system did not behave what they expected. It seems to be no meaning to show body language to the system but the user are learning something about that primary object in his / her world of the how to communicate with the system.

The third stage is that emerges the *secondary circular reactions* which direct their activities toward objects and events outside themselves. It produces the result in the circumstance with the user's own body. For example, if the user finds out the preferable outcome from the system by certain utterance, the user repeats that speech or words to the system.

The fourth stage is that to form new behavior throughout to *coordinate secondary schemes.* The users try to move the objects to accomplish the goal. For example, the user set the goal of the outcome of the system interaction and acts in logical way to gain the preferable outcome through intentional behavior. It is rather purpose oriented behavior than disorganized interaction.

The fifth stage is that the appearance of *tertiary circular reactions* that is repetition with variation behavior to provoke new results. It produces novelty interest and curiosity by continuous growth and changing cognitive processes. For example, if the user found some words or speech command that bring the expected result, the user tries out different utterance to gain the same result.

The sixth stage is that the developing the *internal representation*. It is a kind of the primitive representation appearance. According to Piaget's observation, one day his one years old daughter approached a door that she wished to close. But she was carrying some grass in each hand. She put down the grass on the floor in order to close the door. However, she realized that if she closed the door the grass would blow away then she moved the grass away from the door's movement and closed it. That is a good explanation of have a plan before acting.

All those six processes are considered for human-computer interaction and it would better to apply for SUI design.

3.4 Feedback in SUI

It is common to use technical terminology in special field to aid communication. For example, psychology use "superego" as technical terminology in describing one of human's thought structure of layer, the psyche structure out of three according to Freud [Freud, 1955], "unconscious desire of the human" comparison to "conscious desire of the human", not "extremely selfish." In the computer science has been applied similar things far away from natural language interaction. It is used to print "Del" on the

keyboard as abbreviation of “delete”, not “deliver”. If the user knows how to communicate to the PC application in sufficiently, the user use “Del” key as deleting the message. But if the user is a novice to the computer, there is in danger of using “Del” key as delivering (sending) the message.

Talking about the benefits of speech user interface, speech is the most natural way to communicate to human-human interaction. The human – computer interaction require the speakable language in terms of human behaviour. The commands for PC have been developed as assembler language like command line prompt. The main idea is that train the human to the system understandable language rather than training the system to understand the human. Here come to the point, here is necessity of some degree of compromise between the system oriented approach and human behaviour oriented approach. The relationship of those approaches is exactly inverse proportion between the system and the user. The more the system demands the user to learn how to use the system, the more the user has to learn artificial language rather than natural language. Vice versa the more the user demand the system to understand the natural language, the more the system hardware needs to invent higher technology in order to catch up the user’s high expectation. In order to covey the benefit of both sides, middle approach is the suitable solution.

3.5 SUI design

Speech recognition and synthesis provide an important user segment that will benefit both user and application developers. The speech interface has been used primarily to augment applications with an existing visual interface (e.g., VoiceNotes [Wtifelman *et al.*, 1993]). There are a couple of reasons why a SUI to a system might be desirable. First, the application might require a non-visual channel mode interaction that is free from compulsive engagement within computer screen. Second, telephone service is one of the few truly robust and ubiquitous network technologies, so it makes sense to extend information services away from the desktop by providing a telephone interface. They are developing an experimental system, which like MailCall [Marx *et al.*, 1996] and SpeechActs [Yankelovich *et al.*, 1995] provides speech-only access to desktop and network-based information services. Since the speech is the one of the main factors of the human behaviour interaction channel, there is the necessary to talk about human communication factors to clarify the benefit of speech interface.

3.5.1 Phenomenon in SUI

As long as we live on the earth, we do communicate someone to alive. The communication caused by response perception one to another. Almost every guideline for speech interface mention that the importance of feedback factor in regard of interaction input and outcome. It deal with the behavioural considerations that refer to a user's intentions and the system interpretation regarding what the user wants the system to do with the information contained in the message. The user expects the system to provide a direct response that are usually an outcome. For example, the user expect to send email, the user say "Send message" as direct explicit intention. But there is implicit additional information to relate the message "now". Hence the user has an assumption that the system interpret the prompt in the message is new to the system. The assumption is one of the main elements of the human behavioural interaction provides to play the role of speaker and listener to both the system and the user vice versa. The speaker conducts the communication by expectation of the utterance to the listener's response. The human-computer interaction is taken control of the speaker's utterance that expect the listener's action in regards of the hearing the speaker's prompt.

3.5.2 Assumption in the SUI

The word assumption carries out two types of perception, assertion and supposition in terms of linguistics. The assertion means that assumption of the speaker's information is new to the listener or justifies emphasis. The supposition means that assumption of the speaker's information is past of the listener's prior knowledge of the world [Cole, 1980].

Therefore, the assumption factors in the speech user interface imply mutual response time in feedback between computers and users.

Assertion and supposition affects the time it takes a listener to comprehend a sentence and react to the information. It creates the more sufficient and effective interaction.

In terms of human behaviour in the human-computer interaction, there is expectation based on communication. Since the human dialogue requires the flexibility and complexity in the interacting of sufficient conversation. In a way, the computer system has set the standard in how the user views the advent and progress of speech recognition and synthesizes technology. SUI of the computer system could hear the user and understand the user' commands based on these technologies. When the speaker takes control of the dialog, the listener provides that can be the both hearing state and reaction according to speaker's utterance.

Feedback is the one of general principle of user interface guideline that has been developed by over decades of research. It handles the direct control over an application's action in terms of interaction medium between the system and the user.

It is tried to apply those general principles through working within many important well-know voice-environment lessons that hope to help further discussion Voice BBS development.

The Observation of reduced vocabulary is almost same result in WOZ [Wizard of Oz] usability tests [Fraser *et al.*, 91]

This is one of cognitive issues of user perception and communication. If users believe the system has a small working vocabulary, they will interact using a small vocabulary, to the speech recognition system's benefit. However, speech recognition is made more difficult if users believe the system has a large working vocabulary because they will interact using a large vocabulary.

3.5.3 The matter in SUI

When it comes to the workable system in speech interface, it is the question of the effectiveness of the system: is this working or not? If we think about working the system, the complex function should be taken away as long as it doesn't influence the accomplishment of the system purpose. The system is able to add on fancy functions as long as the system is built on in a way of flexible structure. The core system structure should take into account of main gaol of the system itself, rather than the fancy function that comes later as decoration of the system. The capability of the spoken language enabling adds the naturalness and efficiency to human-computer interaction in terms of the part of immersive multimodal interaction. In the past, there are many speech interface application had been implemented by using that potential technology e.g. speechAct [Yankelovich *et al*, 1994]. It is not an independent-sequential multi-modal system.

3.6 Technology in Speech application

The speech recognition and synthesizer are core technology of speech user interface. It is used to generate the process of the speech input and producing the words spoken in this paper. In the SUI, it deals with the interface design rather than larger spoken language understanding system of speech technology. The speech recognition system use computer hardware to convert speech input into digital data which can then be analysed using the Digital Signal Processing (DSP) The DSP output is used to choose the best word match among an application's vocabulary. Speech recognition system focuses on the main obstacle of automated human speech. In this sense the

recognition system distinct and acknowledges who and what by extracting from speech signal alone. This input can be extremely variable from speaker to speaker, even when saying the same words. Because the methods is used to articulate speech, the types of sounds made while speaking and the ways used to process the acoustic signal.

3.6.1 Type of speech recognition

There are two major factors in speech recognition systems that categorize four general areas. The one is the recognition training and speech flow control. The speech recognition package can either be speaker dependent or independent. The speaker dependent system needs to be trained to a specific person's speech patterns and characteristics. The voice system can then either accept continuous speech, or it must work with discrete speech where the speaker needs to insert a small pause in between each word. But there are other important factors which come into play for an interface designer. These include the size of the vocabulary a speech system can choose from at any time, the acceptable error rate, and the ratio of processing time to speech rate, etc. These factors must be considered in order to create a workable application and have a more casual dialogue e with the speech system. The speaker independent systems are usually command and control system that requires quick response and very low rate of error.

3.6.2 Natural language processing

One of the aspect of the speech user interface is the natural language processing that is closely related speech recognition. The natural language processing refers that spoken and written language in based on a given culture for using speech input and output rather than technical terminology in terms of speech technology. For the human-computer interaction point of view, SUI field are interested in the practical adaptation. The usage of language controlling the computer in terms of every day manner without special terminology is most concern. It includes the environmental factors that is not required the special circumstance e.g. noise free setting, special device, etc.

The speech interface design looks at natural language processing rather in view of applying human interaction behavior pattern than literally applying human conversation language. The hardware side of the speech recognition is required the continuous recognition technology that allow the user rather to speak to the system in an everyday manner than using specific technical words and pausing. On contrast to that, discrete recognition technology that allows the user to speak limited each words and phrase by pausing between the utterances.

The speech synthesis has longer history e.g. interactive voice response (IVR) than speech recognition. It divided mainly two categories like concatenated synthesis and formant synthesis. The concatenated synthesis is natural speech and controls prosody that affects additional meaning of the human conversation. It creates meaningful speech utterance by assembling the recorded voice sounds. Since the concatenated synthesis record human voice sound, it requires the storage space and high computation power for assembly. Therefore, the small vocabulary works best for the concatenated synthesis system.

On the other hand, the formant synthesis represents unnatural speech and artificial voice. It generates audio waveform of machine speech by simulating human speech on the base of phonological rules. It articulates each word separately and producing unlimited amount of speech because of free from storage issue. It applies for Text-to-Speech (TTS) application that generates speech output from text date input. Considering two categories of speech synthesis, a synthesizer can produce an unlimited vocabulary with natural human voice speech output by combination of both technologies. The naturalness is a subjective measure and perspective of the user interpretation. The pin-point technique of the hybrid of synthesize can cover each short point and take advantage of merits of both system.

The system behavior structure is able to design by SUI but it is difficult to determine all the vocabulary that will be used in the system in advance. If the system starts with formant voice, the user feels the in-empathetic and uncomfortable to communicate the system from the beginning. The importance of the first impression in the human behavior interaction can apply for human-computer interaction. The user usually build up the interaction mental model at the first contact. It is good outcomes comes better first impression. If the user realizes the evidence that the system is hard to communicate during the first session, the user opt to avoid using the system to find another alternative way to achieve the goal of the system, since there is many way to communicate. For example, use concatenated voice at the welcome message of the system and use the formant voice for unexpected data.

3.7 Techniques in SUI

Speed and the amount of time-spent waiting were important to users. They did not want to wait a long time while data was being retrieved. A delay of more than approximately 5 seconds caused concern because they were given no interim feedback. Delays in voice applications cause more frustration than in GUI applications [Marx *et al.,* 1996]. If the prototype is designed to provide no feedback telling users that it was retrieving data, whereas with a GUI

application, the user may get feedback by looking at the display. Users require constant interaction in voice applications because they have no other method of control or feedback. With a voice-only interface, only audio feedback can be used. Different cues can be used to tell a user when they should speak, when the system has or has not understood their requests, when it is fetching information (if long delays), and so on. Distinctive sounds and intuitive sounds are an effective way of providing audio feedback to users, but their duration is the second essential parameter.

The appropriate time of duralation and precise response of the user's prompt prodives greater feeling of control and reduce the burden of user's short memory consumption. According to Dey's [Dey, 1997] study, the users feel a lack of control when waiting for system responses and the burden that voice output places on cognitive load. The example of experiment described as below:

> *As For example, with some users, they replied to a request for information about a particular stock with a 30 second long discourse on the current price, annual high and low, price-to-earnings ratio, and so on. With other users, they replied to the same request with a short reply simply stating the current price and the daily change, and providing options for obtaining the more detailed information, if desired. The second group of users was more satisfied than the first group. ([Dey et al., 1996], pp. 3)*

Users feel a lot of stress during interaction with SUI application especially, long duration of time response. Since speech is temporal, the events should occur one after another. Otherwise, users become frustrated and try to speed up the operations that cause more errors. Vice versa, too fast speech pacing interaction might create stress to users that not knowing what is being spoken and what they are supposed to react.

The greatest design problem was the lack of a speech recognition system. Even without computer controlled speech recognition, a useful system was developed using a Wizard of Oz approach. As with GUIs, the important design considerations with speech interfaces deal with users' perception of the interface: how much control they have and how easy it is to use. There are essential elements techniques in speech interface designs to cover those issues that are error modeling, time out, barge-in, and wizard of oz study (WOZ).

3.7.1 Timeouts

Many speech interfaces treat the lack of response after a certain time window as a timeout error. A common strategy for a timeout in many

interfaces is to repeat the last played prompt, in hope that the participant will be able to respond, in case they did not hear the prompt the first time.

3.7.2 Error Modelling

The error handling is one of critical element of speech interface design. If a random error happens repeatedly, a user might be frustrated and leave the system before completing the purpose of using. Robust speech interface designs handle these types of errors.

3.7.3 Barge-In

The complex recognizer and synthesizer technologies carry on in a speech user interface require a high level of technical competency to understand. The barge-in, to allow the user interrupt the system utterance to take control or move to the next phase, is a fairly sophisticated speech interface technique. It is especially important in conversational interfaces where the system utterance or prompt might be long and repetitive.

3.7.4 Wizard of Oz in Speech Interface Design

There are many factors behind the incorporation of SUI (speech user interface) into everyday use though a number of researchers and industry analysts believe that SUI will become popular.

Fraser and Gilbert describe the wizard-of-oz technique as "the simulation of a computer system which takes spoken natural language input, processes it in some principled way, and generates spoken natural language responses."

Since there is no speech recognition system underlying this Wizard of Oz test, some of simulation application done by WOZ tool e.g. SUEDE [Klemmer et al., 2000] in the early stages of design to accept several alternative inputs. In that example, the wizard might accept the user's utterance in a many way like "Send. " and "I want to send." as valid utterances for the "Send email" response link. This is the one of classical benefits of word-spotting technique. The key-word is "send."

3.8 Enhanced behaviour

This chapter presented the requirements of the speech interface architecture and the problems faced in designing and evaluating proposed improvements to its key communication protocols. The challenge is to develop a method for designing improvements to architecture such that the improvements can be evaluated prior to their deployment. A project that focuses on multi-modal input for computer access, in general, the classic example of experimental multi-modal system that intended to replace the

existing input device to another device: keyboard to speech, keyboard to pen-tapping, etc.

My approach is to use an architectural style to define and improve the design rationale behind the Web's architecture, to use that style as the acid test for proving proposed extensions prior to their deployment, and to deploy the revised architecture via direct involvement in the software development projects that have created the Web's infrastructure.

The next chapter introduces and describes the Voice Bulletin Board System (Voice BBS) architectural style for distributed resource systems, as it has been developed to represent the one of example model for how the independent-sequential modality system should work through modern Web technology. Voice BBS provides a set of architectural constraints that, when applied as a whole, emphasizes different modality interactions for common data, generality of conceptual interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce-security, and encapsulate legacy systems.

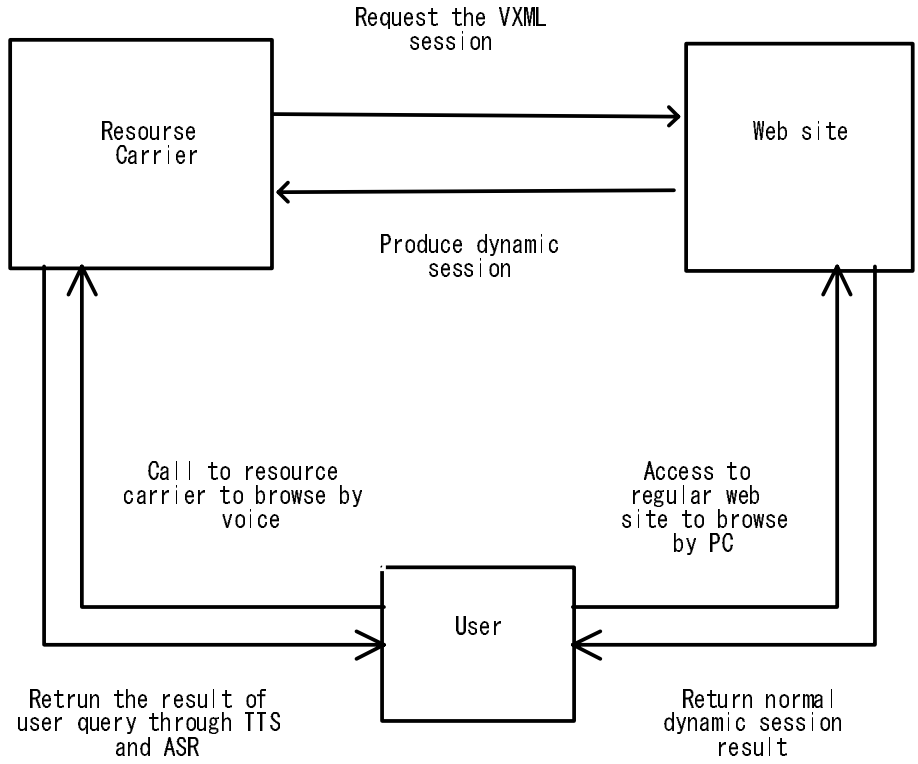
4. System Architecture

This section explains concept of voice application, "Voice BBS" including system and brief speech design.

The system calls "Voice BBS". BBS stands for "Bulletin Board System" and it is equivalent meaning as "discussion board system." The design of system is based on distributed system. There are three sites and each site has its own dedicated task.

4.1 Overview

The aim of this application is to provide a user for multi-modality interaction through the web site. A user is able to write, read, and browse his/her favorite normal BBS (Bulletin Board System) through the web and telephone (both fixed and mobile). Talking about voice browsing through telephone, it considered only read and browsing at this moment. It gives a user to additional freedom to access web site when the normal PC is not available.



4.2 Application procedure

A user opens any web page that he/she likes to join BBS and write, read, and browse it. Later, user calls to the BBS site through the phone to browse the site.

4.3 System view

This application aims to re-use existing resource on the web rather than re-build entire new system. It can be added on existing site and run same as before for normal BBS side. It can be adapted to several languages because it depends on the resource carrier side (such as Application Service Provider: ASP) to implement the language technology part including TTS (Text to Speech) and ASR (Automatic Speech Recognition).

4.4. Functionality

It describes brief voice browse function of this application.

4.4.1 Basic Functionality

There is parent menu item and child menu items on voice browsing part.

A user is able to choose menu to sort items from parent menu and child menu runs recycled order except contents part.

4.4.2 Parent menu

It consists of four parts that is title, name, email, and date. A user chooses one out of those four categories in order to sort contents of items.

4.4.3 Child menu

There are four functions available that are read first, read next, read previous, and read last. Once sort item has been decided at parent menu, a user is able to browse items by four commands.

4.4.4 Sub-child menu

There is a function inside of each item to skip each field to contents.

4.5. User Interface requirement

There is not strict rule to use this system for user side as long as following rules are met.

4.5.1 User side:

1) Ordinary browser for PC or browser phone.
2) Fixed phone or mobile phone to browse by voice.

4.5.2 Server side:

1) The site must be able to use database connection and CGI
2) It must be allowed to redirection from other site.
3) It must be connected to desirable language resource carrier.

4.6 VoiceXML Dialog Collation Chart

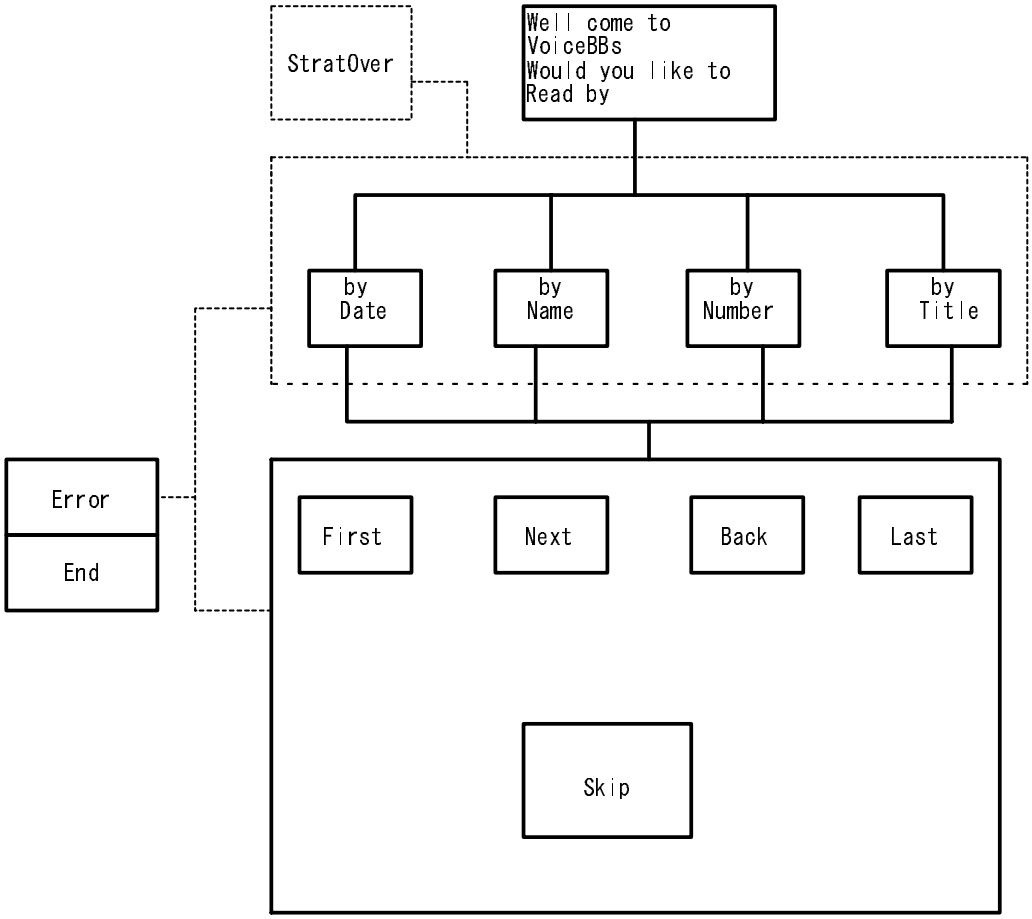


Figure 1. Including “confirm”

4.7 Dialog label

The simple BBBS is based on “saliency” approach. The user doesn’t necessary to utter exact words to operate expect keyword.

4.7.1 System

Initial:

Welcome to Simple-VBBS.

Would you like to read by date, handle name, e-mail address, or title?

Welcome to the Voice BBS Reading Service.

Please choose the service. Would you like to read by: number, name, title, and date?

Error handle:

Uhh, I didn’t get that.

I am sorry. I did not understand what you said.

Start Over:

Please start over your query.

Going back to the first message.

Read By:

You need to read by number, name, title, date.

Welcome to the Voice BBS Reading Service.

Please choose the service. Would you like to read by: number, name, title, and date? (Same as initial)

Confirm:

Ok, Read by date

Date, name, e-mail, title, contents

Reading by name. You can say Skip, First, Next, Back and Last to retrieve message.

Say 'start over at any time to go back to the main menu.

End:

Thank you for visiting SimpleV-BBs. See you again!

4.7.2 User

END:

Good-bye (key word is "bye.")

Start Over:

Main menu (go back to main menu)

4.8 Example of flow dialog

Case1: error incidence during reading.

User: Read by date.

VBBS: uhh, I didn't get that.

VBBS: I am sorry. I did not understand what you said.

User: Go by date.

VBBS: Ok, read by date.

December 20, posted by Tom, no email address, subject is hello.

Contents are: …….

VBBS: Reading by date.

Date 2002 May 26, 20:50:37. number 2, name Tom…….

4.9 Sicons

There are sicons (functionality sound indication to the user) for each label.

Welcome

It sounds when the initiation of the application. It sounds only one time per session.

Good-bye

It sounds when the application ends. It sounds only one time per session.

Error

It sounds when the system couldn't understand the user utterance.

Start Over

It sounds when a user try to start over the search.

Read by

It sounds when the user choose to category which to choose.

Skip

It sounds when user skip a field in the each item.

First

It sounds the first item of articles

Last

It sounds the last item of the articles.

4.10 Data base table design

The database is cue to precede this application property.

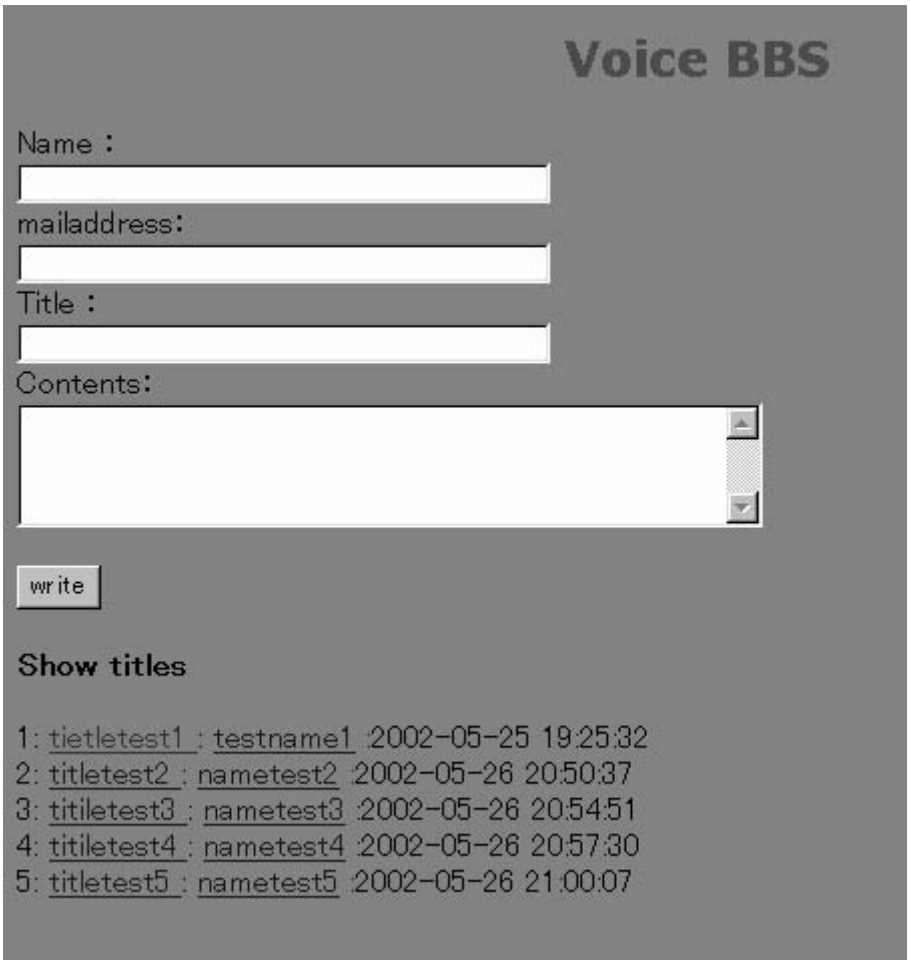
This database is shared between VoiceXML web site and regular web site.

Field Name	Filed Type	Attribute
number	Int(5)	PRIMARY KEY, AUTO_INCREMENT
parent_number	Int(5)	
title	Varchar(255)	
contents	Text	
name	Varchar(255)	
datetime	Datetime	
IP	Varchar(255)	
email	Varchar(255)	
deletekey	Int(5)	

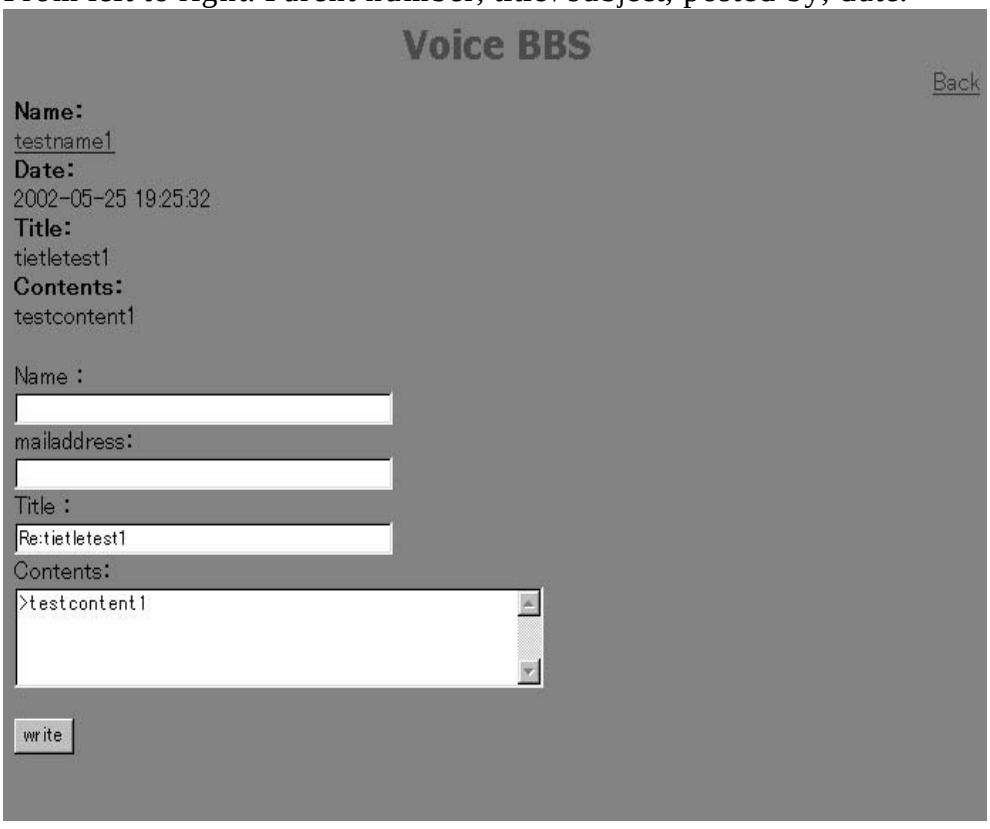
4.11 GUI BBS web part

This might be changed for look and feel.

The field for Name and contents are required to fill out by user. Another field is filled out automatically if a user leaves it blank.



From left to right. Parent number, title / subject, posted by, date.



4.12 Environment

This application is implemented with php4.1.2, VoiceXML 1.0 and HTML. MySQL3.23.49 is used for database server and Apache is for web server. Resource carrier for VoiceXML execution will be English site. There are a lot of possible combinations of this system. However, the combination of php, vxml, MySQL and Apache are expected handy and first response performance at this moment. PHP script and MySQL database is provide fastest response. In the speech application, it is required the fast response to the user.

4.13 Future concern

It is needed to server space to execute this application on real network. This VoiceXML application is a part of BBS system, not main purpose of application. Therefore, the function is limited to read only, not for writing. The VoiceXML application's part plays a different accessibility to a user rather than regular PC access.

However, it is nice to have writing function through VoiceXML application to consider about independent parallel modality.

5. Evaluation

Looking at new technology, VoiceXML, makes possible to provide a different form of service to users through already-existing technology. [Hughes, 1987] wrote: "Technological systems contain messy, complex, problem-solving components. They are both socially constructed and society-shaping."

A broader institutional-level perspective- technological study expanded *usability* beyond its traditional definition of *information-processing capabilities* to include its effectiveness in the conduct of work and social transactions [Kling and Iacono, 1989]. My definition of "usability" examines the effectiveness of technology in people's lives as a function of their comprehension of the societies. Looking at usability of my system design, technologies like telephones and Internet-enabled personal digital assistants (PDAs) are alternative possible access device (e.g. MiPad) for comparable usability. Currently, there are a lot of mobile phone services and PDA. Those technologies are usually required users t o engage in service providers with various policies. I consider my application system is a part of telephony service, VAS (value added service).

Contextual inquiry and design are for naturalistic approach technological study as applied a process to analysis as well as design [Beyer and Holtzblatt, 1998] In the development of a process that can be followed by non-social scientists, the examination of work practices and social and institutional factors

that affect the design, and ultimately the use of a new technological system have considered in the development of a process. For the purpose of my analysis, the lesson can be applied to the engineers of the technology as well: The development and deployment of a technology can reflect-for better or worse-the social organization and various incentives of the business units within the technology provider. This chapter examines the usability of the Voice BBS step by step through the experience and lessons learned during implementation. It discusses the first intention of the examination, inspection of SUI (Speech User Interface) and flexibility of the Voice BBS as a whole.

5.1 First intention

Direct observation allows the investigator to observe object activity use as it really happens, but when tracking particular participants, requires getting access to the many places participants spend their time while also involving a large time commitment for all parties. Instead of having participants record their activities in a paper diary, I was supposed to plane to invite participants to call in to a dedicated SUI line and talk about their experiences. However, there were some socio-technical problems beyond the Voice BBS. For example, toll-free is only limited in certain area or inside of the United States. The calling to that “toll-free” number from outside of the states is charged as international long distance call.

I would ask them to the effort would be low cost for participants: Since their calls would go straight to a voiceXML service line (voie server), they could call at any time of the day or night, and they knew their time investment would be only the length of the message they wanted to listen. It would be free as long as inside of the U.S. I could not find any ASP site except in the States. It is considers as international call to reach that site outside of America. This effort would be also low cost for me. Since Palen and Salzman study says 42% of participants made daily report as a result of giving $1 for every day they called in during their usability test [Palen et al 2002], motivation of the user is the key to succeed in the system. This hypotheses estimate is based on my using discussion board system. The discussion board is for a kind of amusement club. That discussion board system that I use quite often has a tendency to be used; certain days of the week (most of the cases are on weekend and holidays) and day after some event happen in the club. What is more, not all member are active on the board even the member participated the event.

5.2 Practical approach –heuristic evaluation

A lot of usability studies have done by direct observation based on task-based evaluation with the system. However, this section examines the system in term of technology aspect and former usability studies. Therefore the conclusion of this section would be different as the result of direct observation test like usability test. The direct observation study should be carried out for the future development of the system.

5.2.1 Socio-technical factors in SUI

Evaluation of usability is often measured as a function of the performance of the hardware and software of a device. In this sense, hardware and software is as easy to measure to focus attention. However, the examination of this section focus on novices' use of the system through mobile telephony, they struggled to understand how the larger technological system worked. In other words, the user should not puzzle to figure out the system to use. The acceptability of the users to the system plays a lot of means. People usually prefer natural voice but prefer synthesized voice for warring because it sounds different from other voices in the immediate environment [Cohen and Oviatt, 1994]. This included wrestling with socio-technical aspects of the system as well. I saw that multiple other technological, service, and agreement policy issues that users struggled to weave together into a plausible mental model of mobile telephony operation complicated the usability of the mobile phone device itself. However, there are some technology problems for this system such as socio-techno-environmental, not design system. For example, fire wall problem prevents for testing environment. Therefore, direct observation method is not suitable evaluation though it looks effective for SUI usability study for this time.

5.2.2 Expression possibility in inspection method

It is possible to apply heuristic evaluation to usability inspection, as defined by Jakob Nielsen, is "the generic name for a set of methods based on having evaluators inspect or examine usability-related aspects of a user interface." [Nielsen and Mack, 1994] The list of recognized usability principles is the inspection methods cataloged by Nielsen. The heuristic evaluation is the least formal and involves having usability specialists judge whether aspects of a given interface conform to a list of established usability principles, known as the heuristics.

Heuristic evaluation, along with the other inspection methods, differs from more conventional empirical usability testing in significant ways: evaluators are not drawn from the user community, evaluations take less time, evaluations are easier to set up and run, and evaluations cost less. "It is easy (can be taught in a half-day seminar); it is fast (about a day for most evaluations); and it is as cheap as you want it." [Nielson and Mack, 1994] The promise of ease, speed, and low cost attracted me. It came to the conclusion that; it evaluates the prototype and explores the method to modify this project. The objective was to determine the usability shortcomings of the prototype so they would not be repeated in the final product, and to determine whether heuristic evaluation had promise for future projects within Voice BBS and similar system projects.

It is expected systematic usability improvements through findings and lessons learn during implementations. In adopting a systems analytical framework, these usability findings constitute reverse salient of a sort: As a result of putting mobile telephony into VoiceXML practice, it helps the users to reveal the shortcomings of the technology for everyday use, which reverberate throughout the socio-technical system, and prevent advancement to technological closure.

Since the heuristic evaluation approach could be applied to design, code and deployment stage, it is ideal approach for my current project status. Heuristic evaluation, developed by Jakob Nielsen, is a method for structuring the critique of a system using a set of relatively simple and general heuristics. Nielsen describes that heuristic evaluation is a discount usability engineering method for quick, cheap, and easy evaluation of a user interface design [Nielsen and Mack, 1994]. Overview of heuristic is a guideline or general principle or rule of thumb that can guide a design decision or be used to critique a decision that has already been made. Nielsen's experience indicates that around 5 evaluators usually results in about 75% of the overall usability problems being discovered.

The general idea behind heuristic evaluation is that several evaluators independently evaluate a system to come up with potential usability problems. It is important that there be several of these evaluators and that the evaluations be done independently.

What is evaluated? Heuristic evaluation is best used as a design time evaluation technique; because it is easier to fix a lot of the usability problems that arise. But all that is really required to do the evaluation is some sort of artifact that describes the system, and that can range from a set of storyboards

giving a quick overview of the system all the way to a fully functioning system that is in use in the field.

For this study, to follow the classic evaluation methodology in order to discover usability problems in terms of objective perspective as much as possible, a list of ten legacy evaluation heuristics is used to examine the system properties which can be used to generate ideas while evaluating the system. Here is a list of heuristics: Visibility of system status, Match between system and the real world, User control and freedom, Consistency and standards, Error prevention, Recognition rather than recall, Flexibility and efficiency of use, Aesthetic, and minimalist design, Help users recognize, diagnose, and recover from errors and help and documentation.

The next section elaborates each property and examines them to the elements of Voice BBS's system followed by Nielsen's list.

5.2.3 The heuristic evaluation analysis in Voice BBS in terms of SUI

1. *visibility of system status* means that the system should always keep users informed about what is going on, through appropriate feedback within reasonable time. This property can find in the *confirmation* feature. It gives assurance to the user what interaction has been occurred between the systems. Human speech has been associated a lot of recognition errors. You should use confirmation questions to assure the system has heard the right message from the user. For example, the system confirms the user next action when user asks query. The user says that "I would like to read by date." The system responses "Read by date".

2. *Match between system and the real world* means that the system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in natural and logical order. It can find in the user *voice command* such as help and start over. Since this system use word-spotting technique to match the action, the user says any natural language what he/she expect to receive action from the system. For example, if the user needs a help to understand what is going on, he/ she can say that "Help!" or "I need a help." or "Give me a help menu", etc. The user gets the desired action from the system as long as the user's utterance matches expected action.

3. *User control and freedom* means that users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo. It can be found in *barge-in* feature. The system starts the action that a user not wants to or the user change the mind to cancel the request, when the user interrupts system's action. For example, the users ask query the message by date and the system start to read by date. However, the user says that "Oh, stop that. Read by name". The system immediately stops to read by date and start to query by name. The user feels the sense of control in using the system.

4. *Consistency and standards* mean that users should not have to wonder whether different words, situations, or actions meant the same thing. Follow platform conventions. It can find in s*icons.* Each sicon has different indication to give a cure to the user. For example, user asks query and the system starts query. The system gives query sicon during the query. But the user would like to back to main menu. When the user ask to go to main menu, the system gives start menu sicon. The user recognizes that the system stops the query and goes back to the start menu. If the user could not receive the going back menu sicon, he/she might start to wonder whether the system is still doing query or something wrong the system.

5. *Error prevention* means that even better than good error messages is a careful design which prevents a problem from occurring in the first place. It can be found in *status message* feature. Since the Voice BBS system gives confirmation message, the user realizes the situation of the system. It prevents to go wrong direction for user's undesired operation.

6. R*ecognition rather than recall* means that make objects, actions and options visible. The user should not have to remember information from on part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. It can find in *voice command* feature and *word-spotting* technique. Since the key word for word-spotting is intuitive, the user easily associates the action and the system behaviour. The word choice is the one of the biggest challenges in the speech interfaces. Even though technique and

technologies has been considered and advanced, there are still significant human factors in what the user can understand. In the each speech application, the types of errors are various. However, there is still room for the interface to attempt an interpretation of the user's speech or actions based on context. In order to make the progress in the accuracy of the communication between the user and the system relies on speech interfaces. It is the easier the job to be improved the interface. Therefore, the interface dialog design must remove the uncertain factor by being even more attention to in interpreting the user actions and speech. In the voice only system session, a common error is the user misinterpretation of what the user should says to the system in order to make action. The more closely the user can recognize, the more usable the system dialog. For example, the user wants to hang up and the finish up the system, he/she says that "Good bye." It finishes the operation of the system and exit the system. Even the user could say "I want to exit this" to end the system. There is not system-oriented terminology but the user will and action.

7. *Flexibility and efficiency of use* means that accelerators, unseen by the novice user, may often speed up the interaction for the expert user to such an extent that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. It can be found in the *barge-in* technique. Since the user is able to interrupt the system's action, the user reaches the destination phase of the application quickly. In this sense the bare-in feature also works as a short cut operation to omit the unnecessary system's dialog. When the user meets the system at the first time, he/she can create or realize the mental model or how the interface will work. It might exist before they begin to use the system.

It is related the conservation that associates previous experience to the cognitive level in order to realize or understand the exiting the world. If they have heard or read that the new interface will be like using a voice mail box, they might have the mental model of a voice mail box in stead before even use the new system. In this sense, the user always has a conceptual mental model operating. It might mislead the users what they meant by receive the out come through the input. Because the users' conceptual mental model is not always match the conceptual model of the actual interface. But the user do change he/her mental model of specific

interface through experiences with the product itself and providing help to the user accelerates to adjust their mental model.

Speech only interfaces are particularly difficult to develop a conceptual model for because they are usually momentary event. If the interface is not multi-modal, the information the system provides “exit” for a short time even it is only long enough for it to be spoken by the system. This feature adds to the challenge to carry out enough information about conceptual model. To guild the user’s mental models in order to adapt the system is an important concept in interface design and usability evaluations. If a user’s mental model does not modify well with the interface’s conceptual model, the interface will be hard to use.

Consideration of design and communication a conceptual model that fits well with the user’s mental model are two important tasks of the interface design process and usability inspection. Therefore, the interface will make reasonable guesses about what the user is trying to do. The user should not have to figure out how the system works or how will response to his / her input. The interface of the system should be able to observe user behaviours and make a reasonable guess as to the action or speech the user is trying to modify. The guiding the user’s need is key to meet this seventh requirement. It applies the barge-in feature to meet enough to build up quick conceptual model how the system works. Because it allows the user interrupt the system any time, any user is able to reach out directly his /her desirable phase of the application.

8. *Aesthetic and minimalist design* means that dialogues should not contain information which irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. It can be found in the corresponding *voice command* feature. Since command and action matches exactly the user’s mental model, there is no confusion of the semantic communication and syntax. The user can understand and realize the real time interaction. For example, if the user would like to jump to next message, the user can say that “Next”.

9. *Help users recognize, diagnose, and recover from errors* means that error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. It can be found in the error handling feature and barge-in technique. Once the

user has caught in the error event, there is sound icon (sicon) and error message. This is considered one of the self-detect and self-correct models. The users recognize their error before the system detects it. It is more important to allow the users to detect and correct an error before the system interrupt than modify the programming in the system. For example, if the user says that “go to next” and realize that the user realize that he/she should have said that “skip”. The user can say that “Oh, no. Go back.” Letting the user know what the error and what he /she can do to correct it is important fact in SUI. Because one of common problems in speech applications is that the users do not know when the system is ready to hear from them. In other words, they are not sure when it is “their turn to speak.” The users need to know what the system is doing even though the system does not have to let the user know what it is doing in order to correct the system in being function. Providing the feedback to the user about the status of the system is the one of ways of communication between the system and user.

It is hard to know what caused an error. We wonder that whether the user never leaned how the menus works or the system interface problem for not having better menu structure when the user choose wrong menu consistently. The user needs feedback on his/her actions and feedback on the system’s actions. Feedback plays a lot of rules when the user expects that he /she are having “communication” with the system.

It is considered that one of the rules of the interface design in the system is to help reduce the number and serious errors the user can make, as well as to help minimizing the effects of the errors the system can make. Therefore, the error messages should be specific. There is another example. If the user mess up and the system starts to say that “I am sorry. I do not understand what you say. Say ‘start over’ to back to main menu or help to get guide.” The user realizes that something wrong in the interaction to the system and get rid of the situation by saying “Start over.” The user can change the course and correct it.

10. *Help and documentation* means that even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not

be too large. It is found in help menu in this SUI system. It is not large documented help file but the guide for the user to assist in order to use the system. The detail information of this system should provide the same site as Voice BBS for GUI site not for SUI site. Some of the traditional ways of providing support such as help message on screen, user documentation are not suitable in the system in temporal applications. The help must be built in the application in order to correct error. Those should be considered in terms of increased design of the system and usability. Defining resource of the documentation such as help file or menu a concept rather than a document leaves us with another question: how does a user access, manipulate, or transfer a concept such that they ca get something useful when an interaction event has occurred? Voice BBS answers that question by defining the things that are manipulated to be representations of the identified syntax voice command interaction, rather than semantic interaction itself.

According to heuristic evaluation guideline, it is useful to have another person evaluates the interface that I have been designed because it is difficult to recognize flows when the design is myself. But I did not have opportunity to find a person to do for me and the Voice BBS is prototype and still under development to improve the system itself. I considered inspection by another person is next version step and it is useful to examine the system myself in order to enhance in the view of usability interface design to improve the system itself. At last, it should be noted that the heuristics are meant to help the evaluators to find usability problem, but not to restrict them to find only the problem justifiable by the heuristics.

5.3 System flexibility

The theoretical constructs, Law's [1987] concept of "network," Kling and Iacono's [1989] "web," and Hughes' [1987] "system" can help broaden traditional definitions of usability. In this sense, technology is always the combination of conceptual system to invent new technology. Looking at usability analysis for service-based technologies like mobile telephony, would include addressing additional socio technical elements that affect use beyond the handset hardware and software alone. Helyar [Helyar, 2001] and Silfverberg [Silfverberg et al., 2000] described as observation in traditional usability lab studies that usability evaluations with a systems-level scope can

complement the kinds of detailed interaction findings. It indicates that the usability analysis done by in terms of each systems component.

Talking about the system as the adaptation of mobile technology, there are several usability issues when the system put into practice. In adopting a systems view of mobile telephony technology, although a carefully scoped one, I have identified several usability issues that arise when the technology is put into practice. The fact that even the non-technical aspects of the technology are nevertheless a part of the larger technological system that emphasize the system components categorize as hardware, software and network. I have also implicated to the relationships that socio-technical factors with an analysis that weaves together many of the findings presented through this project, and how the decisions or designs affect another portion of the component domain. In this section, I discuss this component level, socio-technical relationships on user expectation and "system-level usability" based on mental model rather than focus on VoiceXML application itself alone. Because my framework is consider as additional service technology of mobile telephony. The back bone of the network system can't accept the break down. It conducts the distributed model system to add in and upgrade the each component.

5.3.1 GUI vs. SUI

An origin server for GUI maintains a mapping and generates dynamic VoiceXML scripts from Voice servers to the set of response corresponding to database information. A database resource is manipulated by transferring through GUI site server interface and SUI site server. Therefore it is possible to implement the database resource server independently. Voice BBS database resource drives from the requirement of the Web server such as SUI site and GUI site server: independent authoring of interconnected script across multiple trust domains. Forcing the interface definitions match the interface requirement because the resource to seem vague because the SUI interface being manipulated is only an ISP site, not implementation site. The ISP interface is specific about the intent of an application action from voice server through generated dynamic VoiceXML script from GUI site server, bet the mechanism behind the voice server interface must decided how the intention affects the underlying implementation of the generated VoiceXML mapping to representations.

Independent-concurrent modality of the system operation is one of the key that motivates the Voice BBS. Because the user is restricted to the manipulation of the representations rather than directly accessing the implementation of resource, the system can be constructed in whatever form is desired by Web internet access without impact the clients that may use its

authority log-in. If multiple representations of the resource exist at the time to access the resource, the SUI side updated after the GUI side of the database has been updated. It is the premise that Voice XML is slave to the GUI site. The file of GUI site has locked when it has multiple accesses at the time. Therefore the content of the SUI site is always delay compare to the GUI site. The date source of both sides is not straight forward of the original data. There is always slight delay data update between the site (between the SUI and GUI).

Database resource does not always map to a singular file but not all resource that is not static is deriving from some other resource. The user can edit the contents of the BBS by finding address of URI. However, the user is not able to edit configuration of the system itself. The Voice BBS is not only targeted only for CPU embedded object but it is conceptual modality model proposal in order to enhance socio-technical factor in these days.

The method of generating response for all the contents does not need to change is that the Web interface: if the system designed properly, the modality of the interaction between user and the system can retrieve the common resource, meaning that from the client's perspective, that knows about the suitable situation to apply for and about how the system offer to the user without changing the aside from the system conceptual structure.

The semantics interactions in the system are a by-product of the act of assigning resource information. The user or SUI need to know the meaning of syntax. They act or interact as a result through that creator of resource content. In other words, there are no resource for top of the GUI or SUI; it just mechanism that supply the interface across the shared database that it is operated by remote setting. It is the exactly what the Web work across so many different implementation which makes different modality possible to operate the system.

5.3.2 Environmental aspects in modality issue

It is the common characteristic among the computer science that it tries to define things in terms of components that will be used to finished product that the engineer predicted how the user interact. The user dependent modality product doesn't work that way. It is decided by the user's situation and environment which modality will be used. The RPG (role playing game) is a kind of system changing the content by how the user chooses the story. The story of the content is changed by user's operation. The same rule exists in Voice BBS - the user's freedom of choice. If you talk about the user's choice, it means a lot. There is one dimension in RPG that the user has a right to choose the story end - so do Voice BBS. There are two dimensions in Voice BBS that is modality and environment that in table 1. (modality issue)

Modality (UI)	Environment	
	Stable	Mobile
GUI	Read / Write	Read /Write
SUI	Read	Read

Table1: Function scale in modality and environment.

There are two type of modality in Voice BBS that are GUI and SUI. The environmental vectors are mobile and stable. The main sensory perception of the human communication has done by visual, auditive, tactile [Charwat, 1992]. If you look at legacy language learning process at current educational system, there are categorized two modalities such visual (reading and writing) and auditive (hearing and speaking). The sense of sight is the main sensory perception for GUI, both sense of hearing and touch are sub-sensory perception. The Voice BBS has limited the modality as visual (GUI) and auditive (SUI).

The current implementation is only for independent-exclusive modality but it is possible to become independent-concurrent modality as whole system. But those modality operations are substitute function in order to accomplish the purpose of the system-main function of BBS that is using same resource by different modality operation. Because the SUI is always sequential, independent-exclusive modality while GUI can be simultaneous, combined-synergistic. The GUI could be combination of three modalities such as visual, auditive, tactile. It might be possible to add several modalities such as

The combination of GUI and SUI in the mobile environment is ideal implementation in terms of synchronized update and building up seamless operation mental model between stable and mobile. The sense of hearing is the main sensory perception for SUI. It might be possible to add SUI to another sensory perception such as sense of touch in order to enhance usability. The vibration function is the most possible tactile cue in these days conceptual technology. Talking about ideal implementation, the key of succeed the system relies on minimizing function and corresponding available function between mobile and stable setting.

One of the main goals of this system is that offering modality freedom of choice to the user for any situation by providing same expectation outcome. Hence the vector of the modality has limited GUI and SUI by examining the environmental variables in terms of human communication [Charwat, 1992]. The user is able to expect equivalent operation at any environment. If you look at GUI as axis, you see read-write function in both stable and mobile environment. It is same things that you can see in SUI. This matter provides the user assurance at any environmental factors. Once the user build up the

mental model for system operation, it is convenient for the user to follows that conceptual model in order to communicate that system. It is one of benefits for the user to omit the learning time at various environment setting.

5.3.3 Practical adaptation

However it is still considered as independent-sequential model because the SUI function is limited such as retrieve the data by SUI. Using SMS (short message service) [ETSI, 1996] to text input most ideal way in terms of using existing resource system by add in implementation to expand functionality of the system. The alternate goal for this Voice BBS system is table 1 and it is possible to add the implementation one by one. The current implementation is table 2.

Modality (UI)	Environment	
	Stable	Mobile
GUI	Read / Write	N.A.
SUI	Read	Read

Table2: current implementation

The GUI in mobile setting is not available current implementation. The first step is SMS implementation in order to enable writing function in mobile setting. The standard GSM and PDC's (Personal Digital Cellular -It is used mainly in Japan as second generation cellular phone.) MT (mobile terminal) has already implemented SMS interface, all the Voice BBS has to do is to implement SMS-GSM/PDC interface in order to access Voice BBS database. This implementation involved in service career. However, if you would like to be closed the implementation only inside of Voiced BBS, you have another choice - browser enabled MT. For example, WAP (Wireless Application Protocol) phone is one of the choices. As long as the MT is browser phone, the implementation is WAP proxy or i-mode server in order to access the database. (See figure 2 for the implementation of GUI axis.)

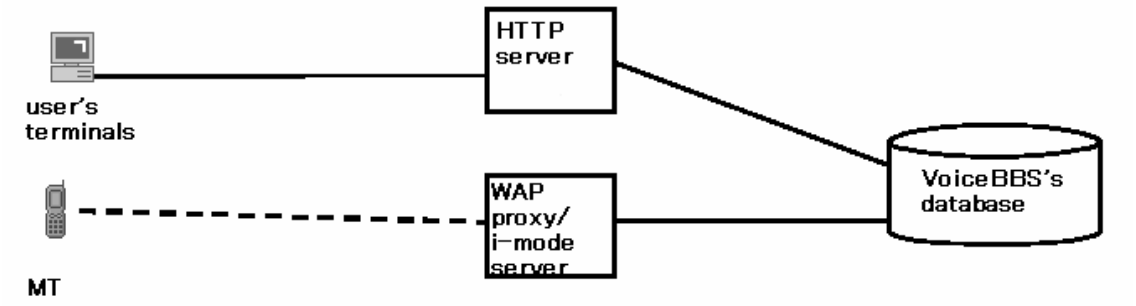


Figure 1: GUI-read and write implementation in both stable and mobile environment

The user's terminal and MT are representatives of stable and mobile environment in table 1. The user's terminal could be a desktop or notebook.

For the mobile terminal could be the PDA (Personal Digital Assistance) or any type of browser enabled cellular.

The SUI implementation is different structure from GUI (see figure 2). The user's terminal could be the desktop or notebook that are enabled browser in order to install VoIP (Voice over internet protocol) or SIP (Session Initiation Protocol) enabled software-phone and fixed phone. The cellular phone and browser enabled PDA is possible mobile terminals.

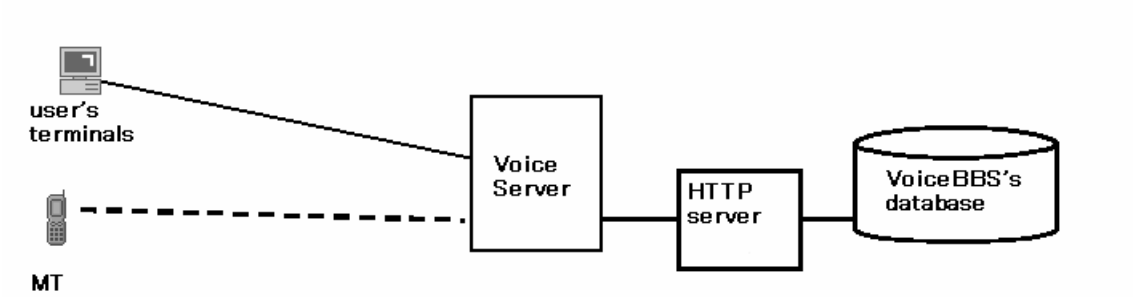


Figure 2: SUI-read implementation in both stable and mobile environment.

Voice server

The implementation of the voice server is various in order to handle multi-environment. For example, voice server should be enabling to SIP gateway for browser access in addition to PSTN access. The voice server should take care of annotation of voice server behavior and type of script enabled -which script is available to generate dynamic code. The voice server is the key for SUI in Voice BBS system. It is reusable the voice server once it has been builds up. It increases the SUI engineering productivity by multi-purpose-using the voice server.

The system structure of Voice BBS is possible to use low power MT for SUI because it doesn't necessary to be required speech recognizer and synthesizer engine on the device itself besides the voice server on the web. The terminal devices are able to access the voice server through SIP or PSTN. It helps to reduce massive computation work for hardware. If the Voice BBS system model has applied for other systems, it is able to share voice server among the projects. Because all the terminal devices have to do is to access the voice server on the web for SUI.

5.3.4 Security issue

It is considered that one of aspect of the Voice BBS feature is unusual for architectural style in order to provide different modality situation which it influences the definition of the user modality choice. Voice BBS model defines the expected application behavior that supports simple and robust applications that are largely immune from the partial failure conditions that beset most web based applications by implementing each server separately. Hence it is able to

update each site independently and reduce the entire crash of the system. For example, if the HTTP server has down the user can use SUI instead of GUI. However, it remains the security problem in SUI side.

Shouting the password to the device in the public is not desirable behavior in terms of both manner and privacy protection. Enabling DTMF (Dial Tone Multi Frequency) feature (VoiceXML supports DTMF feature.) in SUI is the solution. The current Voice BBS has not added in writing feature yet the reason that we have discussed earlier about speed issue. Even if the DTMF feature has added to SUI, it is the question to add writing feature in SUI for this system in terms of speed flexibility. It is required more complex dialog in order to handle recording error. It causes to increase the user confusion and frustration to the system. The Voice BBS use DTMF feature only initial entrance of the system in order to recognize the service ID for multiple-use. It is difficult to change the user mental model that the user learned once the system is not user-friendly in terms of socio-technical use. Therefore the Voice BBS system doesn't have writing feature in SUI and concentrates on the data retrieve in order to use casual resource checking or reference.

6. Summary

If we think about in terms of mobility access in different modality, there are several possibilities. Moreover, it might be considered to input data feature of mobility for my framework in the future implementation through existing technology. The SMS (short message service) [ETSI, 1996] is the most likely to meet the requirement of my system design. My system, "voice discussion board system - Voice BBS," is based on distributed resource model. The current implementation is focused on independent-sequential modality input and output. GUI side is considered both input and output operation whereas SUI side is only output (message retrieving); However, the combination of SMS and VoiceXML technology make possible for full independent-parallel modality to both stable and mobile environment. It will increase the ability to be mobility in those days socio-cultural technology.

The Voice BBS system elaborates only one aspect of the modality problems like voice only command operation that are considered essential for multimodality in SUI interaction side. Areas for improvement of the SUI can be seen where it exists mental model fail to express all of the potential semantics for based on human factor interaction and where the details of syntax can be replace with more efficient models without changing the system

architecture and capacity. Likewise, proposed extensions can be compared to smart phone type of device applications.

Voice BBS sees if they fit within the architecture; if not, it is more efficient to add that functionality to a system running in parallel with a more applicable architectural style in order to make up the some features. For example, implementation of input feature trough mobile device to Voice BBS is the one of extension in the future. In an ideal world, the implementation of a software system would exactly match its design. Some features of the previous multi-modal system architecture do correspond exactly to their design criteria in Voice BBS, such as MASK, the use of URI to retrieve the data as resource identifiers and the use of Internet media types [SmartKom] to connect different mobile interface representation data resource. But there are also some aspects of the modern Web protocols that exist in spite of the architectural design, due to legacy experiments that failed extensions deployed by developers unaware of the architectural style, the fire wall. Voice BBS behaves differently in the each environment. It states same behavior for the same setting. Some of environment behaves as the design of the system has expected but some of them don't. Hence, one of solutions is to provide a model not only for the development and evaluation of new features, but also for the identification and understanding of broken features for the moment. Understanding the key architectural principles underlying the interaction can help explain its technical success and may lead to improvements in other distributed applications, particularly those that are amenable to the same or similar methods of interaction. Just as in human-human dialog, grounding the conversation, avoiding repetition, and handling interruptions are all factors that lead to successful communication.

Voice BBS attempts to contribute both the rationale behind the modern Web's software architecture and a significant lesson in how software engineering principles can be systematically applied in the design and evaluation of a real software system. For network-based applications, system performance is dominated by network communication. For a voice only interaction system, component interactions consist of performance of speech recognition and synthesize computation-intensive tasks for date transfer. The Voice BBS interaction modality model was developed to fulfill to those requirements. Its focus upon the modality independence for the user required situation to situation in order to retrieve the data as a connector interface of resources. The representations in Voice BBS feature have enabled intermediate processing of common data and substitutability of components, which has allowed Web-based applications to access the user requests any time.

The World Wide Web is arguably the world's largest distributed application. It is huge theme to deal with paper in detail. However, the internet technology itself is also growing fast and improving day by day, it catches up those issues in the feature. At this moment, we should keep in mind and take into account this fire wall problem in order to avoid undesired situation when we design the network-based system.

References

[Alm, 1987] Norman Alm, Alan Newell, and John Amott. A communication aid which models conversational patterns. In Richard Steele and William Gerrey, editors, Proceedings of the Tenth Annual Conference on Rehabilitation Technology, pages 127-l 29, Washington, DC, June 1987.RESNA.

[Ballard, 1997] DanaH.Ballard, An Introduction to Natural Computation, MIT Press, Cambridge, MA, 1997

[Beyer *et al.*, 1998] H. Beyer, K. Holitzblatt, 1998. Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann, San Francisco, CA.

[Bijker *et al.*, 1987] W. E. Bijker, T. P. Hughes, T. Pinch, Eds. The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology. MIT Press, Cambridge, MA, 1987.

[Black, 1990] Black, A., Visible Planning on Paper and on Screen: The Impact of Working Medium on Decision-making by Novice Graphic Designers. Behavior & Information Technology, 9(4): p. 283-296, 1990.

[Bolt, 1984] Richard Bolt, The Human Interface: Where People and Computers Meet. Life long Learning Publications, Belmont, CA, 1984.

[Bos et al., 1994] Edwin Bos, Carla Huls, and Wim Claassen. Edward: Full integration of language and action in a multimodal user interface. International Journal of Human-Computer Studies, 1994.

[Brecher, 2000] Brecher, E. Power to the people. *Miami Herald* (Feb. 2, 2000), 1E.

[Buxton and Myers, 1986] William Buxton and Brad A. Myers. A study in two-handed input. In CHI '86, 1986.

[Brown and Harper, 2001] Brown, B., Green, N., and Harper, R. Wireless World: Social and Interaction Aspects of Wireless Technology. Springer-Verlag, London, 2001.

[Cairns et al., 1994] Alistair Y. Cairns, William D. Smart, and Ian W. Ricketts. Alternative access to assistive devices. In Mary Binion, editor, Proceedings of the RESNA '94 Conference, Arlington, VA, 1994. RESNA Press, pages 397- 399.

[Carletta and Isard, 1999] Carletta, J. and Isard, A.: The MATE Annotation

Workbench: User Requirements. In Proceedings of the ACL Workshop: Towards Standards and Tools for Discourse Tagging, University of Maryland, p.11-17, June 1999.

[Charwat, 1992] H. J. Charwat, Lexikon der Mensch-Maschine-Kommunikation, Oldenbourg, 1992.

[Churchill, 2001] Churchill, E. and Wakeford, N. Framing mobile collaborations and mobile technologies. In *Wireless World: Social and Interaction Aspects of Wireless Technology,* B. Brown, N. Green, and R. Harper, Eds. Springer-Verlag, London, 2001..

[Clark, 1994] Clark, H. "Managing Problems in Speaking," *Speech Communication 15,* 3-4 (December 1994), 243-250.

[Cohen, 1995] Cohen, P.R. and S.L. Oviatt, The role of voice input for human-machine communication. *Proceedings of the National Academy of Sciences*, **92**(22): p.9921-9927, 1995.

[Cole, 1980] Cole, Ronald A., Perception and production of fluent speech, Hillsdale: Lawrence Erlbaum Associates, 1980.

[Coutaz *et al*., 1993] J. Coutaz, L. Nigay, D. Salber, J. Caelen: The MSM Framework: A Design Space for Multi-Sensory-Motor Systems, InterCHI'93 Workshop on Multimedia &Multimodal Systems, Amsterdam, The Netherlands, 1993

[Decia and Trecodi, 1997] Decina, M. and Trecordi, V. "Convergence of telecommunications and computing to networking models for integrated services and applications", ' *Proceeding of the IEEE,* vol. 85, no. 12, pp. 1887-1914, Dec. 1997

[Demasco and McCoy] Patrick Demasco and Kathleen McCoy. Generating text from compressed input: An intelligent interface for people with severe motor impairments. Communications of the ACM, pages 68-78, May 1992.

[Dey, 1997] A. K. Dey, L.D. Catledge, G.D.Abowd, C. Potts., "Developing voice-only applications in the absence of speech recognition technology", Technical Report GIT-GVU-97-06, GVU Center, Georgia Institute of Technology, March 1997, Submitted to DIS'97.

[Edwards *et al*, 2001] Edwards, A. D. N, Carey, K., Evreinov G.E., Hammarstrom, K., Raskind, M. Information and Communication Technology in Special Education: Analytical Survey. Moscow: Unesco, Institute for Information Technologies in Education (IITE), 2001, pp. 15-17.

[ETSI, 1996] European Telecommunications Standard Institute, GTS *GSM 01.02,* European Telecommunications Standard Institute, fifth edition, Mar. 1996.

[Fasbender *et al.*, 1999] Fasbender; A., Reichert; F., Geulen; J., Hjelm; J. And

Wierlemann, T. "Any network, any terminal, anywhere," *IEEE Personal Communications,* pp. *22-30,* Apr. 1999.

[Fraser *et al.,* 1990] Fraser Shein, Nicholas Brownlow, Jutta Treviranus, and Penny Pames. Climbing out of the rut: The future of interface technology. In Beth Mineo, editor, Augmentative and Alternative Communication in the Next Decade, Applied Science and Engineering Laboratories. Wilmington, DE, March 1990, pages 36-39.

[Fraser *et al.*, 91] Fraser, Norman M., and G. Nigel Gilbert. "Simulating Speech Systems," Computer Speech and Language, Vol. 5, Academic Press Limited, 1991.

[Freud, 1955] Freud, S., Totem and taboo, In J. Strachey (Ed. And Trans.), The standard edition of the complete psychological works of Sigmund Freud (vol. 13), London: Hogarth Press.

[Goel, 1995] Goel, V., *Sketches of Thought*. Cambridge, MA: The MIT Press, 279, 1995.

[Gould and Lewis, 1983] Gould, J.D., and C. Lewis, Designing for Usability -- Key Principles and What Designers Think, in Proceedings of ACM CHI'83 Conference on Human Factors in Computing Systems. p. 50-53, 1983.

[Grinter *et al.,* 2001] R. Grinter, GRINTER and M. Eldridge, why do tngrs luv 2 txt msg? In *Proceedings of the Seventh European Conference on Computer Supported Cooperative Work* (Sept. 2001, Bonn, Germany). 219-238.

[Hartson and Gray, 1992] H.R. Hartson, P.D. Gray: Temporal Aspects of Tasks in the User Action Notation, Human-Computer Interaction, 1992, VO1.7, pp. 1-45.

[Helyar, 2001] V. Helyar, Usability of portable devices: the case of WAP. In *Wireless World: Social and Interactional Aspects of Wireless Technology,* B. Brown, N. Green, and R. Harper, Eds. SpringerVerlag, London, U.K., 195-206.

[Huang, 2000] X. Huang and et al., "MiPad: A Next Generation PDA Prototype," ICSLP, Beijing, China, 2000.

[Hughes *et al.,* 1987] Hughes, The evolution of large technological systems. In *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology,* MIT Press, Cambridge, MA, 1987, 51-82.

[Hutchins, 1995] E. Hutchins, *Cognition in the Wild*. MIT Press, Cambridge, 1995

[Kate, 1980] J.H. ten Kate, E.E.E. Frietman, F.J.M.L. Stoel, and W. Willems. Eye controlled communication aids. Medical Progress through Technology, 8: 1-21, 1980.

[Kazi, 1995] Zunaid Kazi, Marcos Salganicoff, Matthew T. Beitler, Shoupu

Chen, Daniel Chester, and Richard Foulds. Multi-modal user supervised interface and intelligent control (MUSIIC) for assistive robots. In IJCAI-95 Workshop on Developing Al Applications for People with Disabilities, Montreal, Quebec, IJCAI, 1995, pages 47-58.

[King and Iacono, 1989] KLING, R. AND IACONO, S. The Institutional Character of Computerized Information Systems. *Office: Techno. People 5*, 1, 1989, 7-28.

[Klemmer et al., 2000] Scott R. Klemmer Anoop K. Sinha Jack Chen James A. Landay Nadeem Aboobaker Annie Wang Suede: a Wizard of Oz prototyping tool for speech user interfaces, Publisher ACM Press New York, NY, USA, 2000, Pages: 1 - 10.

[Koons, 1993] David B. Koons, Carlton J. Sparrell, and Kristinn R. Thorisson. Integrating simultaneous input from speech, gaze, and hand gestures. In Mark T. Maybury, editor, Intelligent Multimedia Integrates, AAAI Press I the MIT Press, Menlo Park, CA, 1993, pages 257-276.

[Law, 1987] J. LAW, Technology and heterogeneous engineering: the case of Portuguese expansion. In *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*. MIT Press, Cambridge, 1987, MA, 111-134.

[Ling and Hyper, 2002] Ling, R. and Yttri, B. Hyper-coordination via mobile telephones in Norway. In *Perpetual Contact: Mobile Communication, Private Talk, and Public Performance*, J. Katz and M. Aakhus, Eds. Cambridge University Press, Cambridge, U.K., 2002.

[Marconnay, 1993] P. De Marconnay, J. Crowley, D. Salber: Visual Interpretation of Faces in the NEIMO Multimodal Test-Bed, IJCAI 93 Conference, Chamb&y, France.

[Marx *et al.*, 1996] Marx, M., and Schmandt, C. MailCall: Message Presentation and Navigation in a Nonvisual Environment. In Proceedings of CHI ‘96 (Vancouver, Canada, April 1996), ACM Press, 165-172.

[McNeill, 1992] D. McNeill, Hand and Mind: What Gestures Reveal about Thought, University of Chicago Press, Chicago, 1992

[Messerschmitt,1995] Messerschmitt, D.G. "The convergence of telecommunications and computing: What are implications today?", *Proceeding of the 1EEE*, vol. 84, no. 8, Aug. 1996, pp. 1167-1186.

[Nielsen and Mack, 1994] Jacob Nielsen and Robert L. Mack, “Usability Inspection Methods,” John Wiley and Sons, Inc. 1994

[Nigay and Coutaz, 1993] Laurence Nigay and Jo/“elle Coutaz. A design space for multimodal systems: Concurrent processing and data fusion, INTERCHI ’93 Conference Proceedings, ACM Press, 1993, pages 172-178.

[Nigay *et al.*, 1991] Nigay L. and Coutaz J. Building User Interfaces: Organizing Software Agents. In Proc. ESPRIT91 Conference (Bruxelles, Nov. 1991), pp. 707-719.

[Oviatt et al., 1992] Oviatt, S., P. Cohen, M. Fong, and M. Frank. A rapid semi-automatic simulation technique for investigating interactive speech and handwriting. In Proceedings of the *International Conference on Spoken Language Processing*. Banff, Canada, October 1992.

[Palen and Salzman, 2002] L. Palen and M. Salzman, (to appear 2002), *Voice-Mail Diaries for Naturalistic Data Capture under Mobile Conditions.* To appear in *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW '02, New Orleans, LA).

[Palen et al., 2000] L. Palen, M. Salzman and E. Youngs, Going wireless: behavior and practice of new mobile phone users. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (CSCW '00, Philadelphia, PA). ACM Press, New York, NY, 2000, pages 201-210.

[Palen, 2000] Palen, L., Salzman, M., and Youngs, E. Going wireless: Behavior and practice of new mobile phone users. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'00*) (Philadelphia, Dec. 2-6). ACM Press, New York, 2000, 201-210.

[Piaget, 1967] Piaget, J., six psychological studies, Random House, New York.

[Portnoff, 1978] M. R. Portnoff. Time-Scale Modification of Speech Based on Short-Time Fourier Analysis. PhD thesis, MIT, April 1978.

[Richard, 1980] Richard Foulds. Communication rates for non-speech expression as a function of manual tasks and linguistic constraints. In Proceedings of the International Conference on Rehabilitation Engineering, pages 83-87, Toronto, Canada, June 1980.

[Rieman, 1993] J. Rieman, The diary study: a workplace-oriented research tool to guide laboratory efforts collecting user-information for system design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (INTERCHI '93, Amsterdam, The Netherlands). ACM Press, New York, NY, 1993, pages 321-326.

[Salber and Coutaz, 1993] D. Salber and J. Coutaz: A Wizard of Oz Platform for the Study of Multimodal Systems, InterCHI 93 Conference Adjunct Proceedings, Amsterdam, the Netherlands, 1993.

[Sherry, 2001] Sherry, J., and Salvador, T. Running and grimacing: The struggle for balance in mobile work. In *Wireless World: Social and Interactional Aspects of Wireless Technology,* B. Brown, N. Green, and R. Harper, Eds. Springer-Verlag, London, 108-120, 2001.

[Silbernagel, 1979] D. Silbernagel, Taschenatlas der Physiologie. Thieme, 1979

[Silfverberg et al., 2000] SILFVERBERG, M., MACKENZIE, S., AND KORHONEN, P. 2000. Predicting text entry speed on mobile phones. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI '00, The Hague, The Netherlands). ACM Press, New York, NY, 9-16, 2000.

[Suchman, 1987] L. A. Suchman, *Plans, and Situated Actions: The Problem of Human Computer Communication.* Cambridge University Press, New York, NY, 1987.

[Taylor and Harper, 2002] A. Taylory and R. Harper, Age-old practices in the "New World": A study of gift-giving between teenage mobile phone users. In *Proceedings of the Conference on Human Factors in Computing Systems* (CHI '02, Minneapolis, MN). ACM Press, New York, NY, 2002, 439-446.

[Townsend, 2000] Townsend, A. Life in the real-time city: Mobile telephones and urban metabolism. *J. Urban Tech. 7,* 2, 85-104, August 2000.

[Treviranus et al., 1991] J. Treviranus, F. Shein, S. Haataja, P. Pames, and M. Milner. Speech recognition to enhance computer access for children and young adults who are functionally non-speaking. In Jessica J. Presperin, editor, Proceedings of the Fourteenth Annual RESNA Conference, Washington, DC, RESNA Press, pages 308-310, 1991.

[Verton, 2001] Verton, D. Staying in touch through two hours of hell. Computerworld, September 14, 2001.

[Wagner, 1990] Wagner, A., Prototyping: A Day in the Life of an Interface Designer, in *the Art of Human-Computer Interface Design*, B. Laurel, Editor. Addison-Wesley: Reading, MA. p. 79-84, 1990.

[Walker et al., 1997] M. A. Walker, Litman D., Kamm C. and Abella A. PARADISE: A Framework for evaluating Spoken Dialogue Agents, in Proceedings of ACL '97 (Madrid, Spain, July 1997), MIT Press.

[Walker et al., 1998] M. A. Walker, J. C. Fromer, and S. Narayanan., Learning

[Walker et at., 1997] M.A. Walker, Hindle D., Fromer J., Di Fabbrizio G, and Mestel C., "Evaluating competing agent strategies for a voice email agent", In Proceedings of the European Conference on Speech Communication and Technology, EUROSPEECH97, 1997.

[Weilenmann and Larsson, 2001] A. Weilenmann and C. Larson, Sharing the mobile: mobile phones in local interactions. In *Wireless World: Social and Interactional Aspects of Wireless Technology*, B. Brown, N. Green, and R. Harper, Eds. Springer-Verlag, London, U.K., 2001, pages 92-107.

[Whittaker *et al.*, 1995] Whittaker, S., Hyland, P. and Wiley, M. Filochat: Handwritten Notes Provide Access to Recorded Conversations. *In*

CHI '94, pp. 271-277. ACM, 1994.

[Wtifelman *et al.*, 1993] Stifelman, L.J., Arons, B., Schmandt, C., and Hulteen, E.A. VoiceNotes: A Speech Interface for a Hand-Held Notetaker. In Proceedings of INTERCHI '93 (Amsterdam, The Netherlands, April 1993), ACM Press, 179-186.

[Yankelovich *et al*, 1994] Yankelovich, Nicole and Eric Baatz. "SpeechActs: A Framework for Building Speech Applications," *AVIOS '94 Conference Proceedings*, San Jose, CA, September 20-23, 1994.

Optimal Dialogue Strategies: A Case Study of a Spoken Dialogue Agent for Email. In *Proceedings of the 36th Annual Meeting of the Association of Computational Linguistics, COLING/ACL 98, 1998.*