INTERNATIONAL JOURNAL OF SPEECH TECHNOLOGY 4, 227–240, 2001

Voice Portals—Where Theory Meets Practice

ERICA L. GOLDMAN, ERIN PANTTAJA, ANDY WOJCIKOWSKI AND ROBERT BRAUDES
Comverse, 10 Fawcett Street, 4th Floor, Cambridge, MA 02138

Received November 6, 2000; Revised March 21, 2001

Abstract. This paper describes the underlying aspects of networked voice portal applications with a focus on a natural language voice user interface. The paper provides a detailed description of a representative commercially deployed personal voice portal solution and reviews results from speech recognition testing of one specific voice portal application—voice-activated dialing. It offers several suggestions on network-based voice portal design in general, and specifically, voice-activated dialing (VAD) applications.

Keywords: voice portal, voice-activated dialing, voice-controlled voice mail, voice user interface, multimodal

I. The Importance of Networked Voice Portals

Revolutionary changes in communications are drawing significant attention to Automatic Speech Recognition (ASR) technology. Three changes are particularly important:

- The emergence of the Internet as a primary reservoir of information, content, entertainment, and commercial transactions;
- The tremendous increase in the availability and popularity of access devices (wireless, wireline, pagers, PCs, handhelds, laptops, Internet appliances); and
- The emerging need for a personalized experience in accessing content or communicating.

Today, most applications accessing the Internet or delivering messaging services require the use of traditional access methods like a Web browser with some type of pointing device or a DTMF (Dual-Tone Multi-Frequency) keypad. This situation is not optimal for the user because:

- Memorization and cognitive load of phone numbers, menu commands, and sequences make many applications cumbersome and inflexible;
- Safety is increasingly becoming a major concern with wireless phone use, as a user may become distracted while driving or performing other tasks;
- Physical disabilities may make using a traditional access method impossible or difficult;
- Different applications, tasks, and environments require a choice of input devices and modes; and
- A telephone keypad is not a suitable input modality for a rich Web browsing experience, just as a mouse is not suited to entering numbers.

Alternatives are needed that combine the idea of a "portal" with more natural access methods such as voice, allowing a user to utilize an integrated set of services in a more flexible, personable manner.

What Is a "Portal"?

In today's information age, a portal is an Internet site provided to visitors which serves as a gateway to other sites on the Internet. Portals generally provide search engines, an address book, e-mail, e-Commerce, personalized home pages, Internet browsing, and chat and instant messaging. In general, a portal may support a variety of input and output modalities that can be auditory, visual, or even tactile. Output media can include graphics, text, speech, or real-time streaming audio. A portal may also be used in a variety of different user environments such as an office, a car, or a street.

A personal *voice* portal brings the portal concept to carrier-grade telephony networks, enabling the end user to access networked messaging and information services, Internet content—such as stock quotes and news—and services from any wireless or wireline handset. New advanced services like Internet browsing, which are difficult to negotiate with a telephone keypad, are now being offered because of recent improvements in natural language speech recognition and voice user interface (VUI) design and development. It is important to understand that a voice portal incorporates two different types of networks: 1) a telephony-based voice network, and 2) a data network, or the IP-based Internet. The convergence of these two networks, which provides voice accessibility to the rich content of the Internet, also requires a robust, carrier-grade infrastructure.

Simply stated, the personal voice portal is the application of a new type of interface that can transfer the attributes of the Web to any telephone. Network service providers derive new sources of revenue from Internet access charges, advertising, e-commerce transaction fees (Lucente, 2000), and advanced intelligent services, while 1) creating an identity by virtue of innovative service offerings, and 2) increasing the ability to attract and retain customers.

Why Voice?

Natural speech is the modality used most often when communicating with other people. This makes it easier for a user to learn the operation of voice-activated services. As an output modality, speech has several advantages. First, auditory input does not interfere with visual tasks such as driving a car. Second, it allows for the easy incorporation of sound-based media such as radio broadcasts, music, and voice mail messages. Third, advances in text-to-speech (TTS) technology mean that text information can be transmitted easily to the user.

Natural speech also has advantages as an input modality, allowing for hands-free and eyes-free use. With proper design, voice commands can be created which are easy for a user to remember. These commands don't have to compete for screen space. In addition, unlike keyboard-based macros (e.g., ctrl-F7, or the use of DTMF), voice commands can be inherently mnemonic ("United Airlines"), obviating the necessity for hint cards. Speech can be used to create an interface that is easy to use and requires a minimum of user attention. It can be used by the blind or those unable to use tactile interfaces (Schmandt, 1994).

Careful interface design can be used to heighten the advantages of a voice user interface. By speaking simple and intuitive requests, through the use of a conversational voice user interface, subscribers avoid hierarchical menus and decision trees and directly access the service they want or shift from one service to another. If rich grammars are used, including variants of each command, the resulting VUI is not difficult to navigate. These commands contribute to the "ease-of-use" of the system; the words expressed are usually equivalent to the functionality requested.

One of the most user-friendly aspects of the natural language user interface is its flexibility in supporting various usage patterns. Users are able to act immediately and directly on the content of messages, conversations, and information. There is no predetermined "call flow," but contexts for applications are retained as users make different requests. For example, while listening to voice mail, a user might shift to another application by telling the system to "*Check my portfolio*." She might then ask for personal news clips, or if she asks to hear the next message, the system will have kept track of what message she was listening to—thereby retaining the vital information needed for an intelligent flow of information—without any extra work on her part.

Additionally, with the explosion in the use of portable devices comes an evolution to newer, more compact models, making physical space a premium. A verbal interface allows the user to eschew bulky equipment and avoid some of the problems of limited screen displays on wireless WAP (Wireless Access Protocol) phones or handheld organizers. Also, natural speech enables users to have a more personalized and customized experience; for example, "*Call Jim on his cell phone*" is more personal than requiring keyed-in numbers or commands. And finally, from a safety and ease-of-use perspective, natural speech allows a user to interact with the required service regardless of the environment. For instance, when the use of hands or eyes is limited, natural speech may be the only good alternative.

As in all user interfaces, good design can make the difference between an enjoyable user experience and a system that is difficult to navigate (Balentine and Morgan, 1999). If an end user cannot remember the commands or gets lost in the system hierarchy, the features available no longer matter, as they are unusable. This is entirely separate from the recent improvements in speech recognizers. People do not speak with one hundred percent accuracy, and phone conversations between two people usually involve error correction (Shattuck-Hufnagel, 1982). Thus error correction dialogues need careful design (Oviatt, 2000). Internationalization and localization of systems require attention to cultural and linguistic differences among target audiences (Nass and Gong, 2000).

What Network Services Are Better Voice-Enabled? Voice, of course has limitations, and is not the best modality for all services. Long documents, for example, are difficult to understand using current text-to-speech systems. Editing is easier using a visual interface. Maps are inherently visual information, though interfaces can be created which use voice to convey directions. CAD programs might benefit from the use of a tactile interface. In addition, it may not be possible, legal, or appropriate to use voice in certain public situations including restaurants, theaters, places of worship, or certain modes of public transportation. Voice recognition accuracy can also be limited by the amount of background noise and variations in volume or inflection (Lamel et al., 2000).

In spite of these limitations, there is a variety of services that can be voice-enabled, ranging from business application-specific services such as call centers (e.g., stock market transactions) to browsing the Web by voice. There is a variety of categories of voice-assisted services. Two major classifications are those for accessing data, which traditionally operate over the IP network, and those for control and management of data and services, which are generally found on the telephony network. These two classes of service tend to have different types of interfaces.

A critical element is the need for an integrated set of services—not just access to Internet and other services through the voice portal—but the ability to do so through a single, consistent, flexible voice user interface that allows the user to have complete control. A single interface is easier to learn. It also makes setup and provisioning easier, providing a single log-in rather than requiring separate access to different services. Additional benefits include the ability to provide new services rapidly and without costly re-engineering.

The target audience for a voice portal can be any group with access to telephones. This research focuses on systems for adult speakers of American English. However, the design ideas, if not the particular solutions presented here, are valid for other languages, cultures, and markets.

Results from internal market studies and focus groups undertaken this year in the United States and Europe suggest end users have a keen interest in:

- A hands-free voice interface to initiate calls;
- Voice and e-mail message management through a voice interface with real-time text-to-speech;
- Calendar and personal organizer management through a voice interface;
- Location-based transactions and browsing; e.g., "*Tell me where the nearest movie theater is*," driving directions, traffic, etc; and
- Multimedia display of content on a wireless phone (voice, text and graphics).

The design process involves a variety of different roles. User interface experts design an optimal interface with the cooperation of language experts who are native speakers of the target language. Once a prototype has been developed, a cycle of testing begins, drawing subjects from the system's target audience. Flexibility built into the system allows VUI designers to redesign the interface taking into consideration information derived from testing. This information helps designers to balance theoretical and grammatical constraints with usability.

II. A Voice-Enabled Solution Example

The following section outlines a practical implementation of a personalized voice portal called Tel@GO. This section focuses on key technologies and components, and follows with an analysis of laboratory results obtained from testing one specific service, voice-activated dialing (VAD), that is currently deployed in a nationwide network.

The voice portal currently includes the following applications:

- Voice browsing of the Web using a set of voice "bookmarks" which can be personalized;
- Voice-activated dialing for voice-controlled call initiation;
- Personal and system address books with personal and shared directories accessible from any telephone; and
- Voice control of messages from any telephone.

Solution Overview

Utilizing advanced natural language speech recognition technology, this personalized voice portal enables individuals to use their voices to access personal information and Web-based services and content from any wireless or wireline telephone. The portal provides access to Internet-based services such as Web browsing for news, sports, weather, stock quotes, Internet radio, and any other carrier-specified information; places voice-activated calls to contacts in a Personal Address Book (PAB); creates, retrieves and manages voice, fax, and e-mail messages; and engages in mobile e-commerce.

The voice portal is a complete hardware and software solution, incorporating advanced human factors and user interface design, a suite of voice-driven applications, and best-of-breed speech recognition engine independence. This open standards-based solution is delivered on a carrier-grade platform, and offers tight integration with standards-based voice mail platforms.

Figure 1 illustrates the networked environment in which a voice portal operates. A subscriber may access the system through either a wireless device or a landline phone.

Architecture

There are two generic types of voice portals—"in-network," which are tightly integrated into the service provider's networks; and "out-of-network," which are loosely integrated. This paper concentrates on in-network solutions. Figure 2 indicates the elements that are typically found in an in-network personal voice portal.

System Areas

The voice portal architecture can be divided into five logical sections—the Voice User Interface, the Applications and Services, the Platform that provides application-independent services, the Service Provider Interfaces (SPI), and the Web Graphical User Interface.

Voice User Interface. The voice user interface is the primary interface to the voice portal for the end users. It enables access to the services and information provided by the voice portal. The VUI provides a consistent, flexible natural language user interface across all applications. The flexibility is built into the voice portal through adaptation of the user interface to

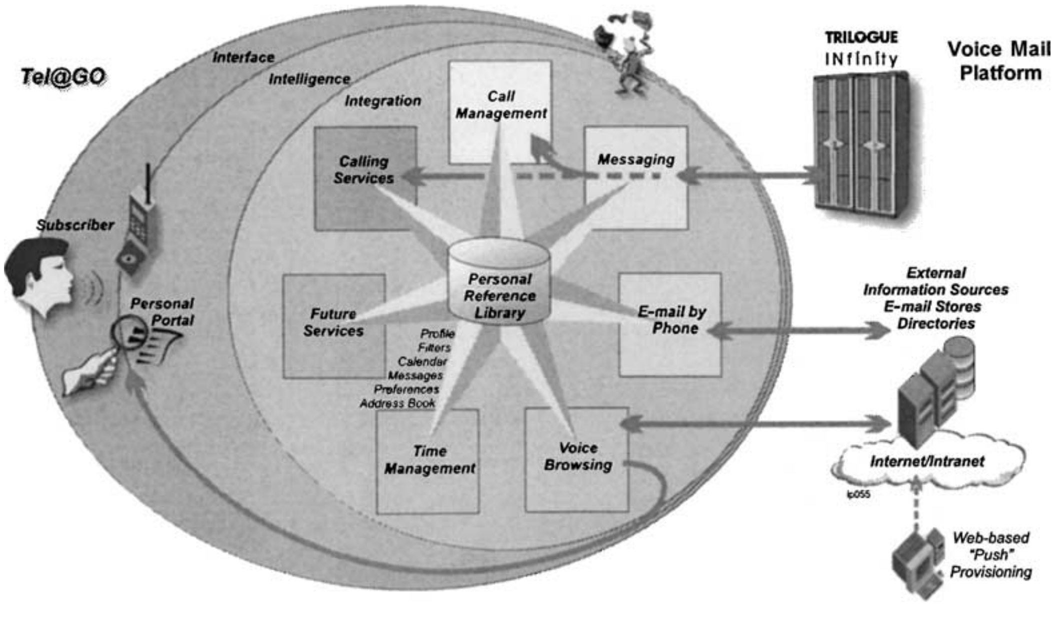

Figure 1. The voice portal environment.

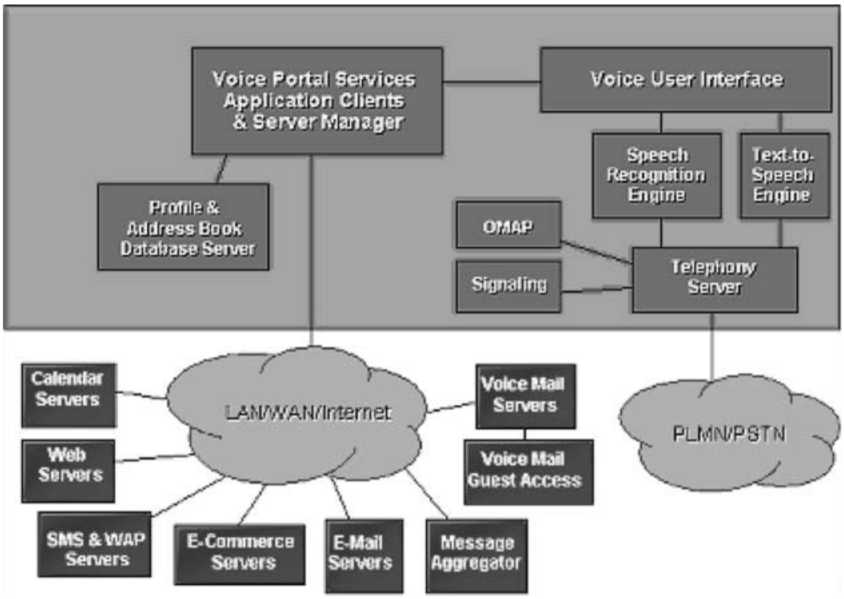

Figure 2. The elements of an in-network voice portal solution.

local languages, speaking patterns, syntax, and cultural norms. However, for implementation efficiency, the underlying applications should be independent of language and culture. Therefore, there should be a layer that translates the utterances received by the VUI into a representation that can be used by the applications. This layer should be reusable across languages. We call the interface layer between the VUI and the applications the Context Application Programming Interface, or CPI, as shown in Fig. 3.

There are two types of grammars used by the VUI. Static grammars are compiled into the system before it is delivered to a customer and are the same for all end users. Dynamic grammars allow individual users and administrators of the system to add their own grammars, such as entries in the PAB. These are loaded into the recognition engine on a per-call basis once the subscriber has been identified. This identification is usually accomplished by the platform, based on the phone number associated with the incoming call. However, when this is not available, the VUI may prompt the user for an account number and password.

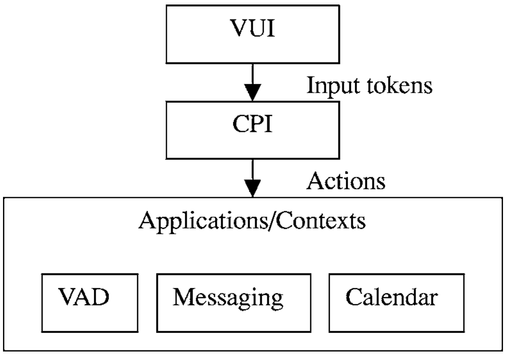

Figure 3. Division between the voice user interface and applications.

Once the subscriber has been identified, the VUI begins the conversation by playing a prompt and listening for input from the caller. Speech input is directed to the ASR engine, which analyzes the speech based on the grammar specified for the caller, while DTMF tones are analyzed by the telephony hardware. This input is returned to the VUI, which passes the input tokens based on the utterances and the DTMF tones to the CPI. The CPI determines the appropriate application or context to handle the input and issues a series of actions to gather the information requested by the user. This information is then passed back to the VUI for presentation. This presentation could make use of pre-recorded prompts, audio files, real-time streaming of audio from external sources such as the Internet, or generated speech from a text-to-speech engine. The platform sends the data over the telephony interface.

Voice Portal Services and Applications. The core services offered by Tel@GO are in this module: Voice-Activated Dialing using Personal and System Address Books; e.g., "*Call John Smith on his mobile*;" Voice Browsing; e.g., "*Tell me the business news*;" and Voice Control of Voice Mail; e.g., "*Play my messages from Susan Smith*."

Many potential applications for personal voice portals were discussed above. These include voice access to messaging systems (such as voice mail and e-mail), calendars, and information from the Internet and other sources, along with voice-activated dialing and other network control services. Applications may be classified into two types—those that are compiled and tightly integrated into the system, and those that are interpreted and more loosely integrated, using a language such as VoiceXML. The main difference between the two classes from a user interface perspective is that the grammar from the first class is integrated with the system grammar, making it easier for the user to seamlessly move among different applications. VoiceXML-based applications, which can be added to the system at any time by adding a new Universal Resource Locator (URL) to the system, incorporate a self-contained grammar, separate from the system grammar. This allows new applications to be rapidly deployed, with only system configuration modifications required. This contrasts with an upgrade to the voice portal software for compiled applications. The cost of the simplicity of the VoiceXML applications is the lack of seamless integration (Lucas, 2000).

Platform. The platform layer is responsible for providing application-independent services to the voice portal. These services include telephony access and control, speech recognition, and text-to-speech translation.

The telephony server (see Fig. 2) is responsible for the signaling and data access using the telephony network. A wide variety of telecommunications protocols are implemented throughout the world, and all of these must be supported. For example, many countries have their own variants of the Signaling System Number 7 (SS7) protocol, while others use ISDN protocols. In North America, T-1 lines are used, while Europe and South America use E-1 lines, and Japan uses J-1 lines. When placing outgoing calls from the system, some central office switches allow the outgoing call to be initiated and then returned to the telephony network, which connects the outgoing call to the incoming call. This integration with the telephony network, which utilizes "intelligent network" protocols normally based on SS7 signaling, has many varieties such as INAP, Release Trunk, and Return-to-Pivot. At the other end of the spectrum, the incoming and outgoing calls are both maintained by the voice portal. This mode is often called "tromboning," as drawings of the lines looping through the voice portal platform resemble the musical instrument. A robust voice portal implements a wide variety of these protocols, and the platform should hide the protocol in use from the applications. The telephony server is also responsible for recognizing DTMF tones and reporting them back to the VUI.

Voice portals need to support multiple languages, often on the same platform. This means that the architecture must allow for multiple grammars, and potentially multiple speech recognition engines and multiple text-to-speech engines. These engines may come from different vendors, and in fact different languages on a single platform may require multiple ASR or TTS vendors on a single platform.

ASR technology is used to translate spoken utterances into tokens, while TTS technology is used to play back information from the Internet that is only available in text form, such as traffic information services. The TTS engine is also used to play back the contact names in the subscriber's Personal Address Book that have been entered by the subscriber using text enrollment (described below). In this case a voice recording of the name is not available. The ASR technology is based on phonetic models of the languages, using large vocabulary (over 500,000 phrases), speaker-independent technology.

Service Provider Interface (SPI). The SPI communicates between the voice portal service application software and the server databases that are required to support the applications. There are many components within a voice portal that may be provided by third parties. Examples of these components include the speech recognition and text-to-speech engines, subscriber databases, personal address books, and external messaging servers that use either standard protocols such as POP3 or IMAP4, or that use proprietary protocols, authentication servers, and notification servers. When interfacing with these components, a voice portal should use open standards whenever possible to allow easy extensibility to new servers as they are integrated into the overall architecture. In addition, service providers usually have "home-grown" back-office systems for operations, maintenance, administration, and provisioning (OMAP) services. OMAP includes services such as billing systems and formats, statistics aggregation and reporting, and provisioning systems for adding, deleting, and modifying the profiles of the voice portal users.

The architecture of voice portals must be flexible to support this wide variety of external interfaces. This flexibility is often provided by a set of SPIs, which allow the same application set and platform to "plug and play" with the third-party products. The SPIs work with a broad range of speech recognizer and text-to-speech applications, implementing best-of-breed technology with the flexibility to change when underlying technologies change, without requiring modifications to each application.

Another approach used in conjunction with the SPIs is database synchronization. Using the example of a PAB, the voice portal adds data such as voice signatures to the typical PAB schema. Voice data require a significant amount of data, typically ranging from 16 to 32 kilobits per second of speech, and so are not normally found in address books. Modifying an existing PAB to incorporate this data is often not a viable option; similarly, modifying existing applications to use the voice portal address book is also usually not viable. The solution to this problem is to synchronize the two address books on a short periodic basis, or on demand using database triggers.

Web Graphical User Interface. Not all user interaction with a voice portal is through the telephone. Certain tasks, such as adding names to a PAB, are sometimes accomplished more easily using a graphical interface over the World Wide Web. Therefore, voice portals usually incorporate both a voice interface and a Web graphical interface. The Web interface is also a convenient way for end users to modify some of their personal attributes such as name and address, or even sign up for the voice portal service in the first place, in a manner that is significantly less expensive for the service provider than having the user contact a customer service representative.

In addition to ease-of-use, current large-vocabulary speech recognition engines usually have greater recognition accuracy from text-enrolled entries than from voice-enrolled entries, which may be counterintuitive. The ASR engines use different techniques for analyzing the patterns of voice data than for textual data, which leads to this situation. Another practical consideration is that the data storage requirement for text entry of data is significantly less than that required for voice entry. As the data storage unit is often the most expensive component of a voice portal, service providers usually offer a much larger number of text-based entries in a PAB than voice entries.

Example—Voice-Activated Dialing

Putting everything together, the following steps occur when a user issues the utterance "*Call Mom at home*." The following assumptions are made to reduce the complexity of the example:

1. The system has already been initialized;
2. The static grammar has been loaded;
3. The user calls from a "trusted" phone, so the system can identify the caller from the Caller ID;
4. Based on the user identification, the user's dynamic grammar, including the Personal Address Book entries, has already been loaded into the recognizer;
5. The system "trombones" outgoing calls; and
6. The user's address book contains an entry for "Mom."

The system flow to the user command is as follows:

1. The VUI plays a welcome prompt, branded with the Service Provider's name;
2. The user says "*Call Mom at home*";
3. The telephony card receives the energy (sounds from the user) and informs the VUI;
4. The VUI passes a pointer to the speech buffers containing the utterance to the speech recognition SPI;
5. The recognizer SPI requests recognition service using the commands appropriate to the particular recognizer being used for that subscriber;
6. The recognizer passes the recognized utterance ("*Call mom at home*") along with a confidence score, back to the recognizer SPI. Note that elements of this phrase are taken from both the static and dynamic grammars;
7. The SPI passes the tokens back to the VUI;
8. The application evaluates the confidence score and determines that it is higher than the appropriate threshold;
9. The VUI passes the tokens to the CPI;
10. The CPI analyzes the tokens;

11. The CPI requests the home telephone number for "Mom" from the PAB SPI;
12. The PAB SPI retrieves the information from the PAB, and passes it back to the CPI;
13. The CPI instructs the telephony server to place an outgoing call to the telephone number retrieved from the PAB;
14. The call is connected, and the telephony server bridges the incoming port (the user) with the outgoing port (Mom); and
15. When the call is completed and Mom hangs up or the user presses a DTMF escape sequence, the telephony server re-connects the VUI to the port connected to the user.

III. Testing a Voice-Activated Dialing Application

A very important part of developing VUIs for voice portals is extensive testing. Some informal testing is done by keeping logs of voice portal use by engineers or trial users. These logs consist of audio recordings of the users' speech as well as a text file listing recognition results, prompts played, and actions performed by the system. These logs can be examined to find trends. More rigorous testing can be done by collecting a large number of users and calls and transcribing all of the logs for analysis. In addition, specific scenarios may be set up to test the usability of particular features (Glass et al., 2000).

Multiple tests of the voice portal have been performed in several natural and structured environments. One such test was performed on a single core service—voice-activated dialing. The results and methodology of this test are described below.

Creating and Using Personal Address Books

The PAB is a secure, user-managed directory of up to 150 telephone numbers. End users may initially create their address books by simply importing the address file from other personal databases. Additional entries may be made either from a PC via the user's Web interface, or from a telephone handset by voice enrollment. These multiple input modes are important. Voice is more convenient for entering a single name away from the computer, but text enrollment provides better recognition and is the only possibility for batch address book entry. Voice and data entries coexist in the PAB. Once a name has been entered, the end user can execute calls by natural voice command, such as "*Call Susan Jones at home*," "*Call my broker*," or "*Call six one seven, four nine seven, nine eight zero zero*."

The Experiment

When designing a voice user interface, vocabulary choice and grammar structure are of utmost importance. One important issue in the design of a voice-activated dialing system is the question of how users should call people in their address books. This involves selection of a word like "call" or "dial," and the issue of whether the command to call John Jones should simply be "John Jones," without any call command at all (referred to here as "bare names"). This latter option is shorter—and therefore faster—for the user. One purpose of the experiment was to investigate the ramifications on recognition accuracy of utilizing bare names in the grammar. In addition, it tested the accuracy of address book and account maintenance commands.

The System. This set of tests was performed in a service simulation scenario by setting up a data collection computer for subjects to call into. Each subject was given a list of utterances to read. The utterances included voice dialing, address book management, and account maintenance commands. The goal of the experiment was to discover which commands had the greatest negative impact on the performance of the system. The utterances were then recognized using several different recognizer grammars, and the results were compared and analyzed for trends.

The Test Grammars. Some examples of grammatical utterances are outlined below. Please see *Appendix A: the complete grammar* for a more comprehensive version of the grammar. The names used in the collection were taken from actual user address books created in a user trial.

1. "Bare names" are utterances of the form "*John Jones,*" in which the user simply speaks a name;
2. Call commands are utterances of the form "*Call John Jones*," "*Call John Jones on his office phone*," or "*Call four nine seven nine eight hundred*" in which a call command and a name or number are specified and a phone type is optionally specified;
3. Address book maintenance commands are of the form "*Delete Cathy Martinez's office phone*." These utterances include a command like delete, add, or

change, a name from the address book, and optionally a phone type;

4. Account maintenance commands are of the form "*Turn expert mode on*"; and
5. Miscellaneous additional commands include "*Help address book*" and "*Goodbye*."

The collected utterances were compared against three different grammar sets:

- G1 included all of the available commands and an address book of 100 names;
- G2 included all of G1 plus an additional 100 address book entries (200 names in all); and
- G3 consisted of G2, with bare name calling and international calling (dialing of numbers beginning in 011 by voice) removed, as well as the address book editing commands altered to require use of the words "phone," "number," or "phone number."

Parameters and Corpus Characteristics. The collected corpus consists of 15,651 utterances (N) collected from 100–150 typical target novice users of the system. The users were divided into groups of men (48% of utterances) and women (52% of utterances). The calls were performed in three different environments: wireline handset with moderate noise (∼35%); earbud in a quiet room (∼33%); and hands-free kit in a car, with no radio and the windows closed, car running, but not moving (∼31%). As the goal of this experiment is a system that can be used in all of these situations, the data were mixed for the analysis.

Evaluation Metric. The metric we chose to use in this particular evaluation was recognition accuracy. With each recognition hypothesis, the recognizer returns a confidence score, which is a value representing the confidence level in the recognition result. We consider that the recognizer has "accepted" when it reports a hypothesis with a sufficiently high confidence level. If no utterance has a sufficiently high confidence level then the recognizer has "rejected." The results were divided into five separate categories:

- True Accepts (TA)
  Accurate acceptance of an in-grammar utterance;
- Substitution (SB)
  Inaccurate acceptance of an in-grammar utterance. For example, this happens particularly in the case of address book names; e.g., the user says "*Call John Jones*" and is recognized as having said "*Call Jim James*," where both John Jones and Jim James are real names from his PAB;
- False Reject (FR)
  Rejection of an in-grammar utterance;
- False Accept (FA)
  Acceptance of an out-of-grammar utterance; and
- True Reject (TR)
  Rejection of an out-of-grammar utterance.

Note that both True Accepts and True Rejects represent the recognizer functioning correctly.

In general, the accuracy measurements (Table 1) are tied to the specific recognizer used and language spoken. However, trends are expected to be generalizable. A speaker-independent grammar-based recognizer was used, which means that the accuracy of recognition of each utterance depends, at least in part, on the size and exact composition of the rest of the grammar. As such, no utterance can be tested in isolation.

Test Results

Bare Names. The numbers show us that disallowing bare names in the grammar greatly increases recognition accuracy. The TA percentage for G3, in which bare names were deleted from the grammar, was 87.3%, higher than either of the other two grammars, even the one with an address book half the size of that used in G3.

In-Grammar Results. In the rest of the discussion, we will be evaluating only in-grammar utterances, so no out-of-grammar statistics will be listed.

Table 1. Overall results.

Grammar	TA%	SB%	FR%	In-grammar	FA%	TR%	Out-of-grammar	N
G1	81.8	12.5	5.7	14639	53.4	46.6	1012	15651
G2	79.9	15.9	5.2	14639	64.5	35.5	1012	15651
G3	87.3	10.2	2.5	11582	60.0	40.0	4069	15651

Table 2. Call commands and bare names.

Grammar	TA%	SB%	FR%	In-grammar
G1	76.3	15.8	7.9	5329
G2	72.0	21.4	6.6	5329
G3	85.4	12.3	2.2	5329

In G2, half of the substitution errors (574 of 1141) were caused by the use of bare names in the grammar. Table 2 shows the results for recognitions of call commands and bare names. Another 455 were instances in which the name or location (e.g., home or office) of the command was incorrectly recognized. These were dominated by instances in which the name was incorrectly recognized. This suggests that in the future more attention should be paid to algorithms for checking names for similarity as they enter the address book, or for confirming user choices. The error rates were higher in G2 because there were more names in the grammar.

The other substitution errors (those not involving call commands and bare names) were distributed evenly among the other system commands. In grammar G3, which had no bare names, there were 657 total substitution errors; 79% of them were recognitions of the wrong name (e.g., recognizing "call Rob Bender" in place of "call Bob Fender").

Expert Mode Results. Prior tests have shown that after a while, users need less feedback in order to use the system. In fact, they may become annoyed by what they feel is excessive instruction. In order to combat this without making the system more difficult to use for the novice user, "expert mode" was created. In expert mode, some of the prompts are shorter and the system requires less user confirmation. However, when creating a grammar for expert mode, a problem arose in that "*Turn expert mode ON*" sounds very much like "*Turn expert mode OFF*." See Table 3 for the test results of subjects using these phrases with "turn" being optional.

The recognition for commands that turned expert mode on or off was very good. Of the 65 substitutions in G3, 15% were misrecognitions of expert OFF for expert ON, and 52% were misrecognitions of expert ON for expert OFF. The rest were distributed among other grammar possibilities. This would suggest that the phrase is a good choice as a user grammar.

Table 3. Expert mode.

Grammar	TA%	SB%	FR%	In-grammar
G1	91.9	6.1	2.0	1526
G2	91.5	6.6	1.8	1526
G3	91.0	6.5	2.5	997

Test Conclusions

Using these tests, guidelines of what works and what does not in this type of user interface were established. Bare names, for example, cause a large degree of error within the system. In this instance, there is a fairly clear tradeoff between creating a slightly more complex interface for the user and creating a system with significantly better recognition. In the case of expert mode, we found that what seemed to be a risky choice (using commands which sounded very much alike) resulted in commands that were recognized with a significant degree of accuracy. One serious problem with this experiment is the fact that subjects read the required utterances. This means their speech was in citation mode, and thus not necessarily indicative of the speech actual users of the system, as read speech is pronounced differently than spontaneous speech.

Other Conclusions

Experience gained in user trials suggests that users much prefer the delivery of Web information as streamed audio rather than text-to-speech. This is because of the human touch of streamed audio as well as the reduced cognitive load on the user. However, text-to-speech may be the only viable option for delivering certain types of information, such as weather reports or e-mail.

Although usually provided with a reference card, users generally do not consult this or a user manual to effectively use the system, and the voice portal was designed with this in mind; system prompts can be used to reinforce preferred grammar items, and careful phrasing can be used to reduce user confusion. For example, in a help prompt about the mailbox functionality, rather than just listing functions that the system supports, the text of the prompt actually gives examples of commands that the speaker can use. Instead of "*With your address book you can look up and call contacts,*" a prompt that says "*Mailbox actions include asking to* '*Look up Mary Jones*,' '*Call John Smith at the office*,' or '*Call 555-1212*,'" actually gives the user an example of exactly what will work. Furthermore, when an end user uses a natural language system to place a call to a person listed in the user's PAB, there are a variety of phrases that could be used, depending on the precise user settings e.g., "*Call John Smith at work,*" "*Dial John Smith at the office,*" "*Call John Smith at his office,*" or even just "*Call John Smith.*" Although there are differences among these phrases, in each case the user has the same end goal in mind, and since all these variations are included in the grammar, the natural language system connects the end user to the office phone of John Smith no matter which phrase was used, exactly as we would expect a live assistant to do.

Equally important in the VUI development process are human factors and design experts who consider each feature of the user interface separately.

- The *Grammar* needs to be created to represent the desired feature set in the appropriate language. In many cases, it needs to include appropriate slang in order to be easily remembered by users. (e.g., "handy" for cell phone in German);
- *Prompts* need to be created which align with a culturally appropriate system voice and personality;
- *Help message contents* need to be translated to the target language, and enhanced to include issues with which the target user population may have difficulties (e.g., in American English, "last message" may refer to either the message just heard or the message in the final position in the queue);
- *Chimes or tones* may be added as earcons or auditory icons to give additional feedback to the user (Gaver et al., 1991; Leplâtre and Brewster, 2000);
- *DTMF command mappings* may be chosen, taking into consideration industry standards and de-facto standards;
- *Numbering plans* must be added to the system. This often involves extensive modification of the system, as phone numbers are broken up in different ways in different parts of the world. ("4 9 7 9 8 0 0 " versus "01 64 64 33 28");
- A *Web user interface* may be required to help with provisioning;
- *Advertising tie-ins* may be required for branding;
- *Handset shortcuts* may be available. (e.g., "*Talk" to get access to portal services); and
- *System Address Books* need to be created which contain all of the numbers to be made available to all of the users of the system. These need to be customized to region and target user population.

Future Research

In the future, more research is needed to examine other issues of grammar choice. For instance, digit-dialing is very important to voice dialing systems and research needs to go into what mechanisms can help users maximize recognition for phone numbers, both the more constrained variety, in which the exact number of digits is known in advance, and less predictable international numbers. Additionally, more analysis of the data from trials like this one should be undertaken in order to draw further conclusions on the design of a grammar from a recognition standpoint. Of course, recognition accuracy of collected speech is only one of many metrics for measuring the quality of a system. Another metric we have used involves the examination of data from actual users of a production system. Instead of having users read commands from a card, users are allowed to interact with the system.

Speech is a medium which has some inherent difficulties not found in visual user interfaces. One of the primary problems is the lack of visual cues. On a desktop machine, for example, end users may want to bring up Web pages. He can open his bookmarks, then scan them looking for the one he wants. In a voice interface, scanning the list of possible commands (all those a user may speak at a given time) is possible, but very slow, since it takes most people a shorter amount of time to read a list than for someone to speak the same list. In addition, if a user wants to listen to all of his options before speaking one (e.g., if he is unsure which is the correct choice), he may also be called upon to remember a command for a long period of time.

This places some rather strong constraints on grammar selection. Commands need to be natural and easy to remember, and the grammar needs to be constrained enough to be learnable, yet flexible enough that a user can easily remember a command without having to memorize a strict list of possibilities. Usability testing gives indications of which commands are easy to remember (and what forms users use when they forget). This sort of testing allows us to broaden the system to match normal patterns of usage. It also helps us to test error handling. In any voice system, there are going to be misrecognitions (Boyce, 2000). Even if the recognition were perfect, there would still be momentary channel corruptions, or user errors ("*call ... um ... John Smy Smith*"). Thus, it is very important to have easy-to-navigate error correction scripts. User testing can help to evaluate these scripts.

Given a large number of transcripts from natural (unscripted) interactions between users and the system, patterns will emerge which can give better feedback on the system than can either focus groups or individual designers. Areas of confusion in the grammar will become apparent, and problems with error correction will show up in either decreased productivity with the system or in user dissatisfaction. The results of these tests facilitate the creation of better voice user interfaces.

IV. Summary

With the evolution of natural language understanding and VUI design, the use of natural voice in delivering communication and Internet-related applications and services will only increase significantly. In fact, natural speech user interfaces will become the de-facto standard for telephone and mobile access. The voice portal solution described in this paper is one example of the practical application of voice. There have been several lessons learned from this deployment:

- The voice user interface must be flexible enough to accommodate a wide range of speakers yet rigorous in its recognition capabilities;
- There must be a smooth integration with an in-network platform in order to manage the convergence of the voice messaging and IP environments; and
- Natural, conversational language is hard. As the test results indicate, "bare names" are more difficult to recognize accurately without context and pose a challenge for future voice interface designers.

In particular, several specific design guidelines have been uncovered including:

- Don't allow bare names;
- Test controversial phrases like "turn expert mode on"—they may work;
- Use one phrase to incorporate several sequential menu choices. This makes for an interface which is easier to use;
- Flatten hierarchical menus;
- Give multiple, predictable options of saying one thing. This reduces a need to rigidly memorize key phrases;
- Spend time to develop good error handling routines ensuring the user does not lose context;
- Reinforce grammar with prompts; and
- Make commands easy to remember or deduce; users don't often use a reference card even if it's available.

Although voice as an access medium will be the wave of the future, several issues remain for conversational applications. Not only does the underlying speech recognition engine need to provide increased accuracy, but the voice portal also needs to improve its ability to understand context. Users will eventually demand the same level of reliability as they take for granted with existing phone networks. This will be a daunting task but is within the grasp of current technology.

Appendix A: The Complete Grammar

A near-comprehensive list of the largest grammar used in the experiment follows, divided into several logical classes. Note that in the accompanying notation examples that "z" stands for possessive, words in square brackets are optional, and words in parentheses that are separated by a vertical bar mean one or the other of the words must be used. The symbol "$" indicates a terminal symbol, and #name represents a name from the address book. (The names used in the experiment were pulled from actual user address books created in a user trial.) In addition, the following "definitions" make the grammar below more condensed and more consistent, so please substitute the elements on the right of the following equations any time an item on the left occurs in the grammar:

- $call = call | dial
- $preposition = at | on | in
- $possessive = the | his | her | their | its
- $phone = phone | number | phone number
- $mobile = mobile | cell | cellular | wireless
- $work = work | office
- $which = home | $work | other | $mobile
- $phone_number = *Note: The full definition of $phone_number is quite complicated and not reprinted here. Simply put, 7, 10, or 11 digit numbers were allowed with few restraints, and had to be spoken as single digits only with the exception of some locations in which the words "hundred" and "thousand" were permissible. International numbers had to begin with "011" and could be nine to fourteen digits in length, inclusive.*

The grammar:

"**Bare names**": for example, "*John Jones.*"

$barename = #name

Call commands: for example, "*Call John Jones on his office phone*," or "*Call four nine seven nine eight hundred*."

$callname = ($call [the] #name [[$preposition] [$possessive] $which [$phone]]) | ($call #name z $which [$phone])
$page = $call #name [$preposition] [$possessive] pager [$phone] | $call #name z pager [$phone] | page #name
$callnumber = $call $phone_number
$redial = redial [last call]

Address book query commands: for example, "*What is Hyatt Long's work number*?"

$lookup = (find | look up) #name [in (my | the) address book]
$whatis = what is #name [z] ($which | pager) ($phone) | what is the ($which | pager) ($phone) of #name
$checkbook = [please] (tell me about my| check [my]) address book [please]

Address book maintenance commands: for example, "*Delete Cathy Martinez's office phone*."

$delete = delete #name z (work|home|office|cell) [$phone]
$delete_name = (delete | remove) #name [from (my|the) address book]
$delete_num = delete #name [z] ($which | pager) [$phone] [from (my|the) address book]| delete the ($which |pager) [$phone] [for #name [from (my|the) address book]]
$edit_default = make [the] ($which | pager) [$phone] the default [for #name]
$add_entry = add (an entry|a name|a person) [to [the|my] address book] [please]
$addnum = add [a | $possessive | #name z] ($which | pager) [$phone] [to (my|the) address book] | add (a|the) [new] ($which|pager) [$phone] [for #name [to (my|the) address book]] | add (a|an) [new] other [$phone] for #name [to (my|the) address book]
$changenum = change [a | $possessive | #name z] ($which | pager) [$phone] [in (my|the) address book] | change the ($which | pager) [$phone] [(of|for) #name])

Account maintenance commands: for example, "*Turn expert mode on*."

$experton = [turn] expert mode on | (switch to | activate) expert [mode]
$expertoff = [turn] expert mode off| (deactivate | stop) expert [mode]

Help commands: for example, "*Help address book*."

$help = help me | what are my options | help [topics | address book | call | dial | redial | page | goodbye | log off | quit | understanding | new user | expert mode | delete | add | change]
$help_web = [please] [tell me [about] my | what is my] web (page | information) [please] | help web [page |information | password] | help password

Miscellaneous additional commands: for example, "*Goodbye*."

$record = record [my] name
$hangup = hang up | goodbye | log off
$cancel = cancel | stop
$sleep = go to sleep
$expertstatus = expert mode

References

Balentine, Bruce and Morgan, David P. (1999). *How to Build a Speech Recognition Application: A Style Guide for Telephony Dialogues*. San Ramon, CA: New York, Enterprise Integration Group, Incorporated.

Boyce, Susan J. (2000). Natural spoken dialogue systems for telephony applications. *Communications of the ACM*, *43*(9):36–43.

Gaver, William W., Smith, Randall B., and O'Shea, Tim. (1991). Effective sounds in complex systems: The Arkola simulation. In *Human Factors in Computing Systems Conference Proceedings on Reaching Through Technology*. LA, USA, pp. 85–90.

Glass, James, Polifroni, Joseph, Seneff, Stephanie, and Zue, Victor. (2000). Data collection and performance evaluation of spoken dialogue systems: The MIT experience. In *Proceedings of the Sixth International Conference on Spoken Language Processing*. October, Beijing, China.

Lamel, Lori, Rosset, Sophie, and Gauvain, Jean-Luc. (2000). Considerations in the design and evaluation of spoken language dialogue systems. In *Proceedings of the Sixth International Conference on Spoken Language Processing*. October, Beijing, China.

Lucas, Bruce. (2000). VoiceXML for web-based distributed conversational applications. *Communications of the ACM*, New York, *43*(9):53–57.

Leplâtre, Grégory and Brewster, Stephen. (2000). Designing non-speech sounds to support navigation in mobile phone menus. In *Proceedings of ICAD'2000*. Atlanta.

Lucente, Mark. (2000). Conversational interfaces for E-commerce applications. *Communications of the ACM*, New York, *43*(9):59–65.

Nass, Clifford and Gong, Li. (2000). Speech interfaces from an evolutionary perspective. *Communications of the ACM*, *43*(9):36–43.

Oviatt, Sharon. (2000). Taming recognition errors with a multimodal interface. *Communications of the ACM*, *43*(9):45–51.

Schmandt, C. (1994). *Voice Communication with Computers: Conversational Systems*. Van Nostrand Reinhold: New York.

Shattuck-Hufnagel, Stefanie. (1982). Three kinds of speech error evidence for the role of grammatical elements in processing. In L.K. Obler and L. Menn (Eds.), *Exceptional Language and Linguistics*. New York: Academic Press.