A Voice Browser Interface for Course Materials

Lisa M. Landgraf
Computer Science Department
Clarke College
llandga@clarke.edu

Abstract

In the last few years, the development of microbrowsers and web appliances has led to a strong interest in ubiquitous computing. For example, some mobile users would like the ability to access the Internet with the same ease of use as a visual browser. The ability to access the Internet via telephone has great potential for exponentially increasing the global use of the Internet.

In the media rich environment of the Internet, vision-impaired people have difficulty using visual browsers. Using voice browsers gives visually impaired users the capability to retrieve information from the Internet by using a telephone and issuing voice commands. No special equipment is required for them to purchase.

Mobile users are another group looking for a safe 'hands-free' browsing environment. A telephone-activated system in which the user must use the dialpad to insert commands has limitations. Using a voice browser with a telephone means the user has virtually an unlimited number of commands and can maneuver in the information space without moving the phone or touching the phone to enter commands.

Non-visual browsers are also known as voice browsers. One definition of a voice browser is a web browser with the capability to generate and/or recognize speech. If a web browser can present a web page in an audio format, then the browser has speech generation capability. If the web browser can interpret spoken input to navigate through some information space, then it has speech recognition capability.

This paper looks at undergraduate students as a possible community of users for a voice-browsing application. Both the traditional and non-traditional college students can be identified as mobile users. If course materials were available by telephone, students would have another alternative to obtaining information about a course that would be accessible 24 hours a day, seven days a week.

Voice-XML (VXML) is a web-based markup language similar to HTML. HTML assumes the user will have a keyboard and mouse for input and a monitor for output. By

using VXML, voice-generation software and a server that recognizes VXML, a voice browsing information space can be created which allows the user to speak commands to jump from page to page. Each page can be rendered by a computer synthesized voice or pre-recorded audio files.

The intent of the paper is three-fold. The first goal is to explore the feasibility of using voice browsers to provide another means for students to obtain information about a course. A second goal of the paper is to determine how much time and effort it requires of an instructor to support a voice-enabled website. The third goal is to experiment with guidelines of speech-based systems in the development of the system. A prototype of a voice-enabled website of course materials has been developed by using the website provided by Be Vocal, Inc. The results of the effort are presented in the paper.

Introduction

In the last few years, the development of microbrowsers and web appliances has led to a strong interest in ubiquitous computing. Some mobile users would like the ability to access the Internet with the same ease of use as a visual browser. The ability to access the Internet via a telephone has great potential for exponentially increasing the global use of the Internet. [5]

[2] defines a voice browser as a web browser with the capability to generate and/or recognize speech. If a web browser can present a web page in an audio format, then the browser has speech generation capability. If the web browser can interpret spoken input to navigate through some information space, then it has speech recognition capability.

Voice-XML (VXML) is a web-based markup language similar to HTML. VXML assumes a voice browser with audio output (computer-synthesized and/or recorded), and audio input (voice and/or keypad tones). [11,3]. By using VXML, voice-generation software and a server that recognizes VXML, a voice browsing information space can be created where the user speaks commands to jump from page to page and each page can be rendered by a computer synthesized voice or pre-recorded audio files.

When developing voice-browser applications, there are several design considerations to think about. In an audio environment, the information is serial and one-dimensional. It must be easy to use, consistent and well structured or users will become discouraged and stop using the application. [8]

Besides the vision-impaired community, mobile users are becoming more interested in voice browsers. Mobile users are looking for a safe 'hands-free' browsing environment. [4] A telephone-activated system in which the user must use the dialpad to insert commands has two problems. First, there are a limited number of commands, due to the limited number of keys on the dialpad and entering a command requires that the phone must be moved away from the face in order to be able to enter the command. But using a voice browser with a telephone means the user has a virtually unlimited number of commands and can maneuver in the information space without moving the phone or touching the phone to enter commands. In the literature, the automobile driver is most often referred to as the mobile user requiring the hands free environment. Other mobile users need to be identified as having legitimate uses for voice browsing.

This project looks at undergraduate students as a possible community of users for a voice-browsing application. Both traditional and non-traditional college students can be identified as mobile users. If course materials were available by telephone, these students would have another alternative to obtaining information about a course that would be accessible 24 hours a day, seven days a week.

Many faculty choose to make their course materials available to students on the Internet. This allows the students the ability to review due dates and descriptions of assignments outside of class. Undergraduate students are often reluctant to talk to instructors. A voice-browsing application for retrieving course materials gives students another way to retrieve materials 24 hours every day. However, it may mean more work for the instructor to prepare the materials.

For this project, a voice browsing application was developed for retrieving course materials. The target audience was an advanced microcomputer applications course for undergraduate students in the Computer Information Systems major. The application was developed using VXML and a free service offered by BeVocal, Inc. Course materials such as the syllabus, assignment descriptions, due dates and the ability to retrieve current standing in the class were all incorporated into the website. Access to the application was accomplished by dialing a toll free number and issuing voice commands as directed to retrieve information.

The intent of the study was three-fold. First it was to explore the feasibility of using voice browsers to provide another means for students to obtain information about a course. A second goal of the project was to determine how much time and effort it would require of an instructor to support a voice-enabled website. The third goal was to adhere to [8] guidelines of speech-based systems in the development of the system and to test some of these guidelines by purposely developing part of the system using pre-recorded audio files and some with voice-synthesis files to determine if users have a preference.

The results indicate that students liked the idea of retrieving *grades* using this application but other parts of the application such as assignment descriptions were not preferable to more traditional ways of retrieving this information. The next section reviews some of the literature about voice browsing, including a discussion of Noonan's design guidelines. The third section describes the prototype development and usability studies that were conducted. The fourth indicates some suggestions for improvement as well as further research possibilities.

Review of Literature

[7] names three types of hyperlinks found on web pages. First, there are structural navigation links that outline the structure of the information space. The second type of hyperlink is the associative link within the content of the page that points to additional pages about the information space. Third are "see also" links that take the user outside of the current information space to other spaces with similar material. In a visual world, hyperlinks are easily identified by underlined blue text.

In a non-visual browser, vocal commands act like hyperlinks to jump the user from one page to another. If the commands are given over a telephone, then the only types of files that can be retrieved are audio files and text files that are converted to synthesized

speech. All three types of links that [7] identifies can be used in a non-visual browser as long as the information is coded to be accessible by a non-visual browser.

[13] predicts that in the next five years there will be a widespread distribution of voice browsers for retrieving information from the Internet. Rather than loading down PCs and PDAs with software that converts speech to text and vice versa, voice portals will become the norm for businesses. This is especially crucial for wireless devices like cell phones and PDAs because of their limited storage capabilities. [3]

The organization of the information and how much information is presented at one time are key factors in the structure of a voice document. Designing usable information spaces on the Internet is very important. As [7] indicates "if the customer can't find the product, then he or she will not buy it." This is also true for voice browsing applications as well. Speech is a powerful communication tool but only exists temporally. Speech and audio applications are sequential and lead to different design considerations as opposed to designing for a visual environment. [1] When using a voice browser, the concept of navigation and layout needs to be conveyed in a non-visual way. It is imperative to convey the structure to the listener or the listener will find the application useless.

Some of the design considerations suggested by [8] are listed in Figure 1. These considerations were used as guidelines in developing the prototype for this study. These design considerations can be categorized as navigational, organizational and cueing issues.

- Always announce a function and the key that activates it. This helps avoid confusion.
- When there is a list of functions, present the more common ones first. Then the user does not have to listen to the entire list to make a selection.
- Keep messages short and include prominent key words. Keywords should be verbally emphasized as a replacement for visual highlighting.
- Restrict the maximum number of items in a menu to between three and five. The user's short-term memory can easily handle that amount.
- Use silence to convey structure. Use short pauses between menu items and longer pauses between menus. However avoid long silences as this can cause the user to feel concern that the system is not working properly.
- Use careful wording, tone of voice, audible tones and logical sequences to convey context, errors, menu structure and the relative importance of the material being presented.
- Use terms and metaphors, which relate to spoken communication. Avoid computer terms.
- Confirm choices verbally so the person is more confident about what is happening.
- Use keys and commands consistently.

Figure 1. Design Considerations [8]

By using a simple test-to-speech system, the rendering of the document structure will likely be lost. [4] Navigation in the audio environment is more difficult. There is no highlighting or "to the right of". Menu selection can be accomplished but requires more information in order to work. There is no "click here" ability because by the time the user decides that is the direction they want to go, "here" has passed by and no longer exists in the audio world. [1,6].

One way to overcome this navigation problem is to use a scripting language such as VXML that allows the design flexibility to present and organize the information in an order that the listener can take in as well as adjust the amount of information presented at any one time. Other features that enhance the voice browsing experience include multiple synthesized voice choices, prosody, announcements and pausing to help form a structure for the listener. [4]

Other navigation strategies can be traced back to a physical-based system used in geographical systems. [12] names five of those navigation strategies that can be adapted to the browsing experience. Three of those strategies relate to voice browsing. These are using key identifiers, using a path or using a direction strategy.

[14] talks about the path strategy too and indicates that a path or order of traversal is helpful for some users. It helps users feel less disoriented and reduces the cognitive overhead needed to determine next direction. By using paths, the designer has the ability to present a structure to the audience. [14] goes on to identify three major characteristics of a good path mechanism. Those are 1) give power to the designers, 2) help authors create and modify existing paths and 3) help readers find paths and follow them in flexible ways. The goal is to keep the user from becoming lost in the information space.[9] By using menus, physical command maps and descriptive audio files, the user will have several ways to obtain an overview of a system and have a general idea where certain information resides in the application.

Another design consideration issue to consider is audio cues. Just as it is good to inform a sighted person that selecting a hyperlink may require a longer amount of time to render, it is important to give audio cues as well about lengthier audio files. [1] Another important aspect is the ability to interrupt. An interruption should lead to either a normal termination of a document or jump to the start of another document with relative ease.[6] VXML has a barge-in ability so that if the listener is ready to jump to another location, the command can be issued and it interrupts the current document and jumps to the location identified by the command. [4] also suggests that spoken feedback is helpful too so the listener knows that the command spoken was the command that was activated. Most speech recognition software operates on a principle of "most likely match" where the word spoken is matched as close as possible to the commands that are available. By choosing similar sounding commands, in the same information space, the system is likely to execute a recognition error and jump the user to the wrong space. [2] This also relates to the problem that occurs in applications written for other languages when non-natives speakers attempt to use the system.

The amount of material rendered in a given file is another design issue. In a study conducted by [9] with visually impaired subjects, users requested the ability to control the amount of information presented to them at one time. Information needs to be presented in meaningful chunks. Additionally, when using pre-recorded audio files, it is important that the files are relatively small in size or they will take too long to retrieve from a server. This is similar to the problem that occurs when large graphics files are viewed in a web browser.

Voice browsing has some specific problems too. As [13] points out, the major obstacle to voice browsers is that HTML does not support it. There are no standardized speech recognition grammars or voice prompts. There are standards in the making. These are principally being undertaken by World Wide Web 3 Consortium (W3C). The primary focus is in using VXML as the standard markup language.

Another problem is that pre-recorded audio can be noisy. It requires good recording equipment in order to produce quality, easy to hear files. Additionally, in a hands free environment, microphones are not often close enough to the speaker's mouth to obtain consistent quality input. Excessive background noise reduces input quality. In some environments, it may require additional microphones to triangulate a voice command in order to get good quality. [3]

Finally, there is also research which indicates that issuing voice commands for a web interface can take twice as much time as issuing the same commands using a keyboard and mouse. One reason this occurs is that most systems using voice commands are driven by software and it takes software longer to respond to an error correction. Visual browsers accept input from the keyboard and mouse. The user in this case is quicker at correcting errors. Another reason it takes longer is that the types of errors users make with voice browsers are not as easily detected or corrected as with a visual browser. [2]

Methodology and Research

Two websites that offer designers free service to develop VXML applications are http://café.bevocal.com which is supported by Be Vocal, Inc. and http://studio.tellme.com which is supported by Tellme Networks, Inc. After testing both sites, the designer chose to develop the application at the Be Vocal site.

Be Vocal's site includes a short tutorial and a lengthier manual with VXML script commands described in detail. Scripts can be stored on any server and retrieved as needed. This is also true for audio files. All scripts for the prototype reside on a server that the designer has access to in Iowa. The Be Vocal site has a utility to debug VXML scripts and was used extensively in order to make sure files had the correct syntax. There were a total of 66 VXML scripts written for the prototype and 53 audio files were recorded. The VXML scripts are text files written using an ASCII editor. All the scripts together required about 176 kilobytes of storage. The audio files were recorded with an

inexpensive microphone and the Sound Recorder software available in Microsoft Windows 98. These files required over 18 megabytes of storage. Besides the storage space on the local server, no additional software or hardware was required for the prototype.

The materials for the prototype were the course syllabus, a calendar of assignments, homework assignment descriptions and the portfolio assignment description. As this material was reviewed, the designer identified four main areas of information that students would be able to obtain from the prototype.

1. Syllabus. The information from the syllabus would further be subdivided into content areas.
2. Assignments. Descriptions of homework, quizzes, exams, the portfolio assignment and the final exam would be presented in this section.
3. Calendar. Students could get a list of assignment due dates for each month, for specific content areas or for a specific date.
4. Progress. This would give the student their current standing in the course.

These four areas became the main menu choices for the application. A final command added to the main menu was the "Exit" command which would exit and disconnect the listener from the application.

As noted by [8] in the design considerations listed above, messages need to be short. [10] also suggests using the "magic seven" rule. People tend to only remember seven (plus or minus two) ideas at a time. Not only would the number of menu options have to be limited but the amount of material that was presented would need to be short as well. Menus would also need to include commands to return to major areas of the application. "Main" and "Exit" commands were placed on all submenus to insure that the listener could always return to the top of the menu structure or quit the application.

Menu commands were kept simple and related to the information that would be retrieved. This adheres to [8] design considerations listed in chapter 2 of this paper. [8] second design consideration suggests that common functions be listed first. The word "common" was confusing. Did it mean commands common to this part of the menu or commands that were universally common throughout the prototype or the most likely commands that students would use? The designer decided that "common" in this case referred to the commands that related to the subtopic area. Universal commands such as "Main" and "Exit" were always listed last.

An unexpected problem arose when the recording of audio files began. A file that had 40 – 50 seconds of audio is approximately one megabyte. These files took too long to load. Therefore, most audio files are no longer than approximately 30 seconds. Only one audio file is over 900 kilobytes in length. The average size of each audio file is approximately 351 kilobytes. This problem of file size actually turned out to be helpful in chunking the information for the prerecorded audio files.

Considerable time was spent on chunking the syllabus material. Each area needed further subdividing. The designer wanted to carefully chunk the syllabus material, since it would be pre-recorded. Chunking the assignment material was less rigid because it would use the synthesized voice generator. The goal was to see what reaction the listener would have to the difference. The syllabus information was further subdivided into course information, course materials, course outcomes, course assessment tools and course policies.

Assignment material used the voice generated with the Be Vocal system. The voice generator simply repeats text that is typed into the script. The scripts were simple ASCII files. Therefore, more information could be chunked into a file without the file size restrictions of the pre-recorded information. Although more information could be chunked using the voice generator, it was not clear how much more the listener could hear and still absorb. Longer files were purposely created for the assignment descriptions to see what response the listeners would have.

In order to be understood, some of the words were phonetically spelled in the voice-generated texts. The designer discovered the VXML command <prosody> late in the development and was unable to test it in the prototype. This command controls the speed and volume, among other characteristics of the voice generated texts.

The purpose of the calendar menu choice was to provide students with a mechanism for finding out due dates. Three options for getting dues dates were created. The first was by content. Because other students may be interested in knowing the assignments for a given month, a second option for obtaining due dates was by month. Finally, some students might be interested in determining if there was an assignment for a specific date. The script files for all three options used the voice generated by the Be Vocal site. These files were shorter and more in line with the length of the pre-recorded audio files in the syllabus area.

Finally, the progress area is a way for students to be able to retrieve their current grade. Students are asked for a pass code. If the pass code they give matches one of the pass codes in the script, a pre-recorded audio file gives the grade information. Five fictitious names and grades were created for the prototype. Since pass codes can be any length, it is reasonable to create a variety of pass codes, each of different lengths so that students would have difficulty guessing other student pass codes.

If this prototype was used for a course, an instructor would spend several hours at the start of the semester to chunk the information and record it. During the semester, an instructor would require additional time to record updated audio files with grade information and to type up directions for any assignments that are not be present in the existing prototype. Once the prototype is fully developed, there would be no need to change the VXML scripts other than to change the names of audio files that are accessed. If the same audio file names were used as are in the current scripts, then the time for an instructor is solely in recording audio files.

There are two characteristics of the prototype that need to be addressed at this point. The first is that during testing, it was evident that certain times of the day were better for accessing the prototype than other times of the day. Recall that the scripts and audio files reside on a server in Iowa. The Be Vocal server is in California. Afternoons and early evening typically had very poor response time. Scripts and audio files took considerably longer to load. The second characteristic is that when a file is loading, there is "dead air". This was more noticeable when the files took longer to load. VXML does have the ability to have sound files play during loading to fill that air but the designer was not able to get that feature to work. On a more positive note, files are cached, so once a prototype script is accessed, returning to the script in the same phone call takes less time.

Results

Table 1 shows the questions that related to student use of the application and the subject's responses. The results clearly show that students prefer a visual browser to a voice browser. It is likely that students would use the voice browser for retrieving some course information. That is reiterated in the written comments submitted. Every student suggested that the application could be used to retrieve grades and some suggested getting the due dates for assignments. Overwhelmingly, this would not be a good choice for descriptions about assignments however. [8] design consideration regarding short messages appears to be affirmed by their comments. Assignment description passages were too long.

Table 1. Student usable application.

Question	S1	S2	S3	S4	S5
3. I/Students would use this application to retrieve course information. (1 = Definitely, 7 = Not at all)	2	2	7	4	5
4. I/Students would use this application to retrieve grade information. (1 = Definitely, 7 = Not at all)	3	2	3	1	1
5. I/Students would use this application more often than using a visual browser to locate the same information. (1 = Definitely, 7 = Not at all)	4	4	7	6	4
7. I/Students would use this application before getting in touch with the instructor if there was a question about the course. (1 = Definitely, 7 = Not at all)	2	5	7	3	7
8. I/Students would use this application before getting in touch with the instructor if wanting to know grade information. (1 = Definitely, 7 = Not at all)	2	1	2	1	1
13. I/Students would rather retrieve course information from a visual browser. (1 = True, 7 = False)	2	1	1	2	1
20. In my opinion, students would use this system. (1 = Would use, 7 = Would not use)	2	3	6	4	3

There were four questions that asked for opinions about the pre-recorded audio files and the voice generated information. These results are shown in Table 2.

Table 2. Voice preference questions.

Question	S1	S2	S3	S4	S5
9. I did not like the pre-recorded audio of the instructor's voice. (1 = I disliked it, 7 = It was fine)	6	7	4	7	6
10. I did not like the female synthesized voice audio. (1 = I disliked it, 7 = It was fine)	5	7	6	5	7
14. The pre-recorded audio voice of the instructor was difficult to understand. (1 = Very difficult, 7 = Not difficult)	5	7	7	7	7
15. The female synthesized voice was difficult to understand. (1 = Very difficult, 7 = Not difficult)	5	7	7	7	7

These responses seem to indicate that students did not have a real preference. Only one person noted in the written comments that the voice-generated information came too fast. This indicates that if an instructor were to adopt a voice browser, using the voice generator from the Be Vocal site would not be a problem for students. For example, if an instructor were to make grade information available over the telephone, then those files could be typed into the scripts. This would mean the instructor would not have to make audio recordings of grade information and would therefore save the instructor considerable time in creating those files.

Table 3. Comprehension of information.

Question	S1	S2	S3	S4	S5
16. Most of the course information took a long time to hear. (1 = Long time, 7 = Short time)	2	4	1	2	3
17. I got tired of listening. (1 = True, 7 = False)	2	6	1	6	4
18. The time between giving commands was too long. (1 = Too long, 7 = Not too long)	2	5	1	2	2
19. I liked how the information was divided. (1 = Good division, 7 = Poor division)	4	1	2	2	2

Some got tired of listening. As noted above, the waiting time for loading files was too long. Assuming this application was installed with all files residing locally, this may not be such a big problem. Also, if music or "commercials" could be played during file loading, the user might perceive that the wait time was not so long.

Questions 1 and 22 addressed training. All responses indicate that the training document that was provided was enough for learning how to use the system. In written comments, one student noted that the command maps were essential in understanding how the system worked. If students were to use such a system, a command map would likely need to be provided.

The other questions on the survey dealt with general usability. Those questions are listed in Table 4. Most of the students indicated that the application did understand the commands given. Students self-reported that they made few errors and were non-

committal on whether it was difficult or easy to make errors. The application itself seems to have a mixed review on whether or not it is confusing. Students did not seem concerned about privacy issues as question 11 shows. And overall, most feel that the application is usable.

Table 4. General usability questions.

Question	S1	S2	S3	S4	S5
2. The application did not understand my commands. (1 = Easily understood, 7 = Did not understand)	3	2	6	2	2
6. This application is confusing. (1 = Definitely, 7 = Not at all)	3	5	2	7	7
11. I would be concerned if grade information was available from a telephone application such as this. (1 = Very concerned, 7 = Not concerned)	5	7	6	5	7
12. I seldom gave a wrong command. (1 = Few errors, 7 = Many errors)	3	2	4	2	3
21. It was easy to make errors. (1 = True, 7 = False)	4	4	1	6	4
23. Overall, this application is usable. (1 = Very usable, 7 = Not very usable)	1	2	7	1	1

Conclusions and Further Research

There were two good suggestions for improving the application. The first was to provide additional commands that would summarize the information a student would most likely want from an assignment description. For example, on a homework assignment, list the page and problem numbers for assignments from the book in addition to a longer description of the assignment that is obtained from a different command. The second suggestion was to include a "Repeat" command. Currently the only way to hear the information that the user just heard is to return to the previous content menu (syllabus, assignment, calendar or progress) or return to the main menu and traverse through the system again. The designer also suggests that "pause" and "resume" commands would be helpful so that longer passages could be stopped allowing the listener to write notes before continuing.

At this point, providing all course materials to students from a voice browser does not seem very effective. Most students would likely use a voice browser to retrieve grade information and due dates for assignments but it would not be their first accessing tool choice. If assignment descriptions were to be included in the application, more attention would need to be made to chunking the information and organizing it so that students would be more willing to listen to it. It seems evident that the assignment part of the application was not presented in a usable format for students. Overall, the voice browsing application would probably be considered a backup system to what is available through a visual browser.

It is difficult to assess how much retrieval times for scripts and audio files affected the usability of this application. If the voice browsing system with scripts and audio files were accessible on a local server, it is likely that the access time would be reduced. Further research is suggested with an application that has only grade information and assignment due dates available to students.

Appendix A

Student Usability Exit Survey

Exit Survey

1. The application was easy to learn.	Easy 1	2	3	4	5	6	Hard 7
2. The application did not understand my commands.	Easily Understood 1	2	3	4	5	6	Did not understand 7
3. I/Students would use this application to retrieve course information.	Definitely 1	2	3	4	5	6	Not at all 7
4. I/Students would use this application to retrieve grade information.	Definitely 1	2	3	4	5	6	Not at all 7
5. I/Students would use this application more often than using a visual browser to locate the same information.	Definitely 1	2	3	4	5	6	Not at all 7
6. This application is confusing.	Definitely 1	2	3	4	5	6	Not at all 7
7. I/Students would use this application before getting in touch with the instructor if there was a question about the course.	Definitely 1	2	3	4	5	6	Not at all 7
8. I/Students would use this application before getting in touch with the instructor if wanting to know grade information.	Definitely 1	2	3	4	5	6	Not at all 7
9. I did not like the pre-recorded audio of the instructor's voice.	I disliked it 1	2	3	4	5	6	It was fine 7
10. I did not like the female synthesized voice audio.	I disliked it 1	2	3	4	5	6	It was fine 7
11. I would be concerned if grade information was available from a telephone application such as this.	Very concerned 1	2	3	4	5	6	Not concerned 7

	Few errors						Many errors
12. I seldom gave a wrong command.	1	2	3	4	5	6	7
13. I/Students would rather retrieve course information from a visual browser.	True						False
	1	2	3	4	5	6	7
14. The pre-recorded audio voice of the instructor was difficult to understand.	Very difficult						Not difficult
	1	2	3	4	5	6	7
15. The female synthesized voice was difficult to understand.	Very difficult						Not difficult
	1	2	3	4	5	6	7
16. Most of the course information took a long time to hear.	Long time						Short time
	1	2	3	4	5	6	7
	True						Not true
17. I got tired of listening.	1	2	3	4	5	6	7
18. The time between giving commands was too long.	Too long						Not too long
	1	2	3	4	5	6	7
	Good division						Poor division
19. I liked how the information was divided.	1	2	3	4	5	6	7
20. In my opinion, students would use this system.	Would use						Would not use
	1	2	3	4	5	6	7
	True						Not true
21. It was easy to make errors.	1	2	3	4	5	6	7
22. I needed more training.	More training						Adequate training
	1	2	3	4	5	6	7
	Very usable						Not very usable
23. Overall, this application is usable.	1	2	3	4	5	6	7

1. Generally what did you think of the application?
2. Do you think the chunks of information were too long? Too short?
3. Was it hard to use?
4. Would students/you use it?
5. Are there parts of the system you would use more than others? What parts?
6. Do you have any suggestions for improving the application?
7. Any other comments about the system?

References

1. Arons, B. (1991). Hyperspeech: navigating in speech-only hypermedia. *Proceedings of the 3rd annual ACM conference on Hypertext.* San Antonio, TX. pp. 133 – 146.
2. Christian, K., Kules, B., Shneiderman, B. & Youssef, A. (2000). A comparison of voice controlled and mouse controlled web browsing. *The fourth international ACM conference on assistive technologies.* Arlington, VA. 72 – 79.
3. Clark, D. (2001). Speech Recognition: the wireless interface revolution. *IEEE Computer*. 34(3).16 – 18.
4. Goose, S., Newman, M., Schmidt. C. & Hue, L. (2000). Enhancing web accessibility via the Vox Portal and a web-hosted dynamic HTML to VoxML converter. *Computer Networks*. 33(2000). 583 – 592.
5. Goose, S., Wynblatt, M. & Mollenhauer, H. (1998). 1-800-Hypertext: Browsing Hypertext with a telephone. *Proceedings of the ninth ACM conference on hypertext and hypermedia: links, objects,* time *and space – structure in hypermedia,* Pittsburg, PA. pp. 287-288.
6. Muller, M.J. & Daniel, J.E. (1990). Toward a definition of voice documents. *Proceedings of the conference on Office Information Systems.* ACM. pp. 174 – 183.
7. Nielsen, J. (2000). *Designing Web Usability*. Indianapolis: New Riders Publishing.
8. Noonan, T. (2000). A strategy and information of designing user-friendly automated telephone services which incorporate text-to-speech. 15th Annual International Conference, Technology and Persons with Disabilities, Los Angeles, CA. Available: www.csun.edu/cod/conf2000/proceedings/0190Noonan.html. Accessed: July 24, 2001.
9. Petrie, H., Morley, S., McNally, P., O'Neill, A., Mojoe, D. (1997). Initial design and evaluation of an interface to hypermedia systems for blind users. *Proceedings of the 8th ACM conference on Hypertext.* Southampton United Kingdom. pp. 48 – 56.
10. Shneiderman, B. (1998). *Designing the User Interface*. Reading, MA: Addison Wesley.
11. Tober, E. & Marchand, R. (2001). VoiceXML Tutorials. *The VoiceXML Forum*. Available: http://www.voicexml.org/tutorials/intro1.html. Accessed November 17, 2001.
12. Van Dyke Parunak, H. (1989). Hypermedia topologies and user navigation. *Proceedings of the 2nd annual ACM conference on Hypertext*. ACM. pp. 43 – 50.
13. White, J. (2000). Voice browsing. *IEEE Internet Computing.* 4(1). 55 – 56.
14. Zellweger, P.T. (1989). Scripted documents: a hypermedia path mechanism. *Proceedings of the 2nd annual ACM conference on Hypertext*. ACM. pp. 1 – 14.