# AUDIO BROWSING OF AUTOMATON-BASED HYPERTEXT

A Thesis

by

SELEN USTUN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2003

Major Subject: Computer Science

# AUDIO BROWSING OF AUTOMATON-BASED HYPERTEXT

A Thesis

by

SELEN USTUN

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

| | |
|---|---|
| Richard Furuta<br>(Chair of Committee) | Frank M. Shipman III<br>(Member) |
| Ergun Akleman<br>(Member) | Valerie E. Taylor<br>(Head of Department) |

December 2003

Major Subject: Computer Science

# ABSTRACT

Audio Browsing of Automaton-Based Hypertext. (December 2003)

Selen Ustun, B.S., Bogazici University, Turkey

Chair of Advisory Committee: Dr. Richard Furuta

With the wide-spread adoption of hypermedia systems and the World Wide Web (WWW) in particular, these systems have evolved from simple systems with only textual content to those that incorporate a large content base, which consists of a wide variety of document types. Also, with the increase in the number of users, there has grown a need for these systems to be accessible to a wider range of users. Consequently, the growth of the systems along with the number and variety of users require new presentation and navigation mechanisms for a wider audience.

One of the new presentation methods is the audio-only presentation of hypertext content and this research proposes a novel solution to this problem for complex and dynamic systems. The hypothesis is that the proposed Audio Browser is an efficient tool for presenting hypertext in audio format, which will prove to be useful for several applications including browsers for visually-impaired and remote users.

The Audio Browser provides audio-only browsing of contents in a Petri-based hypertext system called Context-Aware Trellis (caT). It uses a combination of synthesized speech and pre-recorded speech to allow its user to listen to contents of documents, follow links, and get information about the navigation process. It also has mechanisms for navigating within documents in order to allow users to view contents more quickly.

To Esin and Guldem Ustun,

For all the love and support that they have given me.

# ACKNOWLEDGMENTS

There are several important people who have helped me accomplish my master's degree. First of all, I would like to express a profound appreciation for my committee chair, Dr. Richard Furuta, who guided my research. I would like to acknowledge that his suggestions, encouragement and patience have made this research possible. I would also like to thank my other committee members, Dr. Frank M. Shipman and Dr. Ergun Akleman, for their help and time.

While implementing the Audio Browser, Unmil Karadkar answered my endless questions with patience. Turkan Aydogmus helped me find subjects for the usability tests. I would like to thank them and everyone who participated in the tests.

During the course of my study at Texas A&M University, I worked at the Computing Support Group (CSG) at the Department of Computer Science. I learned a lot and met great people there. I would like to thank everyone at CSG, and especially Adam Mikeal and Kathleen Flores for making my experience at CSG a memorable one. I would also like to give my thanks to Dr. Valerie E. Taylor, Dr. Jennifer Welch, Dr. Nancy Amato, and Dr. John Leggett for their support. During the last stages of my research, I started working for the Adaptive Technology Services Lab with David Sweeney. I am thankful to him for his confidence in me and the wonderful work environment that he provided for me.

I feel blessed to have friends who stood by my side even when they were thousands of miles away in Turkey. I would especially like to thank Koray User and Didem Giranit for being the wonderful people they are and letting me feel loved and supported wherever I was.

I was also lucky that two of my best friends, Onur Buk and Omur Koken, came to the United States for their studies. I turned to Onur whenever I felt down and he miraculously managed to make me feel great every time. I could not thank him enough for his encouragement and company. I had the pleasure of visiting Omur in California and later having a trip to New York City. Both of those trips were some of the best in my life and they would not have been nearly as enjoyable without his company. I would like to thank him for his friendship. I am also thankful to all my friends from College Station who have been like a family to me.

I am grateful to my professors, family and friends for their support, love and encouragement.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER I

# INTRODUCTION

Hypertext systems are used extensively for non-sequential data browsing. The most widespread hypertext system is the World Wide Web (WWW) due to its simplicity and large-scale availability. However, the simplicity that makes it so popular is a result of the lack of structure and therefore it is not suitable for more sophisticated applications accessed by a wider user base.

One of the new presentation mechanisms is audio-only presentation, which is different than visual navigation due to the inherently sequential access in listening to audio and the lack of visual cues for skimming and extracting key points for faster navigation. These problems require more than simply converting the textual content to audio in order to facilitate the ease and pace of navigation, and still need to be addressed. Therefore, there is a need for an efficient mechanism for audio presentation and navigation of dynamic hyperlinked networks.

This thesis proposes a novel solution to the problem with an Audio Browser that is based on the Petri-based hypertext system called Context-Aware Trellis (caT) [Na et al. 2001, Furuta and Na, 2002a, Karadkar et al. 2002, Furuta and Na, 2002b]. The Audio Browser presents mechanisms to convert textual contents to audio, listen to contents, and navigate between and within these contents while allowing faster and more efficient browsing. The underlying Petri net engine allows the creation and authoring of a context-aware dynamic and structured net.

---

This thesis follows the style and format of *ACM Transactions on Information Systems*.

In order to overcome the problems related to audio-only browsing, the usage of navigation aids within documents, careful selection of commands and keys, combination of synthesized and pre-recorded speech, and author-created summaries are investigated and implemented in the Audio Browser. Then the usability of the browser is evaluated with a user test conducted at Texas A&M University.

The Audio Browser is expected to be useful in several applications. The first one is in hypertext networks for blind users who can use this browser to navigate within a network making use of the properties that are designed specifically for audio-only browsing instead of screen reading systems which have to extract content from visualization-oriented documents that are not created with audio-only browsing in mind. Therefore, the audio browser is expected to be significantly better than these systems.

The second area is with remote applications, which can easily be automated with the proposed automaton-based hypertext system and can produce speech results for remote users of the system. With this browser, users may connect to the Audio Browser and listen to the contents and follow links while using their devices such as regular phones, cell phones or PDAs.

This paper first gives examples of related work in research in the Related Work section. Then, the proposed system architecture and features are described in detail followed by the evaluation test plans. The evaluation results and their interpretations for the effectiveness of the developed system, therefore the correctness of the initial hypothesis, are discussed in the Conclusions and Future Work section, which also includes directions for future work in this field.

# CHAPTER II

# RELATED WORK

## caT Hypermedia System

Several hypertext systems have been developed and they have evolved to those that respond to user and environment characteristics as well as presenting documents and links. The characteristics involve the time of day, location, device and network properties, user preferences, etc. One of the earlier responsive hypertext systems is Trellis [Stotts and Furuta, 1989], which investigated hypertext applications specified by Petri nets.

A basic Petri net is a directed bipartite graph like the ones shown in Fig.1. The circular nodes are called places, rectangular bars are called transitions and dots within nodes are called tokens. When all the nodes that have outgoing arcs to a transition have tokens, that transition is fired. When a transition is fired, the tokens in the input places of the transition are removed and placed in the output transitions. Fig.1 shows three sample cases before and after transition firing.

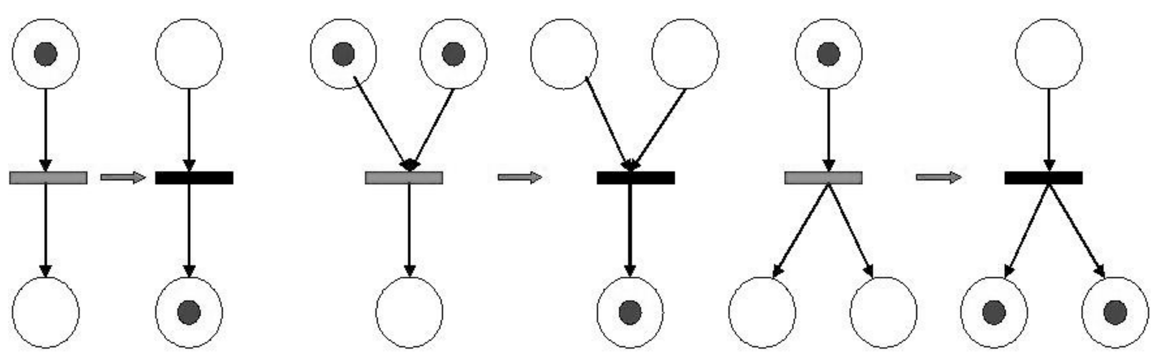

Fig. 1. Petri Net Basics

A more enhanced form of Petri nets is the timed and colored Petri Net, where tokens are separated from each other by color and arcs may have time constraints for transition activation. Trellis used timed, colored Petri nets in order to describe contents and links among these contents. In Trellis, contents were associated with places and links were associated with transitions. Also, different colored tokens represented different users and timing on the arcs allowed auto-firing of transitions.

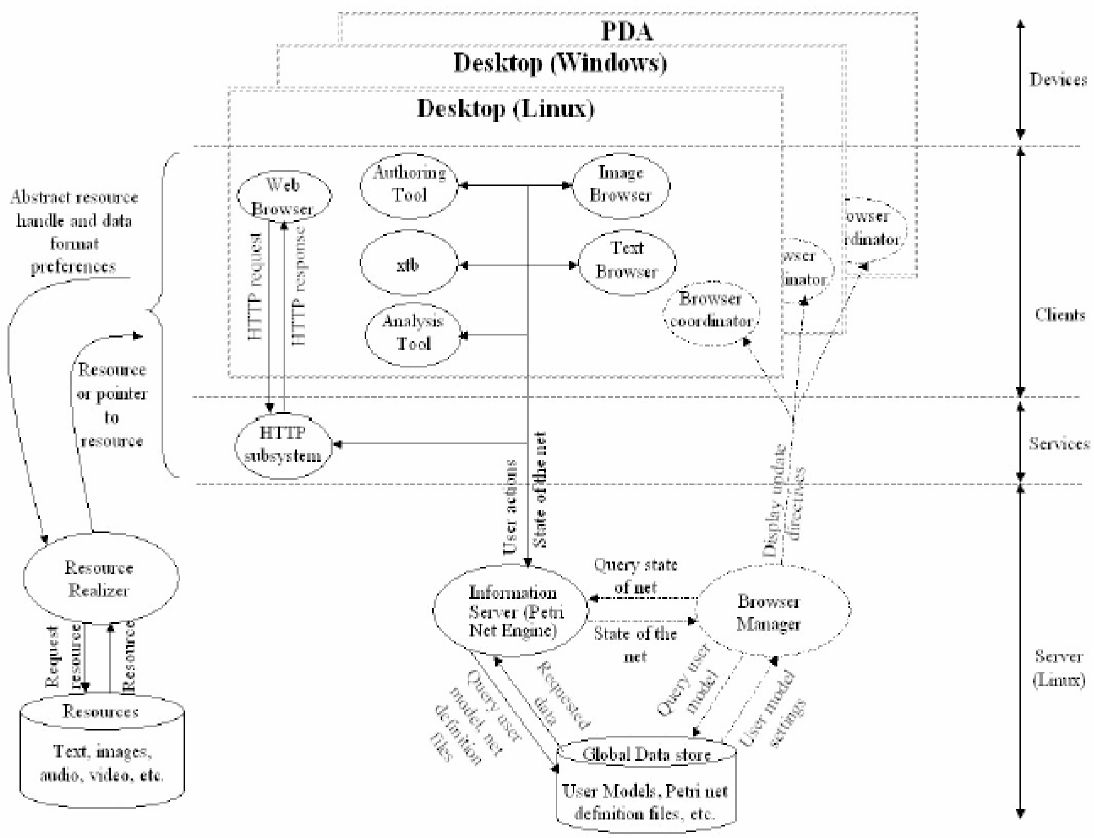


Fig. 2. caT Architecture

Trellis was later extended to caT, which incorporated characteristics of the environment, and therefore became context-aware. These characteristics also included user preferences, and support for representing classes of users. With caT, users may define networks, place contents within nodes, and specify transitions and conditions for transitions. This allows users to synchronize and easily manage large quantities of different types of documents and provides means for viewing and navigating among them.

As seen in Fig. 2, caT has a modular architecture providing several features for authors and viewers [Karadkar et al. 2002]. This thesis is related to the Editing and Browsing sections of caT. The editing takes place in the editing tool called xTed (Fig. 3). It allows authors to create nets, subnets, nodes, transitions, and assign attributes to different sections of the net. Browsing in caT is realized with separate browser for different types of media. This section is explained in detailed in Chapter IV.

**Speech Synthesis and Presentation**

There are several available speech synthesis engine and this research uses the Festival Speech Synthesis program [FSSS 2003] in order to synthesize voice from contents of the net. It is a multi-lingual speech synthesis system developed at the Centre for Speech Technology Research at University of Edinburgh. It is written in C++ and has a Scheme-based command interpreter for control. It provides US and UK English and Spanish voices and allows the integration of MBrola voices [Mbrola 2003]. It is selected since it is open-source, provides several APIs and is readily available in Linux Red Hat distributions.

A major aspect of research that this study is related to the characteristics of audio browsing and properties of audio, which need to be considered. As stated in [Arons 1993], listening to a speech recording is significantly harder than visually skimming a written document because of the transient and temporal structure of audio. This sequential access of audio that is inherent in its nature is one of the major differences between written and audio documents.

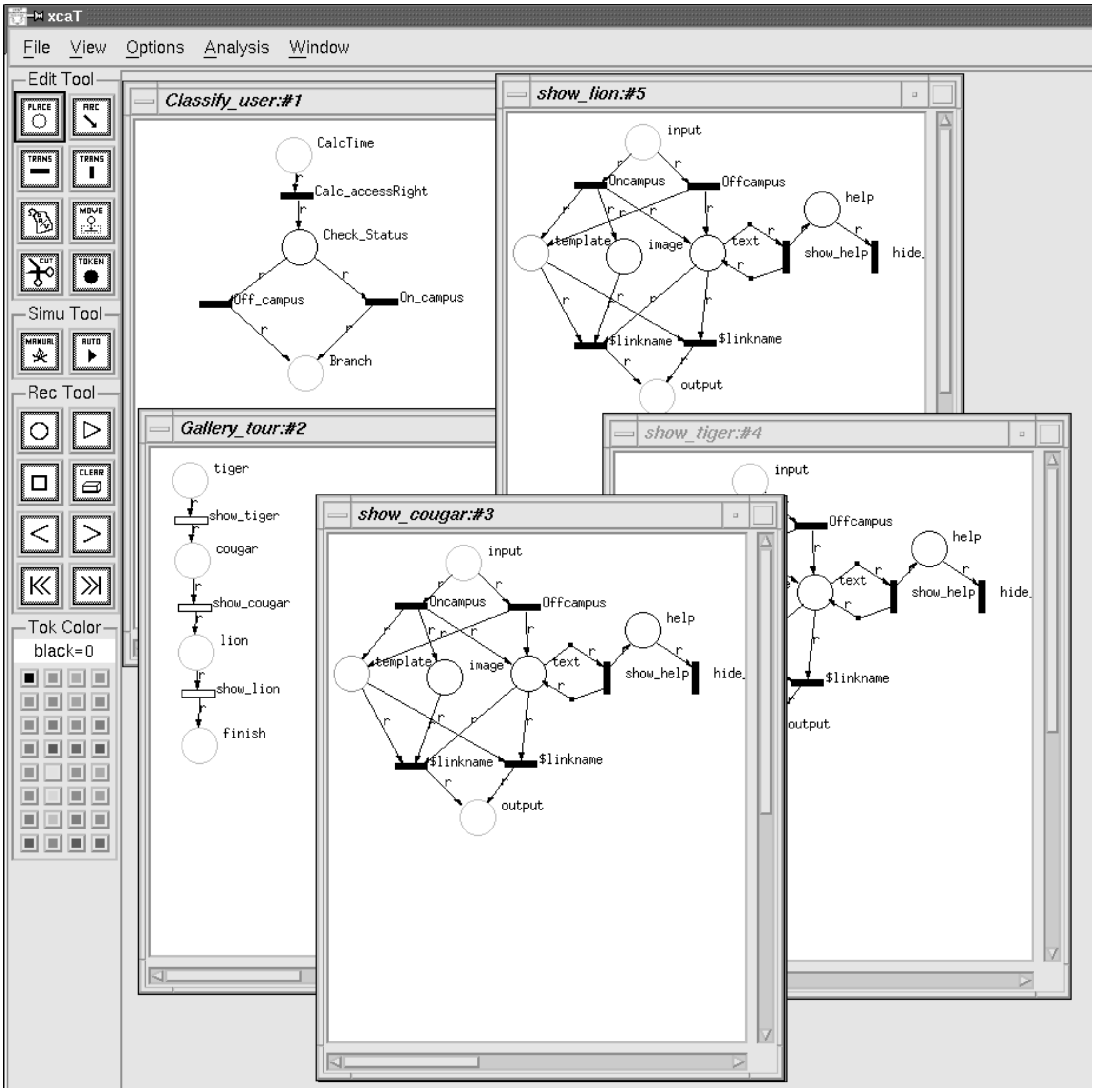


Fig. 3. xTed Screen

SpeechSkimmer [Arons 1993, Arons 1997] considers these properties of audio and proposes methods for structuring, filtering and presenting audio so that users can navigate easily and find points of interest within speech data. In order to do this, reducing the time to listen to speech is proposed through the use of time-compressed audio, pause shortening, automatic emphasis detection and non-speech audio feedback.

Time compression is considered in several other studies after the SpeechSkimmer study [Omoigui et al.1999]. These studies are about the audio characteristics of data and the means to manipulate the characteristics of speech data rather than the textual characteristics of data.

Another approach depends on the textual characteristics of the data. A study at Microsoft Research [He et al. 1999] is related to auto-summarization of audio-video presentations. The study stems from the point that the multimedia content on the net is becoming increasingly large and there is a need for quickly examining this content. For this purpose, they evaluate three techniques for automatic summarization of presentations. The first one examines information in the audio signal such as pitch and pause information. The other methods are knowledge of transition points in presentation and access patterns of previous users.

Although this is a very restricted study in that they consider only presentations, and some of the findings are related only to audio content, it also relates to any multimedia content on the net and shows that summarization significantly decreases the amount of time required to listen to and understand audio content.

**Audio Browsing and the Web**

The Auditory Navigation in Hyperspace [Morley et al. 1998] is a non-visual hypermedia system for blind users and is significantly related to our proposed research. This study does not consider the characteristics of the audio signal, but considers the use of non-visual interfaces, non-speech sounds, three input devices and a 37-node hypermedia module.

While this research provides audio navigation of an alternate hypermedia system, another approach is to provide an audio hypertext interface to existing WWW and to build audio applications on top of that. The main approach here is to develop screen readers for enhancing normal browsers [Edwards and Mynatt, 1994]. Another approach is to build voice browsers for the Web [Raman 1996, pwWebSpeak 2003].

Also, there are studies branching in several directions ranging from creating mark-up languages for audio [Sproat et al. 1997] to converting HTML content to audio with the use of auditory cues [James 1996]. The best-known and elaborate effort for creating a mark-up language is the development of VoiceXML. It is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations [VoiceXML 2003]. There are several VoiceXML browsers that present VoiceXML documents in audio format but these applications use the telephone system to access to their portals.

# CHAPTER III

# MOTIVATION

With the advancement of technology to wider audiences, there are several exciting opportunities for the Audio Browser, which are explained in the following scenarios.

**Scenario I – Navigation for the Visually-Impaired Student**

Hypermedia systems are used increasingly in education for allowing non-sequential means of reaching information. Combining the network of information with user preferences and the Audio Browser, visually-impaired students may benefit from these complex and context-aware systems.

In a school or university where class material is kept as a caT net, a visually-impaired student can access the system from any computer and run the Audio Browser to listen to the announcements related to a specific course. Once the student logs into the system, the user profile and preferences are loaded by caT for further personalization.

The student can navigate to the course material by listening to the position information and following links using simple keystrokes. Once she reaches the course information, the context-aware system checks the user profile and finds the section of the course that the student is enrolled in. That section is automatically selected by the system and news and announcements are extracted according to the date.

When the content is ready, the student can listen to the announcements while controlling the listening process as she wishes. She may pause, replay, stop, listen to a summary, skim the content using the commands provided in the Audio Browser for more sophisticated playback control.

**Scenario II – Remote Telephone Application**

A visitor travels towards Texas A&M University and does not know how to get to the university. From her cell phone, she calls the number that connects her to the Audio Browser. She navigates to the travel directions using the Audio Browser and following links that she hears. caT requests the position information from the cell phone and provides her the appropriate driving directions accordingly.

She listens to the directions and replays them to make sure that she understands correctly. The system also provides parking information as a link from the driving directions. The system checks to see if she has a parking permit. Depending on the permit that she has, the links to the relevant parking lots become active. She listens to their locations using the Audio Browser and makes a choice based on what she has heard.

**Scenario III – Intelligent Audio Application**

Today in many museums of the developed world, there are audio applications that allow the user to hear about the particular instances in the museum or gallery that they are visiting. Usually, users are required to receive a listening device as they enter the museum and wear it for listening to the contents available to them.

The audio content reaches the user in two ways. Either the audio broadcast to all users at the same time, or they can listen to prerecorded contents. When the audio is broadcast, the user may listen to it either by selecting it with one of the keys available on the headset she is given or the radio signals are picked up automatically and listening starts immediately as the user reaches the particular area where the broadcast is audible.

When the audio is broadcast, it is not possible to control the beginning or the end of the audio and the user is totally passive in the listening process. In cases where the user may select the number to listen to as in the case of New York City Natural History Museum, the user has to decide what item to listen to and the only control she has over the listening process is stopping and starting the audio.

All of these types of listening are insufficient in two major directions. First of all, the listening is generally not interactive and the system is not intelligent enough to adapt according to the environment and the user. There are efforts to make more intelligent and interactive systems, but there is yet no one system that effectively handles these situations.

With the use of the caT structure and the Audio Browser, the contents in the museum may easily be converted into a net in the system. Since there are relationships between the components in a museum, the net is actually a hypermedia system. caT incorporates this structure and also permits additional more sophisticated features such as controlling the exhibition displays according to the state of other exhibitions or the number of users that are accessing the system.

With the Audio Browser, the user may have extended control over the listening process as she may play, pause, go forward and backward within each document. This way, the user would never have to miss any part of the listening and if it is not as interesting, it may be skipped or skimmed. This allows the user to know how much is left and how much she has listened to.

In the case of a museum system where the net is not as changeable as the Web, for example, it might be better to record the audio instead of synthesizing voice from different components. Fortunately, the Audio Browser allows different types of media to be presented even if it is not synthesized.

**Scenario IV – Audio Books**

Hypermedia novels and stories are gaining more popularity every day. One of the reasons is that they give interactivity to the user making them a part of the story and make their choices important. While this allows different individuals to have different perceptions, it may also allow one user to have a different experience based on some properties in the system. For example, some nodes may be reachable though a certain number of iterations.

These systems are either displayed over the Internet or some propriety systems are developed allowing access for user to only one specific story or novel. When they are displayed over the Internet, the ability to reflect characteristics of the environment and to gather information is highly restricted.

In order to overcome these problems, one solution is the usage of caT, which provides content-awareness for users. It detects environment variables such as the time of day, position if a GPS device is attached, and the type of device used, and also user preferences such as the synthesized voice, rate of playback, and personal characteristics that may be important for the hyperlinked story.

In order to realize this, the author creates a net with xTed and sets the conditions on the transitions and nodes. Based on the diverse set of context-awareness that is provided by caT, the author may add as much interactivity and non-determinism as desired. Then, for the audio sections of the book, the author either adds voice recordings or just leaves the text for voice synthesis.

The person who has access to the story starts reading that story on his tablet PC but he needs to drive his car and cannot continue reading. He connects a Bluetooth headset to the tablet PC, which is detected by the caT system as a sign to transfer the information to the headset as audio.

The browser coordinator at that point starts running the Audio Browser which continues from the point he left reading by producing either synthesized voice from the written contents or by playing the prerecorded audio. With the commands available to the user, the user may easily listen to the contents interactively. This gives an example of hyperlinked literature, but the audio browser may also be used with normal online books in order to treat them as audio books. Even though the recordings are not available as in the case of hypertext books, voice may be synthesized and presented to the user, making any kind of content available as audio.

# CHAPTER IV

# BROWSING IN caT

As in any hypermedia system, browsing is an essential part of caT that allows users to navigate between nodes in an easy and meaningful manner. In order to achieve this, caT provides specialized browsers, each of which is capable of handling one type of document content. This gives users more flexibility by providing tools to traverse the net only viewing some types of documents or view the whole net using multiple browsers at the same time. The structure of caT allows developers to add additional modules into the system so that different visualizations may be provided.

**Text Browser xtb**

The main browser provided by the older xTrellis implementation was capable of handling textual content. This browser, called xtb, was also included in caT.

xtb is invoked from the command line by writing “xtb” with optional parameters such as content type and net number. When the browser starts, it connects to the Net Engine using RPC and retrieves the document names representing active contents. The names of active nodes and transitions are presented in the SELECTOR window (Fig. 4). The active content files are opened and their contents are written on the xtb windows. These windows also have buttons for active transitions (Fig. 5).

When the name of the node is selected from the SELECTOR window, the window representing the contents of that node receives the focus. When one of the transitions is selected either from the transition list in the SELECTOR window, or by pressing the related button in the xtb window, that transition is fired.

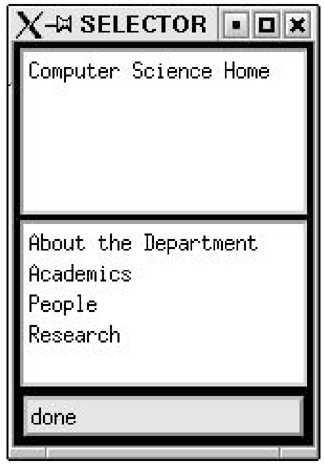


Fig. 4. SELECTOR Screen

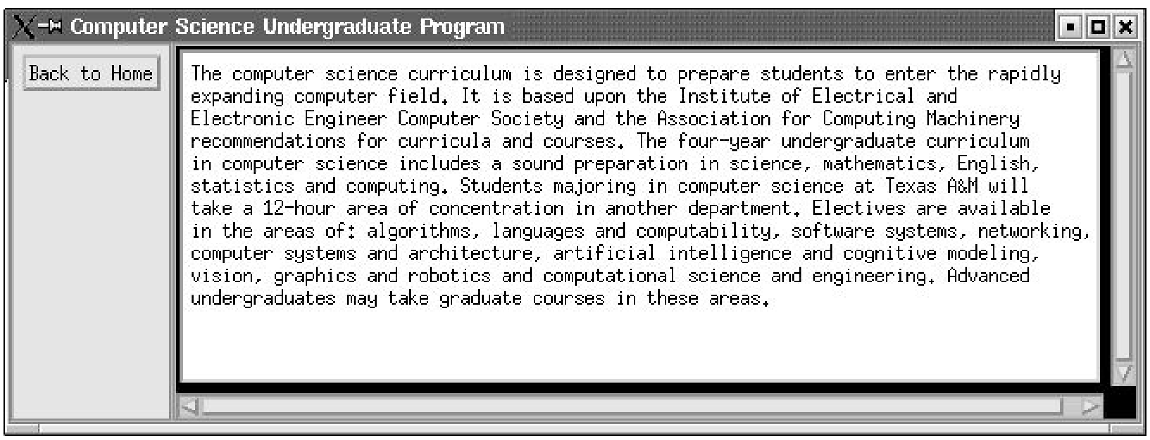


Fig. 5. xtb Screen

According to the specified polling time, the browser checks for updates in the underlying net. Whenever a transition is fired explicitly by the browser user, or when the net is updated from another browser or on the editor, these changes are detected and conveyed to the browser through RPC. The browser, in turn, compares the new active nodes to the displayed one in order to add new nodes and delete the ones that are no longer active.

**Image Browser xtb2**

The second browser that was added to the system is called xtb2 and it can handle image content, which allows users to view image files with several different extensions. This browser is also present in the current version of caT.

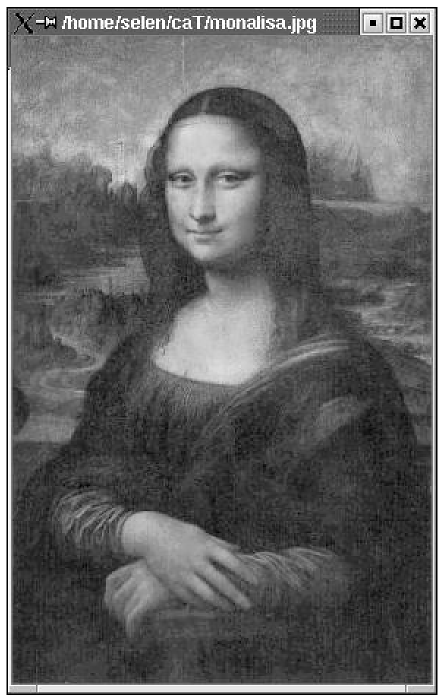


Fig. 6. xtb2 Screen

The SELECTOR screen is the same as in xtb and it gives the user to focus on specific screens or fire transitions. Unlike xtb, however, xtb2 does not handle image files by itself but calls the xv image viewing program [xv 2000] in order to display images (Fig. 6). Also, there are no buttons in the image windows for firing transitions. Similar to xtb, when the net is updated elsewhere or when the user selects a transition to fire in the SELECTOR window, the new active elements are displayed and the old ones are deleted from the screen.

**Audio/Video Browser xtb4**

A completely new browser that was created as part of this research is called xtb4 and is capable of handling audio and video content in document files. It is an important addition to caT since no modern hypermedia system can be considered complete without a wide range of multimedia support.


Fig. 7. xtb4 Screen

Although text and image browsers are similar in nature, this audio/video browser has different features. One of the main differences is that unlike the textual and image content, there should be only one document active at any given time. Therefore, multiple browsers and multiple nodes activated simultaneously require more sophisticated handling in order to prevent concurrent audio and visual displays.

The SELECTOR window is the same as in xtb, and xtb2 in order to keep different parts of the system consistent with each other. The split windows show the name of the active nodes and transitions that can be fired. The display of audio and video content is made available with the usage of the xine program [xine 2003]. xine is a free, gpl-licensed player for UNIX-like systems. It is distributed with Red Hat 7.3, which is the development system for caT.

Like in the previous browsers, the active nodes of the net are traversed and for each active audio/video content element, a new xine window is opened with that file (Fig. 7). However, the presentation of contents does not automatically start in order to prevent the clash of multiple files at the same time. For all the active nodes that have video content, a new window is created and placed on different locations on the screen so that they do not overlap when they are created. The user may move, resize, or close these windows at any time using the mouse. At the time of creation, video files are not played so that the user can choose the file to play and play all the files in any order she likes.

Similar to xtb and xtb2, this browser also polls the net engine periodically in order to detect changes in the net structure. Another important event is the updating of the net. When the net is updated by the user or by any other user, all users of the video browser get a message saying that there is a change in the net. The update is not enforced in the net immediately in order not to terminate the video prematurely in the middle of execution or change the contents of the screen without notifying the user. The message asks the user to click on a button when she wants the update to be propagated to the video browser.

The parent process that calls the xine program for the display of video files also keeps track of when the playing of the files is over. In the case of automatic displaying, the end of the last process is waited and control is returned back to the program in case there is an autofire option selected. If the user is selecting the files to display, then the user is supposed to close the xine windows in order to indicate that the viewing is over. When all the windows are terminated, the end of playing information is conveyed back to xTed, which may in turn fire some transitions in case the auto-fire option is selected. This is particularly useful for applications where the net needs to be traversed in a certain order taking care of synchronization either audibly or as a combination of audio and visual files.

# CHAPTER V

## AUDIO BROWSER FOR caT

The audio browser for caT is a browser that allows users to navigate any net within caT by issuing commands with simple keystrokes and to listen to the available contents. It is designed as a new module that is incorporated to the caT architecture (Fig. 2).

### xTed for Audio Browser

The editor for authoring caT nets is modified in order to better accommodate the needs of the Audio Browser. Although, the existing nets may be used without further modifications, full functionality may be accomplished with the new editor.

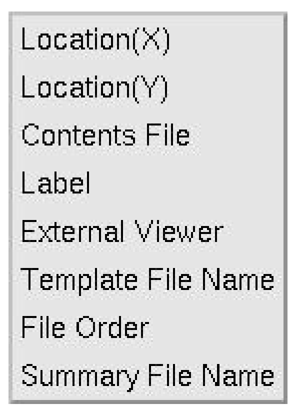


Fig. 8. xTed Attributes

One of the two main areas where xTed is modified is related to the ordering among the pages in the net. In order to allow nodes that are active at the same time to be ordered relative to each other, a new attribute called File Order has been added to every node in xTed (Fig. 8). This attribute determines the ordering among nodes where the node with the smaller File Order has precedence over another node with a larger File Order.

In the new version of xTed, when a new node is added to the net, in addition to the default X and Y location of the new node, the attribute File Order is automatically set to 1. If the attributes are not later modified, this makes the order of the nodes equal to each other. Therefore, if the ordering among documents is important, then it is suggested that these File Orders be set for each document. Since there can be several different combinations of simultaneously active nodes, it is recommended that the file order numbers are unique within a net.

The File Order attribute can be modified by the user by selecting the attribute and manually changing the attribute number in the Attribute Properties dialog box (Fig. 9). When the user changes the value of the order, it may be equal to any other node in the net. In that case, these nodes are considered to be equivalent to each other from the precedence perspective and no further action to correct this is taken.

The ordering of the nodes using the File Order attribute is specified manually by the user at the time of node creation or any other time afterwards. However, this is not intuitive enough since the orders of other files are not visible unless their attributes are viewed individually. Therefore, in the future implementations, this can be improved by adding a feature to select nodes and move them in the list of nodes by dragging them with the mouse. Another feature might be to include the order number of the nodes so that they are visible from the xTed view.

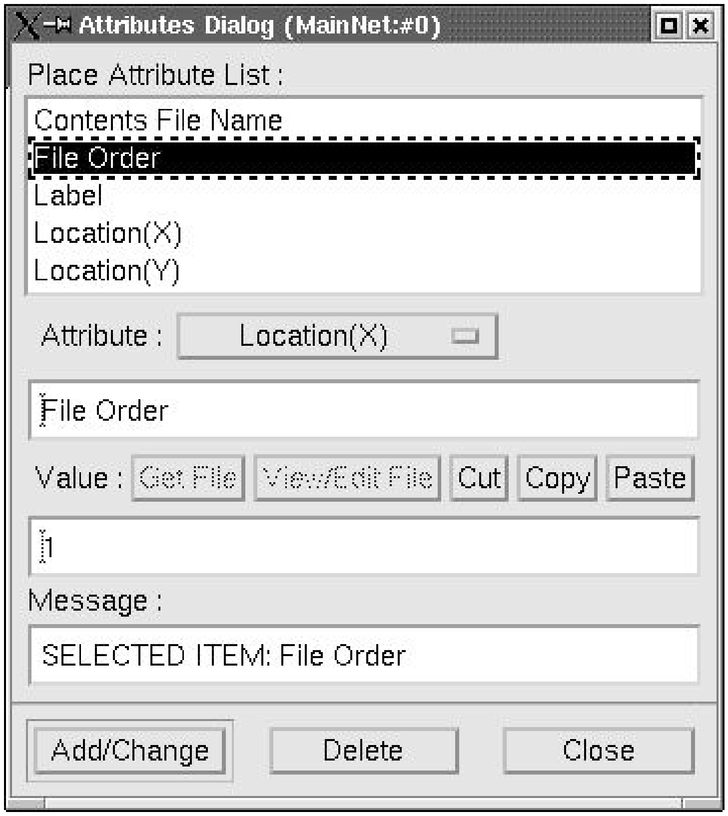


Fig. 9. Setting File Order Attribute

Another feature added to xTed is the Summary attribute which holds the address of the file that contains the summary of the selected node (Fig. 10). This attribute allows authors to specify summaries for their content to be later listened to by users with the Audio Browser. Users select this attribute from the attributes dialog box and may specify the summary file similar to the content file selection.

In the case of existing nets and nets that do not use these new attributes, all simultaneously active nodes are considered to be at the same precedence level and the order among them is assigned by the Audio Browser in an arbitrary fashion and the Audio Browser can operate undisturbed by the absence of the attribute. Likewise, if summary is not available for a particular page, and the user wishes to listen to the summary, the user is simply prompted with a message indicating the absence of a summary.

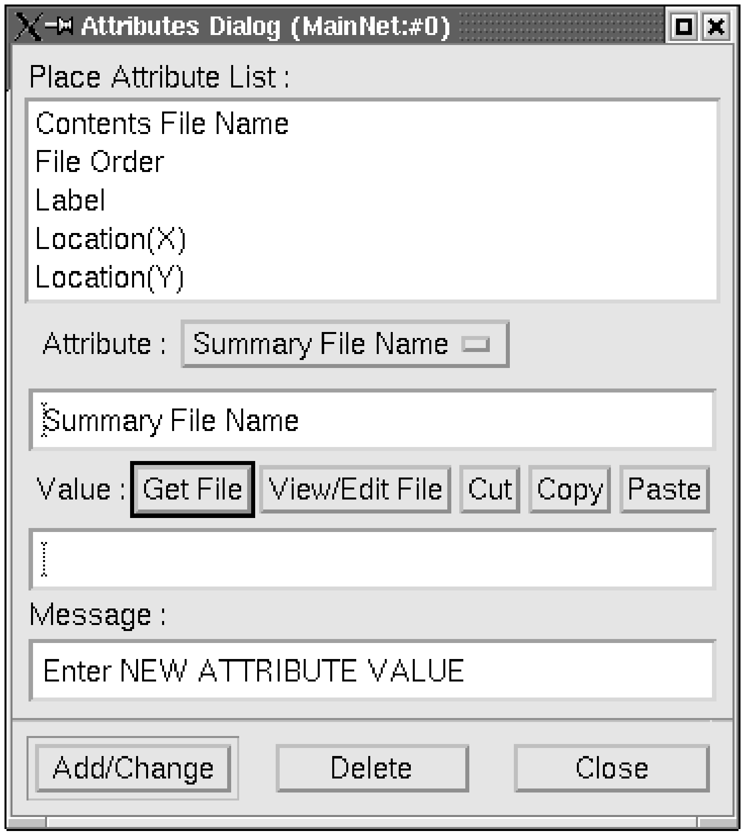


Fig. 10. Setting Summary File Name Attribute

**Architecture**

The Audio Browser consists of three modules (Fig. 11). On top of the architecture is the I/O handler layer, which captures user keystrokes and interprets them for invoking related functions in the service layer, which consists of two separate engines. At the bottom of the architecture is the network layer which communicates with caT to receive and update the requested net structure.

The synthesizer engine uses the festival speech synthesis system in order to synthesize speech from textual resources. The audio playback engine is responsible for the playback of selected nodes, menu items or status information. This engine in turn consists of two modules. The I/O handler captures user commands form the keyboard and calls related functions in the playback module. The playback is achieved through the usage of arts libraries.

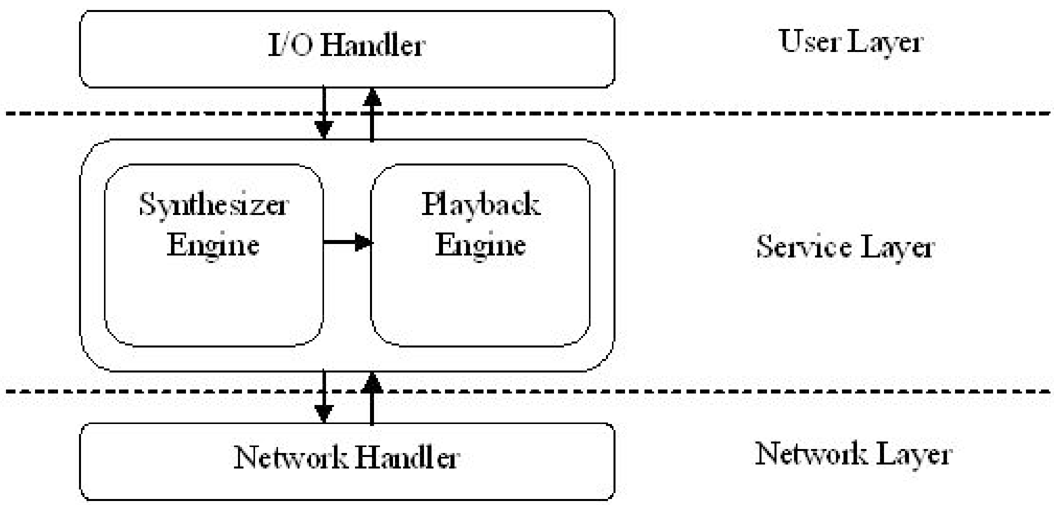


Fig. 11. Audio Browser Architecture

When the browser is invoked from a terminal, it is invoked by typing xtb3 followed by an optional net and server number. The content type of the document to be handled is set to be of SOUND type and the net is traversed page by page in order to determine the active nodes and transitions.

In the first version of the Audio Browser, active nodes were then preprocessed in order to synthesize voice from the contents of the documents. These documents were saved in wav files that had the id of the node as the name of the document in order to preserve the uniqueness of names. After the user evaluation, this structure was changed so that node are processed after the user selects them in order to give users more power over the choice of synthesis voice and rate of speech.

When xtb3 is initiated, the I/O handler level interacts with the user through the initiation of several commands in the net traversal menu. The commands are initiated through the keys in the numerical pad in order to let them be intuitive and be close to each other. These commands can be initiated both to traverse the nodes and to traverse the transitions.

The Audio Browser can handle two types of documents. First, it can display audio files in several different formats such as wav, mp3, and midi. Second, it can synthesize speech from text files. All other types of files are ignored since they can be displayed using different types of available browsers such as the text browser, image browser and the video browser.

When an audio file is encountered as the content of a node, the file is flagged and the contents are kept intact. When a text file is encountered, voice is synthesized with the contents using the Festival speech synthesis program, which converts the contents into wav files to be presented by the playback engine.

The network layer communicates with the underlying Petri engine in order to receive the list of active nodes, transitions, and content and summary file names and locations. It also retrieves the changes to the net when the net is updated and conveys transition firing information to the net.

**Issues and Considerations**

The commands are selected carefully in order to encompass as much functionality as possible while making the system easy to learn and use. Ease of learning and usage are particularly important because due to the nature of the program, the user has to learn the commands and the keys that correspond to them at the beginning and use them.

Especially for normal-sighted users who are used to the self-explanatory buttons and labels found in graphical user interfaces, the lack of these visual representations eliminates all further clues as to what the commands may be. Therefore, the commands are designed to be as intuitive as possible. For this purpose, arrow keys are selected for forward and backward navigation and initial letters of commands are selected as keys for the rest of the commands.

Choice of sounds and voices were selected carefully in order to allow the user to differentiate between notifications and actual listening processes. Therefore, both recorded speech and synthesized voice are used together. Prerecorded speech is used for notifications that may be turned on and off by users, whereas voice is synthesized for all information gathering and content listening purposes.

There are two main functions included in the Audio Browser in order to prevent the users from feeling lost in the system. One of them is the information command, which tells the user where she is and what pages and links are available. If the user is in the navigator menu of the program, it tells that it is in the main section and reads the list of pages and links. If the user is in the audio section of the browser, then the user is presented with the name of the file, how long is the file and how far in the file the listening has processed.

This function in conjunction with the help option is designed in order to decrease the feeling of being lost where there are no visual clues as to where the user might be. The help option in the main and the audio section is to help the user remember the commands. Therefore, the majority of the help menu tells the user about the commands and the keys to issue those commands.

The help menus are also context sensitive in that the system recognizes whether the user is in the browsing or the listening mode and gives descriptions of the commands accordingly. The user may stop listening at any time.

The audio playback engine may be initiated in two modes, being interruptible and uninterruptible. The uninterruptible mode is used when the user is presented with a voice message indicating the issued command or the help menus. The user can stop listening at any given time by pressing Escape but does not have more sophisticated controls such as pausing, replaying, and repositioning, which are only available for the active node contents. This strategy is employed in order to provide better differentiation among pre-recorded voice messages or status information and synthesized node contents.

**Features**

There are two main sections of the audio browser. Once the program is started, the user is in the navigator menu where navigation among pages and links takes place. This is where the user may traverse the net and select particular instances to listen to or links to follow. The second section of the browser is the audio menu where the actual listening of the contents takes place.

Whenever the user enters the navigator section either by initiating the Audio Browser or after listening to contents in the audio section, she is automatically presented with information about the place that she is currently visiting, in order to let the user know that she is in the navigator menu and inform her about the available pages and links in that position.

The position information is presented to the user with synthesized voice that is created at the time of user request and may be interrupted at any time by the user. The user may also listen to the lists individually by pressing the appropriate keys.

When the list of pages is selected in the navigator section, the synthesized voice says the number of the page followed by the label of the corresponding node in the caT net. The number of the page corresponds to the relative position of the node within the active nodes and is calculated according to the order information specified in xTed.

Similar to the list of pages, the list of links consists of the numbers of links followed by the labels of the corresponding transitions in xTed. The difference is that the numbers are arbitrary since there is no ordering among transitions. They are included in order to provide numerical selection of links as an alternative selection method.

Navigation within the net is realized by the selection of pages and links. For this purpose, the navigator menu provides two selection mechanisms to the user. The arrow keys and the numerical keys are the two ways of selecting a page to listen to or a link to follow. While the arrow keys allow the user to navigate between pages and links one by one, the numerical keys allow the user to select pages and links directly.

The numerical key selection is based on the number of the page or link as presented in the lists. This requires that the user remember the number of the page or link to select that element. Therefore, it is suggested that the nets should be designed carefully so that the number of pages and the number of links active at any given time should not be minimal. [Miller 1956] states that short-term memory has the capacity to hold 7 plus or minus 2 items. More recent studies have even reduced this number to 4 plus or minus 2 [Cowan 2000]. Moreover, studies find that the first and last items have a better chance to be remembered than the middle ones [Madigan et al. 1992]. This shows that human short-term memory capacity and the remembering characteristics have to be taken into account when designing nets.

Also the length of the list puts more pressure on the users who need to listen to the links and nodes carefully in order to keep track of their numbers. However, the system at hand does not put any predefined limit on the number of links or pages since it is likely to integrate this system into preexisting nets, such as the Internet and there is no limit on those nets. However, it is suggested that future nets be created with this in mind.

In the current version, there is no correlation between the audio and the position within the read contents. Therefore, there is no way of knowing what link or page is read to the user at any given time. This prevents the system to have the feature of selecting links or pages as they are read by the synthesized voice. It is planned in the future that as the user listens to the list of pages and links, she should be able to select one item by just pressing the return key.

Link selection corresponds to the Petri net transition firing in caT and takes the user to a different place in the net structure. Therefore, the user is again presented with the page and link information related to the new position immediately following the transition.

Page selection takes the user to the audio menu, where browsing within a page is provided. In this menu, the user may continue listening. At the end of the listening the system waits for user input where the user may select to exit the listening and go back to the navigation menu or choose to listen to the contents again. The audio browser also gives more power to the listeners by allowing them to control the listening process.

The user may pause, replay or stop listening at any time and while listening, may chose to go forward and backward in the audio file. The unit of going forward or backward is a percentage of the audio file, which can be altered with commands provided by the audio browser. Since currently there is no correlation between the playback and the structure of the actual content file, the unit of skipping was set as a percentage. In the future versions of the browser where the relation between the playback and file is established, then the unit of playback may be set as word, sentence, or line within that file. Especially word by word traversing within the document may increase the perception of the users in cases where the user may wish to listen to a word that she does not understand or wishes to listen to the whole sentence again.

It might also be the case that if the system can understand the structure of the document that is played such as the page information within PDF and PS files, the user may be able to use the functions provided within those programs such as going page by page, a certain number of pages or going to the beginning or end of the document.

Another feature of the audio browser is the summary option that allows users to listen to a summary of the content page if the summary is made available in xTed by setting the Summary File Name attribute. Currently, this summary is created by the author and there is no automation within that creation.

**Commands**

The commands are categorized according to where they may be issued. This results in a division into commands in the navigator menu, audio menu, and those that are common to both. The common commands are those that allow users to change user settings and receive help about commands and current position. The menu specific commands are navigation commands in the navigator menu and playback control commands in the audio menu. Since there is a distinction between commands in different menus, the same keys may correspond to different commands in these menus.

Table I. Common Commands and Keys

| *Key* | *Command* |
|---|---|
| H, h | Help |
| I, I | Information |
| T, t | Notification on/off |
| Esc | Exit |

Common commands and their corresponding keys are shown in Table I. The first of the common commands is “help”, which is issued by the key ‘h’ selected as the initial letter of the command. Although the command is common, help is context sensitive and gives help on the commands depending on the menu that the user is in. When the user is in the navigator menu, help presents information about the commands that are available in that section of the Audio Browser. When the user is in the Audio Menu, pressing ‘h’ results in the presentation of help on the audio playback commands. Both these help menus also include help on the common commands. Help is provided as synthesized voice compiled from the written description of the commands at runtime.

The second common command is the information command, issued by pressing the ‘i’ key on the keyboard. This is also context sensitive and provides information that is specific to the browser’s state. When the user is in the navigator menu, selecting the information command is followed by the presentation of a synthesized voice saying the total number and the lists of available pages and links. This is the command that is issued automatically when the user enters the navigator menu. When the user is in the audio menu, issuing the information command gives the name and size of the selected page followed by where the listening process is within that page. In other words, the command lets the user know what percentage of the file she has listened to.

Another command that is available in both the menus is the toggle command that turns extra notifications about navigation on and off. In order to give better sense of direction and avoid confusion in the space, the user is presented with the confirmation of every action that she performs. This might be too time-consuming for expert users, whether they access the system remotely or they are visually impaired. In that case, this key allows them to toggle between extra notifications and limited notification. When the notifications are turned off, the user is not given the audio feedback on the selected action such as “next page”, or “previous link”.

In the navigator menu of the Audio Browser where navigation takes place, the available commands are related to listing and selection of links and pages (Table II).

Table II. Navigator Menu Commands and Keys

| *Key* | *Command* |
|---|---|
| Right arrow | Next page |
| Left arrow | Previous page |
| Up arrow | Next link |
| Down arrow | Previous link |
| P, p | List of pages |
| L, l | List of links |
| S, s | Summary |
| Return | Select page or link |

The right and left arrow keys are used in order to go from page to page. The left arrow allows going to the previous page, and the right arrow key goes to the next page. As stated previously, this order is determined by the File Order attribute of the node defined in xTed.

When the user is at the beginning of the list of pages, if she selects the left arrow key to go to the previous page, although there is no page to go back to, the pointer goes to the end of the list and the page at that location is presented as the previous node. Similarly, when the user tries to go to the next node while she is at the last page, the list points back to the beginning of the list. Initially, these operations were prohibited and the user was presented with a beep to inform about the invalid command. However, the user evaluations showed that it was better to present the list of pages as a circular list since the beeping confused some of the users.

When the user selects either of the arrows, she is prompted with a voice recording saying which action she has selected followed by the name of the page. Therefore, pressing the left and right arrows causes a voice to say “Previous page”, and “Next page”, followed by the name of the previous and next page, respectively. The usage of prerecorded audio shows that these notifications may be turned off and on by the user when necessary.

Link navigation is similar to the page navigation. This time, up and down arrow keys are used in order to go from link to link. The up arrow allows going to the next link, and the down arrow key goes to the previous link. The order among the links is arbitrary and the previous and next relationship follows the one in the list of links.

Similar to the list of pages, the list of links is presented as a circular list to the user so that the user may select the previous link when she is already at the beginning or the next link when she is at the last. In those cases, she goes to the last link or the first link, respectively. This is also a divergence from the beeping solution in order to minimize the amount of discomfort that the users may have with beeping sounds.

When the user presses the up arrow key to go to the next link, she is presented with a prerecorded audio saying “Next link” followed by the synthesized voice saying the name of the next link. Likewise, when the user presses the down arrow key to go to the previous link, she is presented with a prerecorded audio saying “Previous link” followed by the synthesized voice saying the name of the previous link. As with the usage of recorded voice, the “next link” and “previous link” notifications may be turned on and off using a toggle command.

When the user wishes to listen to any of the active nodes, the file location is passed to the synthesizer engine which synthesizes voice from the file if the file is a text file. If the file is already an audio file, the contents are kept intact. The synthesizer engine then calls the playback engine with the location of the audio file.

When the audio playback engine is called, the playback of the file starts immediately in order to save the user from having to issue more commands. After the start of the playback, the user has several options to control the playback (Table III). In order to keep the parts of the browser similar to each other, arrow keys are used for these control functions. This also results in a lesser number of commands to be memorized.

Table III. Audio Menu Commands and Keys

| *Key* | *Command* |
|---|---|
| Right arrow | Forward |
| Left arrow | Backward |
| Up arrow | Increase skip time |
| Down arrow | Decrease skip time |
| P, p | Play / Pause |

The left and right arrow keys are used in navigation within the content that the user is listening to. The left arrow moves the file pointer to go back a fraction of the file, and the right arrow goes forward a fraction of the file. This fraction is initially set to be ten percent of the file. Therefore, when the user hits the left arrow, this repositions the file pointer to a point in file that is ten percent in the back. When the user selects a seek value that extends past the limits of the actual file, then the file pointer is moved to the beginning or the end and the user hears a beeping sound in order to notify that there is no more content to seek back or forward.

Percentage values in going forward or backward in the file is based on the total duration of the files. This is selected to be the unit of seek since the duration of the files may vary significantly and the user might be presented with too long or too short jumps depending on the actual duration if absolute seek values were employed.

The granularity of the seek unit can be altered by using the Shift and arrow combinations as in transitions. When the user presses Shift and the up arrow key at the same time, a prerecorded voice is displayed by the audio player saying “Make seek time longer”. When the down arrow key is pressed simultaneously with the Shift key, the user hears “Make seek time shorter.” The seek times are always percentage values which can be 1, 5, 10, 20, or 25 percent.

When the user presses the ‘p’ key, the state of the audio playing is checked to see if the audio is currently playing or paused. If the audio is playing, then it is paused. Similarly, when it is paused, it continues playing from the point it was paused. This allows the user more control on the audio playback.

Playing of the audio can be terminated in two ways. First is the actual termination of the audio file and second is the deliberate termination by the user. When the user wishes to terminate the audio playback, she presses Esc, which stops the playing of the audio and releases the resources.

When the playing is over, the user is informed with a prerecorded message that the end of file is reached. However, control is not returned back to the Navigator Menu and the user stays in the Audio Menu until she presses Esc and exits that section. This allows the user to replay the audio without having to select the page again.

As stated earlier, one of the most important differences between the Audio Browser and other browsers such as the text and the image browsers is that in visually-based browsers, any number of windows can be active at any given time depending on the number of active pages. The only concern is to place these windows in different places on the screen so that they do not completely overlap and they are more visible for the user. Because of this difference, in those browsers, the update operation is also different.

When the update function is called, the browser keeps track of the list of presented documents and the new list is compared against this previous lists. If a document is not present in the current list, then the application that is displaying is killed and the slot representing its place on the screen is made available so that a new one can be placed in its place.

In the Audio Browser, however, only one document is active at any given time and since all operations are handled from the keyboard and there is no visual representation, there is no need to keep track of where that document is displayed. When an update in the net is encountered, user is warned of the change in the net without interrupting the current listening operation. After the listening operation is over, the user is prompted of the change in the net and this change is propagated to the browser.

The update operation is one of the most important and complicated parts of the system. As described in the previous chapter, the files are preprocessed in order to prevent the system from synthesizing the same files over and over again. However, both periodically, and when an update occurs, the changes in the files need to be considered in order to prevent an older version of the file from being read to the user. In this case, the last modified date of the file is compared to the creation date of the corresponding wav file and if the file has been modified after the audio file is created, then the file is synthesized again.

# CHAPTER VI

# USABILITY EVALUATION

The Audio Browser was designed with the users' ability to learn and interact with the system in mind and selection of features and commands were determined according to usability criteria. The usability of the resulting system was then tested by Texas A&M students who had no prior knowledge of the system. The tests were carried out individually in an informal manner in order to determine the comfort level of new users with the system and determine their satisfaction related to specific parts of the Audio Browser.

## Expectations

The study was conducted in order to observe new users and find out the ways to make it more user-friendly and easy to learn and use. Users were observed and interviewed in order to determine their overall satisfaction with the Audio Browser, as well as the more specific thoughts and suggestions about the modules within the Audio Browser.

One of the most important aspects of the Audio Browser is the choice of commands, which were selected carefully in order to minimize the number while providing all the necessary functionality. Therefore, one purpose of the usability test was to determine if the choice of commands makes them easy to remember and the assignment of keys to those commands are received favorably by both normal-sighted and visually-impaired users.

Especially for untrained ears, synthesized voice requires some time to get used to the quality of the audio. Another purpose of the usability test was to see how users react to the synthesized voice and whether this reaction differs in visually-impaired and normal-sighted users. One of the most important disadvantages of the audio browsing is that it requires additional means of information in order to prevent users from feeling lost in the hypertext. Although it is fairly easy in visual hypertext to give an idea of where the person is, audio requires users to have a clear understanding of current position in the lack of visual cues.

Since the program was run by users who learn about the browser for the first time, they were likely to make mistakes and try to recover from them. The tests were expected to show how the program would react to unexpected input in order to find programming bugs if there were any.

The free-form conversations with the interviewer were to reveal if there were any additional suggestions for the Audio Browser. These are likely to be incorporated into the future versions as new features.

**User Characteristics**

The usability tests were carried out with nine Texas A&M students. Eight of the students were contacted directly by the principal investigator and one blind student was contacted through the Adaptive Technology Services Coordinator, David Sweeney. The students were selected specifically to represent a wide range of A&M students, being graduate, undergraduate, male, female, from different majors, ethnic and national backgrounds. Therefore, the study was expected to show an unbiased picture of potential users of the system.

At the beginning of the usability test, users were asked to fill out eight questions on a questionnaire about their background related to the study. As seen in Table IV, all users had used computers and accessed the Web for more than two years, and with the exception of User#4, all were using a computer and accessing the Web almost every day. User#4 said that she would use the computer and access the Web weekly.

Table IV. User Characteristics

| *Question* | *Answer* | *Response* |
|---|---|---|
| How long have you used computers? | Less than 6 years | |
| | 6 months to a year | |
| | A year to 2 years | |
| | More than 2 years | 9 |
| How often do you use a computer? | Daily (almost every day) | 8 |
| | Weekly (2 to 3 times a week) | 1 |
| | Monthly (2 to 3 times a month) | |
| | Less than once a month | |
| What type of computer do you use? | Apple Macintosh | |
| | Windows 95/98/ME/2000/NT/XP | 9 |
| | Unix (Sun, HP, Linux, SGI, etc.) | 4 |
| | Other | |
| How long have you used the WWW? | Less than 6 years | |
| | 6 months to a year | |
| | A year to 2 years | |
| | More than 2 years | 9 |
| How often do you access the WWW? | Daily (almost every day) | 8 |
| | Weekly (2 to 3 times a week) | 1 |
| | Monthly (2 to 3 times a month) | |
| | Less than once a month | |
| What browser do you use to access the WWW? | Netscape | 5 |
| | Internet Explorer | 8 |
| | Other | |

In general users seemed comfortable with computers and browsing among pages. Only User#2 and User#4 said that they were not "computer savvy" and were initially more hesitant to experiment with the application. The majority of the users were using the Windows operating system and Internet Explorer to access the Web. Prior to the usability test, none of the users had used hypermedia systems other than the Web and the only person who had used speech synthesis programs was the blind student.

**Test Procedure**

A sample net was prepared for the usability test in xTed (Fig. 12) based on the Texas A&M University Department of Computer Science web site.

The description of the caT system given to users was simplified since the purpose of the test was to evaluate the usability of the Audio Browser and not the caT system as a whole. Extra features of caT that were available, such as context-awareness, different number of tokens, and auto-fire, were not described.

The nodes in the system were described as similar to Web pages on the Internet and the transitions were described as the links between them. Although this seemed similar to the Web, there are two major differences. First of all, the links are not part of the pages themselves but rather imposed on the pages. This gives more flexibility to the links. The page needs not be authored again in order to add, delete, or change a link. And anyone may easily change links without having to be the author of the page.

The flexibility that is provided to the links also allows links to have multiple anchors and targets. This is the second major difference. At any given time, there is not one page but a set of related pages that are available to the user and the user is able to select any one of them to listen to its contents.

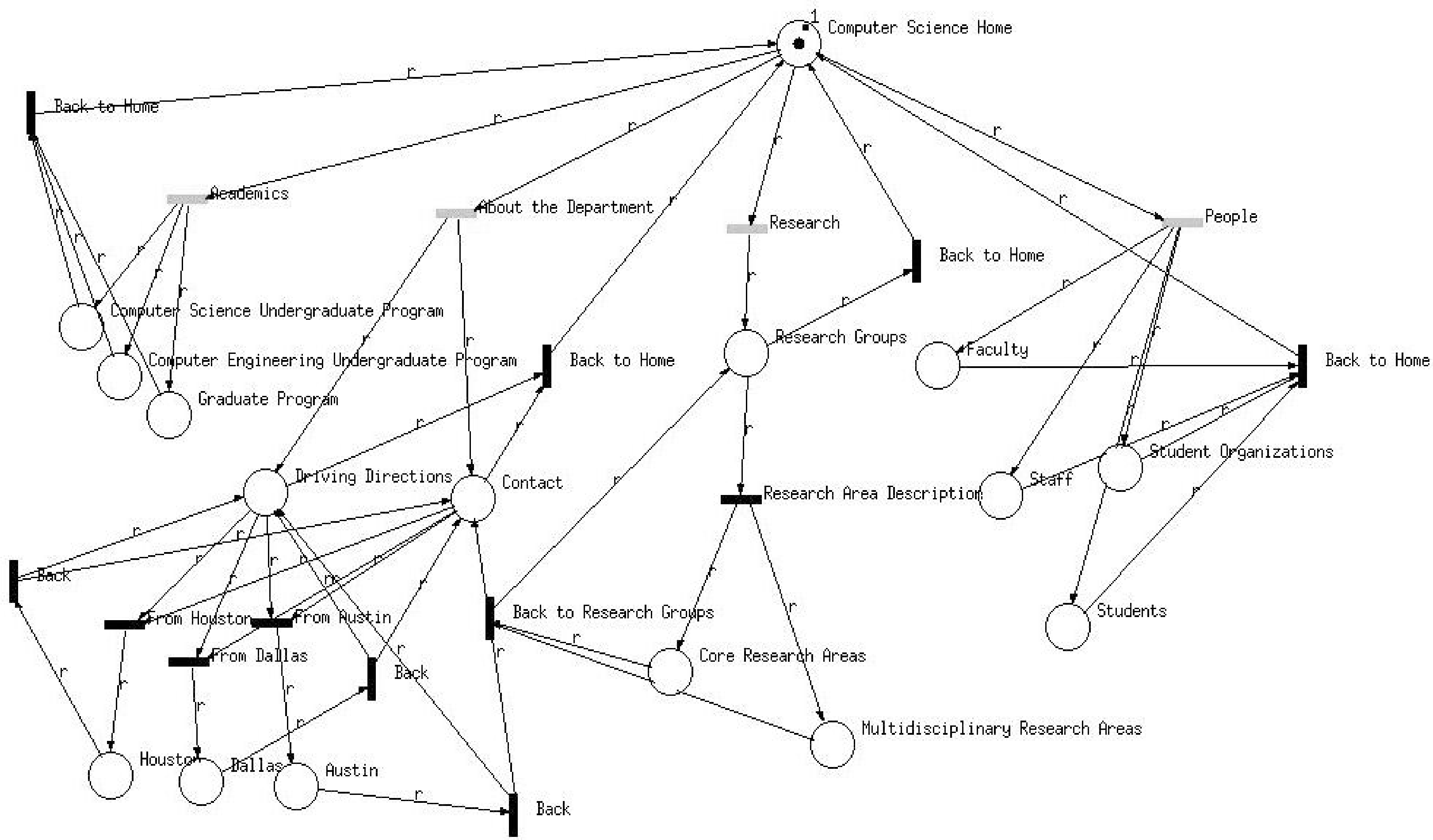


Fig. 12. Test Net

After the explanation of the caT system and how it is similar to and different than the Web, the Audio Browser was introduced. Its purpose and features were described followed by a demonstration by the principal investigator on the test net. Commands were explained and used while following links to the page containing the driving directions from Austin and listening to the contents.

Following the initial training session, users were asked to experiment with the system themselves while they were observed and their questions were answered. When they felt comfortable with the system, the visual representation on the xTed window was turned off and the users were requested to find and listen to the graduate program information and driving directions from Dallas. At the end of the tasks, users filled out the evaluation section of the questionnaire, which asked questions about their satisfaction with the software.

# CHAPTER VII

# EVALUATION RESULTS, ANALYSIS AND DISCUSSION

## Overall Satisfaction

Users were observed during their interactions with the Audio Browser, interviewed and asked to fill out the evaluation section of the questionnaire that is available in Appendix A. In the first six questions, users were requested to evaluate the statements in questions using a scale where 1 corresponded to strongly disagree, 3 to disagree, 5 neither agree nor disagree, 7 agree and 9 strongly agree. One question asked them to compare this browser to any other similar browsers and the last four questions allowed users to express their satisfaction in free form.

### Satisfaction of Normal-Sighted Users

Tests with the eight normal-sighted users lasted between half and hour to an hour and in general, the Audio Browser was favorably received. From the interviews and their responses to the questionnaire (Table V), the users found the system simple and effective. Some of the users suggested new usage areas such as the integration of the Audio Browser and an in-vehicle GPS system by User#3.

### Satisfaction of the Blind User

The usability test with the blind user lasted for ninety minutes and more than thirty minutes was spent on explaining the underlying caT system and the difference between pages and links.

The initial training with the sighted users was carried out with the advantage of the visual representation of the net that they were traversing in xTed. This proved to be significantly important in the understanding the system and familiarizing with the system. In the case of the blind user, the initial oral description was insufficient and the user needed significantly more time to understand the system and how it differs from the Web. The user was also interested in why this system was selected instead of the WWW and comparison of these two systems.

The blind user was definitely the most enthusiastic user and was thrilled with the concept of having an open-source audio browser available for Linux. He expressed interest in working with the development team in order to enhance the Audio Browser. He was interested if the product would become marketable and was also interested in using the system if it became available.

He found the currently available features useful, but different than the rest of the users, he asked for more advanced features for expert users. Some of these features were later added to the system so that they can be turned on and off by the user at any given time.

Table V. Evaluation of Normal-Sighted Users

| *Evaluation Questions* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* |
|---|---|---|---|---|---|---|---|---|---|
| The training provided an overview of the Audio Browser and how it can be used. | | | | | | | | 3 | 5 |
| The training was sufficient to get an idea of all the components and use them in tasks. | | | | | | 1 | | 2 | 5 |
| The commands were easy-to-learn. | | | | | 1 | | 2 | 2 | 3 |
| It was easy to use the Audio Browser to perform the tasks. | | | | | | | 3 | 5 | |
| The software runs properly (bug-free). | | | | | | | 3 | 3 | 2 |
| The Audio Browser is useful for hypertext navigation. | | | | | | 1 | 1 | 2 | 4 |

**Analysis of the Audio Browser**

Commands

Commands proved to be the hardest part to get used to for the users and the most important reason was not the actual difficulty of the learning but the difference between pages and links.

Some of the users used the page and link listings extensively and expected to do the selection from that listing. Although, it was expected that the usage of arrows would be more intuitive, it was surprising to find that the numerical selection was preferred by the majority of the users.

Visually normal-sighted users were instantly comfortable with the convention that the keys are the initial letters of the commands that they represent. Only the blind user thought that the arrows were far away from the letters and suggested that either the numerical pad should be used without the initial letter convention or some letters should be assigned to work as arrow keys. He also suggested the usage of a mouse or a joystick, which would be more difficult to learn for average normal-sighted users but would be beneficial for people with one hand or with those who are used to using these systems.

Navigation Quality

There was a difference in the navigation quality of the users. The sighted users had the advantage of visualizing the system before having to use audio clues only. Actually, after they started using the program, they said that the visual representation was confusing them and that it was better to turn it off completely and found it easy to navigate within the system. This shows that even after a short period of time with the Audio Browser, users can easily navigate within hypermedia.

The blind user had more difficulty getting used to the navigation since he tried to think of a Web-like system and had to spend more time to understand the structure of the net and what pages and links are in particular. After he understood the underlying structure, he was very comfortable browsing the system and he was one of the first to realize that it was the Department of Computer Science site that he was browsing.

Getting Help

Users did not express feelings of being lost in the hypertext. When they did not know where they were, most of the users preferred using the list of links and pages and two users tried to listen to the help menu. One user suggested that the link and page information be read to the user whenever it was the beginning of the program or when there was no action. This feature was added to the system after these comments.

Navigator and Audio Menu Transitions

In the evaluation system, the audio listening was set such that the system would return to the main menu after the end of normal listening. Then it was possible to re-listen to it if that was desired. The blind user suggested that the user stay in that menu until she wants to get out manually. On the other hand, normal sighted users did not have problems with the end of listening probably because they were more interested in the navigation concepts than the actual listening process.

Synthesized Voice Quality

It was agreed that users got used to synthesized voice as they listened to more contents and this shows that even normal-sighted users who have no prior speech synthesis experience can use the system and easily understand the contents.

The selection of the voice was criticized by some users and they wanted to know if there were more voices available. They were given a list of voices to choose from and the female voice from the MBrola library was liked most by the users. The most important concern was how continuous the speech was. The more continuous the speech was, the better they could comprehend.

The blind user also cared about the accent of the speaker and said that he would not be comfortable with the British accent. He also wanted the ability to be able to change the voice at runtime. The addition of this ability is in process.

Rate of Audio Playback

The speed of speech was set at a speed that was faster than most telephone applications and slower than most screen readers, such as Jaws in order to make it more understandable for normal-sighted people. The only concern was for the blind student who had prior experience with screen readers. However, the study showed that the blind user was comfortable with the speed of the speech and did not feel any different than Jaws. He did not think that it was slow and realized that it was slower only after listening to a sample speech from Jaws.

**Discussion**

The training session proved that it was very important to familiarize the users with the system before actually starting to experiment with the browser software. Users' perception of the system was directly related to the time of the initial training session. This was especially apparent in the case of the blind user who

Although it was assumed that the two differences, being the availability of multiple pages and the separation of links from pages, seemed to be minor from the perspective of first-time users, it proved to be a major issue for the users. If the system is not going to be used as a front end to a Web application but instead to the caT system, then it is necessary to explain the system very carefully in order for the commands to be well understood and used.

Some of the users had a hard time understanding the difference between pages and links. This was caused by the multiple nature of both of pages and links. Understanding was harder for users with lesser computer knowledge. Users that were used to opening multiple browser windows or tabs could understand the multiple pages concept better.

The study shows that the browser is welcomed by new users and the majority find it very useful. Some of the users thought of new places to use the browser and some empathized with blind users. The commands are easy to learn and use. The notification voices were very welcome. The users familiarized themselves with the system approximately in half an hour. This shows that the system is easy to learn and use.

The study was also very productive in user suggestions. The voice quality was found to be satisfactory. Different voices were presented and the one most liked was later set to be the default. An expert user feature was suggested to allow the runtime change of synthesized voice choice.

As a result, the program was found to be useful. The blind user especially was very enthusiastic about the browser. The majority of the suggestions was related to minor modifications and have been changed or added after the usability test.

# CHAPTER VIII

# FUTURE WORK AND CONCLUSIONS

There are two major areas where this research may be extended to make the system a novel sophisticated browser for wide-spread use. The first one is the establishment of relationship between the playback of the audio and the pointer on the file that is being played. The second one is the intelligent analysis of textual content for better presentation.

In the current version of the Audio Browser, there is no direct relationship between the synthesized voice and its corresponding file once the synthesis is over. Therefore, at the time of audio playback, there is no way of knowing where the file pointer is at any given time. This is due to the lack of such provision in the Festival speech synthesis module.

It is suggested that further research should be conducted in order to develop an algorithm that would calculate the file pointer position from the analysis of file content and current playback position. This will involve parsing of the file and language analysis to determine approximate rendering time of file units. Another approach would be to place markers in the file so that each marker represents a unit within the file, such as a word, sentence or a line. With this approach, it would be possible to jump to different sections of the file without having to synthesize the file again.

The addition of this backward direction from audio to file brings several interesting opportunities. One of them is the selection of pages and links in the Navigator Menu. Currently, the fastest way for the user to select a particular page or a link is to listen to the related list and type the number of the page or link. Unless the user remembers all the numbers from listening to the list once, this procedure adds an additional step of typing the number before the actual listening or link following can take place. If the backward direction is established, this would allow the users to select the page or link directly from the list since the user response time would point to the desired page or link in the list file.

Establishment of this direction from audio to text would also be beneficial in the Audio Menu. In that case, when the user changes the rate or the voice, the position of the file pointer would be calculated in order to synthesize the rest of the file according to the new preferences and listening would resume with the new settings from where it was stopped.

The second area of improvement is about the analysis of textual structure. In the current version, when the requested file is synthesized, no information regarding the file structure is used or computed. Therefore, the file is considered as a chunk rather than a combination of smaller units such as words or sentences. If this information can be extracted from the file and combined with the backward direction from audio to file, then the more advanced features such as going backward and forward in units rather than percentages may be possible as was suggested by the blind user.

Enhanced text analysis would also enable parsing of structured documents such as HTML, and XML files. If the structure of these files can properly be examined and content extracted from them, then any Web page would be available from our Audio Browser without having to change any content. Considering the wide adoption of Web all over the world, this would be beneficial for everyone wishing to combine the Web with the capabilities of the Audio Browser.

Another future work suggestion is to combine the text analysis with natural language processing, which would be useful in two major aspects of the Audio Browser. First, this would allow the system to determine the importance of specific parts of the textual content and present the user with a more condensed yet understandable and meaningful portion of the content, which would decrease the amount of time required to listen. Second, this may be used as an alternative to human-created summaries.

Even without the suggestions added to the system, the Audio Browser is very promising as the audio solution for complex, context-aware hypermedia nets. It provides an easy-to-learn interface, high-quality speech synthesis and control over the listening process, which allows the user to interact with the system. The user evaluations show that both visually-impaired and normal-sighted users find the product useful and are satisfied with the design decisions that were made. It is expected that the Audio Browser would become the choice for audio-only browsing.

# REFERENCES

ARONS, B. 1993. SpeechSkimmer: Interactively skimming recorded speech. In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology.* ACM Press, New York, 187-196.

ARONS, B. 1997. SpeechSkimmer: A system for interactively skimming recorded speech. *Transactions on Computer-Human Interaction (TOCHI). 4*, 1, 3-38.

COWAN, N. 2000. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*. *24*, 87-185.

EDWARDS, W. K., AND MYNATT, E. D. 1994. An architecture for transforming graphical interfaces. In *Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology*. ACM Press, New York, 39-47.

FESTIVAL SPEECH SYNTHESIS SYSTEM (FSSS). 2003. http://www.cstr.ed.ac.uk/projects/festival/

FURUTA, R., AND NA, J.-C. 2002a. Applying caT's programmable browsing semantics to specify World-Wide Web documents that reflect place, time, reader, and community. In *Proceedings of the ACM Symposium on Document Engineering*. ACM Press, New York, 10-17.

FURUTA, R., AND NA, J-C. 2002b. Applying programmable browsing semantics within the context of the World-Wide Web. In *Proceedings of the 13th Conference on Hypertext and Hypermedia.* ACM Press, New York, 23-24.

HE, L., SANOCKI, E., GUPTA, A., AND GRUDIN, J. 1999. Auto-summarization of audio-video presentations. In *Proceedings of the 7th ACM International Conference on Multimedia*. ACM Press, New York, 489-498.

JAMES, F. 1996. Presenting HTML structure in audio: User satisfaction with audio hypertext. In *Proceedings of ICAD'96*, 97-103.

KARADKAR, U. P., NA, J.-C., AND FURUTA, R. 2002. Employing smart browsers to support flexible information in Petri net-based digital libraries. In *Proceedings of* *the 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2002).* ACM Press, New York, 324-337.

MADIGAN, S. AND O'HARA, R. 1992. Short-term memory at the turn of the century: Mary Whiton Calkins's memory research. *American Psychologist*, *47*, 170-174.

MBROLA. 2003. The MBrola Project Homepage http://tcts.fpms.ac.be/synthesis/mbrola.html

MILLER, G. A. 1956. The magical number seven plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, *63*, 81-97.

MORLEY, S., PETRIE, H., O'NEILL, A.-M., AND MCNALLY, P. 1998. Auditory navigation in hyperspace: Design and evaluation of a non-visual hypermedia system for blind users. In *Proceedings of the 3rd International ACM Conference on Assistive Technologies.* ACM Press, New York, 100-107.

NA, J.-C., AND FURUTA, R. 2001. Dynamic documents: Authoring, browsing, and analysis using a high-level Petri net-based hypermedia system. In *Proceedings of the ACM Symposium on Document Engineering.* ACM Press, New York, 38-47.

OMOIGUI, N., HE, L., GUPTA, A., GRUDIN, J., AND SANOCKI, E. 1999. Time-compression: Systems concerns, usage, and benefits. In *Proceedings of ACM Conference on Human Factors and Computing Systems*. ACM Press, New York, 136-143.

PWWEBSPEAK. 2003. http://www.soundlinks.com/pwgen.htm

RAMAN, T. V. 1996. Emacspeak-direct speech access. In *Proceedings of the 2nd Annual ACM Conference on Assistive Technologies*. ACM Press, New York. 32-36.

SPROAT, R., TAYLOR, P., AND ISARD, A. 1997. A markup language for text-to-speech synthesis. In *Proceedings of Eurospeech*. 1747-1750.

STOTTS, P. D., AND FURUTA, R. 1989. Petri-net-based hypertext: Document structure with browsing semantics. *ACM Transactions on Information Systems*. *7*, 1, 3-29.

VOICEXML. 2003. The Voice Browser Activity http://www.w3.org/Voice/

XINE. 2003. xine – A Free Video Player http://xinehq.de/

XV. 2000. Official Homepage of xv
http://www.trilon.com/xv/xv.html

# APPENDIX A

## AUDIO BROWSER EVALUATION

**Background Information:**

---

**Instructions:**

- Please select the answer that you think is the best answer for a given question.
- Please do not answer a question if it makes you uncomfortable or you would not like to answer it for any other reason.

---

1. How long have you used computers?
   a. Less than 6 months
   b. 6 months to a year
   c. a year to two years
   d. more than two years

2. How often do you use a computer?
   a. Daily (almost every day)
   b. Weekly (2 to 3 times a week)
   c. Monthly (2 to 3 times a month)
   d. Less than once a month

3. What type of computer do you use? (Please select all that apply)
   a. Apple Macintosh
   b. Windows 95/98/ME/2000/NT/XP
   c. Unix (Sun, HP, Linux, SGI, etc.)
   d. Other (Please Specify) ____________________________________

4. How long have you used the World Wide Web (WWW)?
   a. Less than 6 months
   b. 6 months to a year
   c. a year to two years
   d. more than two years

5. How often do you access the World Wide Web (WWW)?
   a. Daily (almost every day)
   b. Weekly (2 to 3 times a week)
   c. Monthly (2 to 3 times a month)
   d. Less than once a month

6. What browser do you use to access the WWW? (please select all that apply)
   a. Netscape
   b. Internet Explorer
   c. Other (Please Specify) ________________________________________

7. Have you used other hypertext/hypermedia systems? ___ Yes ___ No

If yes, please specify _________________________________________________________

______________________________________________________________________________

______________________________________________________________________________

8. Have you used speech-enabled applications? ____ Yes ____ No

If yes, please specify _________________________________________________________

______________________________________________________________________________

______________________________________________________________________________

**Evaluation**

---

**Instructions**

- Please select the answer that you think is the best answer for a given question.
- Please do not answer a question if it makes you uncomfortable or you would not like to answer it for any other reason.
- Please answer all questions using the following scale:

| Strongly Disagree | | Disagree | | Neither Agree Nor Disagree | | Agree | | Strongly Agree |
|---|---|---|---|---|---|---|---|---|
| ↓ | | ↓ | | ↓ | | ↓ | | ↓ |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |

---

1. The training provided an overview of the Audio Bowser and how it can be used.

   1 2 3 4 5 6 7 8 9

2. The training was sufficient to get an idea of all the components of the Audio Browser and use them in tasks.

   1 2 3 4 5 6 7 8 9

3. The commands were easy-to-learn.

   1 2 3 4 5 6 7 8 9

4. It was easy to use the Audio Browser to perform the tasks.

1 2 3 4 5 6 7 8 9

5. The software runs properly (bug-free).

1 2 3 4 5 6 7 8 9

6. The Audio Browser is useful for hypertext navigation.

1 2 3 4 5 6 7 8 9

7. How would you rate your experience with the Audio Browser and any other similar software that you have used? (Select the number that best matches your level of satisfaction)

| Software | Extremely Dissatisfied ↓ | | Dissatisfied ↓ | | Neither Satisfied Nor Dissatisfied ↓ | | Satisfied ↓ | | Extremely Satisfied ↓ |
|---|---|---|---|---|---|---|---|---|---|
| Audio Browser | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| _______________ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| _______________ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| _______________ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| _______________ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| _______________ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

8. What are the best aspects of the Audio Browser?

9. What are the worst aspects of the Audio Browser?

10. Will you use the Audio Browser if it is available to you? _____ Yes _____ No

If no, why?

11. Please provide your suggestions regarding how the Audio Browser may be improved.

# VITA

Selen Ustun was born in Istanbul, Turkey on August 2, 1979. She earned her bachelor's degree in computer engineering from Bogazici (Bosphorus) University, Istanbul, Turkey in 2001. She enrolled in the Department of Computer Science at Texas A&M University, College Station in 2001 and started her research under the guidance of Dr. Richard Furuta. She earned her master's degree in computer science from Texas A&M University, College Station in 2003. Her permanent address is:

Nispetiye Cad. Toren Sok.
Ozkonak Apt. 13/12
1. Levent 80600
Istanbul, Turkey