

IBM Research Report

The World Wide Telecom Web Browser

Sheetal K Agarwal

IBM India Research Lab
4C, Institutional Area, Vasant Kunj
New Delhi-110070, India

Anupam Jain

IBM India Research Lab
4C, Institutional Area, Vasant Kunj
New Delhi-110070, India

Arun Kumar

IBM India Research Lab
4C, Institutional Area, Vasant Kunj
New Delhi-110070, India

Amit Anil Nanavati

IBM India Research Lab
4C, Institutional Area, Vasant Kunj
New Delhi-110070, India

Nitendra Rajput

IBM India Research Lab
4C, Institutional Area, Vasant Kunj
New Delhi - 110070, India

IBM Research Division

Almaden - Austin - Beijing - Delhi - Haifa - T.J. Watson - Tokyo - Zurich

LIMITED DISTRIBUTION NOTICE: This report has been submitted for publication outside of IBM and will probably be copyrighted is accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties). Copies may be requested from IBM T.J. Watson Research Center, Publications, P.O. Box 218, Yorktown Heights, NY 10598 USA (email: reports@us.ibm.com).. Some reports are available on the internet at <http://domino.watson.ibm.com/library/CyberDig.nsf/home> .

The World Wide Telecom Web Browser

Sheetal K Agarwal, Anupam Jain, Arun Kumar, Amit Anil Nanavati, Nitendra Rajput
IBM India Research Lab
4C, Institutional Area, Vasant Kunj
New Delhi, India

sheetaga,anupamjain,kkarun,namit,rnitendra@in.ibm.com

ABSTRACT

The World Wide Telecom Web (WWTW) consists of interconnected voice applications (called as VoiceSites) that can be accessed by a regular phone call. A user can interact with a VoiceSite only through speech. This is extremely relevant in developing regions where the literacy levels are low, Internet penetration is low and phone penetration is high.

In this paper, we present the WWTW browser, that enables these users to experience the same browsing experience as is available on the World Wide Web. Since the WWTW can be accessed through a dumb phone instrument, the WWTW browser actually resides in the network. The browser can be accessed by making a phone call to a specific number, and standard browsing features such as back, forward, bookmarking, history are supported. This paper presents the concept, design, implementation and limited usability feedback of the WWTW browser.

Categories and Subject Descriptors

H.4.3 [Information Systems]: Communications Applications—*information browsers*; H.5.2 [Information Systems]: Hypertext/Hypermedia—*Navigation*

General Terms

design, human factors

Keywords

developing regions, world wide telecom web, browsers, voicesites, mobile phones

1. INTRODUCTION

Telephony voice applications have been used by enterprises to provide an alternate access channel to their customers. The technical sophistication of these applications vary a lot, where the simplest ones use the phone keypad to navigate the voice menu, while the sophisticated systems can process natural language in a human voice. Such voice applications find a wider acceptability since the access interface is a phone device – so people can access these applications on the move. The modality of interaction being voice, it is fairly easy to use and navigate through these applications. The implications of phone access and ease-of-use are even higher in developing countries where the alternative of

Copyright is held by the author/owner(s).
WWW2009, April 20-24, 2009, Madrid, Spain.

Web access is not penetrative yet. The Internet penetration in developing regions is still (in the year 2008) less than 10% [28] as shown in Figure 1.

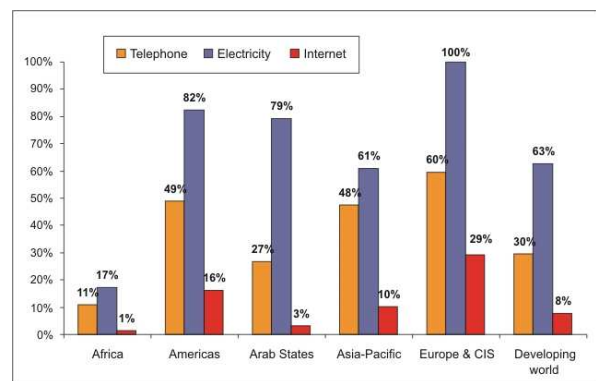


Figure 1: Comparison of Internet and phone penetration rates in rural areas. Source: ITU/BDT research.

However the use of telephony voice applications is still restricted to a handful of enterprises. One major reason for this is that the voice applications, unlike the websites, are still isolated applications that have a specific piece of information or service. Interconnection of voice applications (call them VoiceSites) is therefore a critical requirement to enable their spread as WWW has been able to achieve through hyperlinks. The World Wide Telecom Web (WWTW or Telecom Web or T-Web) [15] provides a mechanism and the advantages to developing the web of voice applications. There is therefore a need to develop a browser to navigate the connected VoiceSites in the WWTW. However unlike the standard Web browsers, a T-Web Browser is a non-trivial extension of its counterpart in the Web world. This is because:

- The VoiceSites can be accessed by a dumb phone, so the browser cannot reside on the client device.
- Voice interaction is a sequential process, unlike a visual interaction. Therefore supporting browsing features in parallel to the customary VoiceSite interaction is challenging.

1.1 Problem

Currently, voice applications are accessed using a telephone device as shown in Figure 2. A Voice Browser is used

to access multiple applications A_1, A_2, \dots, A_n that are hosted on the same Application Server (*App Server* in the figure). Since all the voice applications are on the same server, and are accessed by the same Voice Browser, it is possible for the Application Server to maintain a browsing history of the applications being browsed. The applications A_1 to A_n can provide a link to each other [1].

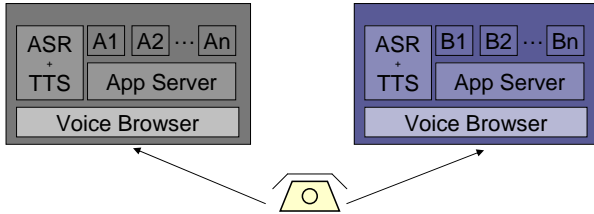


Figure 2: The current voice application deployment and access model.

However, in this architecture, the Voice Browser at the first site will not be able to surf a link from an application in set A to any application in B . *How to enable surfing such cross-browser links is the problem we address in this paper.* A solution to this problem can enable browsing of interconnected voice applications in the same way that we browse the interconnect web pages on the Internet. This can extend the use and applicability of the current voice application space to become the next web – the World Wide Telecom Web.

1.2 Paper Contribution and Outline

In this paper, we present the WWTW Browser that can extend the use and applicability of voice application to reach the level of the WWW. The WWTW Browser can support standard browsing features that we use on the WWW. The browser is designed as a server side application that can be accessed by a dumb phone device over the voice channel. The browser uses call-control features and VoiceXML [6] tags to manage a user session. It provides a seamless navigation experience to the end users. We position this niche work with respect to the research done in the areas of speech-enabled browsers, browsers on mobile devices, technologies for developing regions and the WWTW (Section 2). Later we present the functional specifications of the WWTW browser (Section 3). We provide the high level architecture and the details of the call manager, session manager and the underlying Hyperspeech Transfer Protocol (HSTP) [2] (Section 4). We present the implementation details of the Browser (Section 5) and report the usability of the WWTW Browser (Section 6) before concluding the paper.

2. RELATED WORK

The work presented in this paper can elicit learnings from the following four areas of research:

Browser on the phone: Mobile Internet penetration has reached beyond 240 million users [19]. Not surprisingly, researchers are actively working to improve the Internet experience on mobile devices. WEST [4] is a Web browser for small terminals, that uses text reduction and context visualization techniques to fit a web page in the limited device display. In [7], the authors present a method to reconstruct

PC-optimized webpages for mobile devices by using a segmentation based approach. A usability test for navigation on the mobile device suggests that not all navigation guidelines of the fixed Internet are applicable to the mobile Internet [9, 11]. An HTML to WML conversion proxy server is presented in [10] to render webpages on mobile browsers.

The focus of most of the work done in this area is on presenting the content on the small device. Something to adopt from this area of research is the use of a proxy server that converts webpages to the device form-factor.

Non-Visual Web-Browsers: Research in the area of non-visual interaction with Web browsers is inspired from two different sources: (a) to provide accessibility to visually-impaired users, and, (b) to minimize load on the small screen size in a mobile device. CSurf [16], aiBrowser [18] and HearSay [23] are examples of browsers that render webpages for the visually impaired. They use different analysis tools to identify the right content and then the right presentation format for end users. Authors in [24] use audio in the feedback channel to minimize user attention required to browse a webpage. They focus on delivering content rather than a specific focus on navigation. The PhoneBrowser [20] proposes to browse the WWW using a phone instrument where the user hears a verbal description of each visited Web page using a synthesized voice. Here too, the focus is on transcoding the content from visual to an acoustic modality so that it can be accessed by phone over voice.

Even though significant work has been done to render a webpage content to voice, exact features of the web browser such as back, forward, bookmark and history have either not been discussed in detail, or there has not been a need for a novel solution to provide these features on mobile devices.

Technologies for developing regions: There have been several efforts to provide information to people in rural areas. There are solutions in the network space to provide low-cost Internet access for rural areas [17], [25]. Community Radio systems have been studied as an alternative solution to provide information in rural areas. A sustainability study of the community radio service is presented in [8] for Nepalese radio stations. Extensions to community radio service to involve audience participation have been presented in [27]. The CAM system [21] provides an interface to capture local data using the mobile phone camera. It has been applied for microfinance data capture [22] and for procurement and quality control in a rural cooperative [26].

While there are several point solutions that provide innovative technologies to improve the living standards in developing regions, these solutions are not easy to generalize to different geographies and to different applications.

VoiceSites and the World Wide Telecom Web: VoiceSites are voice driven applications that are created by the subscribers and hosted in the telecom network [13]. They are represented by a unique phone number and can be accessed from any phone instrument, mobile or landline through an ordinary phone call to that number. The phone does not require any extra features or software to be installed on the device. VoiceSites are therefore analogous to websites in the WWW but can be accessed by dialing a phone number and information can be *listened* rather than being read or seen.

Creation of a VoiceSite is made easy by the VoiGen system [13] through which anyone can call up the VoiGen sys-

tem and interact with it through voice. This can enable any illiterate person to create her VoiceSite. Such a system enables easy local-content creation. All information in the VoiceSite is stored as audio messages that are recorded by making a phone call to the system. A sample interaction for creating a VoiceSite is shown in Figure 3.

S: Welcome to the VOiGEN system. Please speak your welcome prompt
 C: My name is Ratan and I work as a hindastani classical vocal teacher in south extension, malviya nagar and nearby areas
 S: The welcome prompt has been recorded.
 S: Say 1 if you wish to specify your references, Say 2 to specify the charges or Say 3 to specify your working hours.
 C: One
 S: Say the people and their contacts who can vouch for your work.
 C: You can ask about my teaching abilities from Mr. Sanjay, his number is 9484042433. Or you can ask from Mr. Gaurav - 9844740248.
 S: References have been recorded. What next?
 C: Two
 S: What are the charges that you ask for your service.
 C: I charge fifty rupees for an hour and one thousand rupees for a month. However the rates really depend on the type of job.
 S: Charges have been saved. What next?
 C: Three
 S: What are your working hours.
 C: I am available from 3 pm to about 6 pm - all seven days a week.
 S: You have specified all the preferences. Should I save them?
 C: Yes.
 S: Your VOICESITE has been created. Thanks for using the system.

Figure 3: A sample interaction of service provider with the VOiGEN system.

By answering simple questions as the ones shown in this interaction, an illiterate person can also create a VoiceSite. A person who needs a classical vocal teacher can access this VoiceSite and get her information. Such a simple mechanism can improve the business of the VoiceSite creator.

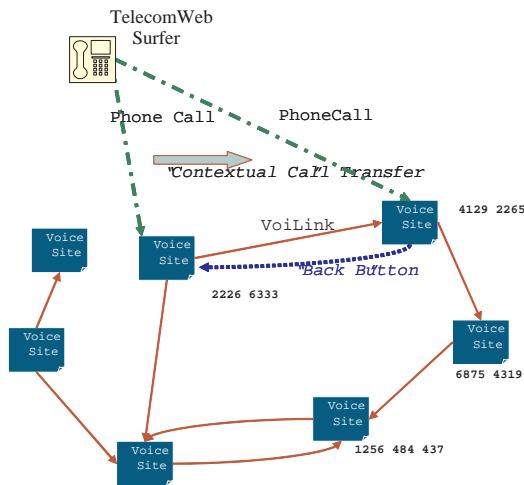


Figure 4: The World Wide Telecom Web.

A VoiceSite can link to other VoiceSites through Hyper-

speech Transfer Protocol (HSTP) [2]. A network of such interconnected VoiceSites results into a World Wide Telecom Web (WWTW) [14] as shown in Figure 4.

System level technology development work on WWTW has been presented in [13] where the authors present technologies for VoiceSite creation. The Hyperspeech Transfer Protocol (HSTP) has been presented in [2] that enables linking of one VoiceSite with other in the Telecom-Web. A solution based on these technologies [12] proposes to organize the unorganized urban poor businesses. The concept of the WWTW Browser has been presented in [3]. In this paper, we extend the underlying technologies of the World Wide Telecom Web [14] by presenting the WWTW Browser to enable navigation of VoiceSites akin to the Visual Web Browser in the WWW domain.

3. FUNCTIONAL SPECIFICATIONS

Drawing parallels from the Web browser, a T-Web browser needs to provide the following features

- **History:** A user should be able to browse through the previous VoiceSites that s/he has visited in the past. The browser therefore needs to store the phone number of the VoiceSite and the date and time when it was visited by the user.
- **Bookmarks:** The browser needs to allow the user to bookmark a particular VoiceSite that s/he is currently browsing. In the T-Web, the browser will have to ask the user to speak a label for the VoiceSite. It should also provide an interface to present the bookmarks to the user so that a user can choose one of the stored bookmarks.
- **Navigation:** The browser needs to provide standard navigation features such as *back*, *forward* and hyperlink navigation to other VoiceSites. For this, the browser should maintain a session for the current browsing of a user and should store the navigation paths.
- **Tabbed Browsing:** Over the last few years, Web browsers have been providing the feature to open multiple tabs for browsing different websites simultaneously. Such a feature allows the user to navigate multiple sites within the same browser window.
- **Search through Browser:** Current web browsers provide a mechanism to search the content of a rendered web page. This is especially helpful when a page contains a large amount of text.

The first version of our browser can support the first three features mentioned above - which are in fact basic to a browser. Advanced features such as tabbed browsing and search within a browser are even more complicated in the voice domain. The work presented in this paper will enable development of a server side T-Web browser that can provide the features of navigation, bookmarking and history to the end user. The working model of the T-Web browser involves a user making a phone call to the Browser. Once the user is connected to the browser, s/he can either (a) choose to view history and go to a previously visited VoiceSite by speaking the specific commands, (b) Open a new VoiceSite by providing the phone number of the VoiceSite, (c) listen

to bookmarks and choose a particular bookmarked VoiceSite and connect to that. When a user is connected to the VoiceSite through the T-Web browser, s/he can issue the following spoken commands to the browser while navigating the VoiceSite (a) bookmark this VoiceSite, (b) traverse a hyperlink to connect to the other VoiceSite, (c) go back or forward to the previous/next VoiceSite, and, (d) access history or stored bookmarks. These features will enable the navigation on the Telecom Web, through any phone instrument using just the voice commands.

4. THE TELECOM WEB BROWSER

In this section, we describe the architecture of the WWTW Browser. We first present the operational model that highlights the role of a browser in the WWTW. Then we present the detailed architecture and the application flow.

4.1 Architectural Overview

A traditional Web browser such as Internet Explorer or a Firefox browser is a client side software that resides on the user's personal computer (PC) or a high end phone that has Internet access. In the World Wide Telecom Web, the client device is typically a low-end phone¹ and the access to a VoiceSite is through voice interaction over a telephone call rather than a data channel. This implies that the Telecom Web browser needs to be designed as a server-side technology. Based on this implication, Figure 5 shows the operational model of the WWTW Browser. A user makes a phone call by dialing the WWTW Browser phone number. Once s/he is connected to the browser, s/he can then access the VoiceSites in the Telecom Web by different browser commands that are mentioned later in this paper. Being a server-side technology, it is evident that a single T-Web Browser can support multiple end users.

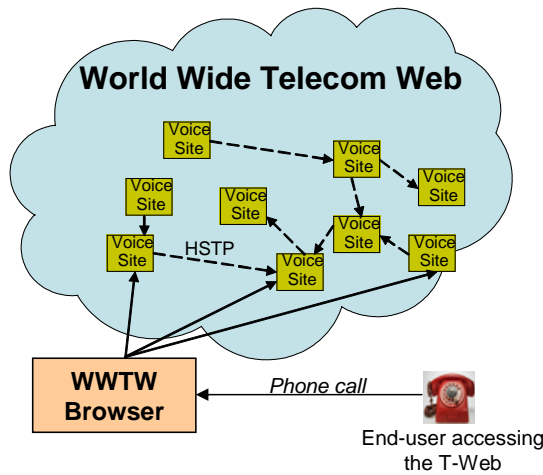


Figure 5: Telecom Web browser operational model.

Given this operational model, the browser commands issued by a user can either be spoken word commands or keypad (often known as DTMF - Dual Tone Multiple Frequency) commands since a basic phone instrument and the telephone link between the phone and the browser cannot support any other modality. The interpretation of these

¹It is possible to use a PC to access VoiceSites through a soft phone.

commands happens at the browser. Therefore the browser needs a speech recognition system. A VoiceSite also typically uses the same interaction modalities.

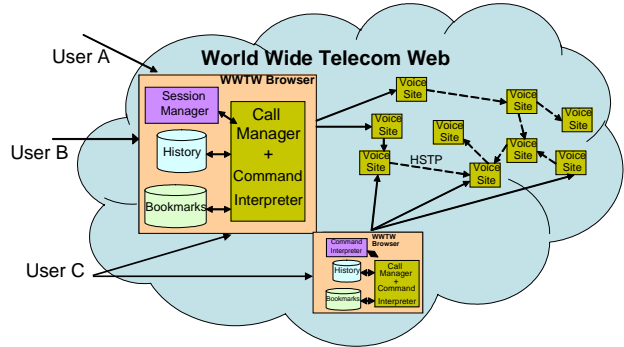


Figure 6: Multiple users and multiple browsers.

As shown in Figure 6 the Telecom Web browser serves as the window through which a user accesses the VoiceSites hosted in the Telecom Web in a seamless fashion. A user can access individual VoiceSites directly too, without involving the Telecom Web browser. However, in that case, the user either needs to know the address² of the VoiceSite to be visited or the current open VoiceSite should contain a hyperlink to that VoiceSite. The figure also shows that multiple users (user A, B and C) can surf the Telecom Web through the same browser instance. Similarly, a single user (user C) can choose to browse through different Telecom Web browsers at different times. The browser itself can be considered to be a special VoiceSite that provides navigational features to the user.

4.2 Detailed Architecture

The operational figures above illustrate that a user connects to the T-Web browser VoiceSite and then accesses any VoiceSite of her/his interest. Once a user is connected to a VoiceSite, s/he should be able to interact with the VoiceSite and the browser simultaneously. This is non-trivial since the browser commands can be issued at any point of time while navigating a VoiceSite. Therefore it is crucial that the browser is available and accessible to the user at all times. This is relatively a simple task in Graphical User Interfaces where traditional browsers split the screen space to accommodate browser's command menu and the website content as shown in Figure 7. However, in the absence of this spatial dimension in the WWTW, it is non-trivial to ensure that the user can communicate with the Telecom Web browser and the VoiceSite being browsed.

The detailed component level architecture of the WWTW browser is presented in Figure 8. The browser uses CCXML [5] to conference two call legs (one connection between the browser and the user, second connection between the browser and the VoiceSite). This allows the browser as well as the VoiceSite (and therefore their corresponding speech recognition systems) to remain active in the same telephone call. This does not require the end-user phone to support conferencing. The phone call mentioned in Figure 8 between the phone instrument and the browser and between the browser

²In the WWTW, the equivalent of an address or URL is the phone number of the VoiceSite.

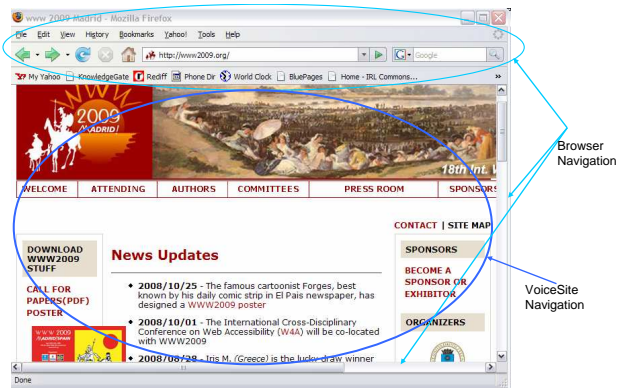


Figure 7: Browser and content separation in a visual browser.

and the VoiceSite is a three party conference call through the CCXML. Drawing parallels from the visual browser, if a speech command is specific to the browser, then the browser takes action on it. On the other hand, if the speech command is specific to the VoiceSite, the browser takes no action. This is similar to a mouse click in the browser part of the pane vs. a mouse click in the website content in the visual world. In the present implementation, any command that starts with the word *browser* is a browser command.

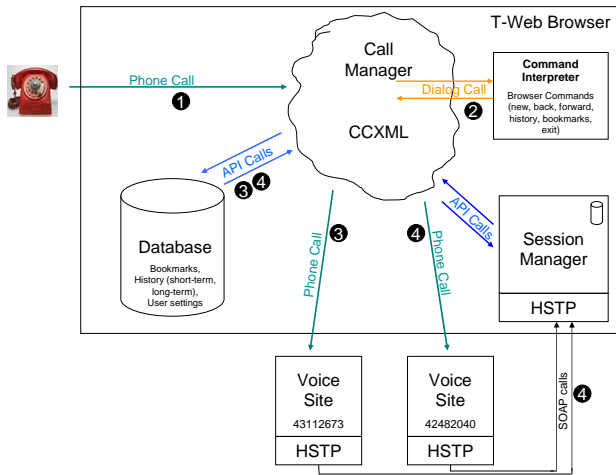


Figure 8: Detailed architecture of the WWTW browser.

As shown in the figure, the Call Manager, Command Interpreter, Session Manager and the Data Model in the Database form the core components of the Telecom Web browser. The interaction between these components is explained through the numbers in the figure and the corresponding explanation below:

1. All calls from the end-user land to the Call Manager through a simple *phone call*.
2. The Call Manager makes a *dialog call* to the **Command Interpreter** which then presents the browser commands to the user. The Command Interpreter is responsible for all interactions with the end user. It

provides the user with the choice of commands and interprets any command issued by the user. When a command (to access a new VoiceSite with the number 43112673 in the figure) is issued by the user, it is passed on to the Call Manager.

3. The Call Manager then makes the *phone call* to the VoiceSite, while retaining the original call with the browser. Thus a three-party call is established by the Call Manager. The Call Manager also makes relevant *API calls* to register events in the browser database. A Data Model (explained in Section 5) is designed to store data related to the browsing session of the user.
4. While browsing this VoiceSite, a user can issue a browser command to be transferred to a new VoiceSite (42482040). This command is again interpreted by the Command Interpreter and the HSTP layer transfers the relevant information (explained in Section 5) through a *SOAP call* to the Session Manager. The Session Manager manages the session information for the current user browsing session. It then passes on the message to the Call Manager through an API call. It also sends a *SOAP message* through the HSTP to the new VoiceSite. The Call Manager then establishes the *phone call* connection of the to the new VoiceSite. It also repeats the step of making the appropriate *API calls* to register the navigation in the browser database.

4.3 Design Discussions

The benefit of the design presented above is that the browser becomes omnipresent and user need not be aware of it. This brings in the possibility, however, that browser becomes a bottleneck especially if it is shared among various users. It would keep a telephone call active for every user for the entire duration of their browsing sessions and also bear the overhead of managing call transfers while they follow hyperlinks. In an alternative design, the Telecom Web browser may not actively be involved in the direct call path of the user and could simply receive the status of the browsing session from various VoiceSites visited by the user. The HSTP layer at each VoiceSite visited could perform the task of updating the Telecom Web browser VoiceSite. This implies that the browser VoiceSite and the visited VoiceSite are not available simultaneously but accessible over different telephone calls. In such a case, to issue a browser command, the user would be required to put the current VoiceSite session on hold and switch to or make a fresh call to the browser for issuing a command. The browser could then instruct the HSTP layer at the currently visited VoiceSite to carry out the necessary action. The benefit here is that the browser VoiceSite need not act as a call switch and thus rule out the possibility of being a potential bottleneck. The drawback is that the user interface aspect suffers since we need two separate calls to use the browser in conjunction with the browsed VoiceSite and the user becomes aware of the presence of the browser.

We adopted an approach that is a mix of both the above. In this approach, the browser still remains a single control point and initiate the VoiceSite connections but the call transfers and session transfers (for voiLinks) will still happen between the VoiceSites themselves with updates about all actions sent to the browser. Also, it would reduce the overload on the browser for managing session transfers and helps retain

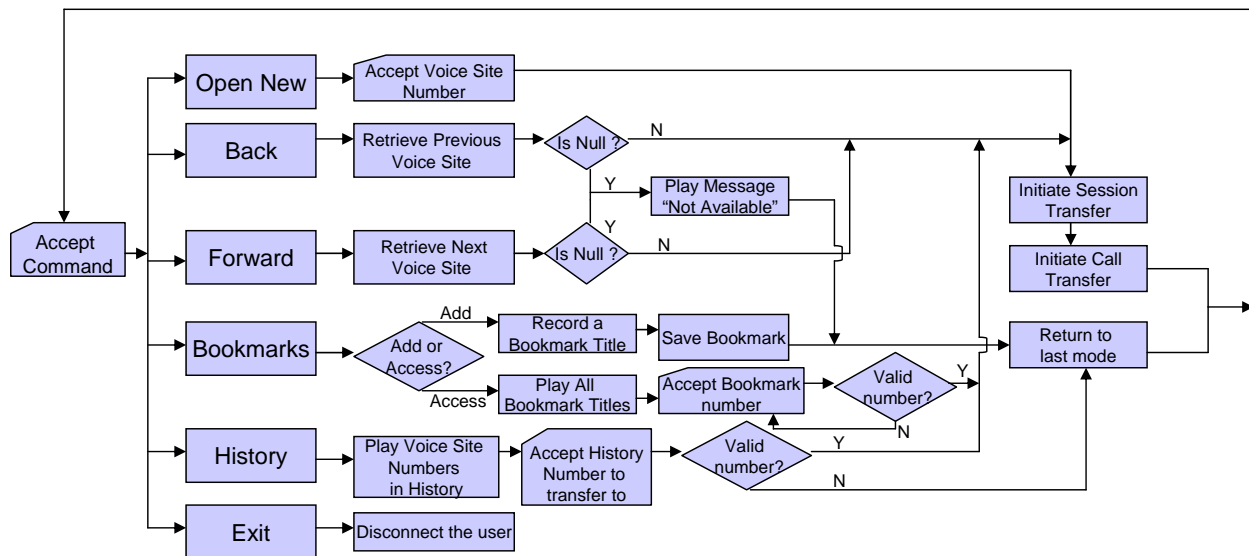


Figure 9: Telecom Web browser application flow.

efficient usability since the user does not have to switch to any other interface to issue browser commands. This keeps the user interface simple and the browser scalable.

Another important design point to consider when two call legs are active at the same time is to identify a mechanism to distinguish between the browser commands and the VoiceSite commands since the audio input channels are open on both legs. One approach is to ‘switch’ to the browser-command mode by saying a unique browser specific keyword like ‘T-Web Browser’ or pressing special keypad sequence like ‘*####*’. This keyword would take the user to the browser command menu where the user can enter the browser commands such as *back*, *forward* and *bookmark this site*. The assumption is that this keyword would not be present in any VoiceSite’s grammar. Other approach is to append a prefix word *Browser* to all the browser commands such as *Browser back* and *Browser new*. This would not require the user to switch any separate command mode for issuing browser commands. The assumption is that these phrases (with the prefix) would not be a part of any VoiceSite’s grammar. *We have adopted the second approach to give a seamless browsing experience to the user with minimum effort and wait time.*

4.4 Specific Scenarios and Application Flow

The application logic of the Telecom Web browser VoiceSite is responsible for providing its core functionality. Figure 9 shows various commands supported by the Telecom Web browser along with the application flow triggered by each of those. Following commands are accepted by the browser:

- **Open New:** When a user requests connection to a new VoiceSite, the Command Interpreter accepts the phone number of the VoiceSite and passes this to the Session Manager. The Session Manager then initiates a session transfer to the new VoiceSite through the HSTP. Once a session is established, the Call Manager then initiates a phone call transfer to the new VoiceSite and the user is connected to the VoiceSite. In

parallel, the Call Manager publishes the information of the VoiceSite to the navigation history of the user in the Database. While navigating the VoiceSite, the user can speak any browser command as shown in the figure.

- **Back:** If a user issues a *back* command, the Call Manager makes an API call to the database and retrieves the details of the previously visited VoiceSite. If there is no VoiceSite that was visited by the user in the current browsing session, the Command Interpreter plays a *Not Available* message to the user and the session is transferred back to the same VoiceSite that the user was browsing. If there was a previously visited VoiceSite, the Call Manager gets the details from the Database and passes the information to the Session Manager and the flow proceeds in the same way as explained in the *Open New* case above.
- **Forward:** The flow for the *forward* command is exactly similar to the *back* command, except that the Call Manager retrieves entries from the forward column of the database.
- **Bookmark:** When a user issues a *bookmark* command, the Command Interpreter asks the user whether s/he wishes to add a bookmark or access the previously stored bookmarks. To add a bookmark, the user records an audio label for the VoiceSite and the details of the VoiceSite along with the audio label are stored in the Database. The interaction is then returned to the previous state. If a user wishes to access previously stored bookmarks, then all the audio labels of the stored bookmarks are played to the user sequentially by the Command Interpreter. The user can select a specific bookmark and then the Call Manager checks in the Database if it has the required details of the VoiceSite. If the details exist, then the session and the call is transferred to the bookmarked VoiceSite. If not, the Command Interpreter interacts with the user to choose another bookmark.

- **History:** A user may choose to look at previously visited VoiceSites. The browser retrieves the information from the Database and the Command Interpreter presents the results to the user. As is the case with *bookmarks*, a user can then choose to browse a specific VoiceSite from this list. The application flow for *history* is similar to the *bookmarks* for establishing a session.
- **Exit:** At any point during the browsing, a user may issue an *exit* command to close the browser session. The Call Manager drops the call and this disconnects the user from the browser.

When any of the above commands are invoked, the VoiceSite is disconnected by the browser and hence the session information of the VoiceSite is sent to the browser Session Manager by the HSTP layer.

5. IMPLEMENTATION

Even though T-Web Browser is invoked and implemented like a VoiceSite, it's requirement for managing multiple call connections and sessions across VoiceSites makes it an ideal application to be implemented using CCXML [5]. CCXML is a W3C standard for Voice Browser Call Control which has extended capabilities for maintaining multiple call sessions and also the ability to integrate VXML applications within the call flow. However, due to lack of availability of resources during the implementation of our prototype, we have simulated the entire call flow using VXML-jsp applications and Java Beans. In this section, we describe the classes, interfaces, data model and the modifications to HSTP that were performed in the development of the T-Web Browser.

5.1 Classes and Interfaces

The following classes have been used in the implementation:

1. *CallManager()* class manages all the connections to VoiceSites and makes API calls to SessionManager and the DB class explained below. Since we have used Voice XML in our implementation, we are using bridged transfer to connect to the VoiceSites. In a VoiceXML bridged transfer, a grammar can be kept active which listens to the browser commands while the user is connected to the VoiceSite.
2. *SessionManager()* class maintains the user's session information while the user is browsing different VoiceSites. The transferSession() method is invoked by the CallManager to indicate the SessionManager to transfer session to a VoiceSite before initiating a call transfer to it. The HSTP layer of the Session Manager in turn, triggers a SOAP request to the target VoiceSite's HSTP layer with the corresponding session information. The session information related to a particular VoiceSite can be of two types:
 - (a) User's session information pertaining only to the VoiceSite that the user is visiting. (Example - login or state information).
 - (b) User's context information that needs to be transferred to a different VoiceSite when a hyperspeech link [2] is accessed. (Example - for transferring payment information).

3. *DB()* class maintains the database for history, bookmarks and user settings. It has methods for inserting a new voice site to the short term and long term history, for lookup of short-term and long-term history, insert and lookup of bookmarks to/from the database and for cleanup to cleans the user's session information if the user disconnects from the browser.

The classes described above also implement methods to provide the following logical interfaces for communication across these classes:

1. *Call Manager-Session Manager:* The Call Manager indicates the Session Manager to transfer the user session before moving back or forward to any VoiceSite.
2. *Call Manager-Database:* Call Manager issues commands to the database such as *Insert VoiceSite*, *Set Setting*, *Lookup History*, *Insert Bookmark*, *Lookup Bookmarks* to update and/or retrieve user history, settings and bookmarks.
3. *Session Manager-HSTP:* A Session Manager indicates the HSTP layer to transfer session information to the VoiceSite (as and when required by the Call Manager).

5.2 Data Model

The database for the browser has been implemented on Apache Derby. Browsing history is stored in two separate databases:

Short Term History is maintained 'per user session'. If the user issues a back/forward command, the short term history is looked upon to find out the corresponding VoiceSite number. This history is deleted after the user's browser session is over.

Long Term History is maintained 'per user' across all her sessions. If the user issues a *history* command, she can browse all the VoiceSites visited over a certain period of time.

Bookmarks are stored for each user with an audio file which contains the title of the bookmark recorded by the user.

5.3 HSTP Modifications

To efficiently use HSTP along with the browser framework, a couple of modifications had to be made to it's functionality and operation.

HSTP Message Format. The HSTP message that is sent to other HSTP layers is as follows:

Table 1: HSTP message format

ID	Description
CALLER_NO	phone number of the person who called the browser
SESSION_ID	session ID for the user's browser session
VA1	current VoiceSite's number
VA2	target VoiceSite's number
AC1	session info from current VoiceSite
AC2	session info for target VoiceSite

VA2 and AC2 will be sent in case a voiLink is being traversed or when the browser's session manager is sending session information to a VoiceSite. Hop number is no longer needed since the browser will be maintaining the user history. Also, 'Type' parameter has been removed because all VoiceSite traversals will be standard call transfers and the reconnection attempts will be monitored by the browser.

To efficiently maintain user's session information HSTP will now send the session information to the **Session Manager** of the browser as soon the user's connection to the VoiceSite is broken (this also takes care of the use case when the browser connection disconnects while user is still browsing a VoiceSite)

6. USABILITY TESTING

We tested the usability of the Telecom Web Browser with 10 subjects by making them use the browser to browse different VoiceSites. All of these subjects had used a WWW browser like Internet Explorer or Mozilla Firefox earlier, so were familiar with the concept of a browser. The usability tests were carried out using a soft VoIP phone instead of a regular phone. This was due to the non-availability of a PSTN to VoIP gateway at our lab. However the browsing experience from an end user perspective is the same so we believe that these results will be valid for a regular phone instrument as well.



Figure 10: SJPhone - a VoIP phone used for usability tests.

Of the 10 subjects, 7 said they would definitely prefer using a browser instead of calling the VoiceSites directly. One of the subjects who was not sure about using the browser was worried about the technical limitations of audio interfaces like state of the art of voice recognition etc. For one of the subjects, it was difficult to envision which one he would choose since he hasn't been using VoiceSites very often.

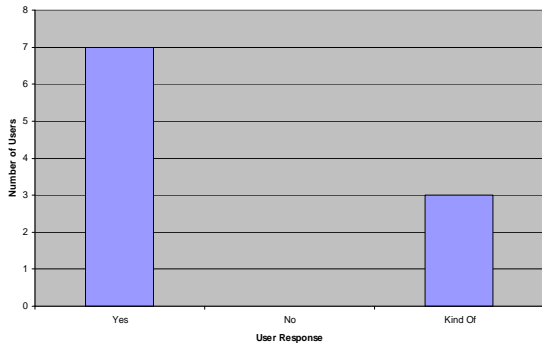


Figure 11: Preference to the browser rather than calling directly to the VoiceSite.

7 of the 10 subjects believed that the call transfer from

one VoiceSite to another happened smoothly for them while using the browser.

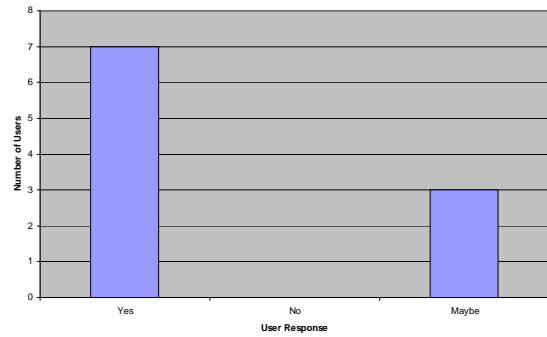


Figure 12: Smooth call transfer experience.

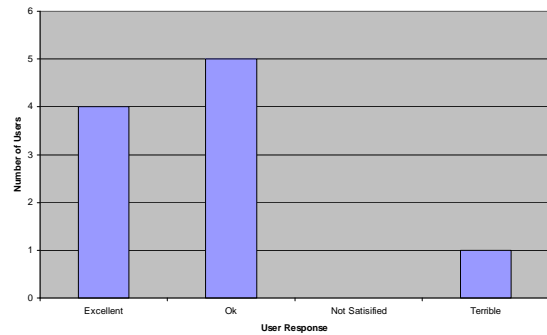


Figure 13: Overall browsing experience.

One of the subjects, whose voice inputs happened smoothly for most of the browser commands, mentioned that he would still prefer DTMF-only input interface for all browser commands. On the other hand, one subject suggested that if the browser is trained by him beforehand for the voice commands, voice recognition can be improved for browser commands. One subject also expressed a need of human recorded voice for browser prompts instead of using text-to-speech to for a cleaner auditory separation from the VoiceSite prompts. 40% of the subjects tagged their browser experience as 'Excellent' while 50% tagged it as 'Ok'.

7. CONCLUSION

A World Wide Telecom Web Browser opens up a lot of possibilities for accessing interconnected voice applications. Rich navigation experience (similar to that available for Web users) can be provided in the T-Web world. The strength of the browser presented in this paper is that it is a server-side implementation and does not have any requirement on the client device. This ensures that the browser can be accessed from a phone instrument of any make, operating system or form factor. The implications of such a browser for the developing world are tremendous – owing to the high phone penetration and low Internet penetration rates. Following are the key conclusions that we draw from the work presented in this paper:

1. We present a WWTW Browser that can provide features such as back, forward, history and bookmarks to navigate interconnected VoiceSites in the World wide Telecom Web.

2. We implement the browser as a server-side technology that can be accessed by any phone instrument. This is unlike the case of a Web browser, but is critical for the Telecom Web world where we expect the end users to access VoiceSites through their phones, mostly low-end.
3. We perform a limited usability study of the implemented browser and are happy to report that of the 10 people who tried this browser, 8 found it to be useful and usable.

The basic concept, architecture, implementation and usability of the T-Web browser presented in this paper opens up a number of features and unique research problems in this domain. Since browsing VoiceSites is itself a novel phenomenon, new requirements will evolve as the Telecom Web grows. A user may want to pause his browsing if he gets an incoming phone call to the phone that he is using to browse. Therefore, a mechanism to ask the browser to pause the current browsing will be required. This is a novel feature and arises due to the sequential and continuous nature of the VoiceSite interaction. Moreover, since a phone is usually used as a two-way instrument, where a user can also expect to be called by a phone (i.e. a VoiceSite), we can provide a feature where the VoiceSite can call a user if the navigation (especially transaction) is incomplete. We intend to pursue some of these directions to harp on the unique and novel requirements in the World Wide Telecom Web Browser space.

8. REFERENCES

- [1] 1800-555-TELL. <http://www.tellme.com>.
- [2] S. Agarwal, D. Chakraborty, A. Kumar, A. A. Nanavati, and N. Rajput. HSTP: Hyperspeech Transfer Protocol. In *ACM Hypertext 2007*, UK, September 2007.
- [3] S. Agarwal, A. Kumar, and A. A. N. N. Rajput. The world wide telecom web browser. In *Proceedings of the World Wide Web*, April 2008.
- [4] S. Björk, L. E. Holmquist, J. Redström, I. Bretan, R. Danielsson, J. Karlgren, and K. Franzén. WEST: a Web browser for small terminals. In *UIST '99: Proceedings of the 12th annual ACM symposium on User interface software and technology*, North Carolina, USA, November 1999.
- [5] W. Consortium. Voice Browser Call Control: CCXML Version 1.0. www.w3.org/TR/ccxml/, 2007.
- [6] W. Consortium. Voice Extensible Markup Language (VoiceXML) Version 2.0. www.w3.org/TR/voicexml20/, 2007.
- [7] G. Hattori, K. Hoashi, K. Matsumoto, and F. Sugaya. Robust web page segmentation for mobile terminal using content-distances and page layout information. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, Banff, Alberta, Canada, May 2007.
- [8] F. Hussain and R. Tongia. Community radio for development in south asia: A sustainability study. In *Proc. Intl. Conf. on Information and Communication Technologies and Development (WWW)*, India, December 2007.
- [9] M. Jones, G. Marsden, N. Mohd-Nasir, K. Boone, and G. Buchanan. Improving web interaction on small displays. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 31(11-16), 1999.
- [10] E. Kaasinen, M. Aaltonen, J. Kolari, S. Melakoski, and T. Laakko. Two approaches to bringing internet services to wap devices. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33(1-6), 2000.
- [11] A. Kaikkonen and V. Roto. Navigating in a mobile XHTML application. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, Florida, USA, Month.
- [12] A. Kumar, N. Rajput, S. Agarwal, D. Chakraborty, and A. A. Nanavati. Organizing the unorganized – employing it to empower the under-privileged. In *Proceedings of the World Wide Web*, April 2008.
- [13] A. Kumar, N. Rajput, D. Chakraborty, S. Agarwal, and A. A. Nanavati. Voiserv: Creation and delivery of converged services through voice for emerging economies. In *WoWMoM'07 Proceedings of the 2007 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Finland, June 2007.
- [14] A. Kumar, N. Rajput, D. Chakraborty, S. Agarwal, and A. A. Nanavati. WWTW: A World Wide Telecom Web for Developing Regions. In *ACM SIGCOMM Workshop on Networked Systems For Developing Regions*, Aug 2007.
- [15] A. Kumar, N. Rajput, D. Chakraborty, and A. A. Nanavati. WWTW: A World Wide Telecom Web for Developing Regions. *IBM Research Report No. RI06010*, Nov 2006.
- [16] J. U. Mahmud, Y. Borodin, and I. V. Ramakrishnan. Csurf: a context-driven non-visual web-browser. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, Banff, Alberta, Canada, May 2007.
- [17] K. Matthee, G. Mweemba, A. Pais, G. V. Stam, and M. Rijken. Bringing internet connectivity to rural zambia using a collaborative approach. In *Proc. Intl. Conf. on Information and Communication Technologies and Development (WWW)*, India, December 2007.
- [18] H. Miyashita, D. Sato, H. Takagi, and C. Asakawa. Aibrowser for multimedia: introducing multimedia content accessibility for visually impaired users. In *Assets '07: Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, October 2007.
- [19] N. Mobile. Critical Mass: The Worldwide State of the Mobile Web. www.nielsenmobile.com/documents/CriticalMass.pdf, July 2008.
- [20] A. Morde, R. S. Kashi, M. K. Brown, D. Grove, and J. L. Flanagan. A multimodal system for accessing driving directions. *Lecture Notes in Computer Science*, 2423(1), 2002.
- [21] T. S. Parikh. Designing an Architecture for Delivering Mobile Information Services to the Rural Developing World. *Ph.D. Dissertation, University of Washington*, 2007.
- [22] T. S. Parikh, P. Javid, S. K., K. Ghosh, and K. Toyama. Mobile Phones and Paper Documents:

- Evaluating a New Approach for Capturing Microfinance Data in Rural India. In *Proc. CHI*, Canada, April 2006.
- [23] I. V. Ramakrishnan, A. Stent, and G. Yang. Hearsay: enabling audio browsing on hypertext content. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, USA, 2004.
- [24] V. Roto and A. Oulasvirta. Need for non-visual feedback with long response times in mobile HCI. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, Chiba, Japan, May 2005.
- [25] U. Saif, A. L. Chudhary, S. Butt, N. F. Butt, and G. Murtaza. Internet for the developing world: Offline internet access at modem-speed dialup connections. In *Proc. Intl. Conf. on Information and Communication Technologies and Development (ICTD)*, India, December 2007.
- [26] Y. Schwartzman and T. S. Parikh. Using CAM-equipped Mobile Phones for Procurement and Quality Control at a Rural Coffee Cooperative. In *MobEA V, Mobile Web in the Developing World*, Canada, May 2007.
- [27] S. R. Sterling, J. O'Brien, and J. K. Bennett. Advancement through Interactive Radio. In *Proc. Intl. Conf. on Information and Communication Technologies and Development (WWW)*, India, December 2007.
- [28] I. T. Union. Global ICT Developments. <http://www.itu.int/ITU-D/ict/statistics/ict/index.html>, 2006.