FALA 2010
VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop

A Prototype of Distributed Speech Technologies for the Development of Websites Accessible to the Blind Community

Oscar Saz, Eduardo Lleida, José-Enrique García, Alfonso Ortega

Communications Technology Group (GTC)
Aragón Institute for Engineering Research (I3A)
University of Zaragoza, Spain
{oskarsaz,lleida,jegarlai,ortega}@unizar.es

Abstract

This paper presents a proposal for the use of distributed speech technologies like Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) synthesis to create a paradigm of web development which integrates voice as input and output interface in the navigation experience. This enhanced usability is aimed to provide accessibility to the Internet for the community of blind and visually impaired people, whose access nowadays is difficult and costly. The novel concept in this work is to provide the web developer all the control on the use of the speech synthesis and recognition, i.e. designing the voice experience within the website through the use of a simple set of tags, rules and functions. A pilot experience with Aragón Radio 2, the online radio channel of the Aragonese Corporation of Radio and Television (CARTV), has shown the arising possibilities for this technology and opens the interest for further use of them in collaboration with different institutions.

Index Terms: visual impairments, technical aids, web accessibility, speech synthesis, speech recognition

1. Introduction

Providing universal accessibility to the contents of the Internet is a major issue for all the agents involved in the evolution of the World Wide Web (WWW); the World Wide Web Consortium (W3C) is currently working through the Web Accessibility Initiative (WAI) on defining different scenarios of inaccessibility to web sites and trying to create standards of web development to eliminate barriers for the handicapped community [1]. Due to the visual nature of the web, blind and visually impaired people are the community which finds the largest barrier in their access to this new gate to knowledge.

Screen readers have been the usual tools for blind people to access computers and, furthermore, the Internet. Some of the most popular systems, like JAWS Screen Reader [2], GW Windows-Eyes [3] or the late IBM Homepage Reader, have been running up for many years, achieving great success and providing new versions and languages along time. These systems take all the textual information presented in the screen and provide synthesized speech for it. The blind person can use keyboard commands to navigate through the different elements in the computer screen and to configure the system. Nowadays, more novel solutions are appearing from the academic world trying some of the disadvantages of these commercial software, especially the high price and the need of installing them in a specific platform (mainly Microsoft Windows). For instance, WebAnywhere [4] is a novel solution which provides speech synthesis on the Internet; when a URL address is provided, a Macromedia Flash application connects to a remote server which synthesizes the sequence of texts within the website.

However, the experience of navigating the web with these systems is not as easy as it may seem. Professionals developing websites hardly think on how their site will be interpreted by these speech-synthesis-based applications. This means that the order in which the different texts in a website are synthesized may provide a different feeling than the one when visually inspecting the site. Furthermore, elements which are very common in the development of websites like side bars, search indexes, etc. produce a lot of information when synthesizing the page, which makes difficult for the blind person to separate the important information in the site from all the secondary elements within.

This work proposes to give web developers the tools to not only decide the visual appearance of their websites, but also to define the parameters of a voice input-voice output interaction with the webpage. The inclusion of an active element, i.e. Java Applet, allows the use of distributed services for Text-to-Speech (TTS) synthesis and Automatic Speech Recognition (ASR) as part of the interaction; these elements being handled by the developer via Javascript functions. With the use of a set of tags inserted in the Hyper-Text Markup Language (HTML) document of the website, it is possible to define which elements will be synthesized, to create dependencies among them, to use alternative texts to synthesize and to define a simple oral control of the most usual commands. The prototype for a pilot study will show the possibilities of this new proposal and will promote the development of websites based on this technology.

The paper is organized as follows: Section 2 will describe the use of distributed speech technologies and the active element which permits to include them in the development of a website. Section 3 will describe the HTML tagging system created for the design of the audio-accessible web, as well as the set of keyboard commands which the blind person can use to navigate. Posteriorly, the pilot study with the website of Aragón Radio 2 will be described in Section 4. Finally, Section 5 will present the future lines of work and conclusions to this work.

2. Distributed Speech Technologies

The use of distributed architectures is becoming more and more usual nowadays to take advantage of technologies with a high computational cost on small portable devices which incorporate a fast and reliable network connection. It also helps providing software services on devices and platforms where it is not possible or desirable to install a stand-alone application.

In the specific case of speech technologies like ASR and

TTS, distributed architectures have allowed the massive introduction of these technologies in devices like PDAs or smartphones. In these cases, the client (small device) only captures the audio from the user and plays the synthesized audio signals while the server carries on all the signal and language processing required, including Large Vocabulary Continuous Speech Recognition (LVCSR) or Hidden Markov Models (HMM)-based TTS. Commercial services of the so called Voice Search are operative nowadays achieving a certain success [5].

The EDECÁN Consortium (TIN-2005-08660-C04-01) gave a proposal for the use of speech technologies in distributed architectures [6]. This proposal had a central communications manager which interconnected different services in different remote machines. The services included ASR, TTS, speaker adaptation, speaker verification, dialog managers and any other service which was adapted to the EDECÁN protocol. With this architecture, any client requiring specific speech technologies systems (for instance, to create a full dialog system) would connect to the communications manager and the manager would connect with the services required by the client. This architecture served as the basis of the proposed system for distributed speech technologies on the web.

2.1. Speech Technologies on the Web

The use of advanced technological resources on the web is another case where distributed architectures are shown to be extremely useful. In these cases, using applications directly installed on the user's computer might be uncomfortable for the user and complicated for the developer in order to provide updated versions for all the possible platforms (Microsoft Windows, MAC OS, Linux) and web browsers (Microsoft Internet Explorer or Netscape-based browsers). Hence, the same network connection used to access the remote website can be used to access a resource. Systems like XHTML+Voice (X+V) have tried to provide standards for creating spoken dialogs in the web using the potentially of VoiceXML [7],but the need to include the X+V capabilities on the client side and VoiceXML tagging on the website by the developer have limited its possibilities to become a universal standard.

For the remote use of speech technologies on the web, a schematic diagram like in Figure 1 is proposed. The user accesses to the website via the WWW protocol and the page connects in a client-server protocol with the remote server where all the main processing is done. The page does not require any specific coding aside from the standard HTML, PHP or Javascript languages. The use of speech synthesis and recognition in the web is performed with the inclusion of a Voice-input Voice-output (ViVo) Java Applet in the web source code. The utilities that are provided by this Applet are ASR and TTS.

The ASR is performed with the vocabulary indicated by the website, either in a separate file or as a String. The visual interface of the Applet permits a push-to-talk interface where the user presses a button in the Applet to launch the recognition. Otherwise, the developer can start the recognition when a certain function of the code is called by an event in the page. An audio reinforcement of the start of the recognition phase in the form of a 'beep' can be used to help the user. The result of the ASR is collected by an specific function in Javascript.

The TTS is able to synthesize any utterance in Spanish represented by a String. It provides different voices in Castilian-Spanish and Aragonese languages which can be selected at the moment of invoking the TTS procedure.

A set of Javascript functions are provided to make use of these functionalities:

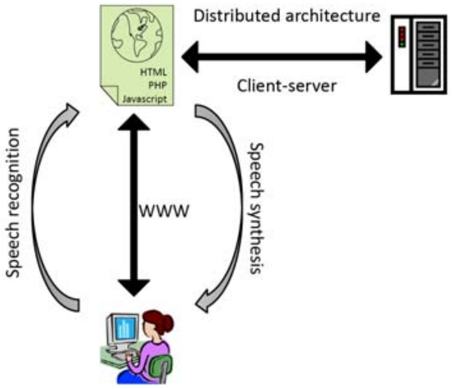

Figure 1: *Scheme of distributed voice interaction on the web.*

- UZSinte(String sentence, String speaker): Launches a TTS instance of a "sentence" using a given "speaker".
- UZSinteStop(): Finishes reproduction of the TTS.
- UZStartRecoGrammar(String url-grammar): Launches an ASR instance using the grammar in a URL.
- UZStartRecoGrammarText(String grammar): Launches an ASR instance using the provided grammar.
- UZStopReco(): Stops the ASR procedures.

The Applet connects with the remote server indicated as a parameter in the HTML code, which performs all the ASR and TTS processing. When ASR is required, the Applet captures the audio from the standard audio input and performs the extraction of the Mel Frequency Cepstral Coefficients (MFCC) features from the speech signal. These MFCC features are quantized and the Applet sends them to the server with with a reduced bit rate without losing ASR performance [8]. When the server decodes the uttered sentence, it sends back the results to the Applet as a string, which is post-processed by a recoend() function which performs the actions programmed by the developer.

When TTS is called, a string is sent to the server with the sentence to synthesize and the desired voice to use. The TTS process is performed on the remote server and, when finished, the audio is sent as a stream to the Applet which plays it through the standard audio output.

3. Proposal for Audio-Accessible Web Development

The functionality for providing the accessibility on the website with the use of the ViVo Java Applet is based on two elements which define the interaction: The HTML tags defined the elements of the site to be synthesized and that can be placed freely by the web developer; and the set of keyboard commands which serve the blind user to navigate thought those different elements and to activate and deactivate different functionalities.

3.1. HTML Tags for Voice Web Interaction

The way in which a web developer provides TTS of a certain section of the website is made by adding attributes to certain HTML tags. Three HTML elements can have synthesized speech associated to them: <span>, <p> (paragraph) and <a> (link). Table 1 shows the templates to apply this feature in these 3 elements with different possibilities. An element with synthesized speech is marked with the attribute $class = “headings - sinte''$ and the text to be synthesized is included in the attribute $title = “Text\ to\ synthesize''$. Links and paragraphs allow for the direct synthesis of the text in the given paragraph or link. The use of spans permits to summarize a large section of the section in a single synthesized text.

Table 1: *HTML tags for voice web interaction.*

span	<span class="headings-sinte" title="Text synthesize"> HTML code </span>
p	<p class="headings-sinte"> Text to appear on screen and synthesize </p>
a	<a class="headings-sinte" title="Text to synthesize"> Text in link </a>
a	<a class="headings-sinte"> Text in link and synthesize </a>

One of the more important features that is allowed by the system is to define two levels of relevance within the different elements to be synthesized. This way, a major heading can contain subheadings which will only be synthesized if the user decides so with a certain key combination. The HTML tagging for this feature is as shown in Table 2 where a major <span> element contains a paragraph and a link which are marked with the attribute $class = “subheadings - sinte''$.

Table 2: *Organizational tree of HTML tags.*

```
<span class=“headings-sinte” title=“Top level text”>
        HTML code
        <p class=“subheadings-sinte”>
                Second level text
        </p>
        HTML code
        <a class=“subheadings-sinte”>
                Second level text
        </a>
        HTML code
</span>
```

When the TTS voice reads the main element, the system indicates orally that there are a certain number of elements and asks whether the user wants to read them or not. At any moment, the user can come back to the initial list of main elements with a keyboard command.

The web developer also has the chance to provide a set of oral commands which create oral shortcuts to the most usual actions on the site. The HTML syntax to define these commands is as seen on Table 3, which is interpreted by the system and extracts the “Command utterances”, that is the sentence to be spoken by the user to be recognized, and the command to be performed as the function toDoOnRecognition(){}.

Table 3: *HTML tags for speech commands.*

```
<span class=“commands-reco” title=“Command utterance”
onclick=“function toDoOnRecognition(){}”>
        HTML code
</span>
```

The system is able to process all these tags in the body of the HTML code and create a grammar with all of them to be used in the recognition stage. When a valid sentence is pronounced, the function or code to be performed is run automatically.

3.2. Use of keyboard commands

Keyboard commands are the main basis that blind people use when accessing computer systems and the Internet. Different combinations of keys allows to move through the different elements in the screen and to configure and use specific software.

The basis that permits the use of the ViVo Java Applet in the accessibility of web sites by blind people are a series of Javascript functions which allow the control of the Applet with keyboard commands and which have already been developed and are ready to be provided to web developers.

Table 4 presents all the available commands for the website control. They are divided into control commands which activate or deactivate the voice aids, synthesis and recognition, or provide an auditive help about the handling of the system; and into commands which serve for navigating around the website, moving back and forth through the different items which are defined to be synthesized.

Table 4: *Keyboard commands for voice web interaction.*

TTS handling commands	
Ctrl+Q	Reads the synthesis help
Ctrl+A	Activates the synthesis of the site
Ctrl+D	De-activates the synthesis of the site
Ctrl+(Right arrow)	Reads next element in list
Tab	Reads next element in list
Ctrl+(Left arrow)	Reads previous element in list
Shift+Tab	Reads previous element in list
Ctrl+(Up arrow)	Reads first element in list
Ctrl+Home	Reads first element in list
Ctrl+(Down arrow)	Reads current element in list
Ctrl+End	Reads last element in list
Ctrl+Z	Switches levels of in the elements tree
ASR handling commands	
Ctrl+E	Reads the recognition help
Ctrl+R	Starts recognition of the user's speech

4. Pilot Study with Aragón Radio 2

A prototype for an audio-accessible website was created in collaboration with the Aragonese Corporation of Radio and Television (CARTV). CARTV has recently launched an on-line radio channel on the Internet (http://www.aragonradio2.com) with the aim of creating a more participative and accessible way of communication with its listeners. The intention of the prototype was to demonstrate the possibilities that the novel techniques explained in this work had for providing Internet access to the blind community [1].

The prototype was build on a static version of the website retrieved on January 2010. The visual appearance of the prototype site is the same to the original one, as seen in Figure 2. The 4 main sites of the AR2 website were adapted to the system (Home, News, Public Service and Podcasts) as well as the radio and podcast players (which work as pop up windows).

The highlighted elements in Figure 2 show the initial part of the organizational tree of the website for synthesis purposes.

[1] http://dihana.cps.unizar.es/ alborada/ar2/ar2_frames.htm

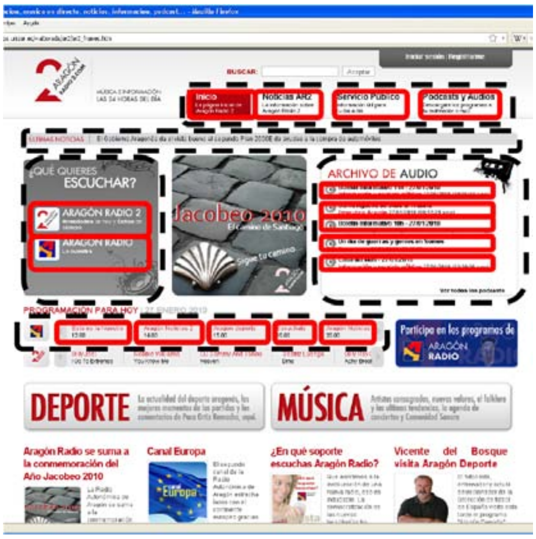
BUSCAR:
Aceptar
MÚSICA E INFORMACIÓN LAS 24 HORAS DEL DÍA
Inicio
Noticias AR2
Servicio Público
Podcasts y Audios
ÚLTIMAS NOTICIAS
¿QUÉ QUIERES ESCUCHAR?
ARAGÓN RADIO 2
ARAGÓN RADIO
Jacobeo 2010
El Camino de Santiago
Sigue tu camino...
ARCHIVO DE AUDIO
Ver todos los podcasts
PROGRAMACIÓN PARA HOY | 27 ENERO 2010
Participa en los programas de ARAGÓN RADIO
DEPORTE
MÚSICA
Aragón Radio se suma a la conmemoración del Año Jacobeo 2010
Canal Europa
¿En qué soporte escuchas Aragón Radio?
Vicente del Bosque visita Aragón Deporte

Figure 2: *Appareance of the pilot study in the Aragón Radio 2 prototype website (synthesis elements are highlighted).*

Elements in dotted line are those which are at the top level of the organizational tree and which are read sequentially until the user decides to enter into some of the subheadings of a main element with the Ctrl+Z command (continuous line in Figure 2). Table 5 provides the contents available for synthesis in the homepage of the prototype and the sub-contents within them.

Table 5: *Information structure in the Aragón Radio 2 prototype.*

Main contents	Sub contents
Main menus	Home page News page Public services page Podcasts page
Latest news	Text of first new Text of second new Text of third new, etc.
Radio launcher	Aragón Radio Aragón Radio 2
Latest podcasts	Description of first podcast Description of second podcast Description of third podcast, etc.
Daily programs	Name of the first program Name of the second program Name of the third program, etc.

With the proposed systems, the understanding of the key elements in the website is better and long lists (programs, podcasts, latest news, etc.) are kept in a secondary level that the user can read when really interested in doing so. Furthermore, elements with little relevance (banners, self-advertisements) are omitted to avoid the user losing focus on the relevant elements.

For providing enhanced interaction, the most common actions in the website were also allowed via ASR. The commands that were incorporated included:

- Navigate to the home page
- Navigate to news page
- Navigate to the public service page
- Navigate to the podcasts page
- Listen to Aragón Radio
- Listen to Aragón Radio 2
- Play radio (when radio pop-up is on)
- Stop radio (when radio pop-up is on)
- Switch radio channel (when radio pop-up is on)

A preliminary evaluation by the experts in technical aids of the Spanish National Association for the Blind (ONCE) gave a positive review on the perspectives of this new technology. Some of their proposals regarding the functionality and use of the system were incorporated in the described system and future evaluation with a set of users will be made to assess how helpful this technology can actually be.

5. Conclusions and Further Work

The conclusion of this work focuses on the great possibilities that the use of speech technologies may have for providing accessibility on the Internet for blind people. The preliminary evaluation of the pilot study carried out showed a great potential of the technology and encouraged to keep working in this line. Among the main features of the proposed system are: Cross-platform use of speech synthesis and recognition, low bandwidth required due to the use of efficient coding features of speech features, absolute power for the web developer to design the audio appearance of the website and full accessibility via simple keyboard commands.

Aragón Radio 2 plans to fully incorporate this system in its website in the near future as part of their effort on providing universal accessibility to their resources. The auditive aid will be started via a keyboard command which can be easily used by a blind user. That way, people without visual impairments will maintain the same experience navigating the site, while their blind peers will have available the new web experience.

6. Acknowledgements

This work was supported under TIN-2008-06856-C05-04 from MEC of the Spanish government.

7. References

[1] W3C. (2010, May) Introduction to “How People with Disabilities Use the Web”. [Online]. Available: http://www.w3.org/WAI/intro/people-use-web

[2] (2006) Jaws 8.0 for Windows, Freedom Scientific. [Online]. Available: http://www.freedomscientific.com

[3] (2006) GW Micro Windows Eyes. [Online]. Available: http://www.gwmicro.com/Window-Eyes/

[4] J.-P. Bigham, C.-M. Prince, and R.-E. Ladner, “WebAnywhere: A screen reader on-the-go,” in *Proceedings of the Cross-Disciplinary Conference on Web Accessibility*, Beijing, China, 2008.

[5] Y.-Y. Wang, D. Yu, Y.-C. Ju, and A. Acero, “An introduction to voice search,” *IEEE Signal Processing Magazine*, pp. 29–38, May 2008.

[6] J.-E. García, A. Ortega, A. Miguel, and E. Lleida, “Arquitectura distrubuida para el desarrollo de sistemas de diálogo hablado: EDECÁN,” in *Proceedings of the VJTH*, Bilbao, Spain, November 2008, pp. 171–174.

[7] R. Beasley, J. O’Reilly, K.-M. Farley, and L.-H. Squire, *Voice Application Development with VoiceXML*. Indianapolis, IN, USA: SAMS, 2001.

[8] J.-E. García, A. Ortega, A. Miguel, and E. Lleida, “Differential vector quantization of feature vectors for distributed speech recognition,” in *Proceedings of Eurospeech*, Brighton, UK, September 2009, pp. 2587–2590.