A Tool for Enhancing Web Accessibility: Synthetic Speech and Content Restructuring

Spyros Raptis *Ilias Spais* *PirrosTsiakoulis*

Institute for Language and Speech Processing (ILSP)
Speech Technology Department
Artemidos 6 & Epidavrou,
GR-151 25, Athens, GREECE
http://www.ilsp.gr, spy@ilsp.gr

Abstract

WebSpeech is a tool developed at ILSP that is intended for people with visual impairments. It tries to combine advantages from both customized and generic web enhancement tools. It consists of a generic engine and a set of case-specific filters. The engine is able to communicate with an external web browser, acquire and parse the content of web pages, synthesize bi-lingual text to speech using ILSP's high quality speech synthesizer, and supports a set of common functionalities such as navigation through hotkeys, audible selection lists etc. The role of each filter is to capture the specifics of a website: the way to determine the title of the current page, the website navigation menus, the current location in the site's overall map etc. However, the most significant task of the filter is to determine the structure of actual textual information in the current page, mainly its paragraph structure and any tabular formulations. WebSpeech poses no requirements on a page and introduces no overhead to the design and development of a website. Based on its engine/filter approach, it can deal with any website without any modifications in the tool's engine. It is only a mater of implementing a specific filter for a website; for consistently designed sites (for example sites that are backed by a content management system) this can be a mater of a few days.

1 Web Accessibility Technology

The web's increasing importance and penetration into every-day activities makes the need for its accessibility more imperative than ever. Information repositories, communication means, and interactive services based on the internet are some of the most active development areas today (Kouroupetroglou & Mitsopoulos, 2000).

As true universal access and inclusive design are now becoming critical requirements, efforts to design and implement browsing helpers, encounter a typical dilemma: generality versus speciality. A general web enhancement tool (such as a screen reader or a text browser) is able to deal with virtually any web page based on uniform, systematic and widely applicable interaction patterns. On the other hand, a custom tool tailored to the specifics of a website will perform better in the sense that it will be able to exploit available a priori information on the site's structure to convey information more coherently to the user. Moreover, it will be able to support richer and more accurate interaction patterns. The choice of the appropriate tool is strongly linked with the intended application and the required use cases.

Web accessibility means access to the Web by everyone, regardless of disability. The following discussion does not intend to be complete nor exhaustive; it is confined mainly to the case of web accessibility technology for visually impaired people.

Web accessibility evolves around content. Content needs to possess a set of attributes that ensure the completeness of the information it is intended to convey. This is directly related to the way content is presented, i.e. the content access, and significantly affects the way it is produced, i.e. content authoring and development.

There is a lot of work in the field of web accessibility and in closely linked fields; fields drawing from or contributing to it. A special reference is due to the W3C Web Accessibility Initiative, and especially its Web Content Accessibility Guidelines, the Voice Browser Activity, and the Alternative Web Browsing considerations (hyperlinks to relevant W3C pages are provided in the References section).

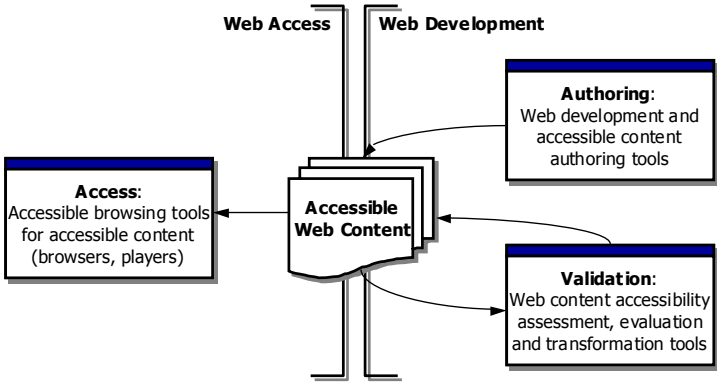

Figure 1: The main parts involved in web accessibility: the content, the authoring tools, and the browsing tools

1.1 Accessible Content

Web accessibility evolves around content. Clearly, the major role of content is to capture and convey information. Content presentation is a separate concern, involving creating views and projections of the content to specific presentational patterns. Much of the content accessibility relies in assuring that no questionable assumptions are hard-wired in the content concerning the way, means and techniques that will be used to present it.

Accessible web content possesses a set of attributes that ensure sufficient consistency and presentational independence (often calling for redundancy) of the information it is intended to convey. In practice, content accessibility suggests conformance to a set of requirements that assure an "accessible format". A widely acceptable specification of such a format is provided by W3C's Web Content Accessibility Guidelines [WCAG]. The primary goal of the W3C's Web Content Accessibility Guidelines is to promote content accessibility. They are summarized as a list of organized and prioritized checkpoints that web pages need to be verified against, along with suggested techniques for implementing them.

Some characteristic examples of specific aspects covered by the WCAG are alternative text of images, tables that make so sense when their elements are read in a sequential fashion, web forms that cannot be navigated into with a meaningful order and no on.

With vision being one of the richest, most appealing and most effective channels for communicating information, the presentational patterns employed for delivering web content significantly rely on images, charts, tables and graphics. However, the sole means that visually impaired persons often have to access such information are aids based on synthetic speech. To avoid cutting such users off, content itself should not be bound to its visual appearance and should provide what is necessary for these sophisticated and attractive page layouts to gracefully reduce to meaningful audible counterparts.

1.2 Content Authoring

Accessible web content calls for content authoring tools that can produce such content as well as tools that can assess, evaluate, and transform web content into an "accessible format".

Content authoring and development tools for accessible content are out of this paper's scope. Detailed information on the technologies and tools can be found through the W3C website.

1.3 Content Presentation and Access

Software used to access web content is usually referred to as called user agent. Obviously, accessible user agents making content truly available to all users are the terminus. These tools must employ efficient content presentation schemes tailored to the special needs of their users enhancing the content perception and understanding. Moreover, they need to implement appropriate user interaction patterns so that the users can effectively interact with the content and navigate in it.

User agents include desktop graphical browsers, text browsers, voice browsers, mobile phones, multimedia players, and so on. Assistive software technologies used in conjunction with browsers such as screen readers, screen magnifiers, and voice recognition software are also of concern. Some of the most relevant user agent technologies for the case of visually impaired persons are shortly discussed in the following.

1.3.1 Alternative Browsing

Text browsers offer an alternative to graphical user interface browsers. They can be used in combination with standard screen readers to render content through synthesized speech. Lynx is a typical example of a text browser.

Voice browsers are about expanding the Web to allow people to interact via spoken commands and synthetic speech. They offer the promise of allowing everyone to access web-based services from any phone, making it practical to access the web any time and any where, whether at home, on the move, or at work. Work in that field of voice browsers is closely related to people with visual impairments since voice is used as a replacement of vision to convey the necessary information, very similarly to accessing the web through the phone.

The term *alternative browsing* refers to all approaches, including the above, that deviate from the typical browser setting and provide specific support for specific types of disabilities.

1.3.2 Content Transformations

There is a number of operations that need to take place for converting a web page into audible form. Some of the operations most relative to this discussion are the web page adaptation, restructuring and, finally, rendering in speech.

1.3.2.1 Adaptation

The adaptation of the web page to the user profile (personalization), involves the alteration of the page's format to fit the user needs and preferences. It includes special processing and transformations that may be necessary to the page so that it becomes more accessible to a specific user.

For example, increasing the contrast or font size for people with low vision, serializing page contents for blind people so that they are read more efficiently by a screen reader, enlarging the active page elements to facilitate their access by motor-impaired people and so on.

Adaptation can take place locally in the user's (client) computer, e.g. as in the case of the AVANTI web browser (Stephanidis, Paramythis, Karagiannidis & Savidis, 1997), or through a proxy server that intervenes between the web server that provides the content and the user's computer e.g. as in the case of WebFACE .

1.3.2.2 Restructuring and Custom Interactivity

Restructuring involves determining the role of page elements and element groups and offering to the user a more efficient access to them.

The *role* of an element is not only related to the type of the element (e.g. an edit control in a web form or a side link), but also its intended use in the site. Examples of elements with specific semantics in a site are the global navigation menus and submenus in websites, elements that are systematically used throughout a site and have a specific purpose, specific formats that signify section breaks or special types of transitions to other pages or sites, and so on.

It is noteworthy that although site templates significantly differ in their aesthetics, they do share a lot of common structural properties. Global site navigation menus, option bars, and copyright notices appear in the vast majority of the sites. Such usable and efficient website design patterns have been widely adopted and people not only have become quite familiar with them but are actually looking for such structures in every new website they visit.

An important property of some of these elements, is the fact that they are appear in *every* page of the site. This makes them *site elements* rather than *page elements*. Reading such elements in each page rarely makes sense. Users always have in mind that these are there and can be called at any time, but don't really want them to be read out along with every page loaded.

Identification and proper handling of elements with specific roles in a website can provide the means for supporting custom enhanced interaction patterns that significantly improve the usability of the site and the quality of the user interaction. However, this can only be accomplished with a priori information about the design of the specific website.

1.3.2.3 Speech Rendering

Speech has become a mainstream in computer technology. Speech synthesis (text-to-speech, TtS) is particularly relative to visually impaired persons since it can be one of the most effective substitutes for vision. TtS is now widely available for many different languages and supported even at the level of the operating system as, for example, in the case of Microsoft Speech API for Windows.

In the context of accessibility tools, a TtS component could be available:

- *Locally* as part of the operating system, an accessibility tool (as a screen reader) or the web browser itself (e.g. IBM's Home Page Reader). This does not pose any overhead in the web connection bandwidth since speech rendering takes place locally. However, it requires the download and setup (and possibly the updating) of a TtS component.
- *Remotely* as a service provided by a speech server (e.g. ReadSpeaker). This alleviates from the need to setup a local TtS component but poses higher requirements on the connection bandwidth since audio needs to be transferred.

1.4 Existing Systems

A very rough overview of the mostly used approaches for the speech-enhanced web access is provided in Figure 2 below. Some characteristics of each case are given below.

- ***Case 1***. This one of the most simple and common settings for speech-enhanced web access. A normal web browser such as Microsoft's Internet Explorer has the role of accessing and retrieving the web content, while a separate tool such as a screen reader (e.g. JAWS), a desktop accessibility tool (e.g. Microsoft's Narrator) or a browser plug-in are responsible for rendering it using synthetic speech.

  In this case, the browser is responsible for performing any necessary adaptations to the web content to meet user needs and preferences. For example, Microsoft's Internet Explorer provides a set of accessibility features.

No restructuring or custom interactivity can be supported in this setting since there is no way to exploit a priori website information.

- ***Case 2***. A slightly different case is that of a custom browser. The browser is responsible for all the tasks: accessing and retrieving the web content, adapting it to user needs and preferences, and rendering it to speech. A typical example of a custom browser with adaptation and speech capabilities is IBM's Home Page Reader. Additionally, it provides support for other media types such as Adobe PDF and Macromedia Flash content.

  Normally, the Home Page Reader does not support web page restructuring. However, special versions of this tools have been deployed to provide enhanced support for specific websites, as for the case of the American Association of People with Disabilities (AAPD) website. Other selected sites which demonstrate specific features are also supported (e.g. the sites of Adobe, Macromedia and W3C).

- ***Case 3***. In this case, the adaptation of a web page to meet user needs and preferences is performed by a remote server. That server keeps the user profiles and intervenes between the web content provider (web server) and the user agent (client computer). WebFACE is an example of this approach.

  Speech rendering can be performed either by a screen reader or a desktop accessibility tool (as in Case 1), or by using a custom web browser (as in Case 2). Restructuring and custom enhanced interactivity are not addressed in this approach.

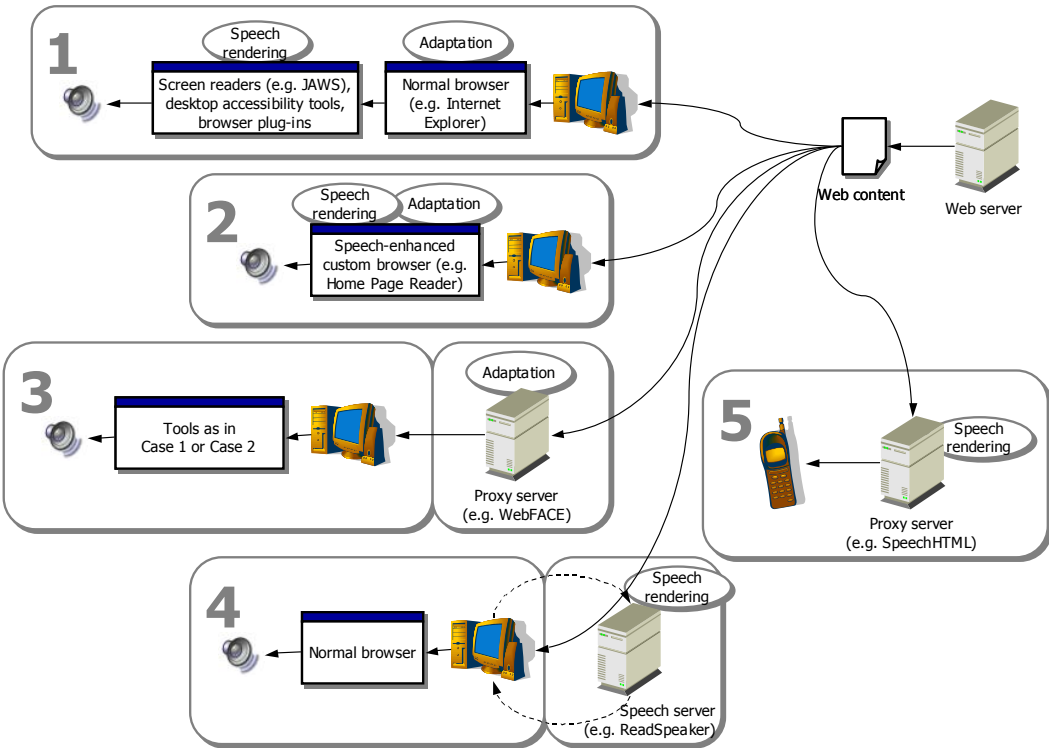

Figure 2. Rough overview approaches for the speech-enhanced web access.

- ***Case 4***. In this case speech rendering is undertaken by a remote "speech server". A normal web browser is used and a synthesized spoken version of the web page is produced and transferred by the speech server on demand, e.g. when the user presses a designated hotkey. The ReadSpeaker system is the main example of this approach.

  The advantage of that is that the users need not keep a TtS component locally at their computers. An drawback is the somehow increased demand on connection bandwidth since audio needs to be transferred

from the speech server to the user's computer. Restructuring and custom enhanced interactivity are not addressed in this approach either.

- ***Case 5***. A last approach that is worth mentioning falls in the category of voice browsing, where a user obtains a speech-based connection with the web content through a proxy server. The SpeechHTML system is an example of this approach.

2 The WebSpeech System

WebSpeech is a tool developed at the Institute for Language and Speech Processing. It is intended to be an accessibility enhancement tool that not only enhances websites using speech but also provides support for page restructuring and enhanced interactivity.

2.1 Functional Description

WebSpeech's main focus is not only to make web content accessible through synthesized speech, but also to offer more efficient presentation and interaction patterns and facilitate the browsing process making it more intuitive and comprehensive.

The requirements when *listening* to the content of a webpage are quite different than those when *reading* it from the screen or a printout. WebSpeech's adopts a quite natural approach for reading a webpage which consists of two stages: the first focuses on conveying higher level info such as the title of the page and the number and titles of its main sections, and the second on reading exhaustively the contents of each section. This permits a fast understanding of the content's structure followed by a detailed reading, and presents similarities to the natural way of approaching a new text.

The most intuitive paradigm for how a web page should be read, is when a person actually reads it to another person, for example through the telephone. For instance, imagine a person being a "Facilitator" or the "access point" of another person, the "User". The User is in his car, the Facilitator is in front of a computer, and the two are communicating through a telephone. The task of the Facilitator is not just to lookup up some specific information for the User, but rather to mediate the User's access to the web. A typical "session" would start with the User asking for a specific web page, probably the starting page of a site. When the page is loaded, the information needs to be conveyed to the User.

To illustrate and clarify things, a specific example site will be used, namely the website of the General Secretariat for Research and Development in Greece (http://www.gsrt.gr), whose English entry page is show in Figure 3.

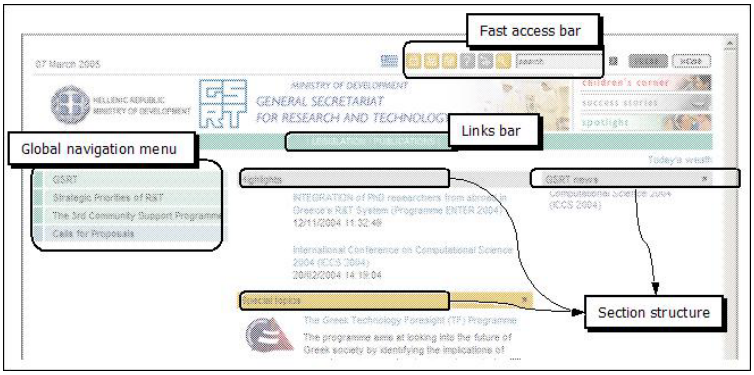

Figure 3. The example of the website of the General Secretariat for Research and Development.

The Facilitator would probably start by an overall account of the web page structure, describing what is available:

- We are at the starting page of the General Secretariat for Research and Development
- There is a global navigation menu with 4 items
- There is a fast access bar with 2 items and an edit field for searching
- There is an links bar with 7 items
- The text in the page is organized in 3 sections.
    - 1. "It interests you…"
    - 2. "It is worth seeing…"
    - 3. "GSRT news…"

With these few phrases, the User on the other end of the line has already gained a general understanding of the current web page and has formed in his mind a rough perceptual model of it. After that, the Facilitator would probably start reading more thoroughly the items in each of the sections.

Of course, the User can understand some commands issued by the User and, for example, provide more information on the available items in the global navigation menu if asked. If an "input device" such as a speech recognition module cannot be assumed, the User would need to use alternative means to ask for that information, e.g. a keyboard. In that case, he would need to know how he can request that, e.g. which are the necessary keystrokes. So, a more appropriate description would be: "There is a global navigation menu with 4 items, accessible by pressing Alt+Ctrl+M".

The aim of WebSpeech is to take the role of the Facilitator in the above example. To this end, it needs to:

- *Identify site elements* and elements with a specific role. A priori information is necessary for this task. Elements such as the global navigation menu, the current position in the site, the current page title, list of documents and other resources accompanying a web page, and so on.
- When appropriate, extract these identified elements from the rest of the page and wrap them in more usable and intuitive *access mechanisms*. A priori information is also necessary for this task. For example, the global navigation menu should be
- *Identify the section structure* of the page content. If content systematically follows accessibility guidelines, this is easy. In other case, heuristics need to be employed also based on the font formats used.
- *Produce a summary* of the page, describing the available role elements and outlining the sections of the content.

These are schematically presented in Figure 4. An additional set of functionalities are supported such as speech-enhanced navigable lists of any global menu, of any fast access links, of all links in the current page and so on.

2.2 Design and Deployment

There are two main alternatives when deploying a local browsing helper or accessibility tool; to build a custom stand-alone browsing environment or to enhance one of the widely used commercial browsers, e.g. though plug-ins or add-ons. When applicable, the first option seems preferable. Designing a custom browser can provide to the application developer full control of the interaction patterns with the user, but it represents a closed solution. On the other side, enhancement tools from different providers in the form of add-ons running on the same browser can provide combined advantages by simultaneously addressing different needs or offering complementary services.

WebSpeech has been developed for Microsoft Windows and deployed as a browser add-on rather than a stand-alone custom browser, as shown in the Figure 5. It is a client-only solution for speech enhanced web browsing.

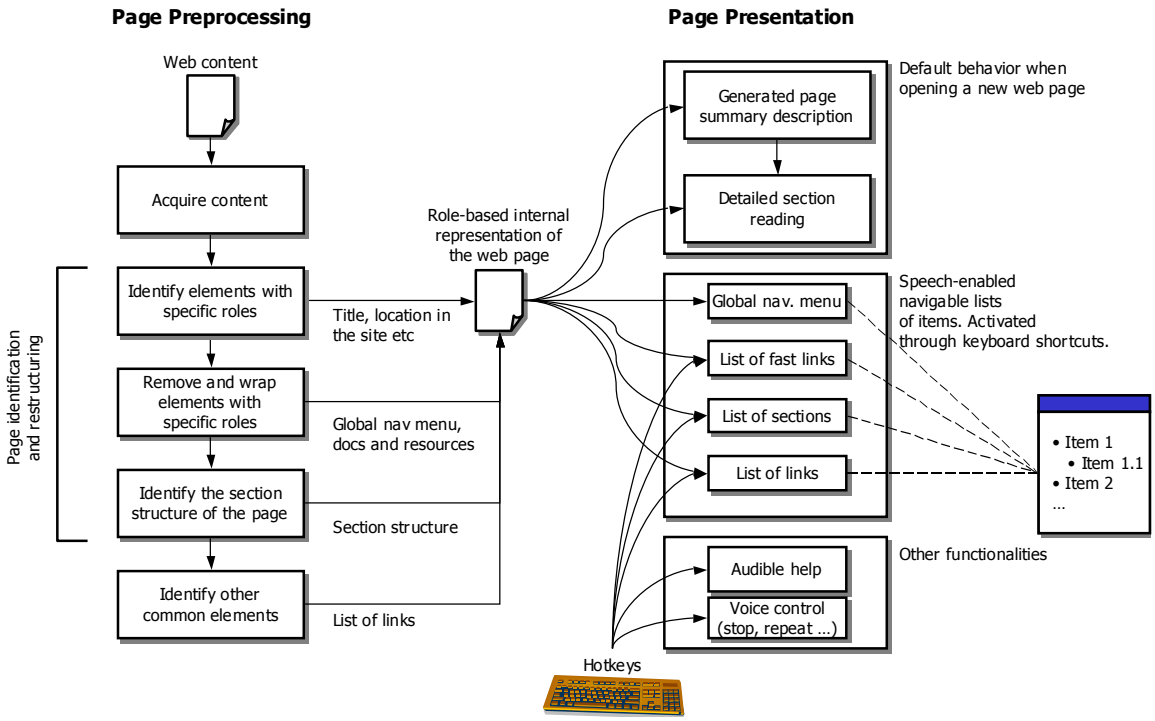

Figure 4. The processing steps and presentation patterns in WebSpeech.

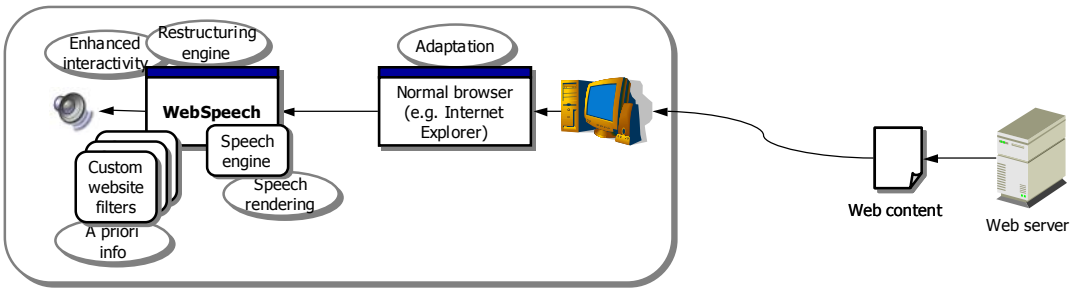

Figure 5. The WebSpeech deployment.

To be able to identify the roles of page elements and provide enhanced interactivity, WebSpeech requires a priori knowledge of a website. These are encapsulated into site-specific *filters*, each containing all the necessary specific information for a website along with specific interactivity patterns when necessary. The core system includes all the generic necessary technological components, such as the speech synthesis engine, the communication with the browser, the support for keyboard shortcuts and other common functionalities.

For Greek, WebSpeech employs ILSP's text-to-speech system (Ekfonitis+). English are be supported through the Microsoft Speech API. Similarly, any other language can be supported if a speech engine is available in the client PC.

WebSpeech maintains an internal history of cached internal representations of pages visited in the current browsing session. This allows it to speed up processing time but also to read content more in context. For example it can understand when users are first entering a page of a site and notify them accordingly. Moreover, it will only read site elements once for the first page entered. Of course, users can explicitly request information on all the available elements.

An important note to be made is that WebSpeech poses no requirements on a page and introduces no overhead to the design and development, and required no changes to the procedures used to store, maintain and update the content of a website. Based on its engine/filter approach, it can deal with any website without any modifications in its engine. It is only a mater of implementing a specific filter for a website; for consistently designed sites (for example sites that are backed by a content management system) this can be a mater of a few days. This also allows it to support existing websites, and even web content that does not follow the accessibility guidelines.

WebSpeech has already been used for speech-enabling the website of the General Secretariat of Research and Development (GSRT) in Greece. A custom filter designed to support GSRT's current web structure along with the WebSpeech tool itself, will be made freely available to the end users. A set of evaluation activities are currently taking place to assess the quality and usability of the tool, while additional experiments involving wider user groups are being planned.

3 Conclusions

In summary, WebSpeech lies in between a generic and a specialised accessibility tool trying to combine advantages from both categories. Its engine/filter design allows it to support most websites and to efficiently compensate for any deviations from the recommendations and standards for accessible design. Common browsing tasks such as following links, navigating back and forth, listening to the page contents again, navigating to site menus, obtaining audible help etc. are supported through hotkeys. Relying on site-specific filters to appropriately identify and restructure the page content, it achieves a natural way of conveying the information to the user, quite similar to the way one would use to read a page to a colleague through the telephone.

Clearly, WebSpeech extends beyond the limits of a desktop accessibility enhancement tool. Its approach can be also used as a basis for implementing voice browsing where users can interact with web pages through devices as simple as the telephone.

4 References

Kouroupetroglou, G. and Mitsopoulos, E. (2000). Speech-Enabled e-Commerce for Disabled a nd Elderly Persons, *COST 219 Seminar: Speech and Hearing Technology*, Nov. 22, 2000, Cottbus, Germany

Savidis, A., & Stephanidis, C. (2004). Unified User Interface Design: Designing Universally Accessible Interactions. *International Journal of Interacting with Computers*, 16 (2), 243-270.

Stephanidis C., Paramythis A., Karagiannidis C. and Savidis A. (1997). Supporting Interface Adaptation in the AVANTI Web Browser, *3rd ERCIM Workshop on User Interfaces for All*, Obernai, France, 3-4 November

Hyperlinks

Home Page Reader from IBM: http://www-306.ibm.com/able/solution_offerings/hpr.html

JAWS from Freedom Scientific: http://www.freedomscientific.com

ReadSpeaker from ReadSpeaker: http://www.readspeaker.com

SpeechHTML from Vocalis: http://www.vocalis.com

W3C links on alternative web browsing: http://www.w3.org/WAI/References/Browsing

W3C Voice Browser Activity: http://www.w3.org/Voice/

W3C Web Content Accessibility Guidelines: http://www.w3.org/TR/WAI-WEBCONTENT/

WebFACE from FORTH: http://www.ics.forth.gr