Enhancing Accessibility of Web Content for the Print-Impaired and Blind People

Aimilios Chalamandaris[1,2], Spyros Raptis[1,2], Pirros Tsiakoulis[1,2], and Sotiris Karabetsos[1,2]

[1] Institute for Language and Speech Processing – Athena Research Center
[2] Innoetics ltd, Knowledge and Mutlimodal Interaction Technologies
Artemidos 6 and Epidavrou, 15125 Athens, Greece
{achalam,spy,ptsiak,sotoskar}@ilsp.gr,
{aimilios,sraptis,ptsiak,sotoskar}@innoetics.com

Abstract. Blind people and in general print-impaired people are often restricted to use their own computers, enhanced most often with expensive, screen reading programs, in order to access the web, and in a form that every screen reading program allows to. In this paper we present *SpellCast Navi*, a tool that is intended for people with visual impairments, which attempts to combine advantages from both customized and generic web enhancement tools. It consists of a generically designed engine and a set of case-specific filters. It can run on a typical web browser and computer, without the need of installing any additional application locally. It acquires and parses the content of web pages, converts bi-lingual text into synthetic speech using high quality speech synthesizer, and supports a set of common functionalities such as navigation through hotkeys, audible navigation lists and more. By using a post-hoc approach based on a-priori information of the website's layout, the audible presentation and navigation through the website is more intuitive a more efficient than with a typical screen reading application. *SpellCast Navi* poses no requirements on web pages and introduces no overhead to the design and development of a website, as it functions as a hosted proxy service.

Keywords: Web accessibility, Speech synthesis, Print-Impairment, Screen Reader.

1 Introduction

This paper presents *SpellCast Navi*, a web based tool that can serve as a web-reader used by blind and print-impaired individuals. Currently, the typical aids for blind computer users, as far as computer usage is concerned, are software programs called screen readers, and they are responsible for converting visual information of the computer screen into speech, providing at the same time the necessary shortcut keys for navigating through programs and screens. These programs, more often, are expensive because of their complexity and the relative small market they are addressing, posing a significant obstacle to the blind computer users to acquire one, through private means.

Similarly to the screen readers, which address mainly blind people as their main target group, there are other applications and tools that aim to simply enhance visual

A. Holzinger and K. Miesenberger (Eds.): USAB 2009, LNCS 5889, pp. 249–263, 2009.

interfaces with aural and spatial information, in order to accommodate cases such as of dyslexia or illiteracy. These programs typically enhance visual features of the interface and often incorporate text to speech synthesizers for producing aural interfacing to the content. These applications are also most of the times commercial, with costs that vary from few Euros to a few hundreds Euros, depending on the complexity of the functions they provide.

In this paper we present a hybrid approach for providing an efficient and generic aid to print-impaired people, which actually can be customized and be adapted according to personal needs, with different settings and preferences for users of different impairments, such as blindness, limited vision, or dyslexia. Additionally to these target groups, one can also include people with temporary inability to read textual content, like for instance while driving a car, or people who face difficulties in reading in a specific language, like for example immigrants.

In the following section of this paper a brief overview of available web accessibility technologies is given, along with a classification of the different approaches met in the industrial and research field of web accessibility software. The *SpellCast Navi* system is described in section 3, with enhanced details when necessary, about its design, operation and the functionalities it offers. Finally, a discussion of a pilot evaluation phase of the systems and of the future plans for further developing the systems are given in the final paragraphs of the paper, in order to provide a more complete idea of the described system and its roadmap for evolution.

2 Web Accessibility Technologies and Tools

The World Wide Web's increasing importance and penetration into daily 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 [1].

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 specialty. 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 and adapted to the specifics of a website will perform better and more efficiently, since by being able to exploit available a-priori information of the site's structure, it can convey information more coherently to the user, while at the same time it can cope more efficiently with different types of impairments, by adjusting its features and its functionalities. Moreover, it will be able to support richer and more accurate interaction patterns than in the case of a typical screen reading program. The choice of the appropriate tool is strongly linked with the intended application and the required use cases.

Web accessibility means access to the World Wide Web by everyone, regardless of the type of the disability [2]. The following discussion in this section is confined mainly to the case of web accessibility technology for visually impaired people and it provides a general idea of the status of the web accessibility currently.

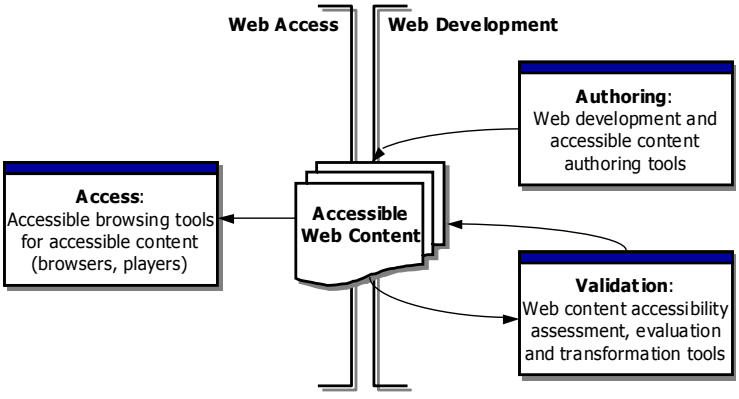

Fig. 1. The main parts involved in web accessibility: The content, the authoring tools, and the browsing tools

Web accessibility evolves around content [3]. Content needs to possess a set of attributes, which ensure the completeness of the information it is intended to convey to the user. This is directly related to the way content is presented, i.e. the content access, and significantly affects the way it is produced, that is content authoring and development.

There has been a lot of research and 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 [4], and especially to its Web Content Accessibility Guidelines [5], the Voice Browser Activity [6], and the Alternative Web Browsing considerations.

2.1 Accessible Content

As already mentioned, web accessibility evolves around web content. Obviously, the major role of the web content is to capture and convey information to the user. Content presentation is of a separate concern, involving creating views and projections of the content to specific representational patterns in order to provide ease or additional spatial information. 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, namely web content desired to be accessible via screen reading programs, possesses a set of attributes that ensure sufficient consistency and presentational independence (often calling for redundancy) of the information it is intended to provide. 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. The primary goal of the W3C’s Web Content Accessibility Guidelines is to promote content accessibility and to set the common ground for consistent behavior of aid-tools, such as of screen readers. These guidelines are summarized as a list of organized and prioritized checkpoints that web pages need to be verified against, along with suggested techniques for adopting them.

Some typical examples of specific aspects covered by the WCAG document are alternative text for images, tables that make no sense when their elements are read in a sequential manner, web forms that cannot be navigated into with a meaningful order and so 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. In order to avoid marginalizing such users, content itself should not be bound to its visual appearance, and should provide what is necessary for these sophisticated and attractive page layouts to significantly reduce to meaningful audible counterparts.

2.2 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 have evolved significantly in the recent years; however they are out of this paper's scope and therefore they will not be covered. Detailed information on the technologies and tools can be found through the W3C website.

2.3 Content Presentation and Access

Software used to access web content is usually referred to as user agent. Obviously, accessible user agents that can make web content truly available to all users are the terminus. These tools, unless they incorporate a-priori information about the layout of the site they are asked to process, they simply convey the textual information of the content with limited spatial and prioritized information. What however is necessary is to employ efficient content presentation schemes, tailored to the special needs of their users, enhancing the content perception and understanding. Moreover, these tools need to implement appropriate user interaction patterns so that the users can effectively interact with the content and navigate through 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 people are shortly discussed in the following paragraphs.

2.3.1 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 allow the people to interact via spoken commands and synthetic speech and navigate through the web, with alternative means than the typical mouse

or keyboard. They offer the promise of allowing everyone to access web-based services from any phone, making it practical to access the web anytime and anywhere, whether at home, on the move, or at work. Work in the field of voice browsers is closely related to people with visual impairments since voice is used as a replacement of the 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 ones mentioned above, and which deviate from the typical browser setting, providing specific support for specific types of disabilities.

2.3.2 Content Transformations

There are several different operations that need to be carried out in order to convert a web page into an audible form. Some of the operations, most relative to this discussion are the web page adaptation, restructuring and, finally, rendition of it into synthetic speech.

2.3.2.1 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 the 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, are only a few of the adaptation processes that need to be carried out.

Adaptation can take place locally in the user's (client) computer, e.g. as in the case of the AVANTI web browser [7], 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 [8] and WebAnywhere [9].

2.3.2.2 Restructuring of the Web Page and Custom Interactivity. Restructuring of the web page involves determining the role of the web page elements and element groups, and provide the user with a more efficient manner to access them.

The role of an element is not only related to the type of the element (e.g. an edit box 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 which 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 worth noting that although website templates significantly differ in their aesthetics and their spatial layout, 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 they become very familiar with them, but also they actually try to identify such structures in every new website they visit, as a fist step of familiarization with the website.

An important property of some of these elements is the fact that they appear in every page of the site providing a tree log of the website. This property assigns to them a more global characteristic, that is, as website's functional elements rather than web page elements. Reading these elements to the user in each page rarely makes sense and it would probably tire the user. Computer users are somehow familiarized with and aware of the website's layout, and they have always in mind that these elements exist in the website and they can follow them anytime they want; however they do not pay attention to them in every web page of the website they visit. Similarly, print-impaired people should not be provided with the same information on every page they visit, but simply they should be notified about their existence only at their request and not in every page they navigate to.

Identification and proper handling of elements with specific roles in a website can provide the means for supporting custom enhanced interaction patterns, which can significantly improve the usability of a website and the quality of the interaction with the user. The navigation through the website is more efficient and less tiring for the print-impaired user. However, this can only be accomplished with a priori information about the design of the specific website.

2.3.2.3 Speech Rendering. Speech has become a mainstream technology in the computers field. Speech synthesis (text-to-speech, TtS) [10] is particularly relative to visually impaired people, since it can be one of the most effective substitutes for vision. Text-to-Speech systems are now widely available for many different languages and are supported even at the level of the operating system as, for example, in the case of Microsoft Speech API for Windows, while their performance has significantly improved during the last few years mainly due to the adoption of statistical methods [11].

In the context of accessibility tools and as far as the location of the software is concerned, 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, like for instance is IBM's Home Page Reader [12]. This does not pose any overhead in the web connection bandwidth, since speech rendering takes place locally, by only consuming local computational resources. However, in this case, the software requires to be downloaded and installed, including regular updates.
- Remotely as a service provided by a speech server like *SpellCast Navi*, ReadSpeaker [13], Talklets [14]. In this case there is no need to install locally a Text-to-Speech software, with the trade-off of higher requirements on the connection for audio files transfer (higher internet bandwidth).

Although in the recent years the local version of TtS software was preferable due to speed limitations as far as the internet connections were concerned, currently they provide a better architectural scheme overall, since they can be maintained more easily and efficiently and they can offer service to the user immediately, to a wider set of devices, without the need of any software installation.

2.4 Existing Systems

A brief overview of the mostly used approaches for the speech-enhanced web access is provided in Figure 2 below. Some features of each category are given below.

- *Category 1*. In this category the most simple and common settings for speech-enhanced web access are included. A normal web browser such as Internet Explorer, Firefox, Safari etc., has the role of accessing and retrieving the web content, while a separate tool such as a screen reader, like for instance JAWS [15], a desktop accessibility tool, like for instance Apple VoiceOver [16], or a web browser plug-in (e.g. FireVox [17]) are responsible for rendering the visited web pages through synthetic speech. In this case, the browser is responsible for performing any necessary adaptations to the web content to meet user needs and preferences. In this category, no restructuring or custom interactivity can be supported since there is no way to exploit a-priori website information.
- *Category 2*. A slightly different approach is that of an entirely custom browser. The new 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, this approach does not support web page restructuring. However, special versions of these tools have been deployed in order 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).
- *Category 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 Category 1), or by using a custom web browser (as in Category 2). Restructuring and custom enhanced interactivity are not addressed in this approach.
- *Category 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 and SpellCast basic are typical examples of this approach. The advantage of this approach is that the users do not need to install any software component locally at their computers while at the same it is cross-browser and cross-platform available. Restructuring and custom enhanced interactivity are not addressed in this approach either and audio files are provided as an independent stream of media.
- *Category 5*. A more recent approach is when a user obtains a speech-based connection with the web content through a proxy server. The SpeechHTML [18] system and WebAnywhere are examples of this approach as well as our approach, *SpellCast Navi*. The reconstruction of the content into distinct elements and its prioritization is also performed in this approach, based on the use of a-priori information

of the website's structure. This approach is the most recent one since it is based on technologies and infrastructures that only recently have shown significant levels of maturity and robustness.

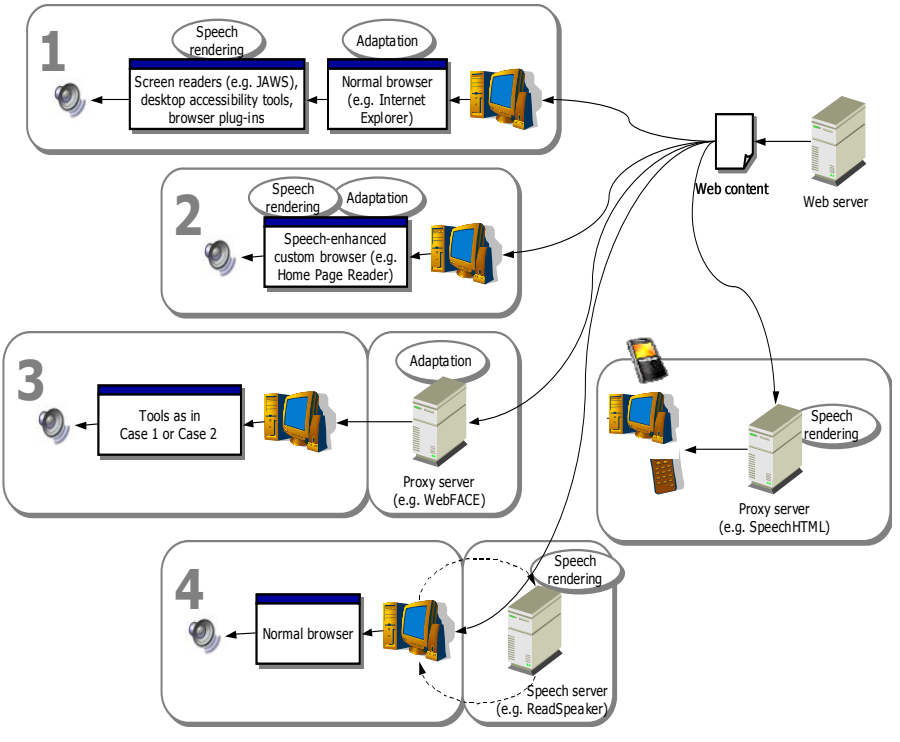

Fig. 2. Overview of different approaches for the speech-enhanced web access

3 The SpellCast Navi System

SpellCast Navi is a tool developed at innoetics ltd, a spin-off company of the Institute for Language and Speech Processing of Research Center Athena, in collaboration with ILSP and it is based on an entirely different approach compared to its predecessor WebSpeech [19].

It is intended to be an accessibility enhancement tool that not only enhances websites via synthetic speech, but it also provides support for page restructuring and enhanced interactivity. It is designed to be a cross-platform and cross-browser application and it operates without the need of any software installation on local computers.

The design of this approach aimed to fulfill a three-fold goal:

1. To provide the functionality of a screen reader for accessing web sites without the need of installing one locally.
2. To provide service in a cross-platform and cross-browser manner, and run on different devices.

3. To offer advanced functionalities and advanced interactivity with the user, depending on the user's preferences and based on a priori information about the website.

By doing so, our approach seems to cover efficiently some of the most important issues concerning the web accessibility and the print-impaired people, while at the same time it constitutes a free, easy to use aid, with intuitive interface for all people. It offers high availability for everyone with advanced functionalities, while, maybe most importantly, it is freely available for all end-users.

3.1 Functional Description

SpellCast's main concern 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 by providing a more intuitive and comprehensive aural interface.

The requirements when listening to the content of a webpage are significantly different than those when reading it from the screen or a printout. Our methodology adopts a more natural approach for reading a web page than when using a typical screen reading application, and it consists of two stages: (i) the first one focuses on conveying higher level information such as the title of the page and the distinct elements on the page, such as menus, links etc, and (ii) the second one focuses on reading exhaustively the contents of each section. This permits a quick browsing of the content's structure of the webpage, followed by a detailed reading of the section the user selects. This approach simulates more efficiently the way a non-impaired person would skim a new web page before starting reading it thoroughly.

The most intuitive paradigm for how a web page should be read is when a person is asked to read a webpage aloud to another person, trying to provide, as efficiently as possible, and any information that is implied in the web page. The role of a content mediator is not just to lookup for some specific information for the user, but rather to mediate the user's access to the web content. A typical "session" would include three different steps:

1. Initially to provide a general outlook of the webpage by identifying the title of the webpage and by providing once, only at the beginning of the session or at user's request, the information and the necessary help about the use of the service and about the options given to the him at any time by the service (i.e. menu lists, links lists, shortcut keys etc).
2. The provided service initiates to read aloud all textual web-content, providing also information about different items that are included in the web page, such as images, tables etc. The prioritization of the content elements is performed through a pre-programmed internal service filter that identifies different page elements based on a-priori information fused by the service provider.
3. At any time during the session, the user is given the choice to navigate through restructured audible lists of the page's menus, the page's links and the page's identified distinct textual items. By doing so, the navigation through the website via this enhanced interface, simulates better the way a non-impaired person would be able to do so in the website, rather than with a simple screen reader. In other words, the visually impaired person can navigate, in a non serialized manner

through the website, without the need to go through potentially unnecessary information that would tire him or make the session less efficient.

In order to illustrate and clarify the processes of the service, a specific example site will be used, namely the website of the Secretariat General of Communication-Secretariat General of Information of Greece (http://www.minpress.gr), of which the website's entry page is shown in Figure 3.

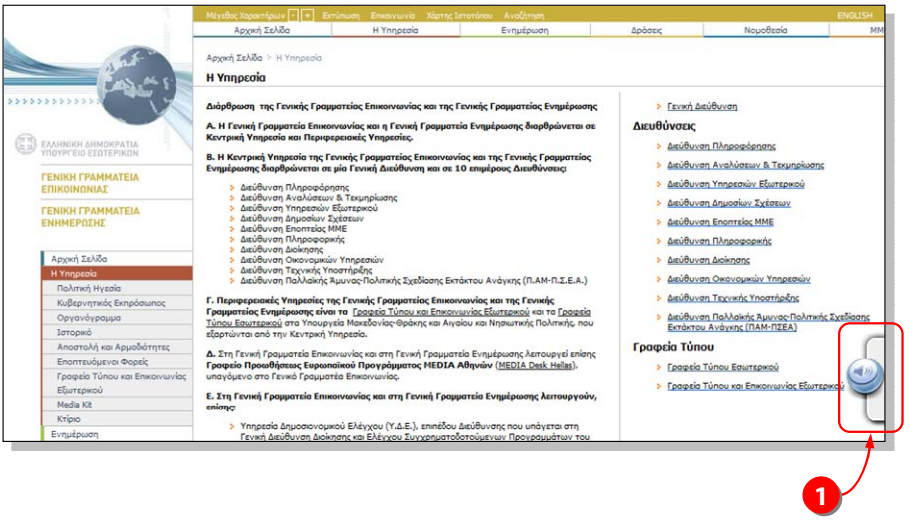

Fig. 3. The example of the website of the Secretariat General of Communication-Secretariat General of Information of Greece (Greek ministry of press)

In the first page and at the beginning of the session the user is notified both visually (for the non-blind people) and aurally, that the website is supported by the *Spell-Cast Navi* service and it can be activated through a combination of hotkeys.

Should the user chooses to activate the service, the user is then given the most essential information about the website, such as the webpage's title, and possible shortcut key combinations for help, menu and link items lists, and navigation inside the webpage elements, such as paragraphs, phrases or tables.

This process is designed to be carried out only once at each user session, via the use of cookies technology. After the first time, the user can listen to this information again at his request, by pressing a hotkey.

Once the process of reading aloud the webpage has started, additional helping tools like highlighting of the current phrase and/or word, and the increased size and contrast of the current cursor position, also provide extra visual aids for people with other print-impairments than blindness, such as dyslexia, limited eyesight, or even illiteracy issues. As it is depicted in figure 4, although the page includes menu items and links that are consistent throughout the website, and they are not read aloud on every page the user visits, but the user can activate any of these menus and browse through them,

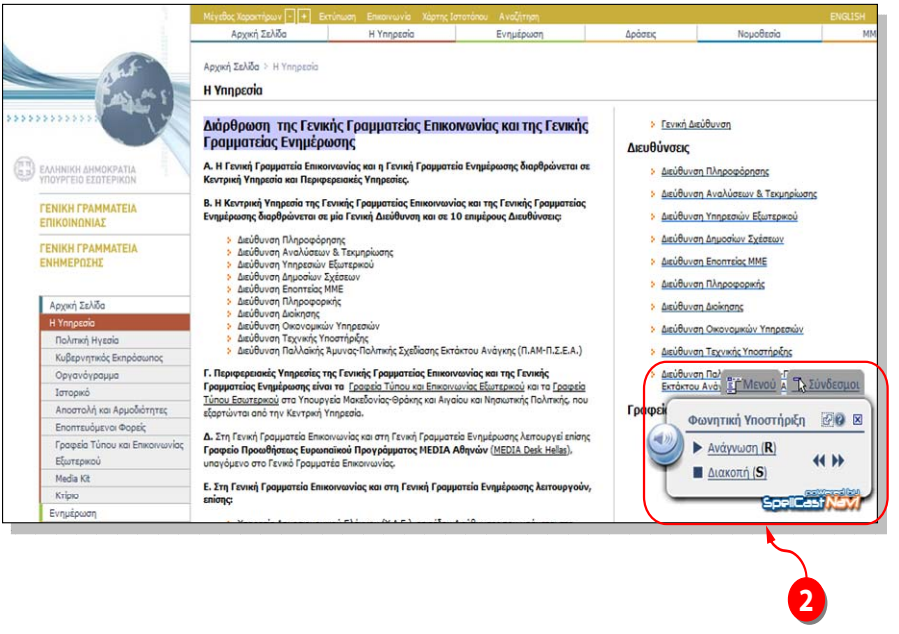

Fig. 4. The example of the website of the Secretariat General of Communication-Secretariat General of Information of Greece (Greek ministry of press). Reading aloud of a webpage.

at his request. This technique improves significantly the interactivity of the user with the website and makes navigation in it more efficient, less time consuming and effortless. One has to note here though that the above information, i.e. of what is a menu and what is the main content item on a website, is fused manually in the service by making through a scripting language especially designed for this service. Although this seems to be one of the drawbacks of this approach, it is only performed once during the adaptation of the service to the specific website, ensuring a more consistent and efficient web reading session for the user.

It remains at the user's request, at any time during his navigation through the website, to stop or restart the audio, jump to next or previous elements, browse in menu items and link items lists and follow one of them by simply selecting one with the arrow keys and the enter button on his keyboard. By doing so the visually-impaired user is gaining quickly an intuitive structure of the website, similar to the one a non-impaired user shapes when viewing the website on a computer screen, without the need for the repetitive pronunciation of items that are present on all web pages of the site.

3.2 Design and Deployment

As already mentioned, the developed system is designed as a web application for speech enhancing websites and for navigating in them, without the need of any software

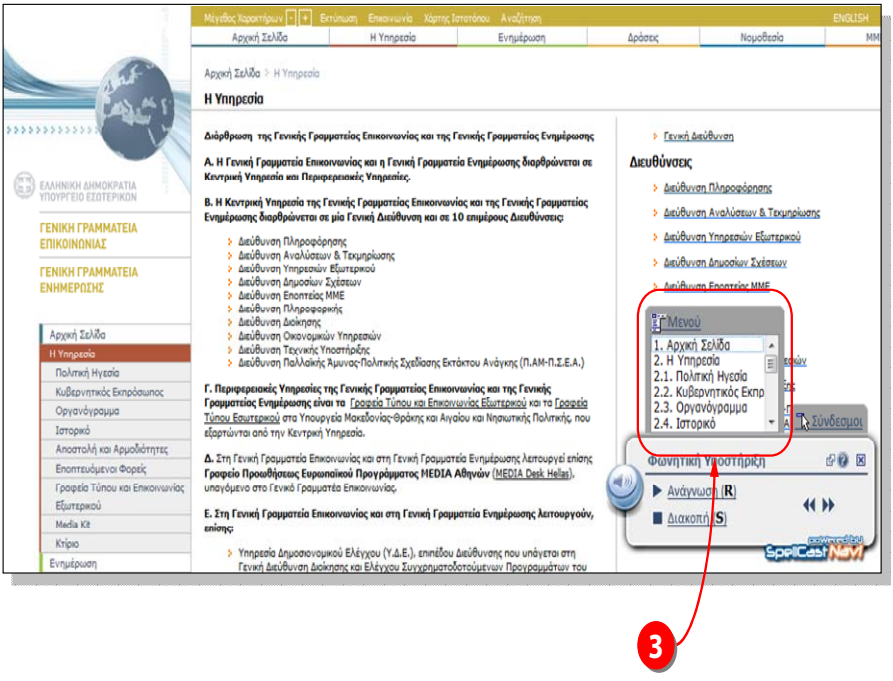

Fig. 5. The example of the website of the Secretariat General of Communication-Secretariat General of Information of Greece (Greek ministry of press). Menu items aural list.

installation on user's local computer. It is entirely developed on a client-side Javascript language, with main concern given to cross-browser and cross-platform capabilities.

The system traverses the DOM of each webpage of a specific website and through a post-hoc approach, based on a-priori information about the website's spatial structure, it re-structures its elements, and via a text to speech rendition, it provides the user with a simplified and intuitive aural representation of the webpage.

In order to be able to identify the roles of page elements and to provide enhanced interactivity, the system requires a priori knowledge of a website. These are encapsulated into site-specific filters; each one of them contains all the necessary specific information for a website along with specific interactivity patterns when necessary.

The system consists of two different components: (i) the client-side Javascript engine which is responsible for the interaction of the system with the user and for the support of the navigation through the website, and (ii) the server side component which is responsible for audio generation, caching and transformation proxy for the web pages. The proposed architecture makes the task of maintenance easier and it ensures high availability and quality levels of the service.

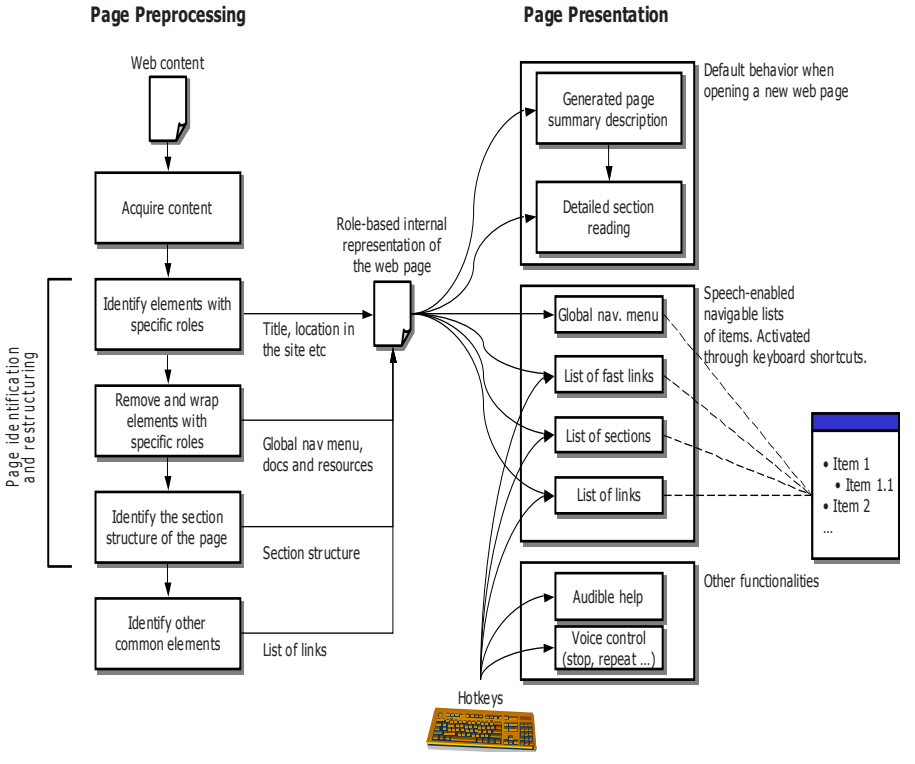

Fig. 6. The processing steps and presentation patterns in SpellCast

The text-to-speech component is incorporated on server-side with additional components such as website specific pronunciation lexica, and a sophisticated production and caching schema of the audio files, for every website's content, ensures high availability and low latency. The latter provides better resources management with improved scalability.

One should note at this point that our system poses no requirements on a website and introduces no overhead to its design and development or functioning processes. Based on its engine + filter approach, it can deal with any website without any modifications in its engine.

For every supported website, there is simply the need to develop a customized filter, which for consistently designed sites (for example sites that are backed by a content management system) it can be a mater of a few days. One of the main advantages of this target-specific approach, as far as the web content is concerned, is that it allows to any web content, even to content that does not conform to the accessibility guidelines, to become accessible through different devices and systems.

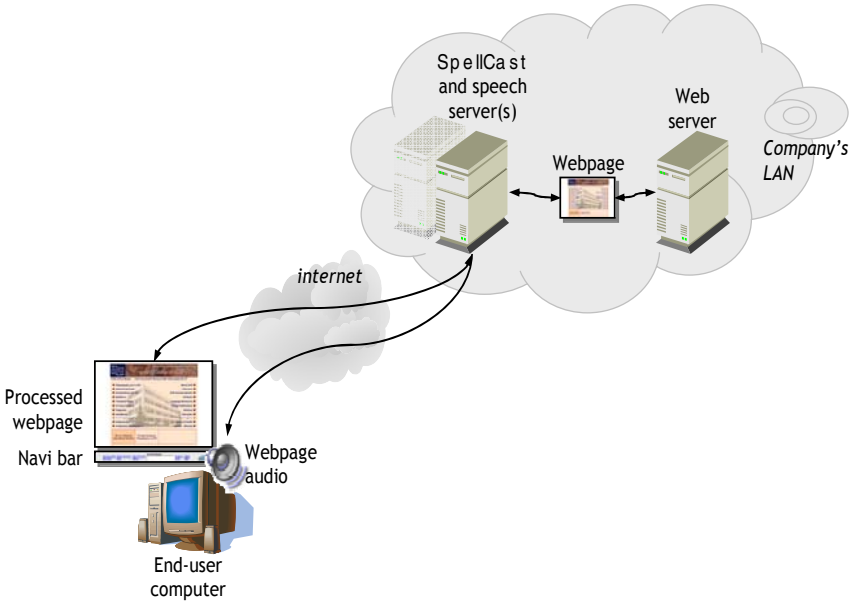

Fig. 7. The *SpellCast Navi* deployment

4 Discussion

SpellCast Navi has already been deployed onto several large websites, mainly of the Greek public sector, and it provides an important tool for rendering vast web content accessible to end users through a web application that can work the majority of computers and web-enabled devices.

Pilot evaluation tests have shown that it offers advanced usability and improved web accessibility compared to typical screen readers and large scale evaluation tests and experiments are currently being carried out. Future improvements have been planned for the development and evolution of the system, such as customizable visual aids for different types and levels of visual impairments, and support for more web elements such as different file types and so on.

In summary, *SpellCast Navi* lies in between a generic and a specialized accessibility tool trying to combine advantages from both categories. Its engine + filter design allows it to support individual websites and to efficiently compensate for any deviations from the recommendations and standards for accessible web design. Common browsing tasks such as following links, navigating back and forth, listening to the page content again, navigating to site menus, obtaining audible help etc., are supported through hotkeys and it provides enhanced aided web navigation.

Relying on site-specific filters to appropriately identify and restructure the web content, it achieves a natural way of conveying the information to the user, by trying to simulate the way one would use to read aloud a website to another person and help him navigate through its pages. It extends beyond the limits of a desktop accessibility enhancement tool and its approach can be also used as a basis for implementing voice

browsing where users can interact with web pages through devices as simply as with the telephone.

References

1. Savidis, A., Stephanidis, C.: Unified User Interface Design: Designing Universally Accessible Interactions. International Journal of Interacting with Computers 16(2), 243–270 (2004)
2. Coyne, K.P., Nielsen, J.: Beyond alt text: Making the web easy to use for users with disabilities. Design Guidelines for Websites and Intranets Based on Usability Studies with People Using Assistive Technology (2001)
3. Bigham, J., Cavender, A.C., Brudvik, J.T., Wobbrock, J.O., Ladner, R.: WebinSitu: A comparative analysis of blind and sighted browsing behavior. In: Proc. of the 9th Intl. Conf. on Computers and Accessibility, ASSETS 2007 (2007)
4. W3C links on alternative web browsing, http://www.w3.org/WAI/References/Browsing
5. W3C Voice Browser Activity, http://www.w3.org/Voice/
6. W3C Web Content Accessibility Guidelines, http://www.w3.org/TR/WAI-WEBCONTENT/
7. Stephanidis, C., Paramythis, A., Karagiannidis, C., Savidis, A.: Supporting Interface Adaptation in the AVANTI Web Browser. In: 3rd ERCIM Workshop on User Interfaces for All, Obernai, France, November 3-4 (1997)
8. WebFACE from FORTH, http://www.ics.forth.gr
9. Bigham, J.P., Prince, C.M., Ladner, R.E.: Webanywhere: A Screen Reader On-the-Go. In: Proc. of the Intl. Cross-Disciplinary Conf. on Web Accessibility, W4A (2008)
10. Black, A., Taylor, P., Caley, R.: The Festival Speech Synthesis System (1998), http://festvox.org/festival
11. Hunt, A., Black, A.: Unit selection in a concatenative speech synthesis system using a large speech database. In: Proceedings of ICASSP 1996, Atlanta, Georgia, vol. 1, pp. 373–376 (1996)
12. Home Page Reader from IBM, http://www-306.ibm.com/able/solution_offerings/hpr.html
13. ReadSpeaker VoiceCorp Website, http://www.voice-corp.com/
14. Talkets Technologies, http://www.textic.com/
15. JAWS from Freedom Scientific, http://www.freedomscientific.com
16. Apple Accessibility VoiceOver, http://www.apple.com/macosx/accessibility/
17. Chen, C.: Fire vox: A screen reader firefox extension (2006), http://firevox.clcworld.net/
18. SpeechHTML from Vocalis, http://www.vocalis.com
19. Raptis, S., Spais, I., Tsiakoulis, P.: A Tool for Enhancing Web Accessibility: Synthetic Speech and Content Restructuring. In: Proc. HCII 2005: 11th International Conference on Human-Computer Interaction, Las Vegas, Nevada, USA, July 22-27 (2005)