CUSTOMIZING HTML BY FILTERING: techniques for making the Internet accessible to the visually handicapped

Djamel HADJADJ[1], Robert AGRO & Dominique BURGER[2]

Institut de la Santé et de la Recherche Médicale
UPMC B239, Quai Saint Bernard
75252 Paris cedex 05

[1]djamel.hadjadj@snv.jussieu.fr
[2]Dominique.Burger@snv.jussieu.fr

ABSTRACT

This paper describes new techniques for making HTML documents more accessible to all users. Since visually handicapped users have great difficulty accessing the Web, this discussion focuses on their special needs. We have developed a software to ensure that adaptation of documents is independent of their production and delivery. The software is compatible with existing recommendations for HTML design [4][11], is readily customised and updated. The software is implemented in the context of Microsoft ActiveX.

INTRODUCTION

Internet servers provide information via the unified HyperText Transfer Protocol (HTTP), and use the HTML [7] language to describe the structure and content of the information delivered. Thus, this information is fairly easy to interpret and transform. Access solutions for the visually handicapped have taken advantage of these features. The text browser, LYNX [5], for instance, adds a label before hypertext links to make them clearly perceived on a Braille bar or when they are spoken by a speech synthesiser. A summary of links can also be provided at the beginning of the document to facilitate rapid access to them, and the WAB proxy server developed at the University of Zurich has also several more features [12]. Both applications process HTML source documents and produce an adapted version. This is referred to as *HTML filtering* in this paper.

Since filtering functions are general, they should be implemented in reusable components that are independent of both the server (being able to process any source document) or the browser (being compatible with all presentation component). This is the main topic of this paper, which is based on our experience of developing a Web browser for the French BrailleNet[1] project, 'Internet for the Education of the Visually Handicapped'.

The presentation briefly reviews the main elements encountered in HTML documents, and gives examples of how they can be made more accessible when " read " via a Braille display or speech synthesiser. The software and its implementation are then presented.

HTML ELEMENTS

The main elements in an HTML document are :

Text

Simple text strings with traditional attributes of font, colour, style and flashing.

[1] BrailleNet is run by the INSERM in co-operation with several universities, special schools, companies, and associations

Headings

Headings define the logical structure of the document. The tag used is <H$_n$> where n is the identifier of the title level.

Anchors & Links

Anchors and links make it possible to navigate within an HTML document. Users encounter them as text segments which can be clicked on to go to a specific position within the current page or to download another page. Links and Anchors are defined by the <HREF> tag.

Images

Several images formats (BMP, GIF, JPEG, AVI, etc.) can be included in HTML pages using the <IMG> tag.

Tables

These are composed of cells forming rows and columns. Defined by the <TABLE> tag.

Forms

A form is composed of standard objects (Button, Radio button, List box, Combo box, Check box, etc.) similar to those used in graphical interfaces. Defined by <FORM> tag.

Frames

Some browsers can display several HTML pages in multiple windows, (2 or 3). Therefore cross references can be defined between pages using frames defined by the <FRAM> tag.

ADAPTATION

The HTML language is flexible enough to provide a variety of presenting HTML elements or clearly describing the structure of a document. Thus, a given semantic content can be presented in various HTML forms. This is the basis of any adaptation method based on processing HTML source code. Several types of transformation can be made :

Simplifying

Some elements of little semantic value, such as tags concerning the size, style or colour text, can be simply removed. Images are also often removed when - unfortunately - no comment is provided to explain them.

Clarifying

Some text attributes can be changed, especially for the visually handicapped, who might prefer enlarged links.

Rephrasing

Links are traditionally shown on the screen using different colours and styles, but these features are not perceived after transcription into Braille or speech. The links can therefore be shown as : `LINK : Design for the Blind.` Links can also be numbered (`LINK 1 : Design for the Blind, LINK 2 : Design for the Deaf).` And many other solutions are possible, such as links delimited by [ ] ; these can still be numbered, e.g. `[ Design for the Blind : 1].`

Restructuring

Even more drastic reformulation is sometimes needed for visually handicapped users to obtain a notion of the global structure of the document and for rapid navigation within the document.

Most access products provide a list of the links contained in a document, or a table of contents showing the headings of a document. This table of contents can be preceded by an anchor making it possible to bypass it. This type of adaptation of links implies the complete rebuilding of a document, since new links or anchors must be added. Thus, different transformations can be used single or in combination for each type of HTML element. Optimal solutions generally vary according to the user’s preference or the display technique used.

ARCHITECTURE

Figure 1 shows the implementation architecture we have used for this adaptation. The filter uses a *transformation database*, a *customisation interface*, and a *filtering module* whose function are described below. It processes source HTML documents and outputs adapted HTML documents .

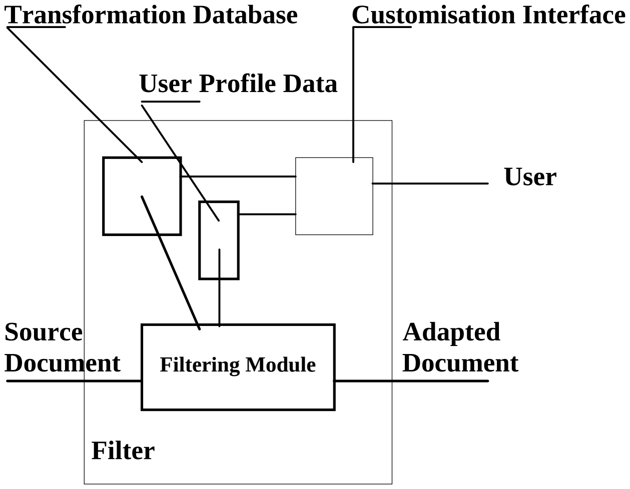

Fig 1 : Filter Architecture

Transformation Database

HTML elements are enclosed in delimiters or tags that clearly indicate the nature of the element. It is therefore easy to trigger an transformation function each time a tag is encountered and to stop it at the end of the element. The process can be applied several times for each transformation. A database is built that defines the relationships between the various HTML tags and the transformation functions (Methods) (Fig. 2).

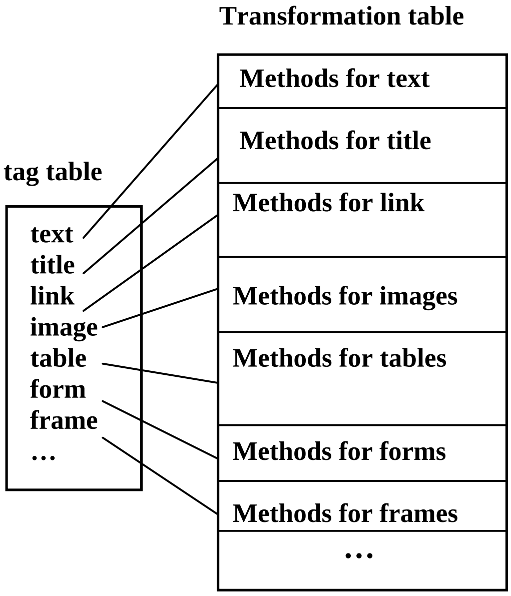

Fig 2 : Transformation Database

Customising Interface

The document is adapted using the transformations appropriate to each type of HTML element. This is done by a customisation module whose user-interface allows or forbids certain combinations of functions, depending on the hardware/ software used.

User Profile Data

The customisation parameters are saved in a data structure and used during filtering.

Filtering Module

The filtering module uses data provided by the database and the customisation module to process the HTML documents. Each elementary transformation can be broken down in three steps: *identification,* in which the HTML tag is identified, *transformation* using one or more adaptation functions, and *replacement* in which the adapted sequence is inserted into the source document in place of the original one.

FILTER INTEGRATION

The filter must be integrated into a real environment where it will act as a specialised agent in co-operation with other components. Figure 3 shows the most general integration scheme where the filter is connected to a document server and an application. Documents from the server are delivered to the application through the filter. The user sends a document request via the interface. The application connects to the server. The server sends the HTML document to the filter, which adapts it and sends it to the application, the user can " read " the document via the user interface.

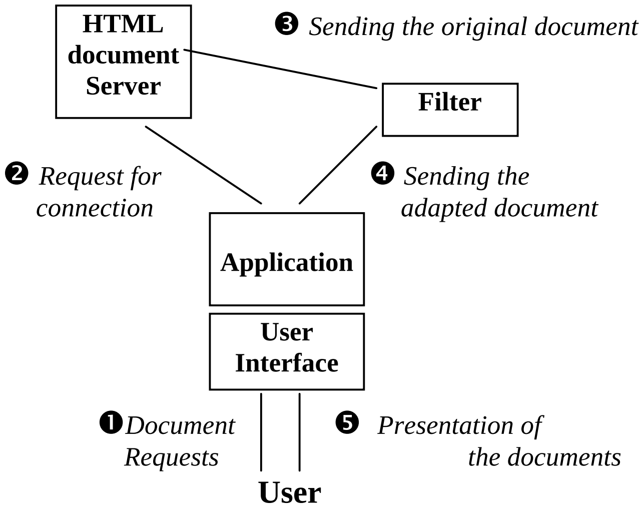

Fig 3 : General Integration scheme

The *HTML Document server* could be an Internet Web server to which requests are sent using the HTTP protocol, a set of HTML files on a hard disk or a CD-ROM, or, lastly, a HTML page generator associated with a search engine on the Web.

The *Application* could be a specific Web browser that provides an optimised non-visual interface for the handicapped users using specific devices, such as Braille terminals and/or speech synthesisers [1][2][3], or a standard Web browser such as Internet Explorer or Netscape used with an access product acting as an interface adapter.

Using high-level components

Figure 4 shows a possible way of integrating the application into Microsoft ActiveX technology [6]. It also illustrates how developers tend to design software using sophisticated components whose functions are defined at a very high level. Powerful components can be controlled through rather simple software interfaces in this software development environment. For instance, Web Browser Objects can be used to develop an application. It is even possible to co-operate with a complete application such as Microsoft Internet Explorer, since any ActiveX application is provided with a standard software interface based on OLE/COM techniques [9]. In Figure 4, the software interfaces have been developed to make the filter communicate with the server and the application, and the HTML application communicate with the user interface.

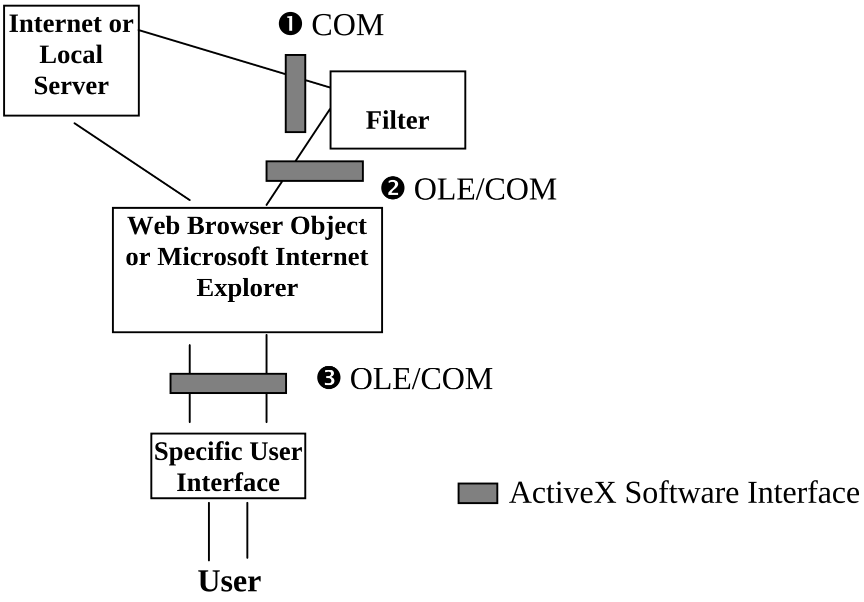

Fig 4 : Integration scheme into Microsoft ActiveX

DISCUSSION

The architecture described here is the basis for developments that have been undertaken at INSERM as a part of BrailleNet [2]. A beta version of a Web browser has been released which combines visual, auditory and tactile communication modes. As the targets of this project are visually handicapped students from primary schools to universities the interface must be very user-friendly and functional. The proposed approach provides a suitable framework for the development of Internet access products whose main features are full compatibility with current and previous HTML versions, and easy updating according as HTML develops by adding new data to the transformation database. The filter is also transparent to the user, who can be unaware that a filter is active.

This approach clearly separates the development work on the adaptation from the work on presentation and browsing [8]. Most browsers that have been developed for the visually handicapped solve the adaptation problem by transforming the HTML source data and adding also specific navigation functions. LYNX, for instance, rephrases and clarifies the links so that they are easily perceived in Braille or speech. It provides functions to go directly from one link to the next or to the previous link, which is a non-standard function in a normal browser. A user using LYNX together with a proxy server having its own method for

adapting links can cause certain problems, since some effects can be duplicated, or the final adaptation is unclear.
Finally, it is to be hoped that standard browsers will become accessible through access products in the near future. This will mean that specific functions are no longer available in the browsing product, and that adaptation filters will complement the standard Internet browsers. We therefore believe that the HTML adaptation problem will have to be completely solved within a well identified software component that other application can co-operate with. Recent trends in software development are encouraging in that direction. The use of high-level components alleviates much of the development work (for instance, developers do not have to take into consideration the problems linked to Internet connection and dialogue, or the basic functionality of applications). Therefore efforts can be concentrated on function and the user interface itself, to the benefits of the end user.

CONCLUSION

The applications of the Internet are so numerous and useful that it should be accessible to all users. Although significant progress is being made almost every day towards making the Web accessible to people who can barely read a video display, the existing access solutions still need improvement to make the Internet services as useful to these users as they are to sighted people. Since Web designers cannot be expected to take into account the special needs of very special groups of users, methods have to be devised to make adaptation as automatic as possible. This paper describes an approach which should prove a useful contribution to this problem, which is a fundamental part of R&D in rehabilitation technology [10].

AKNOWLEDGEMENTS

This R&D project is supported by a grant of the *Fédération des Aveugles de France*.
We express our gratitude to Owen Parkes for editorial assistance.

REFERENCE

[1] Access to information for the print disabled the CAPS project, *http://www.closingthegap.com/library/April-May-95/caps.html*
[2] BrailleNet, *http://www.ccr.jussieu.fr/braillenet/*
[3] Burger D., Improved Access for the visually Handicapped : New Prospects and principals, In: *IEEE transactions on rehab Engin., 1994,* pp. 111-118
[4] Guide to Writing Accessible HTML: Accessibility for Persons with Visual Disabilities, *http://www.utoronto.ca/atrct/rd/html/htmlvis.html*
[5] Lynx, *http://lynx.browser.org*
[6] Microsoft, *ActiveX Accessibility Conference,* Birmingham-UK, 1996
[7] Schwartre J., *HTML,* Data Becker GMBH & Co KG 1996
[8] Stephanidis C., Access to Graphical User Interfaces by blind people, *Concerted Action on Technology and Blindness,* May 1991, p.1-65
[9] Tall E , Ginsburg M, *late Night ActiveX,* Ziff Davis Press, 1996.
[10] The Web and Disabled People, *http://www.w3.org/pub/WWW/Disabilities/*
[11] Unified Web Site Accessibility Guidelines, *http://www.trace.wisc.edu/text/quidelns/htmlgide/htmlfull.html*
[12] WAB: W3-Access for Blind And Visually Impaired, *http://inf.ethz.ch/publications/ea.html*