Apéndice A. Reporte técnico de las herramientas analizadas para su uso en el proyecto.

Technical Report

An Overview of HTML-to-XML/XHTML/WML/VoiceXML Converters

Marisol González Rojas

Universidad de las Américas Puebla – México

is110688@mail.udlap.mx

Abstract

Continous access to information has become a necessity. The Internet has become a main source of information. Users are increasingly accessing the Web from information devices such as PDAs, cell phones, pagers. Since these devices do not have the same rendering capabilities as desktop computers, it is necessary for Web content to be adapted, or *transcoded*, for proper presentation on a variety of client devices. This report analyzes existing transcoders from HTML to other markup languages such as XML, XHTML, WML and VoiceXML, in order to display the original HTML document on PDAs, finding the most convenient transcoder of each language on the basis of advantages of portability and reusability. The goal is to integrate these tools into a graphic conversion environment of the mentioned target markup languages.

Contents

1. Introduction

The World Wide Web provides an enormous amount of data in heterogeneous formats. Beginning with the widely used markup language: HTML, we have different types of documents that need to manage a standard language that will permit the display of this information on portable devices -which have their own language- with the purpose of accessing the information anytime, anywhere.

In order to satisfy the massive predicted growth in mobile devices, such as PDAs and cell phones, companies will use conversion tools to migrate from HTML to the languages handled by devices such as PDAs (XHTML) and cell phones (WML, VoiceXML), without losing accuracy in their services.

But why conversion tools rather than building new sites from the ground up?

The major advantages of conversion are:

- Speed to market and cost: the conversion of existing sites or information takes less programmer time than starting from scratch. Another advantage is that the user of the converting site does not have to be a code expert.
- Also, the content extracted from the original page can be held in a format-independent manner, so formats as WML, XML, XHTML and VoiceXML, can be applied to serve other client types. A converter can be written that delivers content to PCs, WAP phones, PDAs, Internet screen-phones and any other Internet device that appears along the way [Arehart 2001].

2. What are converters?

[Arehart 2001] In simple terms, converters work by extracting text from a source page, in our case an HTML page, and then reformatting that text into the target markup language, in this case, XML, XHTML, WML and Voice XML.

[Arehart 2001] The converter performs the conversion of formatted data to pure data, so we, as the conversion author, decide the format we wish to apply in order to display it on mobile devices.

There are two possible ways of conversion:

- Extract all possible content in the page, such as title, links, and so on. (Fully automated conversion).
- Extract specific parts of the page which are specified by the user. (Configurable conversion).

Content converters are also known as “transcoders”. In other words, transcoding is a method for translating one type of code into a different type. Transcoders exist to convert HTML to VoiceXML, to convert HTML to a format more suitable for display on a PDA and to convert from a UIML document to multiple language documents.

3. Content suitable for conversion

In general, the main features that make existing web sites a suitable target for conversion to another language is that they provide small amounts of timely text-based information [Arehart 2000]. Next, we need our source (HTML) as well-formed as possible, in order to take over the conversion, due to the fact that many browsers

are very lax in how they interpret HTML. This leads to incongruities in how the pages are displayed. So, the best way to correct this is to use an HTML validator.

Focusing on this we will first consider that the document that will be converted is well-formed, after validating it with HTML Tidy, a free open source tool that cleans our HTML documents.

3.2 HTML Tidy: Cleaning up HTML

HTML Tidy is a free utility, created by Dave Raggett that works on markup generated by specialized HTML editors and conversion tools. Tidy is able to fix up a wide range of problems and to bring to the user´s attention things that he needs to work on. Each item found is listed with the line number and column so that the user can see where the problem lies in the markup. [Raggett 2003].

The source code continues to be available under an open source license, which is a bonus on using this tool in order to cleanup the documents before the conversion.

Tidy clean ups

- Detects and corrects missing or mismatched end tags
- Corrects end tags in the wrong order
- Fixes problems with heading emphasis
- Recovers from mixed up tags
- Gets the <hr> in the right place
- Adds the missing "/" at end tags for anchors
- Perfects lists by putting in tags which were missed out
- Adds missing quotes around attribute values
- Reports unknown/proprietary attributes
- Spots tags lacking a terminating '>'

Also, many tools generate HTML with an excess of FONT,
 and CENTER tags. Tidy's *-clean* option will replace them with style properties and rules using CSS. This makes the markup easier to read and maintain as well as reducing the file size.

One of the most helpful advantages of using Tidy is that you can teach Tidy about new tags by declaring them in the configuration file. An additional feature is that Tidy has a library version, TidyLib, allowing the use of HTML Tidy as a callable library, having two versions: a library written in C and in JAVA. We either can integrate the tool to our software or use the API to mediate the conversion.

4. Objective languages

Given the importance of converting HTML documents into XML, XHTML, WML and Voice XML, so they can be displayed on mobile devices, a quick overview of these languages is given.

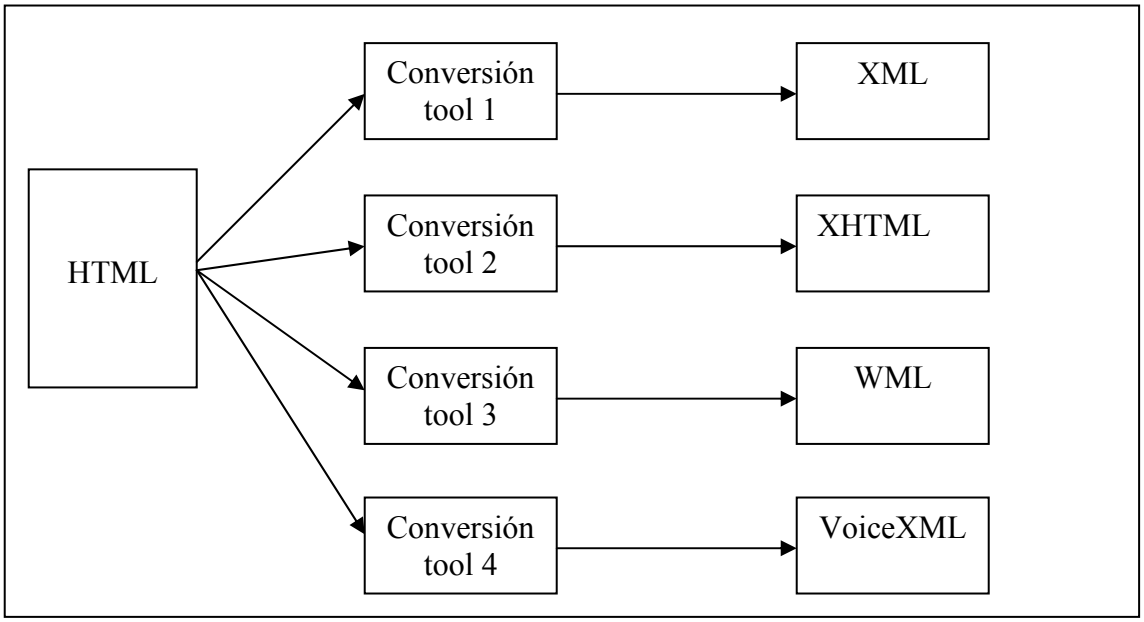

Figure 1. Diagram of the conversion tools integrated.

4.1 XML

[W3C 2000]

4.1.1 Characteristics

- XML stands for E**X**tensible **M**arkup **L**anguage.
- XML is a **markup language** with great similarity with HTML.
- XML was designed to **describe data.**
- XML tags are not predefined. The programmer must **define his own tags,** this is why XML is extensible.
- XML uses a **Document Type Definition** (DTD) or an **XML Schema** to describe the data through a sequence of rules.
- XML with a DTD or XML Schema is designed to be **self-descriptive**.

- XML is a markup language where everything has to be marked up correctly, which results in "well-formed" documents.

4.1.2 Differences between XML and HTML

- XML was designed to describe data and HTML was designed to display data.

- XML does not replace HTML.
- XML and HTML were designed with different goals: XML was designed to describe data and to focus on what data is.
- HTML was designed to display data and to focus on how data looks. HTML is about displaying information, while XML is about describing information.

4.2 XHTML

[W3C 2004]

4.1.1 Characteristics

- XHTML stands for EXtensible HyperText Markup Language
- XHTML is aimed to replace HTML
- XHTML is almost identical to HTML 4.01
- XHTML is a stricter and cleaner version of HTML
- XHTML is HTML defined as an XML application

Therefore, by combining HTML and XML, and their strengths, we got a markup language that is useful now and in the future - XHTML.

XHTML pages can be read by all XML enabled devices AND while waiting for the rest of the world to upgrade to XML supported browsers, XHTML gives the opportunity to write "well-formed" documents now, that work in all browsers and that are backward browser compatible.

4.2.2 Differences between XHTML and HTML

- XHTML elements must be properly nested
- XHTML documents must be well-formed
- Tag names must be in lowercase
- All XHTML elements must be closed

4.3 WAP / WML

[W3C 2004]**4.3.1 WAP**

The WAP protocol is the leading standard for information services on wireless terminals such as digital mobile phones. WML is the language used to create pages to be displayed in a WAP browser. The WAP standard is based on Internet standards (HTML, XML and TCP/IP). It consists of a WML language specification, a WMLScript specification, and a Wireless Telephony Application Interface (WTAI) specification.

- WAP stands for Wireless Application Protocol
- WAP is an application communication protocol
- WAP is used to access services and information
- WAP is inherited from Internet standards
- WAP is for handheld devices such as mobile phones
- WAP is a protocol designed for micro browsers
- WAP enables the creation of web applications for mobile devices.
- WAP uses the mark-up language WML (not HTML)
- WML is defined as an XML 1.0 application

4.3.2 WML

WML stands for **W**ireless **M**arkup **L**anguage. It is a mark-up language inherited from HTML, but WML is based on XML, so it is much stricter than HTML.

WML is used to create pages that can be displayed in a WAP browser. Pages in WML are called DECKS. Decks are constructed as a set of CARDS.

4. 4 Voice XML

[Annamalai 2002] VoiceXML is a derivative of the W3C XML. While XML has become the default standard for representing data and structures on the web, VoiceXML has become the default standard for describing voice applications. VoiceXML is a language for creating Human – Computer interfaces trough telephone. VoiceXML can be thought of as a markup language for voice, like HTML is for text. VoiceXML is used extensively for speech recognition and application development. Two of the main goals of VoiceXML are: 1) to deliver the web content to interactive mobile clients through voice [Annamalai 2202] and 2) make web accessible to people with visual impairments [Pérez-Quiñones 2002].

4.4.1 Characteristics

Just as a user can interact with a HTML page, he/she can also interact with a VoiceXML page.

- VoiceXML documents contain only forms and blocks and each has to be delivered to the user through audio in sequential manner.
- User interaction is provided through forms.

- Output of synthesized speech (text-to-speech).
- Output of audio files.
- Recognition of spoken input.
- Recording of spoken input.
- Telephony features such as call transfer and disconnect.
- The language provides means for collecting character and/or spoken input, assigning the input to document-defined request variables, and making decisions that affect the interpretation of documents written in the language. A document may be linked to other documents through Universal Resource Identifiers (URIs) [VoiceXML Forum 2000]..

5. HTML to XML/XHTML

For some time, the World Wide Web Consortium (W3C) has been looking at the issues involved in providing content to different types of clients, without the need to create many different copies of each page. Under the banner of the Mobile Access Group, many of the new standards, proposals and working drafts – such as eXtensible Markup Language (XML)- which allows the separation of content and presentation - stylesheets and the Resource Description FrameWork (RDF) – are coming together to provide a coherent platform that will support multiple disparate types of clients. [Arehart 2000].

The first step in the process of converting HTML pages, after well-forming the HTML (see Point 3), is to clean them up, so that an XSLT (XSL transformation) or a DOM or a SAX parser can work with the documents. After searching on the web for free conversion tools that could be reused, or more specifically, that could be integrated into a conversion system, a very useful tool was found which is **HTMLTidy.** HTMLTidy not also converts the untidy HTML to well-formed documents but also to XHTML and XML.

HTML Tidy started as a command-line tool, but GUI (Graphic User Interface) versions are available for Windows and MacOS, as well as an API in C and in JAVA, the callable library so it can be integrated to any software.

Because of the grade of difficulty to integrate efficiently the tool as a command-line, the best option is to use the API.

5.1 HTML to XML with HTMLTidy

There are two classes that permit the conversion (Tidy and Configuration) easily with the use of their respective method:

After using Tidy, below, we can appreciate the document source and the resulting output.

Listing 1. index.html (an excerpt)

```
<HTML>
  <HEAD>
    <TITLE>Journey to Windsor</TITLE>
  </HEAD>
<BODY>
<TABLE>
  <TR>
    <TD width=15></TD>
```

```
    <TD><FONT size="3"face="Helvetica">
      Journey to Windsor<BR>
      Beno&icirc;t Marchal<BR>
      July 2003<BR>
      <BR>
      <A href="mailto:bmarchal@pineapplesoft.com">
         bmarchal@pineapplesoft.com</A>
    </FONT></TD>
  </TR>
</TABLE>
<CENTER><TABLE border=3>
  <TR><TD>
    <A href="pages/dscn0824.html">
      <IMG src="thumbnails/dscn0824.jpg" border="0" alt="dscn0824">
    </A><br>
    <FONT size="3" face="Helvetica">
    dscn0824.jpg<br>
    A bright, red mailbox inside the castle. It seems oddly familiar
    in an historic setting.<br>
    Windsor Castle <br>
    &copy; 2003, Beno&icirc;t Marchal
    </FONT>
 </TD></TR>
</TABLE></CENTER>
</BODY>
</HTML>
```

And the consequent output, using the HTMLTidy tool. The changes made by Tidy are in bold letters:

Listing 2. index.xml (an excerpt)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta name="generator" content=
"HTML Tidy for Mac OS X (vers 1st June 2003), see www.w3.org" />
<title>Journey to Windsor</title>
</head>
<body>
<table>
<tr>
<td width="15"></td>
<td><font size="3" face="Helvetica">Journey to Windsor<br />
Beno&#238;t Marchal<br />
July 2003<br />
<br />
<a href=
"mailto:bmarchal@pineapplesoft.com">bmarchal@pineapplesoft.com</a></font></td>
</tr>
</table>
<center>
<table border="3">
<tr>
<td><a href="pages/dscn0824.html"><img src=
"thumbnails/dscn0824.jpg" border="0" alt="dscn0824" /></a> <br />
<font size="3" face="Helvetica">dscn0824.jpg<br />
A bright, red mailbox inside the castle. It seems oddly familiar in
an historic setting.<br />
Windsor Castle <br />
&#169; 2003, Beno&#238;t Marchal</font></td>
</tr>
</table>
</center>
</body>
</html>
```

6. HTML to XHTML with HTMLTidy

The difference between XHTML and HTML might sound trivial (it's only an extra "X" after all) but it is important. XHTML is a version of HTML 4.01 that has been adapted to the XML syntax. The vocabulary is unchanged (XHTML uses the familiar <p>, <b>, and <a> tags, for example), but the syntax is XML, so it merges nicely in an XML workflow [Marchal 2002].

7. HTML to WML

HTML2WML, the choice of converting HTML to WML is a result of the analysis of five pieces of previous work on converting HTML to WML, giving an improved implementation of these [Howard 2001]. These five approaches are described below.

1) David Kapadia, form the Kansas State University, achieved a project entitled *Conversion of given XML data to WML* [Kapadia 2000] Kapadia implements a converter using XSLT, the XML parser Xerces, the XSLT processor Xalan and a Java file to apply the conversion. Results seem good but the project was very limited since input was from known XML structures. The software would have to be rewritten in order to deal with different DTDs or new tags and application to HTML was never considered.
2) Maddingue [Aperghis 2002] published HTML2WML Version 0.4.1, a program registered under GNU License, which is a CGI/Perl on-the-fly HTML to WML conversion tool. The problems encountered were that the input to the program must be valid well-formed HTML and the output is far from valid WML, making it incapable of being rendered on a WML browser. In addition, it does not support frames.
3) The University of Durham describes three approaches of processing XML techniques, but not specifically the conversion to WML.
4) LazyWap v.0.5 is a freeware PHP HTML to WML converter written by a Russian Internet Consultant. It functions as Kapadia´s Java Program and Aperghis CGI/Perl utility but again doesn´t have routines for handling anything than very simple input.
5) The Finland Research Centre. Proposed methods of handling frames and complex HTML conversion but finally their work pointed to the difficulty of converting malformed HTML and the restrictions were still on this software.
6) The Wireless Developer Network published an article which explains how translating XML to WML can be done using XSLT transformations but again restricted to a known type of XML.

After the analysis, HTML2WML´s final implementation was:

HTML → Tidy.exe = xHTML → Saxon.exe + XSLT = WML

As I discussed before, in order to convert an HTML document, it first has to be well-formed HTML, reason of using Tidy to clean it up and transforming it to XHTML. The Saxon utility is a parser, which will convert the XML document into a tree representation, the structure of which is manipulated by XSLT.
Using the XSLT language the design would then follow a very simple structure of pattern matching rules called templates. This also means support for extra modules being added and associated with the other templates that a developer would wish, this is it makes the utility reusable.
The results produced valid well-formed WML and with a success rate of at least 70% the software fulfilled its objectives in handling many of the complex constructs found in HTML. Most importantly these have included routines for handling hyperlinks and presenting framed pages [Howard 2001]. All this with the advantage of being free open source software.

8. HTML to VoiceXML

VoiceXML and the proliferation of voice portal services such as TellMe and BeVocal have enabled mobile access to information through telephone-based voice interfaces. However, most voice portals available today use carefully crafted interfaces designed by expert human professionals to provide access to a select set of information such as top news stories, weather, sports information and stock quotes. [Shao 2003].
Providing voice access to a wide array of information on the web can be a difficult problem. The transcoding of HTML to VoiceXML is a significant challenge. The problem is difficult because a lot of the information stored on the web has been specifically engineered for use by graphical web browsers. The nature of voice interfaces requires different presentation strategies than the high-bandwidth, parallel nature of graphical interfaces. Also, the semantic information that is implicitly encoded in the page contents can be difficult to obtain. An extended approach is to use annotations to help the transcoding process.
Annotations can be used to overcome some of the problems with automatic transcoding. They can be thought of as hints that are added to the code to be translated that provide information about how to transcode particular sections of the code [Shao 2003]. An advantage of well-designed annotations is that they can be added manually and automatically and allow designers to examine how a particular section of code will be translated. Researches at the IBM Tokyo Research Laboratory have used annotations and transcoding to make HTML documents suitable for presentation on small-screen devices.
The most recent researches related to VoiceXML are described below:

1) The IBM WebSphere Transcoding Publisher (WTP) [Lamb 2000] is a commercial product that supports an HTML-to-VoiceXML transcoder. Transcoders can be plugged into a WTP server that can be configured as a proxy. Client-side browsers can use this proxy to obtain transcoded content. This transcoder has a simple translation strategy for converting an HTML-to-VoiceXML: it splits the page into two sections: a main content section and a listing of all the links on the page. HTML heading tags are used to split the main content into subsections and facilitate voice navigation. Links on a page are gathered together in a separate section and can be selected by voice. This approach works for simple web pages with clearly structured heading tags and for documents that the text between the <a> tags is

context independent, but can result in a voice interface with low usability [Perez-Quiñones 2003], because most web pages do not have these characteristics.

2) Another approach that solves the problems of automatic transcoding is what has already been mentioned above: annotations. Hori, Kondoh, Ono, Hirose, and Singhal used annotations and transcoding to make HTML documents suitable for presentation on small-screen devices [Hori 2000]. Hori used annotations to highlight sections of pages that are of interest for later processing. Shao, Capra and Perez-Quiñones explored the types of structures available on web pages, then defined a desired voice interaction style for each structure. Shao et al. defined a set of XML-based tags that are used in a transcoding architecture to generate a VoiceXML document that produces the desired behaviour and at last also defined external annotation files for existing HTML files to make these pages available over a phone-based voice user interface.

3) The third approach is the transcoder implemented by Narayan which is divided in two phases. First phase corresponds to the parsing of the HTML, which takes as input an HTML file and produces an HTML node object which is organized in a tree structure. The second phase is where the HTML node tree is converted to a well-formed VoiceXML file [Annamalai 2002].

4) The Aurora transcoding system adapts web pages based on semantic information and builds an XML document with extracted information. The semantic information used for the transcoding though, must be produced manually.

5) OpenVXI is a portable open source library that interprets the VoiceXML dialog markup language. OpenVXI is only a component of a complete VoiceXML platform. Including simulated speech recognition, prompt and text-to.speech capabilitites, and telephony functions, with users responsible for providing integration to actual components. Nonetheless, OpenVXI provides a strong base that is far preferable to starting from scratch [SpeechWorks 2001].

VoiceXML is still a challenge while converting HTML to it, and the research made up to now, does not yet propose an open source transcoder. The work related above mentioned is either still in research, or has a high cost as a commercial product, so I decided to try applying OpenVXI to XSLT (stylesheets), which is proved to be quite powerful with respect to functionality in order to meet the goals of the thesis: the implementation of a transcoder of HTML to XML, XHTML, WML and VoiceXML. It could be a temporary solution due to the new languages that arrive everyday, but for now, seem an option to try.

9. Conclusions

Technologies seem to facilitate our daily working, and as the main goal of this research, is to implement an open source converter in order to transcode HTML to XML, XHTML, WML and VoiceXML. The tools chosen are all XML based and compatible with Java programming, the other language chosen to the purpose of this research.

10. References

Annamalai N. 2002. *An Extensible Transcoder for HTML to VoiceXML Conversion.* Thesis of Master Science in Computer Science. University of Texas at Dallas, 6-16. Available at: http://www.cs.utdallas.edu/~gupta/narayanthesis.pdf

Apperghis S. 2002. *HTML2WML software*. Article about the HTML conversion into WML format.
Available at: http://maddingue.free.fr/softwares/html2wml/

Arehart, C., Chidambaram, N., Guruprasad, S., Homer, A., Howell, R., Kasipillai, S, Machin, R., Myers, T., Nakhimovsky, A., Passani, L., Pedley, C., Taylor, R., Toschi, M. 2001. *Professional Wap*. Wrox, Birmingham, UK.Baikov, M. 2000. *LazyWAP-HTML-to-WAP converter*, 426-436.

SpeechWorks 2001. ScanSofts´s OpenVXI 3.0. Article about the OpenVXI project. Available at: http://fife.speech.cs.cmu.edu/openvxi/

Chase, N. 2002. *The Web´s future: XHTML 2.0.* IBM- DeveloperWorks. Article about the XHTML language, user´s manual.
Available at: http://www-106.ibm.com/developerworks/web/library/wa-xhtml/

Cornelius, B. 2001. *Processing XML*. University of Durham. Article about XML applications.
Available at: http://www.dur.ac.uk/barry.cornelius/Java/xml.processing/onefile/

Huang, A and Sundaresan, N. 2000. *Aurora: A Concept Model for Web-Content Adaptation to Support the Universal Usability of Web-based Services*. ACM Proceedings on the 2000 conference on Universal Usability. Available at: http://portal.acm.org/citation.cfm?id=355546&coll=portal&dl=ACM&CFID=20897396&CFTOKEN=33240231

Howard, P. 2001. *Converting HTML to WML*. User´s manual.
Available at: http://www.topxml.com/wap/articles/htmlwml/default.asp

Hori, M. 2000. *Annotation-Based Web Content Transcoding*. IBM Tokyo Research Laboratory. Paper that references a thesis project. Available at: http://homepages.cwi.nl/~lynda/www9/www9-ws-hori.pdf

Kaasinen, E., Aaltonen, M., Kolari, J., Melakoski, S., Laako, T. Two approaches to Bringing Internet Services to WAP devices. VTT Information Technology.
Available at: http://www9.org/w9cdrom/228/228.html

Kapadia, D. 2000. *A report on WAP/WML*. Paper about WML conversion. Available at: http://www.cis.ksu.edu/~deep/690/

Lamb, M. Lessons and Horowitz, B. *Guidelines for a VoiceXML Solution Using WebSphere Transcoding Publisher*. Paper that states guidelines when converting VoiceXML. 2000. Available at: http://www-3.ibm.com/software/webservers/transcoding/library.html

Lee, Wei. 2000. *Transforming XML to WML*. Article about XML conversion to WML language. Available at: http://www.wirelessdevnet.com/channels/wap/training/xslt_wml.html

Marchal, B.. 2003. *Tip: Convert from HTML to XML with HTML Tidy*. IBM- DeveloperWorks.
User´s manual. Available at: http://www-106.ibm.com/developerworks/xml/library/x-tiptidy.html

Perez-Quiñones, M.. 2002. *Voice User Interfaces for the Web*. Department of Computer Science, Virginia Tech Paper that states the use of voice interfaces.. URL: http://perez.cs.vt.edu/cs5984/transparencies/Day%2011-06-11.pdf

Raggett, David. 2003. *Clean up your Web pages with HTML TIDY*. Users manual and software download. Available at: http://clk.about.com/?zi=1/XJ&sdn=webdesign&zu=http%3A%2F%2Fwww.w3.org%2FPeople%2FRaggett%2Ftidy%2F

Shao, Z., Capra, R., Perez-Quiñones, M. 2003. *Transcoding HTML to VoiceXML Using Annotation*, Proceedings ofICTAI 2003.