ENABLING VOCAL INTERACTION IN A WEB PORTAL ENVIRONMENT

Federico Bergenti, Lorenzo Lazzari, Marco Mari, Agostino Poggi
Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Parma, Parco Area delle Scienze 181/A, 43100, Parma, Italia

Keywords: Web portals, Vocal interaction, Human-computer interaction.

Abstract: The growing request of innovative, multimodal interfaces for mobile users and the need of different navigation paradigms for impaired Internet users are promoting and driving nowadays research on multimodal interactions. In this paper we present our experiences in the integration of vocal components in a portal engine, namely Apache Jetspeed. First, we discuss the reasons why the integration of a full-featured portal brings significant advantages to the development of a vocal platform. Then, we describe two complementary approaches for enhancing portals with vocal capabilities, and we compare them under various standpoints. The first approach that we investigate is based on a server-side speech synthesizer, while the second relies on client-side technologies, X+V and the SALT markup languages. For each of these approaches we present some examples and we discuss advantages and drawbacks.

1 INTRODUCTION

In last few years the development of the Internet has brought a great amount of information and services to an enormous number of users. At the same time, mobile devices and wireless networks have significantly widened their capabilities from simple voice calls to the hybrid, converged services and fast navigation of the Internet that third generation UMTS devices support. The necessary convergence of the technologies for wireline and wireless networks has provided users with the capability of accessing information and services from anywhere, at anytime. In this scenario, the differences between input/output devices of desktop computers, mobile phones and PDAs are pushing researchers to define new interaction methods, beyond traditional interaction with keyboard and mouse, capable of overcoming such differences. An important example of these efforts is the formation of the W3C Multimodal Interaction Activity, with the aim of extending the user interface of the Web to allow multiple modes of interaction, e.g. vocal, visual and tactile. Moreover, the research in new interaction methods raises to a higher level of importance in consideration of the utility of such methods for the navigation of impaired Internet users, thus increasing the accessibility of the Web to people with visual or other disabilities.

Taking into account the requirements of mobile navigation and accessibility of multiple, personalized contents, portal technologies occupy a key role in future development of ubiquitous and integrated applications. A portal is a central hub that can simplify and personalize access to heterogeneous services, applications and information. These characteristics turn a portal into an ideal gateway to the Internet for impaired navigators. Moreover, everyday portal engines facilitate the access to their contents from mobile devices, and they also provide transparent adaptation of such contents to the characteristics of end users' devices, as we will see in the following section.

In this paper we present our experiences in the direction of enhancing the interaction methods offered by a portal system, namely Apache Jetspeed, in order to provide users with a vocal interface. In Section 2 we discuss some related work, while in Section 3 we summarize the key features of the portal engine that we used and we explain why a portal engine can facilitate the development of a vocal or multimodal platform. In sections 4 and 5 we describe in detail the two complementary approaches that we followed to integrate voice capabilities into the portal. Finally, section 6 is dedicated to a comparative discussion of these two approaches and to an outline of our future work.

Bergenti F., Lazzari L., Mari M. and Poggi A. (2006).
ENABLING VOCAL INTERACTION IN A WEB PORTAL ENVIRONMENT.
In *Proceedings of WEBIST 2006 - Second International Conference on Web Information Systems and Technologies - Internet Technology / Web Interface and Applications*, pages 318-324
DOI: 10.5220/0001239203180324

2 EXISTING STANDARDS AND RELATED WORK

2.1 EMMA, X+V and SALT

The starting point for investigating the actual status of multimodal technologies is without any doubt the W3C Multimodal Interaction Activity home page. The main objective of this workgroup is to extend the Web to allow users to dynamically select the most appropriate mode of interaction on the basis of their current needs, while enabling developers to provide an effective user interface for whichever modes the user selects. The modes that the W3C Multimodal Interaction Activity considers for input are speech, handwriting and keystrokes, while output can be presented via displays, pre-recorded and synthetic speech, audio, and tactile mechanisms, e.g., mobile phone vibrators and Braille strips. An important result of the W3C Multimodal Interaction Activity workgroup is the recent definition of EMMA (Extensible MultiModal Annotation Markup Language), a language used to represent human input to a multimodal application. An example of interpretation of a user input that EMMA facilitates is the transcription into words of a raw signal, for instance derived from speech, pen or keystroke input.

The specification of EMMA is so recent that it is not yet possible to find tools supporting this new language. However, waiting for EMMA-compliant tools, we can still test multimodal (in particular vocal) applications using two standards that have provided a strong contribution to EMMA: X+V and SALT.

X+V (XHTML + Voice) has the aim to integrate two mature standards, i.e., XHTML and VoiceXML, to bring spoken interaction to the Web, i.e., creating multimodal dialogs that combine the visual input mode and speech input and output. X+V is sustained by a consortium comprising IBM, Motorola and Opera Software.

The SALT (Speech Application Language Tags) Forum groups different companies, which includes Microsoft, Intel and Cisco, sharing a common interest in developing and promoting speech technologies for multimodal and telephony applications. SALT specification extends existing mark-up languages, i.e., HTML, XHTML and XML, enabling multimodal and telephony access to the Web.

2.2 Related Work

Multimodal and vocal interaction means are a longstanding research problem. Already in 1992, Cohen (Cohen, 1992) suggested that the focus should not only be on building interfaces that make available two or more communication modalities, rather on developing integrated interfaces in which the modalities forge a productive synthesis, using the strengths of one modality to overcome weaknesses of another.

It is possible to distinguish three kinds of reason (Sharma et al., 1998) for using a multimodal interface in human computer interaction (HCI):

– Practical: traditional HCI systems are unnatural and cumbersome. Moreover, redundant or alternative input sources can help impaired users to access computer applications.

– Biological: human beings, as well as other animals, integrate multiple senses. This strongly suggests that the use of multimodality in HCI would be desirable, especially if the goal is to incorporate the naturalness of human communication in HCI.

– Mathematical: it is statistically advantageous to combine multiple observations from the same source because improved estimates are obtained using redundant observations. In this way, the concurrent use of two or more interaction modalities may improve system reliability.

Multimodality can be defined from human and technology perspectives (Baber et al., 2001). From the human point of view, people can exchange information using different sensory modalities, and so a multimodal system would support more than one sensory and response modality. From the technology point of view, computer systems can receive and present information using different modes, and so multimodality is the capability of a system to allow combination of modes to operate concurrently. Given these definitions, we can point out one of the main problems of multimodality: the design of systems capable of fully supporting concurrent, alternative modes.

Several research works aim to define multimodal interfaces combining different interaction modes. Moran presents an agent-based interface that focuses on voice and pen input, and supported by a gestures-recognition engine (Moran et al., 1997). Again, the SmartKom system (Wahlster et al., 2001) merges three paradigms: spoken dialogue, graphical user interfaces and gestural interaction. For the two latter paradigms, SmartKom does not use a traditional WIMP (Windows, Icons, Menus, Pointer) interface combined together with gesture recognition; instead, it supports natural gestural interaction combined with facial expressions. Another valuable

characteristic of SmartKom is that it spans across a number of different platforms and application scenarios: the kernel functionalities are portable to a wide range of devices. MOUE (Lisetti et al., 2003) is a system whose main characteristic is the ability to build a model of user's emotions by observing the user via multi-sensory devices, i.e., camera, microphone and wearable computer. This system can have a wide range of applications, like tele-monitoring and healthcare. MEDITOR system (Bellik, 1997) applies multimodal interfaces, including speech recognition, synthesis, and Braille terminals, to provide improved computer access to blinds.

Thus some portal solutions (e.g.: IBM WebSphere Portal) offer support for multimodal applications, and despite the amount of research in this field, to the best of our knowledge this is the first paper to discuss the integration of a vocal mode in a portal engine (both for the interface and the contents). The benefits of such a solution will be described in detail in the next section.

3 PORTAL ENGINES

In our vision, the research in innovative and multimodal interaction paradigms has two main goals: *(i)* to facilitate the navigation for impaired users; and *(ii)* to improve the usability of ubiquitous applications. In both cases, the integration of a portal engine can bring significant advantages. In fact, a portal acts as a common layer on which it is possible to integrate heterogeneous applications and contents in a personalized way, without exploring several, different sites, and thus facilitating the navigation for both mobile and impaired users. Moreover, the adaptation of portal pages to the characteristics of mobile devices is easier than the adaptation of generic Web pages because a portal engine provides natively a set of tools for performing such adaptation almost transparently. This last argument has already been discussed in a previous paper (Mari et al., 2004), so in this section we only summarize the reasons why a portal is an ideal platform for providing contents to mobile devices:

- Content adaptation applies to single portlets, not to entire portal pages. Portlets are the base components of a portal, the elements that process a request and generate dynamic contents. Therefore, the content of a portlet is generally shorter and easier to adapt than the content of a whole page.
- The design of the user interface (and therefore the usability of the portal) is separated from the content it presents, therefore a different interface can be specified for each class of devices.
- The disposition of the portlets on the page is separated from the content they present and a different disposition can be provided for each class of devices.
- The portal can easily be extended with the support for new classes of devices.

Therefore, a portal provides an environment in which it is possible to create and adapt separately the contents, the user interface and the content disposition.

Another relevant advantage portals provide is suggested by the works of Dybkjaer (Dybkjaer et al., 2004) and Walker (Walker et al., 2004). The conclusion section of Dybkjaer et al. discusses the benefits of a user modelling facility, in order to adapt the behaviour and the dialogues generated by the system, while in Walker et al. the generation of answers for users is fully tailored on the basis of user profiles. A portal engine always includes a user database that can contain all relevant information for this profiling. Moreover, this database can be updated on the fly, following a possible change of user preferences during system navigation (e.g., a user enters a portion of the portal in which vocal interaction is not desirable).

The portal we used in our research is Jetspeed 1.5, an open source implementation of an enterprise information portal. Jetspeed is part of Portals, a top-level project of the Apache Software Foundation (ASF). Jetspeed provides portal view customization and access to information on the basis of user capabilities: it acts as the central hub where information from multiple sources is made available in an easy to use manner. Therefore, each user can access to a subset of the information/applications available and select the part which she/he is interested in.

The off-the-shelf version of Jetspeed can only serve content to PCs and WAP devices. In a past work, we developed (Mari et al., 2004) an add-on for Jetspeed that provides the portal with the capability of supporting other mobile devices, i.e., PDAs and I-Mode terminals (I-Mode is the Internet access system offered by the Japanese operator NTT DoCoMo). This add-on is useful to test the results of our research on various platforms, both emulators and real devices. We are also developing an add-on that provides real-time user profiling, analyzing user

Figure 1: The vocal mode in the Web user interface of Jetspeed.

actions and changing the system behaviour on the basis of user preferences.

4 ENABLING A VOCAL MODE

The first approach we entailed to enhance the portal navigation experience with voice was to add a vocal mode to all portal content. In this way, each part of the portal can be read or heart by the user. For this purpose, we integrated a speech synthesizer in a servlet, then we created a tool to access the synthesizer from the portal, and finally we modified the portal structure to make the new functionality available transparently to all portlets.

Before describing in more detail this work, it is important to remember that the content of a portal is composed by portlets, and from each single portlet it is possible to access external applications, RSS feeds, Web services, corporate contents, and so on.

4.1 FreeTTS

After an accurate analysis of the available speech synthesizers, we decided to develop the speech synthesis module of our add-on to Jetspeed using FreeTTS. FreeTTS is a speech synthesis engine written entirely in Java and it is based on Flite, a small runtime speech synthesis engine developed at Carnegie Mellon University. FreeTTS has been developed by the Speech Integration Group of Sun Microsystems Laboratories and it is distributed with an open source, BSD-style licence. The main reasons that brought us to the choice of FreeTTS are:

- Support for JSAPI 1.0 specification. JSAPI are the standard Java interfaces for incorporating speech capabilities in a Java program. Moreover, FreeTTS provides its own libraries with added functionalities, e.g., the direct redirection of audio sources to a remote client.
- Being a Java program and an open source project, it is easy to adapt and to integrate in the architecture of our project. Jetspeed is structured as a servlet, and so the best solution is to also have the speech synthesizer available in the servlet itself.
- Among the suggested uses, authors included a sample remote TTS server, whose behaviour is similar to our requirements: FreeTTS can act as a back-end text-to-speech engine that works with a speech/telephony system.
- The internal speech synthesis engine has been recently enhanced to provide optimal performances.

FreeTTS has been integrated in a servlet, and all synthesizer configurations (i.e.: the voice selections) are made available through a Web interface. The integration has been tested with the Apache Tomcat servlet engine, giving good results and performances.

4.2 Portal Integration

For providing text-to-speech functionalities to Jetspeed, we developed a portal module called TextToSpeech Service. This module is designed as a pluggable service of Turbine, the framework on which Jetspeed is built. This service is a set of classes, based on the Singleton design pattern, with utility methods that can be called from every part of the portal. It is worth noting that the architecture of this service is not tied to FreeTTS, so, in the future, we could provide implementations of our service on the basis on other synthesizers.

Having the text-to-speech functionalities available, next step was to introduce the vocal mode in the portal core features. Jetspeed defines a set of standard modes for each portlet: view, customize, info, print, minimize and maximize. On the basis of the portlet type and of the administrator policies, a portlet can have one or more modes available, while the user can select in which the portlet is displayed. A new mode has been defined: the vocal mode. In the Web user interface, the vocal mode is presented as a new icon near the portlet title (see figure 1). As for the other modes, the portlet can declare in its registry if the vocal support is provided. From a functional point of view, the introduction of the vocal mode means that the user can trigger another type of action. This action was defined in the interfaces and in the abstract classes that describe the portlet state and methods. A detailed description of the modifications in Jetspeed code is beyond the scope of this paper. When the user triggers the vocal action, the portlet generates a call to the TextToSpeech service, the service in his turn activates the FreeTTS synthesizer, passing the portlet content to the speech synthesis engine.

By default, the vocal mode is activated only when the user selects the corresponding icon, but it is possible to force the activation whenever a user interacts with a vocal portlet.

4.3 Example Portlets

As an example of vocal-enabled portlet, we implemented the vocal mode for two of the most common portlet types of Jetspeed: RSS (Rich Site Summary) and Velocity.

RSS portlets are used to retrieve news from remote sources: they take an RSS feed from a remote server, apply an XSLT transformation to the feed content and present the result in the portal screen. The sample RSS portlets provided with Jetspeed are BBC Front Page news, ApacheWeek and XMLHack. When a user clicks on the vocal icon, the TextToSpeech service synthesizes the content of the news and sends the audio stream to the client device.

Apache Velocity is a simple, yet powerful, template engine used to develop Web applications following the MVC (Model-View-Controller) model. All Jetspeed screens and most portlet templates are written using Velocity. When a Velocity portlet generates its content, it looks for a template according to the media type of the user (e.g., HTML and WML). Thanks to the new vocal support, developers can also write templates for the vocal mode. The content of Velocity templates is inherently dynamic, and therefore the audio stream sent to the user is not a static description of the portlet, rather it follows the user-portal interaction. For example, the speech can present the results of a query to a database, or it can answer to the inputs of a user compiling a form.

5 A DEEPER INTERACTION USING MARKUP LANGUAGES

Although the vocal mode provided by a simple speech synthesizer like FreeTTS can significantly improve the navigation experience of the user, a more complete vocal interaction between the user and the portal is achieved only with a dedicated markup language. For this reason, our second approach was to include in Jetspeed the support for X+V and SALT: the integration procedure is similar for both languages, and it is described in the next section.

5.1 New Media in the Portal

The presentation of portal pages is strongly dependent on the media type associated with the client browser. On the basis of the media type, the portal selects what portlets can be shown and it finds out the templates to use with such portlets. For example, a portlet could not be suitable for mobile devices, or it could use different templates for desktop computers and PDAs.

All portal elements (portlets, media types, clients, etc.) are stored in local XML registry files. Jetspeed associates the client browser with a media type after analyzing the browser description in the registry. For this reason, we changed the portal registry with the descriptions of three browsers supporting X+V and/or SALT. We have then added two media types, called X+V and SALT, to the registry, and we linked browsers descriptions with the corresponding media types. At this point, it is possible to build the portal for the new media types, creating portal pages and portlets with fragments of X+V or SALT markup in their templates.

5.2 Testing

As a matter of facts, neither X+V nor SALT, are widely accepted standards and they are not supported by everyday browsers. For example, in order to test SALT with Internet Explorer, it is necessary to download the Microsoft Speech Application Software Development Kit, that requires a previous installation of Internet Information Services and of the .NET Framework. We decided to test the portal with three products available free of charge and based on browsers used by a large community:

- Opera Beta 8: it is the first version of a widely used browser that comes with the support for vocal interaction. The support is activated downloading an add-on that enables X+V management.
- Access NetFront for Pocket PC: this version of NetFront supports X+V markup, and it is the result of a partnership between Access and IBM. Thanks to this browser, we succeeded in testing vocal portal navigation from a mobile device.
- OpenSALT: this open source project makes available a SALT compliant browser based on Mozilla.

We have tested the new media added to Jetspeed by including in all portal pages a set of testing portlets. These portlets have been built from the examples provided by the three browsers'.

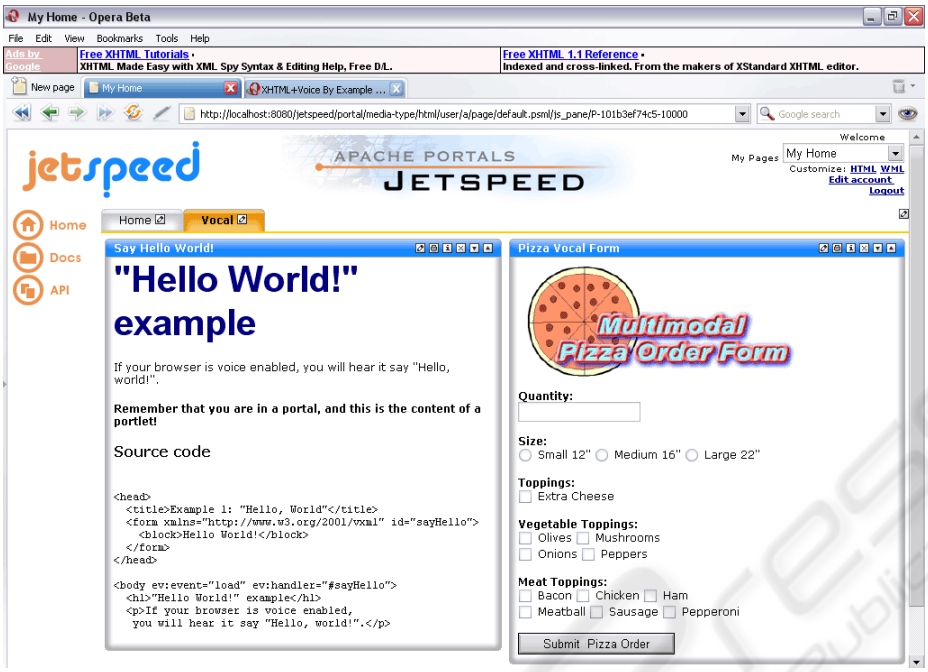

Figure 2: Portal screen with sample multimodal portlets.

developers and by X+V and SALT specifications. The new portlets enable vocal interaction with users: a user fills the fields of a form using its voice, and the portlet can answer with instructions or comments. A portal screen with vocal portlets is shown in figure 2. All tests have been positive, the portal preserves the original behaviour of the examples, and, above all, the results of the interaction (e.g.: the contents of a form) are available from the portlet classes. In this way it is possible to develop complex portlet applications based on the vocal interaction between user and portal.

6 CONCLUSIONS AND FUTURE WORK

In this paper we presented two approaches for providing a portal engine with the capability to communicate and to receive inputs using voice. This vocal interaction is a first, significant, step to reach a more complete multimodal interaction. We believe that the introduction of multimodality will play a significant role in the future of human computer interaction, mainly for two reasons: the growing request of innovative interfaces for mobile devices and the need of different interaction paradigms for the navigation of impaired Internet users.

We focused our research on portal engines, rather than on more generic Web sites, because of their capability of integrating heterogeneous applications and contents, without the need for users to explore manually several, different sites, and thus facilitating the navigation for both mobile and impaired users. Moreover, portal contents are easily adapted to mobile devices, and the interaction with users can be tailored on the basis of user profiling functionalities that portals normally provide.

The two approaches that we followed for enabling the vocal interaction are complementary, and each of them has advantages and drawbacks. The first approach generates an audio stream on the server and sends it to the client, so that the vocal interaction is possible only from the portal to the user. This process drains server resources and bandwidth, but our tests demonstrated that a synthesizer like FreeTTS can grant good performances also with several users connected. The main advantage of this approach is that it does not require a dedicated client, it works with every browser that can receive audio streams. The second approach is based on markup languages designed for client-side vocal interaction: it is the best way to enable a bidirectional vocal communication between portal and user, and it does not employ extra server resources or bandwidth. The drawbacks lie in the need of dedicated, uncommon browsers and in the higher computational power required to the client devices. Taking into account advantages and

drawbacks, we believe that both approaches can coexist in the same portal: the user selects his preferred method, according to the characteristics of the client device and of the connection.

Our future research includes the extension of this work to the EMMA language, as soon as EMMA-enabled browsers would be available. We also intend to provide portal users with other interaction modalities, e.g., pen input and gesture recognition. Moreover, we want to include in the portal engine some tools that can facilitate the multimodal interaction, e.g., an enhanced user profiling system and an improved procedure for adapting contents and interface to mobile devices. Finally, we are planning to port all our work to the new released Jetspeed 2 portal. Jetspeed 2 is compliant with JSR (Java Specification Request) 168, a standard that has enabled interoperability between portal servers and portlets (JSR 168, 2003). In this way, most of our work won't be hooked on Jetspeed, but it will be available to all portals that follow the JSR 168 specification.

ACKNOWLEDGEMENTS

This work is partially supported by MIUR (Ministero dell'Istruzione, dell'Università e della Ricerca) through the FIRB WEB-MINDS project.

REFERENCES

UMTS Forum Home Page. Available from http://www.umts-forum.org

Multimodal Interaction Activity Home Page. Available from http://www.w3.org/2002/mmi/

EMMA Working Draft Home Page. Available from http://www.w3.org/TR/emma/

X+V Specification. Available from http://www.voicexml.org/specs/multimodal/x+v/12/

SALT Forum Home Page. Available from http://www.saltforum.org/

Cohen, P. R.., 1992: The Role of Natural Language in a Multimodal Interface. *In Proc.of the 5th annual ACM symposium on User interface software and technology pp. 143-149, Monteray, California, United States*.

Sharma, R., Pavlovic, V. I., Huang, T. S., 1998: Toward Multimodal Human-Computer Interface. *In Proc. IEEE special issue on Multimedia Signal Processing*, *86(5):853-859*

Baber, C., Mellor, B., 2001: Using Critical Path Analysis to Model Multimodal Human-Computer Interaction. *In Int. Journal Human-Computer Studies, 54, 613-636*

Moran, D. B., Cheyer, A. J., Julia, L. E., Martin, D. L., Park, S., 1997: Multimodal User Interfaces in the Open Agent Architecture. *In Proc. of the 1997 International Conference on Intelligent User Interfaces (IUI97), 61-68*

Wahlster, W., Reithinger, N., Blocher, A., 2001: SmartKom: Towards Multimodal Dialogues with Anthropomorphic Interface Agents. *In Proc. of International Status Conference, Human Computer Interaction, 23-34*

Lisetti, C., Nasoz, F., LeRouge, C., Ozyer, O., Alvarez, K., 2003: Developing Multimodal Intelligent Affective Interfaces for Tele-Home Health Care. *In Int. Journal Human-Computer Studies, 59, 245-255*

Bellik, Y., 1997: Multimodal Text Editor Interface Including Speech for the Blind. *In Speech Communication, 23*

Mari, M., Poggi, A., 2004: A Transcoding Based Approach for Multi-Device Portal Contents Adaptation. *In Proc. WWW/Internet 2004 (IADIS International Conference 2004), pp. 107-114, Madrid, Spain*

Dybkjaer, L., Bernsen, N. O., Minker, W., 2004: Evaluation and Usability of Multimodal Spoken Language Dialogue Systems. *In Speech Communication, 43, 33-54*

Walker, M. A., Whittaker, S. J., Stent, A., Maloor, P., Moore, J., Johnston, M., Vasireddy, G., 2004: Generation and Evaluation of User Tailored Responses in Multimodal Dialogue. *In Cognitive Sciences, 28, 811-840*

Apache Jetspeed 1 Home Page. Available from: http://portals.apache.org/jetspeed-1/

I-Mode Home Page. Available from: http://www.nttdocomo.com/corebiz/imode/index.html

FreeTTS Home Page. Available from: http://freetts.sourceforge.net/docs/index.php

Flite Home Page. Available from: http://www.speech.cs.cmu.edu/flite/

JSAPI Home Page. Available from: http://java.sun.com/products/java-media/speech/

Apache Turbine Home Page. Available from: http://jakarta.apache.org/turbine/

Apache Velocity Home Page. Available from: http://jakarta.apache.org/velocity/

Opera Browser Home Page. Available from: http://www.opera.com

Access NetFront Home Page. Available from: http://www.access.co.jp/english/

OpenSALT Home Page. Available from: http://hap.speech.cs.cmu.edu/salt/

JSR 168 Specification, 2003. Available from: http://www.jcp.org/en/jsr/detail?id=168