2008 Third International Workshop on Semantic Media Adaptation and Personalization

A Speech-Enabled Assistive Collaborative Platform for Educational Purposes with User Personalization

V. Kolias[1], C. Kolias[2], I. Anagnostopoulos[2], G. Kambourakis[2], E. Kayafas[1]
[1]*School of Electrical and Computer Engineering, National Technical University of Athens,*
[2]*Department of Information and Communication Systems Engineering, University of the Aegean,*
vkolias@medialab.ntua.gr, {kkolias, janag, gkamb@aegean.gr}, kayafas@cs.ntua.gr

Abstract

With the proliferation of Web 2.0 applications, collaborative learning has gathered a lot of attention due its potentiality in the e-learning field. Forums, Wikis and Blogs for example are only some of the applications that exploit the collaborative nature of e-learning. However, these applications are originally designed for access from desktop systems and access to them when on the move can prove a challenging task. This paper elaborates on the design and implementation of an assistive collaborative platform for educational purposes that can be accessed by heterogeneous hardware platforms such as PCs, PDAs, mobile or traditional phones due to its capability of representing data in vocal manner. Its main purpose is to provide a platform for collaboration between university students and teachers in a way that enhances students' access to educational resources and their overall learning experience. This is achieved by personalizing its content at least to some degree. Furthermore, its acoustic/vocal characteristics may also prove valuable for learners with visual or kinetic impairments.

1. Introduction

The tremendous penetration of mobile devices has drawn the attention of the research community to wireless learning also known as m-learning. Although still in prime stages, due to its native characteristics, m-learning is expected to have a huge impact on e-learning. For instance, mobile devices have a much more personalized relationship with their owners when compared to normal PCs and they provide ubiquitous access i.e., they are always available for use from virtually everywhere. On the other hand, the inherent deficiencies of mobile devices render such an anytime, anywhere learning experience a cumbersome task. The hardware characteristics of mobile devices like small screens, limited memory and network bandwidth, as well as other reasons i.e., access to the Internet is not everywhere guaranteed and the costs for Internet access are just few of the factors that retard the evolution of m-learning. In addition, the pluralism of standards and technologies the mobile device vendors have adopted makes the migration of existing desktop-oriented learning applications to the mobile world very difficult.

Ideally data should be described and presented in a uniform way among all mobile platforms. Unfortunately, with the available technologies this does not come in the shape of a state-of-the-art standard or working recommendation but rather in the shape of voice and audio. Indeed the acoustic/vocal representation of information may encompass many advantages over traditional visual approaches:

- It can work in conjunction with the visual representation of information to provide a better user experience.
- It does not require an Internet connection. The minimum requirement is a Public Switched Telephone Network (PSTN) line to transmit voice.
- It is platform and vendor independent.
- It may assist persons with visual or kinetic disabilities.

On top of that, existing technologies render this task feasible. Various open source and proprietary Text-To-Speech (TTS) engines that have the ability to produce high quality synthetic speech from text exist in the market today. Speech Recognition Engines (SRE) have been significantly improved and are currently able to recognize spoken words by comparing them with a large set of possible candidates given as input. The Word Wide Web Consortium (W3C) [1] has proposed a set of standards in order to standardize the construction of applications based on voice. VoiceXML [2] is currently the most popular language for specifying audio user interfaces for web applications. VoiceXML is used to control the flow of the dialog between the user and the computer and it is intended to be the audio equivalent of HTML. The Speech Synthesis Markup Language

978-0-7695-3444-2/08 $25.00 © 2008 IEEE
DOI 10.1109/SMAP.2008.31

(SSML) [3] enables developers to specify instructions to the speech synthesis engine regarding the pronouncement of specific words or phrases. By providing tags which control speech attributes such as pronunciation, volume, pitch and rate across different synthesis capable platforms, SSML far improves the overall result of the speech synthesizer. The Speech Recognition Grammar Specification (SRGS) [4] enables developers to specify words or phrases that should be expected or could be recognized by the speech recognition engine. At present, speech recognition engines are not capable of accepting any vocal input from the user, recognize it and possibly translate it into text. The developer must provide a predefined set of words as input that a user is expected to say at various stages of the application, which comprise what is known as a grammar.

In this paper we present the design principles and a prototype implementation of a collaborative learning platform that can be used as an assistive tool in higher education. We argue that the pervasive nature of this platform - made possible by Voice User Interfaces (VUIs) - in conjunction with its collaborative characteristics may prove a valuable resource for both students and educators. Additionally, its adaptive characteristic improves the end-user experience by adjusting its content according to previous user's behaviour.

The rest of the paper is structured as follows. Section 2 provides a brief description of previous work in the field. Section 3 presents and analyses the proposed architecture. Section 4 introduces the personalization method followed by our application, while section 5 discusses two real-life scenarios. Last section concludes the paper and gives pointers to future work.

2. Related Work

The idea of providing some type of voice user interfaces to increase the popularity of e-learning applications or assist the visually impaired individuals is not new.

In [5] the authors explore the prospects that derive from the use of speech in the field of e-learning. They propose a client/server architecture where the production of voice to text and vice versa is done on a central Speech Server and sent towards the client. The clients are implemented as java applets and are embedded on web pages that can be accessed by PCs or mobile devices. These applets also have voice recording and audio reproduction capabilities. Our comment to this approach is that even today not all mobile devices have Java support and thus access to the Internet is not guaranteed when the user is on the move.

The author in [6] discusses an approach that provides audio interface for existing Wikipedia articles with the purpose to achieve ubiquitous access for them. This is actually a very positive idea considering the fact that Wikipedia and generally Wikis enhance the learning process by means of collaboration. In this implementation many interfaces according to the type of client are provided. For instance, a user who utilizes Internet Explorer web browser to access articles acoustically must first install a media player plug-in. After a request, the current version of the article will be transformed to audio, compressed and sent to client with streaming techniques. On the other hand a user who tries to do the same from his mobile phone has to send a Short Message Service (SMS) with his request, wait a time interval (to make sure that the audio representation of the article has been produced) and then place a call and hear the resulting description. Although this approach is able to provide services even without the existence of an Internet connection, it heavily depends on the use of SMS to place requests, automatically throwing out individuals with visual or kinetic impairments. Moreover, users have to listen to the whole article sequentially without having the option to navigate directly back and forth to specific parts of their choice.

The last approach comes from the DAISY Consortium. DAISY is an organization whose mission is to develop, integrate and promote standards, technologies and implementation strategies to enable global access by people with reading disabilities to information provided by mainstream publishers, governments and libraries [7]. The DAISY consortium provides files in a specific format known as Digital Talking Book (DTB) designed for individuals with visual impairments, which can be accessed either from special software installed on a desktop computer or from a device with DTB playback capability. This requirement for some individuals cannot be met, due to the costs involved. From a developer's point of view, DTB is not a widely adopted standard and therefore it is not appropriate for the development of dynamic systems, such as e-learning ones.

3. Architecture Overview

In the following paragraphs we introduce an overview of the application's architecture. The distinct roles of basic modules that constitute the system as well as some technical details regarding the architecture's components and their organization within the system are described.

3.1 Active Modules

The proposed system is organized in four basic independent modules, with which instructors and students are able to collaborate by exchanging information and

knowledge about specific topics. The functionality and characteristics of each module are described in detail further down.

Lecture Notes: This module provides a platform for tutors to share resources for each of their lectures with the students. Educators are able to insert summaries, notes and important milestones for each of their lectures through a specifically designed web interface. At the same time they may add important announcements and upload additional resources such as presentations, documents, external links for further study, podcasts and video presentations. Since all system's modules are connected, this one can also serve as the starting point for further interaction. That is educators may initiate relative activities in other modules of the system and place a reference at the lecture notes module. For instance, this may include relative articles on the Wiki module that would be useful for a student to read, active chat rooms and open discussion threads on the forum. The user may access this module via the Internet by utilizing a desktop or mobile web browser or even a Voice over IP (VoIP) soft-phone and other hardware telephony clients such as mobile or traditional phones. Through a web client the student is able to view lecture notes, download and examine any escorting files. On top of that, through a telephony client, the student is able to listen to the lecture notes in synthesized speech and get informed about the presence of any available resources. If any of these files are in a sound format (for example a podcast) the user will be given the choice to listen to it.

Wiki: Wikis are usually web applications where users can search and share their knowledge on specific topics. The system's Wiki module provides a platform for both tutors and students to contribute and access information on lesson related subjects. By employing the web interface of the module, registered users may create articles, add/edit paragraphs, upload multimedia content relevant to the article etc. Articles can be brought back to previous versions by any registered user if he thinks that the information contained in an article is not valid. Information on an article can be linked to other articles, thus constructing a virtual graph of knowledge. Users who access this module from telephony clients may only search for articles and listen to them. Since the size of articles can be large the user is equipped with browsing commands for quick navigation through different parts of the article. The advantages of this approach are: (a) knowledge is contributed by many users therefore it is expected to be more complete and accurate, (b) knowledge is not static but is constantly updated, (c) students take an active role in the teaching process.

Forum: The forum module provides a platform for problem solving and discussion among the users of the system. Access from web clients is again straightforward i.e., the registered users can create topics in order to initiate a debate and others may reply. Access to this module by telephony clients provides the option for searching and listening to topics and replies via synthetic speech, navigate through replies via voice commands as well as recording voice replies to existing topics. The web clients have the capability of reproducing voice replies via lightweight multi-platform media players embedded in web pages.

Chat: In contrast with the previous modules, the chat module offers the user a synchronous way of collaboration and information exchange. Users can take advantage of the web interface to create or join chat rooms about specific topics. This applies to both web and telephony clients. Web clients can type messages and view or listen to them from their web browsers. Telephony clients can listen to typed messages in synthetic voice or other users' voice messages. Likewise, they can record their own voice messages. Note that the communication is actually pseudo-synchronous since the content is refreshed after a pre-defined (usually small) time interval.

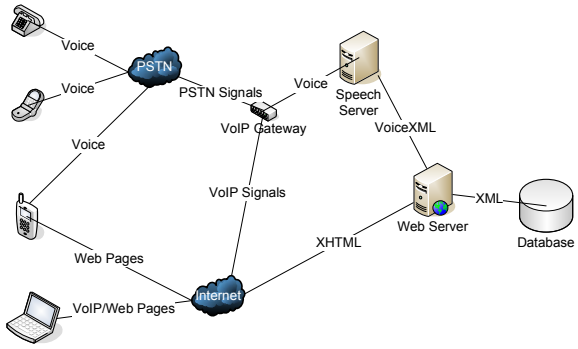

Fig. 1. Architecture overview

3.2 Architecture components description

Figure 1 depicts an abstract view of the system architecture. The system is comprised of five major components.

Database: It is the component where the content of the lecture notes, Wiki articles and forum posts reside. It is also where the user account information is stored, along with other data relative to the various adaptation procedures (see next paragraph). Any data destined for presentation, e.g. Wiki text, is stored along with its presentation markup information. The Microsoft SQL Server 2005 was employed for the implementation of the database component.

Web Server: It is the environment that hosts the application. It is responsible for serving requests that may originate from web or telephony clients. Among its var-

ious duties one critical responsibility is to monitor user behavior and construct user profiles, in order to adapt the content of the application and provide better experience to each user. Specifically, the web server: (a) receives user requests that may originate either from web browsers or phones, (b) retrieves the corresponding data from the database in XML format, (c) transforms the XML data to XHTML or VoiceXML pages based on Extensible Stylesheet Language Transformations (XSLT) scripts, (d) applies adaptive rules in order to personalize the page content and (e) forwards the resulting pages to the client or the Speech Server respectively. Our application was developed using Microsoft Windows Server 2003. An example of the third step , regarding forum data, appears in Table 1 in the Appendix.

Voice Server: The voice server is the component responsible for the transformation of textual information to audio format and vice versa. It consists of a Voice Browser, a Text To Speech (TTS) engine and an Automatic Speech Recognition (ASR) engine. The flow of a voice application is controlled by the voice browser. The voice server: (a) receives user requests and forwards them to the web server for the dynamic production of the appropriate files, (b) receives VoiceXML and grammar files, (c) analyses the instructions contained in the VoiceXML files and executes the voice application, (d) forwards (any) text to the TTS engine, (e) forwards voice (audio) to the ASR. The TTS engine receives any text from the VoiceXML file meant to be spoken, transforms it into synthetic speech and sends it to the VoIP Gateway for further processing. Furthermore, the ASR engine receives grammar files, which are sets of terms along with the client voice prompts and identifies if the prompt corresponds to any word in the grammar. If true it returns the term textually. Microsoft Speech Server was used as the Voice Server.

VoIP gateway: It is the component that receives calls from the Public Switched Telephone Network (PSTN), converts PSTN signals to VoIP signals and forwards them to the Voice Server and vice versa.

Clients: The system supports a wide range of clients. It can serve desktop web browsers or mobile ones, e.g. Firefox, Internet Explorer or IE mobile. Also, VoIP client applications (soft-phones) installed on desktop or mobile machines, like Skype or Xlite. A client might be a wired or wireless connected phone that places its requests through a PSTN network. Obviously, devices like smartphones can support both forms of interaction with the system, acoustic and visual. This feature can prove useful when the user is on the move and has no direct connection to the Internet.

4. Personalizing the student's experience

Accessing content of a relatively big volume via a phone may raise serious problems for the end-user navigating experience. For instance, textual data such as lectures and summaries or chapters from books and articles can occupy several Mbytes even when trying to divide them into categories. When accessing textual information acoustically, users have to hear it sequentially, in contrast to visual access when users can navigate through inter-linked content or skip phrases and sentences and go directly to the part they are interested in. Although the system offers some navigational commands to navigate from one part to its immediate next or previous and directly to the first or last, specific information that a user seeks may reside in a large chunk of data without the user knowing where it is. Additionally, searching into large volumes of text via phone is not an elegant approach since grammar files must include the whole text as searching terms, something that can lead to undesirable delays. Therefore, a different approach that overcomes the previous difficulties must be followed.

In this context we propose content adaptation to users' behaviour as a way to deal with the aforementioned problem. Personalized applications, also known as user-adaptive, are gaining popularity along with the emergence of web applications that attract users having diverse backgrounds of knowledge. Simply put, each user needs to be treated in a unique way, in order to maximize the effectiveness of the acquired service. Adaptive websites or adaptive hypermedia in general, allow their content and structure to be custom-tailored to each user's characteristics, behaviour or environment. Mostly used on e-commerce and e-learning applications, adaptation can take two forms: (a) adaptive presentation which modifies the textual and multimedia content of a website and (b) adaptive navigation which refers to actions such as link hiding, sorting, annotating and guiding. The most important concept of this paradigm is the user profile, which is the reflection of the user to the system. While interacting with an application various data are needed in order for an adaptation to occur. Data entered initially regarding user's preferences and other personal information, like name, sex etc are called user data. Data gathered while interacting with the application regarding the user's behaviour, for example the time one spent on a specific web page, are called usage data. Finally, data revealed by the user's machine like IP address, memory, screen resolution, are called environment data.

In order to apply adaptation in the context of our application various aspects must be taken under consideration. Since the application is meant to work as a sup-

plement of the traditional courses in higher education realms, it follows a specific learning curve, which is decided by the instructor and is rarely changed during the semester. Consequently, the content of the application is bound to the actual content and cannot adopt the individual needs and characteristics of each student. On the other hand, hiding content that students are less likely to seek or removing links that may not be needed is a useful feature for the website and a necessary addition for the audio part of the application. Since students access the application with a unique username and password, it is easy to develop a user profile which will be used in order to personalize the educational content according to previous student behaviour.

When a student accesses the service for the first time, he is not required to fill any personal information, such as name or address, since this information is provided by the institution. So students can begin interacting with the application immediately after they login. Therefore user data are available for a first personalization to occur. For example, for a freshman only the lessons of the first year will appear and the Wiki contents will be adjusted to hide advanced articles. During each interaction with the application, specific usage data are tracked and logged. Such involve data to facilitate the personalization of the web part, like a student's preferable lessons, or most visited modules. Also usage data that aim to smooth the progress of audio interaction are stored, like the time each user spends on a specific part of a module either when it access it via the web or via a phone. In this way the student interest on a specific subject is related with the time spent reading or listening to it. However, tracking this time may be deceiving in some cases. This includes situations like staying on a particular part of a page for long while in fact the user is away from the computer. The constant update of usage data with each user session softens such phenomena over time. Also the system calculates the time to size ratio which is the time a user spent on a specific part divided by the number of words that part had, so that the results are more accurate. That is, if a student reading an article spent equal time in a 1000 word paragraph referring to history, with the 300 word paragraph of the technical aspects, then it is clear that the student is more interested in the technical aspects. As a result, the next time he accesses the same article the technical aspects paragraph will be presented first.

Using the previous adaptive approach a first attempt is made towards a smoother user interaction with the system via phone. Although it saves some navigational time by skipping parts that are considered by the system as immaterial to the student, it does not solve the problem of finding a specific term inside a large text file. However we think that by achieving a satisfactory level of personalization to both content presented via the web and telephone the students' experience is enhanced and the educational procedure is improved.

5. Real-life scenarios

In this section we present two real-life scenarios in order to better understand how the application performs.

According to the first scenario a student wants to retrieve the presentation of the week's lecture on the lesson "Algorithms and Data Structures" and to see if there are any other resources available. He visits the website and he is asked for a username and password. After authentication, the webpage adjusts its content according to the user data that was initially known, and the usage data that was retrieved from the previous interactions with the website. Hence, the website greets the student with a personal welcome message and presents all active lessons in the current semester. He chooses this lesson and he is directed to the corresponding page. Under the "Lecture Notes" module he finds "this week's presentation" and some links to related websites from other universities. He also finds some available e-books, a relative link to an article contained in the wiki module, a link to a topic in the forum module and a link to the lesson "Introduction to Programming with C" which have been taught in a past semester. The instructor also had placed a link to the lesson Artificial Intelligence (AI) but it was hidden to the student since AI is taught in an advanced semester. The student downloads the presentation and he decides to visit the forum to see the active topic for this week's lecture. The topic was started by the instructor and there were various replies both textual and acoustic and he decides to add his own textual reply.

A week later the same student wants to access this week's lecture notes but he has no computer since he had to travel unexpectedly. So he calls the application, say via his 3G mobile phone. The application immediate recognizes the user since he had call once more in the past and his phone number was kept. He chooses to listen to this week's announcements and then a short summary of what was said to the lecture. He runs into a new concept which he does not comprehend so he decides to use the chat module in order to ask for immediate details. Another student is at the time connected to the same chat room, so he asks for his help, by posting an audio message. The other student listens to the post. Since he knows the answer he writes a text reply. The first student listens to the text message in the synthesized voice and then decides to visit the wiki for further details and implementation information. He asks for the concept but since he visited the wiki several times before in order to find some implementations of other algorithms, the application skips the introductory parts

straight to the implementation, thus saving the user navigational time and money.

5. Conclusions & Further Work

In this paper we presented the architectural principles and a prototype implementation of an assistive learning platform that can be accessed by both visual and vocal/acoustical means. Its highly collaborative characteristics and user adaptive features in conjunction with its pervasive nature will render it an identical supplementary tool for educators.

Our objectives are to deploy the application in a full scale enterprise environment and measure the response times of various tasks. Also we would like to evaluate the efficiency of the learning platform by measuring the opinions of students. Our next goal is to enhance the end-user experience by effectively combining visual and acoustic interfaces. Moreover, we would like to solve the problem of voice commands under noisy conditions with the use of requests in Short Message Service (SMS). Finally, we aim to improve the voice user interfaces by the use of natural language.

References

[1] W3C. Retrieved on Aug 31, 2008 from http://www.w3.org/

[2]. Voice Extensible Markup Language (VoiceXML) Version 2.0. Retrieved on Aug 31, 2008 from http://www.w3.org/TR/voicexml20/

[3]. Speech Synthesis Markup Language (SSML) Version 1.0. Retrieved on Aug 31, 2008 from http://www.w3.org/TR/speech-synthesis/

[4]. Speech Recognition Grammar Specification Version 1.0. Retrieved on Aug 31, 2008 from http://www.w3.org/TR/speech-grammar/

[5]. S. Werner, M. Wolf, M. Eichner, R. Hoffman, "Integrating Speech Enabled Services in a web-based e-Learning Environment", Proceedings of the International Conference on Information Technology: Coding and Computing, 2004

[6]. A. Bischoff, "The Pediaphon – Speech Interface to the free Wikipedia Encyclopedia for Mobile Phones, PDA's and MP3-Players", 18th International Workshop on Database and Expert Systems Applications, 2007

[7]. The DAISY Consortium. Retrieved on Aug 31, 2008 from http://www.daisy.org/

Appendix

The Forum Data in XML	Corresponding XSLT Document for Transformation to XHTML	Corresponding XSLT Document for Transformation to VoiceXML
<?xml version="1.0" encoding="utf-8" ?> <topic> <t_title> Does SSL use symetric or asymetric encryption? </t_title> <author> Student11 </author> <category> Security </category> <date> 30-7-2008 </date> <replies>2</replies> <views> 5 </views> <answered> true </answered> <post> <author> Student11 </author> <content> I am not sure if SSL uses symetric or asymetric encryption. Can someone provide help? </content> <date> 20-4-2008 </date> <answer> 0 </answer> <number>1</number> </post> <post> <author> Student45 </author> <content> SSL utilizes both symetric and asymetric encryption methods. </content> <date> 21-4-2008 </date> <answer> 1 </answer> <number>2</number> </post> </topic>	<?xml version="1.0" encoding="utf-8"?> <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"> <xsl:template match="/"> <html> <body> <table border="1"> <tr > <th colspan="2" bgcolor="gray" > <xsl:value-of select="topic/t_title"/> </th> </tr> <xsl:for-each select="topic/post"> <tr> <td> <table> <tr> <td > <xsl:value-of select="author"/> </td> </tr> <tr> <td> <xsl:value-of select="date"/> </td> </tr> </table> </td> <xsl:choose> <xsl:when test="answer = 1"> <td bgcolor="#9acd32"> <xsl:value-of select="content"/> </td> </xsl:when> <xsl:otherwise> <td> <xsl:value-of select="content"/> </td> </xsl:otherwise> </xsl:choose> </tr> </xsl:for-each> </table> </body> </html> </xsl:template> </xsl:stylesheet>	<?xml version="1.0" encoding="utf-8"?> <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"> <xsl:template match="/"> <vxml xmlns="http://www.w3.org/2001/vxml" version="2.0" xml:lang="en-US" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.w3.org/2001/vxml http://www.w3.org/TR/voicexml20/vxml.xsd"> <xsl:variable name="replytotal"> <xsl:value-of select="topic/replies"/> </xsl:variable> <form id="WelcomeFRM"> <block> You have selected to hear the thread: <xsl:value-of select="topic/t_title"/>. <goto next="#Frm1"/> </block> </form> <xsl:for-each select="topic/post"> <xsl:variable name="replynumber"> <xsl:value-of select="number"/> </xsl:variable> <xsl:variable name="formnumber" select="concat('Frm',$replynumber)"/> <xsl:variable name="nextform" select="concat('Frm',$replynumber+1)"/> <xsl:choose> <xsl:when test="number!=$replytotal"> <form id="$formnumber" > <block> User: <xsl:value-of select="author"/>, says: <xsl:value-of select="content"/>. <goto next="nextform"/> </block> </form> </xsl:when> <xsl:otherwise> <form id="$formnumber" > <block> User: <xsl:value-of select="author"/>, says: <xsl:value-of select="content"/>. </block> </form> </xsl:otherwise> </xsl:choose> </xsl:for-each> </vxml> </xsl:template> </xsl:stylesheet>

Table 1. An XML script retrieved from the database and the corresponding XSLT scripts