Design and implementation of a Voice-XML driven wiki application for assistive environments on the web

Constantinos Kolias[1], Vassilis Kolias[2], Ioannis Anagnostopoulos[1], Georgios Kambourakis[1], Eleftherios Kayafas[2]

[1]University of the Aegean, Karlovassi, GR-83200 Samos, Greece
[2]National Technical University of Athens, Zografou campus, GR-15773, Zografou Greece
{kkolias, janag, gkamb}@aegean.gr, vkolias@medialab.ntua.gr, kayafas@cs.ntua.gr

Abstract. In this paper, we describe the design and implementation of an audio wiki application accessible via both the Public Switched Telephone Network (PSTN) and the Internet. The application exploits mature World Wide Web Consortium (W3C) standards such as VoiceXML, Speech Synthesis Markup Language (SSML) and Speech Recognition Grammar Specification (SRGS) towards achieving our goals. The purpose of such an application is to assist visually impaired, technologically uneducated, and underprivileged people in accessing information originally intended to be accessed visually via a Personal Computer (PC). Users may access wiki content via fixed or mobile phones, or via a PC using a Web Browser or a Voice over IP (VoIP) service. This feature promotes pervasiveness to collaboratively created content to an extremely large population, i.e., those who simply own a telephone line.

Keywords: *VoiceXML, Wiki applications, pervasive technologies, assistive environments, collaborative applications*

1 Introduction

Wikis are generally considered as collaboration platforms where users access or contribute knowledge on specific topics. Wikis are mostly implemented as web applications that allow registered (and sometimes even unregistered) users to create, edit, hyperlink and organize their content. Wiki applications are also used in many companies to provide effective Intranets and for Knowledge Management. Beyond doubt, the huge success of Wikipedia [1], which is perhaps the most famous wiki application nowadays, proves in practice the suitability of wikis in exchanging knowledge.

The idea that wiki applications enable the collaborative writing of articles of common interest is simple enough, yet this simplicity entails a profound impact on the flow of information among their users. Their general philosophy is the facilitation of access and contribution to knowledge. However, their collaborative and in many cases, open nature raises important issues such as accessibility, content validity and security [2]. In most of the existing wiki implementations, access is as simple as browsing on a simple web page. Editing is also straightforward and it can be done by inserting information written in a specific (usually very simple) syntax, in the appropriate web forms that the wiki interface offers. Since wiki content can be changed by anyone, depending on the implementation, its validity is consigned either to the users themselves, or to specific users that are considered as experts. As far as security is concerned, it is typically achieved by utilizing standard security mechanisms to accomplish authorization, authentication and integrity, in order to relate changes of wiki content with specific users, or to ensure that the presented information is original.

Most of the existing wiki implementations are primarily dependent on web standards such as the Hypertext Markup Language (HTML). HTML represents information in a visual manner and as a result visually impaired individuals are unable to access their content. Beyond doubt the vast majority of modern web applications neglect the special needs of disabled people. Until now, visually impaired individuals rely on the features of the operating system they use (e.g., ShowSounds for the Windows XP

operating system). While these features comprise a typical feature of most of the modern operating systems, they lack of support for dynamically generated content such as content originating from the World Wide Web. Such individuals are limited in using high cost systems that often require special training as well.

A fundamental requirement of using wiki applications is having some sort of access to the Internet. This automatically makes access to its content prohibited to a large population, such as various semi-literate and illiterate people in cities and rural areas of emerging economies or technologically uneducated people such as the elderly. Thus, the inability of underprivileged people to afford computers or web-enabled handheld devices as well as the disinterest of some people to acquire basic Information Technology (IT) skills, lead to an undesirable blockade to a large amount of knowledge.

Since wikis share a large amount of information among different people, the need to be accessible from the widest range of devices possible, is immanent. However, the integration of wikis to Personal Digital Assistants (PDA)s or smartphones is rather complicated. This is due to the great number of different standards, technologies and operating systems that exist for such devices in the market today. Web based wiki implementations are usually developed focusing on desktop scale HTML based browsers. This fact generates many problems when wiki content is accessed from mobile devices equipped with browsers supporting certain subsets of standard HTML, like Wireless Markup Language (WML) [3] or i-Mode HTML [4].

Speech is the most natural and innate communication mean available to people and when exploited it maximizes the effectiveness of human-machine interaction. At the same time fixed telephony is still considered as the most penetrated communication mean. Apart from its ubiquity in comparison with desktop computers, it obviates the need for special client-side software tailored to the needs of each operating system and standard. In this paper we present a novel wiki implementation which is based on VoiceXML [5] and other W3C standards. The proposed system is accessible not only visually through an HTML web browser but also acoustically through wired and mobile phones. The only requirement, i.e., a telephone line, is generally considered as a low cost requirement and therefore suitable for a wide range of people. The acoustic

representation of the wiki content contributes to the purpose of pervasive learning and offers great help to disabled individuals and people in developing countries.

The remaining of this paper is organized as follows: in section 2, previous work in the area is discussed. Section 3 provides an overview of the technologies and standards used in the proposed implementation. Section 4 describes the system architecture in detail, while section 5 presents some real-life scenarios. Last section offers concluding thoughts and identifies some issues raised by the utilization of audio data. Also, it addresses future work and recommendations that can possibly improve the proposed application.

2 Previous Work

During the recent past, numerous efforts have been made to acoustically access information which was originally intended to be accessed visually. Applications were developed motivated by the need to aid visually impaired individuals and at the same time serve population who do not afford access to the Internet.

In [6] a Wiki application similar to the one described in this paper is presented. It is based on the observation that mobile phones have penetrated more than the Internet into young people, becoming a fashion and in some countries, like the developing ones, has clearly dominated over it. Under this assumption the proposed wiki service waits for a Short Message Service (SMS) signal from the user's mobile phone with the title of the article he wishes to hear. After a while the service calls the user to his mobile phone and speaks the article content via a synthetic voice. During the call the user can navigate to the different sections of the article by pressing keys in his phone. The application is totally based on open source software components such as MediaWiki [7]. This application is addressed to students in emerging economies who usually are not familiarized with the use of PCs, or simply cannot afford one. On the other hand the fact that such a wiki is based on a mobile phone feature it makes it inaccessible to people, e.g. the elderly, who do not own one or cannot or do not know how to send an SMS. Although mobile phones are very popular nowadays, ordinary PSTN phones are still used by the majority of people.

In [8] authors propose a client/server architecture to integrate speech technology into web pages to be used for e-learning purposes. The production of voice to text and vice versa is done on a central speech server where the services speech synthesis, speech recognition and speaker verification are installed. The content produced is presented to the user in a web-browser in which is embedded a Java applet that implements audio input and output capabilities. However, the only requirements on the client side for this approach, i.e., a JavaScript enabled browser and Sun's Java plugin, is uncertain to be supported by current mobile devices. Additionally, access to the Internet is not guaranteed when the user is on the move or simply when he does not have an Internet connection or a PC.

In [9] authors claim that the collaborative nature of wikis is not well served because it limits its users to computer environments or, when deployed in mobile environments, it restricts them by means of input (keyboard stroke or stylus) and output (small screen). They identify that synchronous communication technologies like teleconferences or simple calls are gaining attention as channels to carry out collaborative tasks. Therefore, the combination of the wiki collaboration paradigm with audio communication means can improve the overall usability of wikis. Under this context they propose a wiki implementation to facilitate asynchronous audio-mediated collaboration when on the move. It is based solely on the manipulation of audio files. Despite the fact that it enhances collaboration with a more personalized feeling – each user contributes with his own voice – this implementation does not have any web counterpart and therefore it cannot be accessed by any mean other than acoustic.

In [10] a non-visual web browser is presented. It enables visually impaired people to navigate contents of web pages acoustically. A synthetic speech feature transforms the contents of web pages into sound and the open source Sphinx voice recognition engine [11] transforms the user's voice into signals which are recognized by the system. In this way one can navigate through pages with his voice only, and can avoid the sequential narration of their content. This application provides significant advantages especially to partially blind users that are not willing to invest time and effort to learn new communication means or acquire IT skills. On the other hand, it runs exclusively on PCs, making it inaccessible to users who do not own or do not

know how to use a computer. In addition, there is no (lightweight) version targeting to mobile devices making it inappropriate for roaming users, thus reducing its pervasiveness.

DAISY Consortium [12] 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. A DAISY Digital Talking Book (DTB) is a file with a specific format which can be accessed either from an appropriate software in a PC or from a special device with DTB playback capability. Though feasible, to integrate this approach in a wiki platform would require a conversion of the material to this specific format. Unlike VoiceXML, DTB is not a widely adopted standard and therefore it is not appropriate for developers or suitable for dynamic systems like wikis. Additionally, from a user's point of view DTB requires software which in turn requires a PC and the related skills or a special device. Both lead to extra costs for the end user.

3 Related Standards

Our proposed application is based on VoiceXML in order to become accessible acoustically by a standard phone or by a computer. Besides VoiceXML, our application exploits the power of more relative W3C standards, such as SRGS [13] and SSML [14] in order to increase the overall effectiveness of the application and enhance the end user experience. Hereunder we present a brief description of each one of these standards.

3.1 VoiceXML

VoiceXML is an Extensible Markup Language (XML) based language that aims to function as a tool for the development of interactive voice-response applications. Today, many languages serve that purpose such as Java Speech api Markup Language (JSML) but VoiceXML is the most widespread and widely adopted [15], [16]. VoiceXML is designed for creating voice applications that feature synthesized

speech, audio recognition of voice input or Dual-Tone Multi-Frequency (DTMF) input, recording of spoken input, reproduction of audio files, control of dialog flow and telephony features, such as call transfer. In general terms VoiceXML attempts to constitute the audio equivalent of Extensible Hypertext Markup Language (XHTML) for the voice web instead of the visual web. Applications based on VoiceXML require a voice browser which in turn has to include a speech synthesis and a speech recognition engine, in analogy to HTML applications which assume the existence of graphical browser, screen, keyboard and mouse. Today, VoiceXML is widely used for the creation of voice interactive applications such as: phone banking, ticket reservation, news and weather information. It is an official W3C recommendation and as of 2004 it is currently on its second version.

By utilizing VoiceXML, developers are able to neglect low-level, platform specific programming details and focus solely to the development of the voice application. It constitutes an alternative way of information presentation so existing web applications (for instance Common Gateway Interface (CGI) scripts, PHP scripts) can be extended to produce VoiceXML instead of normal XHTML/HTML markup. This is achieved without changing the logic and structure of the applications themselves. In addition, VoiceXML provides a common platform for application development across heterogeneous platforms, since it's an XML language. Moreover, it is carefully designed and equipped with many features so it can be used to create complex applications.

Voice applications based on the VoiceXML language assume that the following architecture model is adopted: (a) a document server exists, which accepts requests and responds with the appropriate pre-existing or dynamically produced documents, (b) a VoiceXML interpreter processes these documents and translates the instructions contained on those scripts, (c) an implementation platform is responsible for following specific actions in response to specific events that occur throughout the duration of each session. The VoiceXML interpreter is required by the specification to be able to: (a) support audio output either by using audio files or a Text-To-Speech (TTS) engine, (b) be able to recognize spoken input or characters and record the audio input if needed and, (c) support third party connections through various communications networks such as the telephone network.

A VoiceXML application consists of a set of interconnected documents. Every VoiceXML document contains special elements called dialogs which control the flow of voice applications. Each dialog specifies the next dialog or document that the call will transit to, by using Universal Resource Identifiers (URIs). If a dialog does not specify a successor then the execution is terminated. Dialogs can be forms or menus. Forms are used for presenting information and accepting user input, while menus are used for presenting the user with a set of choices and transits to the appropriate dialog based on the result of the user choice. Table 1 presents a sample of a basic VoiceXML document, in which bold and italic text style represents embedded SRGS and SSML elements respectively.

Each dialog must have one or more grammar files associated with it. Grammars contain lists of words that the interpreter may consider valid during the time a user uses that specific dialog. Speech and/or DTMF grammar files exist for recognizing voice commands or digital tones as user input respectively. The role of grammar files is very important for the development of dialogs and a different W3C recommendation exist to specify their structure and purpose. In the following paragraph we elaborate on grammars.

3.2 SRGS

The Speech Recognition Grammar Specification (SRGS) is a W3C recommendation for the creation of grammar documents within voice-response applications [13]. These files are intended for use by speech recognition engine and provide the programmers with the ability to specify list of words to be expected as input and 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 to 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. This set of words is usually described in a corresponding grammar.

In turn, the speech recognition engine is the part of the VoiceXML interpreter which accepts a grammar or multiple grammars as input as well as an audio stream and attempts to figure out if the speech content matches the grammars. The output is

detailed descriptions that indicate if the speech content matches with some of the contents of the grammar files or not. A grammar can be embedded inside the VoiceXML file or in most cases it can be an independent file and just be referenced in the appropriate VoiceXML file. The grammar file may also be static or it can be produced dynamically. A static file may contain a set of predefined words that are not expected to change during the application execution. In the proposed application such words are used as voice commands for browsing, i.e., *repeat*, *back*, *next* to mention just a few. A dynamic grammar file is necessary when the expected terms may change, be added, amended or deleted very often. This is the case for the proposed wiki application in which the article names that the user may search for, change frequently. Currently there are two main formats for writing SRGS grammar files: Augmented Backus-Naur Form (ABNF) and Grammar XML (GrXML). GrXML format is an XML based language unlike the ABNF format which was designed to be more human readable. In our application all our SRGS grammar files are written in the GrXML format, thus the produced grammar files are named GrXML files.

3.3 SSML

The Speech Synthesis Markup Language (SSML) is a W3C recommendation based on the preexisting Java Speech Grammar Format (JSGF) and JSML standards developed by Sun Microsystems [15], [16]. SSML aims to enrich the synthetic speech generation by providing voice application developers with a standard mechanism for controlling various aspects of the produced speech such as pronunciation, volume, pitch, rate etc across heterogeneous platforms. The ultimate goal of the SSML is to improve the aesthetic result of the synthesized speech thus producing a result which is closer to natural human speech. It is stressed that SSML may be embedded in VoiceXML scripts or referenced as a standalone file. In the presented implementation SSML tags are relatively small in number, thus SSML code is embedded in the VoiceXML file.

4 System Architecture

Figure 1 depicts a high level view of the system architecture. The proposed system is comprised of four major components.

Database: The database is a key system's architecture component in which application data are stored and from where they are retrieved. It holds information regarding the contents of the articles, their previous versions, the registered users (the users that have the privileges to create new articles) etc. The complexity of the database scheme is kept low. It is stressed that the content of the articles is stored in the database mixed with presentation information. The latter are nothing but special sequences of characters, usually referred to as wikitext that a user inserts during the editing of an article.

Web Server: On the web server resides the Wiki engine which is a web application responsible for the following tasks:

1. To constantly wait for the user's requests, either from a web browser, thus starting a web page session, or from a voice browser, thus starting a voice session.
2. To communicate with the database to retrieve or update article data.
3. In the case of voice session, the wiki engine has to generate a dynamic grammar of the available article titles, one of which will be the user's choice for presentation. In a wiki application the number of existing articles (and as a result the terms that the user is expected to ask for) might take large proportions. That would lead to the generation of a large grammar file which would affect the overall performance of the system and consequently the user's overall experience. For this reason, articles are categorized and the user is requested to first choose the category of the article and then ask for the article's title. In this way grammar files are kept smaller.
4. To transform wikitext sequences retrieved from the database either in XHTML in the case of a web page session or in VoiceXML in the case of a voice session. An example of the transformation of wikitext into XHTML or VoiceXML is presented in Table 2.
5. To send the generated XHTML files to the user's web browser or the VoiceXML files to the system's Voice Server for further processing.

Voice Server: The voice server is the component responsible for the transformation of text documents to audio data. The voice server consists of a Voice Browser, a TTS engine and an Automatic Speech Recognition (ASR) engine. Additionally, a VoIP gateway is an additional component that plays an important role during the transformation, but is not an actual component of the voice server itself. The Voice Browser:

1. Accepts request from the user and in response proceeds to the appropriate actions.
2. Receives VoiceXML and grammar files from the Web Server.
3. Specifies the execution flow according to the instructions in the VoiceXML file. For instance at a given time it may forward the text meant to be spoken to the TTS engine.
4. Generates the request made by the user and forwards it to the Web Server.

The TTS engine receives the text from the VoiceXML file meant to be spoken, transforms it into streaming sound and sends it to the VoIP Gateway to forward it to the end-user. On the other hand, the Automatic Speech Recognition engine receives a grammar, which is a set of terms that is able to recognize along with the client prompt and identifies if the prompt corresponds to any word in the grammar. If true it returns the term textually. The VoIP gateway receives calls from the Public Switched Telephone Network (PSTN), converts PSTN signals to VoIP signals and forwards them to the Voice Server. It is to be noted that the voice server will accept only VoIP signals. Signals originating from the Internet (from VoIP clients) might be Session Initiation Protocol (SIP) [17] signals, e.g. from Xlite softphone, or signals of some proprietary VoIP protocol, like Skype. The VoIP gateway is also responsible for the transformation of VoIP signals from the Internet to the protocol that the Voice Server recognizes. The Web Server and Voice Server components are depicted in Figure 2.

Clients: The system might accept different types of clients. A client may be a typical web browser installed on the user's PC or PDA for instance Firefox, Internet Explorer or IE mobile. It might also be a VoIP client program installed on a PC, like Skype. Finally, a client might be a fixed telephone or a wireless one that places its request through a PSTN network.

4.1 Implementation Aspects

In the proposed application the MS SQL Server 2005 [18] was used to store the wiki data. A wiki engine application was developed on ASP.NET scripts. MS Speech Server 2007 was selected for the voice server. MS Speech Server contains a powerful TTS engine as well as an advanced ASR engine. The application was tested with the Xlite softphone 3.0. Since this particular software utilizes the SIP protocol and Speech Server 2007 requires the messages to be in SIP as well, the use of VoIP gateway was not necessary.

The proposed wiki implementation converts article text to audio on the voice server and sends it to the client through the Internet or the PSTN. This fact revokes the need for a voice browser with a TTS engine, or any other software that performs analogous tasks, to be installed on the client's PC. Also, it enables clients to access the wiki content via a phone. However, the client will receive voice streams instead of simple VoiceXML documents. Normally, voice streams have a large size resulting to longer response delays. Also the architecture may have large implementation costs because it requires additional hardware and commercial software like TTS and ASR engines.

One aspect that should be take into account is that the twofold nature of this implementation may raise various considerations, regarding its effectiveness. Audio as a temporal medium, cannot be concurrently presented in contrast to text which can be presented in parallel. At the same time grammar files can contain a small amount of terms and this may downscale the searching feature. Therefore users, when accessing an article via telephone, they have to hear its content sequentially, in order to find a unit of interest, something that significantly deteriorates the overall user experience. An approach to deal with this could be to add personalization and adaptive features to the application. Users, through the web interface, could specify which content they would like to hear via phone, and save time when they access it. Additionally, the application itself, could create user profiles by keeping each user's browsing history and infer which content they consider interesting and which not. Then the application could automatically present the content that is considered interesting and hide the irrelevant one.

Other features that could enhance the end user's experience could be:

- Use of natural language: Natural language lets user answer questions set by the application or speak application commands in an open way, that is without limiting the user's responses. Obviously, this yields for sophisticated construction of grammar files but the improvement of the users experience will be significant.
- Multilanguage speech synthesis: The implementation supports only recognition and speech synthesis for the English language. Expansion of the application to support other languages is considered mandatory since one of its main goals is to address and attract users from emerging economies most of who do not speak English.
- Voice annotations provision: In its current form, the application supports pronunciation of the content of existing articles via synthetic speech. It does not support editing or creation of new articles due to the incapability of existing recognition engines to provide full speech recognition instead of recognition of words from a predefined set. Nevertheless, one of the basic characteristics of a wiki is the fact that it can be editable by almost every user. To overcome this, a possible solution is to provide the user with the ability to add voice annotations to the article by recording his voice. However, the large file size of the voice annotations in combination with the frequent article updates might cause problems to users accessing articles via standard web browsers.
- Migrate to open source components: Our application is implemented in Microsoft's Speech Server 2007 which is a commercial software, that demands from users (owners of the wiki application) to acquire a license in order to be used extensively. Open source alternatives do exist. For example, OpenVXI [19] is an open source VoiceXML interpreter toolkit. It is intended to be a component of Voice Browsers and provides APIs for speech recognition, speech synthesis and telephony services. Festival [20] is a general speech synthesis system that offers several APIs for speech synthesis and a rich development environment. Also, Sphinx is an open source speech recognition system. The main disadvantage is the absence of a single platform that

integrates all these functions and thus collaboration or interface problems may occur when all these components are put together.

4.2 Real Life Scenarios

In this paragraph we present a scenario of a typical user – application interaction, in order for the reader to better understand the behaviour as well as the inner mechanics of the proposed application.

In a simple scenario a student with visual impairment wishes to acquire information about the concept of local area networks for his essay at school, but he cannot use his PC, since it is out of order at the moment. He decides to call a well known number that the application is associated with (we consider this an easy task even for fully blind persons) on his fixed line phone and access this information acoustically. After he listens to a short welcome message, he is presented with the existing categories that the articles are currently organized (computers, history, literature, science etc.). He speaks the category he is interested in (in this case "computers") and then speaks the term that his search will be based on (in this case "Local Area Networks"). Since the term is contained in an article title in this category the user will be presented with the corresponding article content. The user has the option to listen to each paragraph of the article sequentially, one after another until it reaches the end, skip paragraphs, return to a previous paragraph or move to a specific paragraph immediately. This happens by giving specific navigation commands such as BEGIN, NEXT, PREVIOUS, 1, 2 etc. At any point the user is able to listen to the user manual by speaking the command HELP. This case of course is very similar when a visitor of a museum wants to quickly know some additional details about the museum exhibits but has no laptop or internet connection. He can use his cell phone to acquire this information wirelessly. Table 3 presents a typical dialog like the one described above between a user and the application.

When a request is sent to the web server, the wiki engine analyzes it and produces the appropriate query to retrieve the corresponding data from the database. Using these data it creates the appropriate VoiceXML and grammar documents. When a call is made to the voice server (by a standard or mobile phone) it starts to interact with its

voice browser component. Just like a normal web browser the voice browser places a request to the web server for a document containing the information requested. The web server responds with a dynamically produced content. Unlike the usual case where the web server produced an XHTML document in this case the server produces VoiceXML and grammar documents. The voice browser receives it, interprets the XML markup, and redirects the result to its TTS engine component. The TTS engine produces audio stream based on the results it received. The voice server then converts the audio signals to packets according to a VoIP protocol such as SIP or to analog voice signals with the help of the appropriate hardware component installed on it. If the client makes the call from a SIP (soft)phone then the voice data will be carried through the Internet to reach the client. If the client makes the call from a fixed phone then the voice signals will be carried through the PSTN network instead. Of course if the access to the wiki is done from a web browser then normal HTML pages will be generated and sent to the browser for display. Figure 3 depicts the corresponding article as presented in the web browser.

5 System evaluation

Voice quality in VoIP systems is a multi-dimensional and a non-trivial problem. For such measurements two kind of evaluation methods exist. The subjective-based methods, in which speech samples are presented to an evaluation group of listeners who rate the quality of the vocal information using an integer opinion score. All scores are then averaged to produce a Mean Opinion Score (MOS) value [21]. Subjective tests are highly reliable, yet they are time-consuming, costly, and the results may be biased to human perspective and the different test settings. Thus, the objective-based testing methods have been developed as a solution for effectively measuring voice quality and dealing in parallel with all these issues. These methods consist of algorithms carried out by devices involved in the VoIP architecture, and do not require any intervention from evaluators. Objective tests are further categorized in the intrusive and non-intrusive methods, based on whether a reference voice signal is used or not respectively.

In this paper we used a non-intrusive objective evaluation method, calculating parameters of the ITU-T Recommendation G.107 Protocol (also known as E-model) [22]. E-model was developed by the European Telecommunications Standards Institute (ETSI) and adopted by International Telecommunication Union (ITU). In the objective-based methods, several QoS parameters and metrics are involved for measuring voice quality such as the Signal-to-Noise Ratio (SNR), packet delays and inter-arrival variations, jitter, etc. Through these international standardized metrics, the protocol provides a transmission Rating factor (R-Factor), which varies between 0 and 100, and can be then interpreted in several subjective evaluation values, such as the commonly used MOS value. Equation 1 provides the estimated MOS value when using the objective E-model, through the calculated R-Factor defined in Equation 2.

$$\begin{aligned} &MOS = 1 + 0.035R + 7 \cdot 10^{-6} R(R-60)(100-R), \quad when\ 0 < R < 100 \\ &MOS = \{1 \ or\ 4.5\}, \quad when\ R \le 0\ or\ R \ge 100 \end{aligned} \tag{1}$$

$$R = R_o - I_s - I_d - I_e + A \tag{2}$$

where R_o: a basic SNR value,

I_s, I_d, I_e: metrics dedicated for the calculation of distortions occurred in the voice signal (combination impairment factor), distortions caused by end-to-end delays and echoing (delay impairment factor), as well as distortions caused by coding-decoding and packet loss (equipment impairment factor) respectively,

A: threshold for fine-tuning the acceptable level of a voice signal (advantage factor).

In our non-intrusive objective method we performed true real-time jitter measurements. The main advantages of a true real-time jitter measurement are two-fold. On one hand, there is no dependence on packets needing to be sent at a known interval, while on the other hand the method can measure jitter on a bursty traffic like the one we expected to have in our application. Also, this method does not restrict test duration. This is because the calculation occurs in real time as packets are received, with no need of packet capture. Finally, this method compensates for lost and out-of-sequence packets, producing results in real-time for instant feedback even when traffic load or device parameters vary [23].

For real time acquisition of the parameters and metrics needed, we use CommView, which is a tool capable for real-time monitoring in Internet and Local Area Networks (LANs) as well as in analyzing activity of captured data network packets (http://www.tamos.com/products/commview/).

Table 4 depicts the results (in average values) regarding a set of evaluation experiments conducted for measuring the call quality of our system. The evaluation procedure was performed as follows. We used 45 individuals, separated in three groups, namely A, B and C, each consisted of 15 users. Group A and C consisted of VoIP users, while Group B consisted of telephone users who either use PSTN or a mobile telephone. During the first sets of experiments, we recorded the averaged values of the R-Factor, the available bandwidth of the voice server, as well as the jitter between the two communication parts, when a single VoIP session was performed from each user. We then performed the same voice sessions ten additional times, by generating traffic in the voice server through protocols UDP, TCP and ICMP. This was made in order to assess the performance of our system under different network stressing conditions. The size of the generated packets was 42, 54 and 106 Bytes for UDP, TCP and ICMP protocols respectively, while their generation rate was 30 packets/sec.

In the second set of evaluation, the user groups involved used either a PSTN and VoIP call or a call from a mobile phone and VoIP call (Group B and A respectively). Calls were performed virtually simultaneously. However, since we were not able to measure the voice quality the users received through telephone calls, we asked from the users to use the 5-points MOS scale for their evaluation. Finally, the experiments ended with the evaluation made for three simultaneously VoIP calls, from users of Group A and Group C. Thus, the total amount of separate experiments conducted for voice quality evaluation over different context and networking conditions were 270 (90 for single VoIP calls, 90 for two simultaneous calls of different context and 90 for two simultaneous VoIP calls).

The values depicted in Table 4 are averaged values from ten different experiments. Figures 4, 5 and 6 illustrate the averaged values for all three different kind of experiments explained above.

It was evaluated that in most cases R-Factor was not significantly influenced by traffic or stressing network conditions. However, a significant reduce of the R-Factor was measured when three users called simultaneously under the heaviest stressing condition (icmp, tcd and udp traffic generation – Figure 4, case v). Now, as far as jitter is concerned, Figure 5 clearly shows that this metric is totally correlated with the amount of simultaneous calls and/or calls of different context, as its averaged values considerably increases in all cases. Once again, the larger increase appeared in case v. Finally, Figure 6 shows that the percentage of available bandwidth presents a similar behavior in respect to the R-Factor variation, and an inverse analogous relation with jitter variation in all cases. Nevertheless, the slope in the R-Factor reduction is notably lower in comparison to the one of the available bandwidth in cases i,ii, iii, and iv. It was assessed that even if the available bandwidth fall down nearly 10% at the most in all three separate experiments (84.61 to 74.58, 68.21 to 59.67, and 69.41 to 62.24 respectively), R-Factor averaged values were dropped only by 2 units at the most (93.3 to 91.4, 91 to 90, and 90.8 to 89.1). This was very encouraging, indicating that our system is quite robust when it comes to provide high quality vocal services with two simultaneously calls. It was also worth noticing that even if three users used simultaneously our system, R-Factor was measured in acceptable levels, even when some stressing network conditions were applied (see cases i, ii, iii, and iv in the last column of Table 4). In these cases, R-Factor varied between 90.8 to 89.6 (4.2 to 4.0 in the MOS scale). Problems appeared when three simultaneous calls were made and the voice server was severely stressed with icmp, tcp and udp generated packets. In that case R-Factor was measured at the levels of 83.4 in average. Even though this number is not so low for having an acceptable communication, some users claimed that didn't hear clearly the system generated voice, probably due to lower R-Factor values. In other words, this means that our system is capable of serving two users simultaneously at any call context. Three users may also be supported but subject to no heavy stressing network conditions.

6 Conclusions and future work

In this paper, a novel wiki application that can be accessed by virtually any wired or wireless phone as well as by a common web browser was presented. By doing so, a strong tool for collaboration, such as a wiki, was made accessible from practically everywhere. Our application can equally support similar collaboration tools like Blikis [24], and other emerging tools like Twitters [25]. Since common telephones are installed in almost every home, our application can bring wikis closer to a wider range of people who do not own or are not comfortable with the use of a PC. The advantages of the proposed implementation over similar existing implementations are: (a) it does not require installation of special software or a PC for someone to listen to wiki articles, (b) it is cost efficient for the end user, and (c) it accepts voice commands for navigation throughout articles and for controlling the application flow.

Currently, we are focusing our efforts on creating a voice interface that will interact with the user in a more immediate way, so that the user will spend as less time as possible searching, locating and navigating throughout the various articles. For that purpose we are exploring various adaptive hypermedia techniques in order to proactively adapt the content of articles to each user's likings. The use of a more natural language for the interaction with the user is also desirable.

References

1. http://www.wikipedia.org
2. C. Kolias, S. Demertzis, G. Kambourakis, Design and Implementation of a Secure Mobile Wiki System, 7th IASTED International Conference on Web-based Education (WBE 2008), Uskov V. (Ed.), pp. 212-217, March 2008, Innsbruck, Austria.
3. WML 2.0 Specification http://www1.wapforum.org/tech/terms.asp?doc=WAP-238-WML-20010911-a.pdf

4. i-mode Compatible 7.1 Specification http://www.nttdocomo.co.jp/english/service/imode /make/content/html/version/index.html#p071
5. Voice Extensible Markup Language 2.0 Specification Retrieved on December 30, 2008 http://www.w3.org/TR/voicexml20/
6. Leinonen, T, Aucamp, FN and Sari, ER. 2006. Audio Wiki for mobile communities: information system for the rest of Us. Workshop on speech in mobile and pervasive environments, Mobile HCI 06 Conference, 12 September 2006, pp 3.
7. MediaWiki Retrieved on May 5, 2008. from http://www.mediawiki.org/wiki/MediaWiki
8. Werner, S.; Wolff, M.; Eichner, M.; Hoffmann, R., Integrating speech enabled services in a Web-based e-learning environment, Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on , vol.2, no., pp. 303-307 Vol.2, 5-7 April 2004
9. Wang, L., Roe, P., Pham, B., and Tjondronegoro, D. 2008. An audio wiki supporting mobile collaboration. In Proceedings of the 2008 ACM Symposium on Applied Computing (Fortaleza, Ceara, Brazil, March 16 - 20, 2008). SAC '08. ACM, New York, NY, 1889-1896
10. Yevgen Borodin, Jalal Mahmud, I.V. Ramakrishman, Amanda Stent, 2007 The HearSay Non-Visual Web Browser, ACM International Conference Proceeding Series, Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A), Vol. 225, pp. 128-129, Banff, Canada, 2007.
11. The CMU Sphinx Group Open Source Speech Recognition Engines Retrieved on December 30, 2008 from http://cmusphinx.sourceforge.net/html/cmusphinx.php
12. The DAISY Consortium. Retrieved on December 30, 2008 from http://www.daisy.org/
13. Speech Recognition Grammar 1.0 Specification Retrieved on December 30, 2008 http://www.w3.org/TR/speech-grammar/
14. Speech Synthesis Markup Language Retrieved on December 30, 2008 http://www.w3.org/TR/speech-synthesis/

15. JSpeech Grammar Format Retrieved on December 30, 2008 http://www.w3.org/TR/2000/NOTE-jsgf-20000605/
16. JSpeech Markup Language Retrieved on December 30, 2008 http://www.w3.org/TR/jsml/
17. Session Initiation Protocol Retrieved on December 30, 2008 http://www.cs.columbia.edu/sip/drafts.html
18. MS SQL Server 2005 Retrieved on December 30, 2008 http://www.microsoft.com/Sqlserver/2005/en/us/express.aspx
19. Vocalocity's openVXI 3.0 Retrieved on December 30, 2008 http://www.speech.cs.cmu.edu/openvxi/
20. The Festival Speech Synthesis System. Retrieved on December 30, 2008 http://www.cstr.ed.ac.uk/projects/festival/
21. L. Ding, Learn about VoIP quality measurements, white paper, Retrieved on May 4 from www.embeddeddesignindia.co.in/STATIC/PDF/200903, EE Times-India, 2009
22. ITU-T Rec. G.107, The E-Model, a computational model for use in transmission planning, March 2005
23. Spirent Communications, Measuring Jitter Accurately, white paper retrieved on May 4 2009 from www.spirent.com/documents/4814.pdf, (2007)
24. WikyBlog, http://www.wikyblog.com/ Retrieved on July 20 2009
25. Twitter, http://twitter.com/ Retrieved on July 20 2009

List of Tables

Table 1 A Sample VoiceXML document

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/ vxml">
  <form>
    <block>
      <prompt>
          Welcome to the <emphasis>Voice Wiki</emphasis>
          <break time='500ms'/>
          Please, provide a category.
            <grammar>
                 <rule id="category">
                    <one-of>
                    <item>history</item>
                    <item >science</item>
                    <item >environment</item>
                     .
                     .
                     .
                    </one-of>
                </rule>
            </grammar>
      </prompt>
    </block>
  </form>
</vxml>
```

Table 2 The Corresponding XHTML and VoiceXML Forms of Wikitext

WikiText	XHTML	VoiceXML
== Local Area Network == '''Contents''' * History * Technical aspects * See also * References === History === The first ''' LAN ''' put into service occurred in 1964 at the Livermore Laboratory to support atomic weapons research. LANs spread to the public sector in the late 1970s and were used to create high-speed	Local Area Network <table> <tr> <td>Contents</td> </tr> <tr> <td> History Technical aspects See also <a href=”# References”> References </td> </tr> </table> History The first LAN put into service occurred in 1964 at the Livermore Laboratory to support atomic weapons research. LANs spread to the public sector in the late 1970s and were used to create high-	You are now listening to article entitled: <emphasis> Local Area Network </emphasis>. <break time=’500ms’/> <menu> <choice next=”#History” accept=”approximate”> Say 1 to move to paragraph: History. </choice> <choice next=”#Technical_aspects” accept=”approximate”> Say 2 to move to paragraph: Technical aspects. </choice> <choice next=”#See_also” accept=”approximate”> Say 3 to move to paragraph: See also. </choice> <choice next=”#References” accept=”approximate”> Say 4 to move to paragraph: References. </choice> </menu> <break time=’500ms’/> <form id=”History”> The first <emphasis> LAN </emphasis> put into

links between several large central computers at one site. Of many competing systems created at this time…	speed links between several large central computers at one site. Of many competing systems created at this time…	service occurred in 1964 at the Livermore Laboratory to support atomic weapons research. LANs spread to the public sector in the late 1970s and were used to create high-speed links between several large central computers at one site. Of many competing systems created at this time…

Table 3 Sample user - system interaction

System	Welcome to the Voice Wiki. Say HELP for navigation instructions or say SEARCH to search for article.
User	HELP
System	Say BEGIN to read an article from the beginning. While reading say CONTENTS to hear the article contents. Say NEXT to skip to the next paragraph. Say PREVIOUS to move back to the previous paragraph. Say HELP to hear navigation instructions.
System	Say HELP for navigation instructions or say SEARCH to search for article.
User	SEARCH
System	Please say one of the following categories. ART, COMPUTERS, HISTORY, SCIENCE.
User	COMPUTERS
System	Did you say COMPUTERS?
User	YES
System	Please provide your search term.
User	LOCAL AREA NETWORKS
System	Did you say LOCAL AREA NETWORKS?
User	YES
System	One article found. Article name is Local Area Network. Please say how to proceed.
User	BEGIN
System	A local area network is a computer network covering small geographic area like home, office or group of buildings …
User	NEXT
System	The first LAN put into service occurred in 1964 at the Livermore Laboratory…

Table 4. Averaged call quality evaluation results

Call context	**sing1e VoIP call [1 user]**			**same time calls (pstn OR mobile phone) AND VoIP [2 users]**			**same time VoIP calls [3 users]**		
Evaluation Parameters	**R-Factor**	**Bandwidth (%)**	**Jitter (ms)**	**(MOS) / R-Factor**	**Bandwidth (%)**	**Jitter (ms)**	**R-Factor**	**Bandwidth (%)**	**Jitter (ms)**
no stress (i)	93.3	84.61	3.97	(4.3) / 91.0	68.21	6.39	90.8	69.41	6.78
tcp+udp (ii)	91.4	73.11	4.87	(4.0) / 90.0	57.91	6.25	89.1	60.78	7.01
icmp+tcp (iii)	92.3	74.12	4.63	(4.0) / 90.3	58.21	6.78	89.3	61.75	6.83
icmp+udp (iv)	92.8	74.58	5.67	(4.0) / 90.1	59.67	6.34	89.6	62.24	7.72
icmp+tcp+udp (v)	88.4	63.32	8.21	(3.8) / 88.6	48.30	8.63	83.4	39.23	10.89

List of Figures

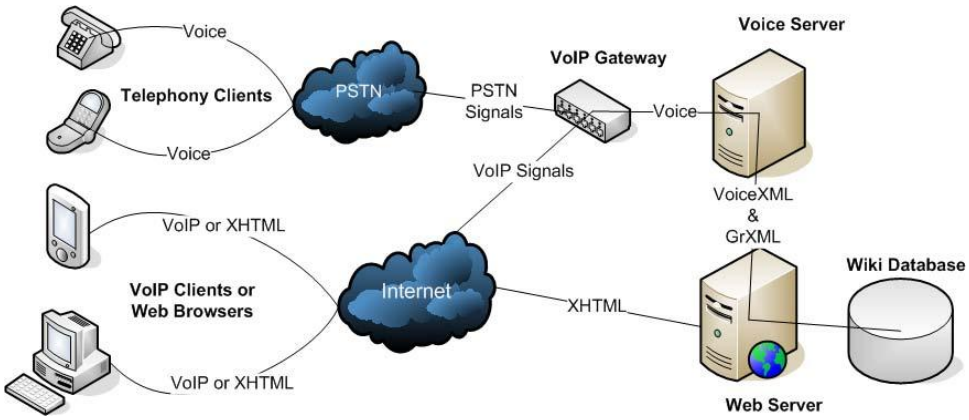

Figure 1 System Architecture

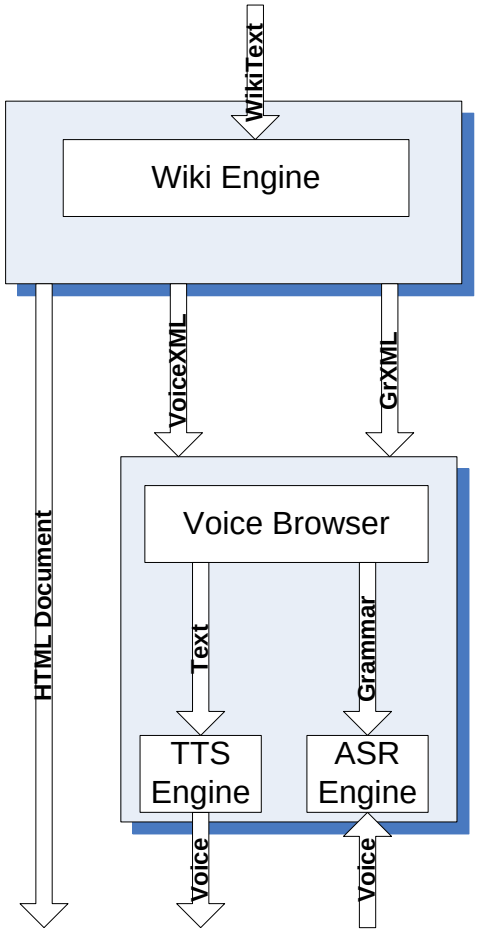

Figure 2 Web Server and Voice Server components

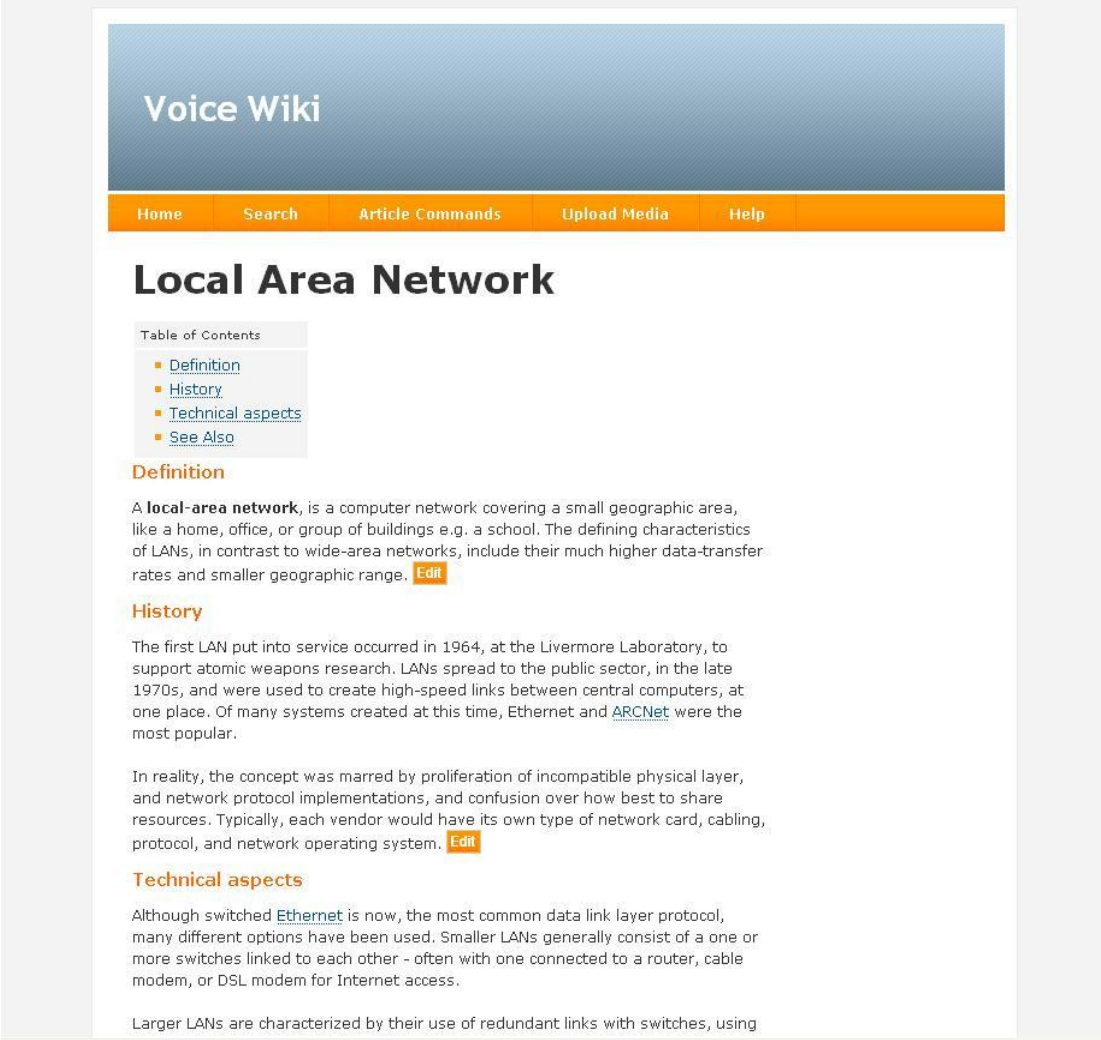

Figure 3 The wiki application accessed from a web browser

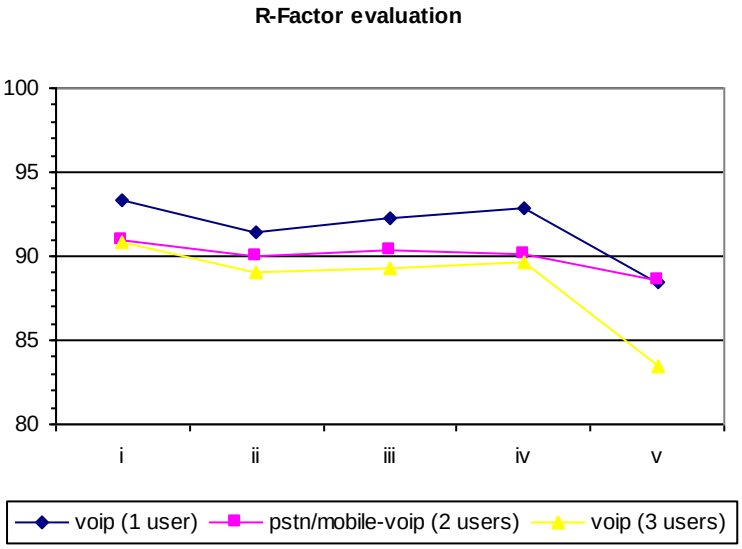

Figure 4 R-Factor evaluation over call context and traffic load

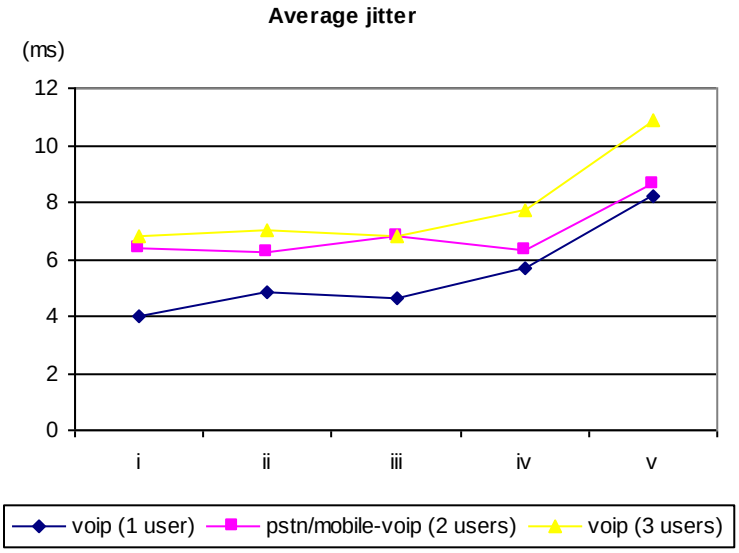

Figure 5. Jitter evaluation over call context and traffic load

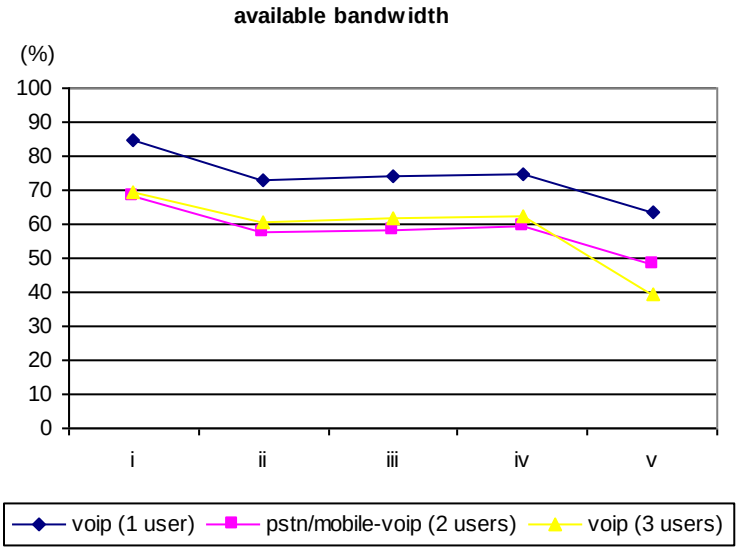

Figure 6. Jitter evaluation over call context and traffic load

Appendix: Acronyms used in the article and their meaning

Acronym	Explanation
ABNF	Augmented Backus-Naur Form
ASR	Automatic Speech Recognition
CGI	Common Gateway Interface
DTB	Digital Talking Book
DTMF	Dual-Tone Multi-Frequency
GrXML	Grammar XML
HTML	Hypertext Markup Language
IT	Information Technology
JSGF	Java Speech Grammar Format
JSML	Java Speech API Markup Language
PC	Personal Computer
PSTN	Public Switched Telephone Network
SIP	Session Initiation Protocol
SMS	Short Message Service
SRGS	Speech Recognition Grammar Specification
SSML	Speech Synthesis Markup Language
TTS	Text To Speech
URI	Uniform Resource Identifier
VoiceXML	Voice eXtensible Markup Language
VoIP	Voice over IP
WML	Wireless Markup Language
W3C	World Wide Web Consortium
XHTML	Extensible Hyper Text Markup Language
XML	eXtensible Markup Language