Structured Speech Control for Semantic Radio Based on an Embedded VoiceXML System

Günther Schatter
Bauhaus University Weimar
Faculty of Media, Weimar, Germany
schatter@uni-weimar.de

Sebastian Schmolke
Bauhaus University Weimar
Faculty of Media, Weimar, Germany
schmolke@uni-weimar.de

Abstract
We present a speech controlled digital broadcasting receiver based on the standardised VoiceXML for embedded systems. In addition to this development we examined and compared several command sets and functionalities of speech-controlled entertainment systems and expanded them with additional information services. The approach aims at reducing the resource demands that traditionally are associated with VoiceXML. The benefit of our approach is the opportunity to apply existing tools and environments for flexible dialogue design and a broader offer concerning the functionality of the media devices. Furthermore, the process of development is considerably accelerated.

Keywords
digital audio broadcasting, user interface, semantic radio, information system, embedded system, markup language, command structure, dialogue design, speech technology, text to speech, automatic speech recognition, data base

INTRODUCTION

It is a matter of common knowledge, the medium speech is the original form of communication between people but to a minor degree between humans and machines. The analogue radio–originally strong voice-based–was invented more than 100 years ago, and continues to be a very important means to communicate with the world. In recent years, digital receivers became soaring processing machines. The idea of talking to devices has been around as long as we computers have. That is why it seems very natural to speak with smart processor-based receivers. The WWW–originally not voice-based–by comparison is very recent, but has swiftly become a competing and successful communications channel.

Convergence of broadcast and the WWW is now delivering the benefits of current web technologies to the classic network medium radio as well. Developers are enabled to create applications that can be accessed via hybrid broadband-broadcasting receivers, we call them *audio managers*. Furthermore, web technologies allow humans to interact with these applications via speech as well. Therefore, the W3C Speech Interface Framework is a suite of markup specifications aimed at realising several goals. It covers voice dialogues, speech recognition and synthesis, and other requirements for interactive voice response applications such as broadcasting devices including radios. These implementations are especially used by people with hearing or speaking impairments or by drivers (*hands-free, eyes-free*).

The technology to make computers recognise voices and generate speech in response has been developing for decades, and it is still imperfect. At this time, we are just becoming able to use speech as a means of limited communication between people and machines. The proliferation of embedded systems in automobiles, in telecommunication devices, and consumer electronic products has brought the previously narrow discipline of speech processing into everyday life. Speech-based features and interfaces are finding their way slow but steady into broadcasting receivers (DAB, DVB) as well. The limiting factors for such applications are fewer and fewer hardware costs, but rather the effort for the dialogue design, especially for functional modifications. Currently dialogue development for embedded systems is done by using programming languages with a considerably overhead when rearrangements are necessary. And typical, speech control systems are characterised by strong limited resources.

We present a speech controlled broadcasting receiver based on the standardised VoiceXML for embedded systems. The approach aims at reducing the resource demands that traditionally are associated with VoiceXML. The benefit of our approach is the opportunity to apply existing tools and environments for flexible dialogue design and a broader offer concerning the functionality of the media devices. Furthermore, the process of development can be considerably accelerated. Additional, there exists no common set of commands and functions for voice control of audio functionalities. This paper will introduce ideas on how to extend these audio functionalities with new concepts of usage.

This paper gives an overview of the project with a focus on a prototypical VoiceXML platform for nearly embedded systems. It is organised as follows. The next section describes related work. Then we show fundamentals such as current and prospective speech-activable radio functions for information access and a survey of data sources in a digital radio. In the following section we explain the concept and implementation of the developed prototype with some tentative results. Finally we compare shortly VoiceXML with other environments.

STATE OF THE ART

Speech-dialogue systems (SDS) are already developed since the 1970ies and are incorporated into more and more devices with the constraint that no security relevant use cases are affected. Speech applications for media devices are available with only few receiver and with several car manufacturers. A simple DAB receiver was able to announce verbally names of radio stations, the time and running text on the basis of stored spoken words and alphabetic characters which admittedly leads to spelling in relation with running texts [1].

In cars are offered as general services mainly telephone, navigation and car comfort functionalities, see Figure 1 (above). For nowadays wide spread car navigation systems exist numerous solutions already for voice communication. Radio-based audio services is the second group of functions activated by speech control, see Figure 1 (middle). Voice control of the entertainment system is standard for high-class automobiles. Now users expect on-demand access to dynamic network services and real-time location-based services such as weather, traffic, and generic information. In previous contributions we reported on the usage of radio-supported information services provided by digital radio [2] [3], see Figure 1 (below). Other speech accessible services are in preparation, offering internet access, e-mailing, and business finder in combination with location-based services on board of vehicles [4].

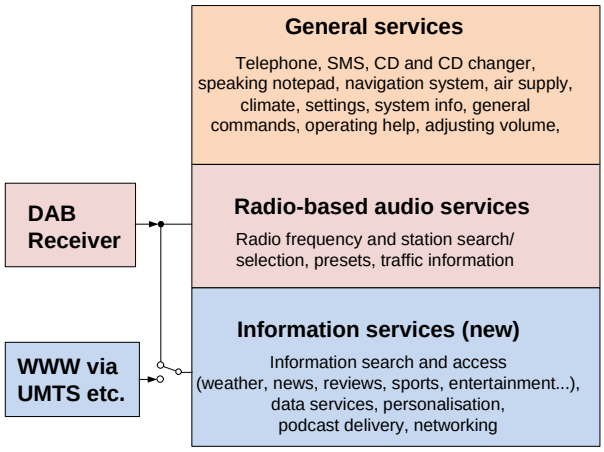


Figure 1: Voice-based audio functionalities for cars

Hence the domain of AV-technology with its inherent information and entertainment orientation is well suited for appliances of speech-dialogue systems. First interactive speech-enabled TV-EPGs were developed [5] [6]. Nevertheless, direct selection of audio content, for example tracks or special news, is scarcely offered by speech in a mobile environment [7]. Resource-related problems of speech-based communication with mobile devices in rough environments are reported on in [8]. Bühler and Hamerich exemplified the needs and conditions of plug-ins necessary for a complete end-to-end speech dialogue system for embedded applications [9]. Zaykovskiy [10] presented certain challenges of automatic speech recognition (ASR) in mobile devices concerning the hardware conditions, here slightly extended:

- limited available storage volume,
- small slow cache and respectively RAM memory,
- low processor clock-frequency,
- no hardware-based floating point arithmetic,
- many signal processing algorithms are not allowed,
- often no access to the operating system,
- cheap microphones and heavy distortions,
- high energy consumption during algorithm execution etc.

Schmitt et al. reported on speech techniques suitable especially for mobile devices with different approaches for providing ASR technology to mobile users. They analysed three principal system architectures with respect to the employment of a wireless communication: embedded speech recognition systems, network speech recognition and distributed speech recognition [11].

SURVEY OF VOICE COMMANDS FOR RADIOS

Table 1 shows the available command set for four types of automobiles with a voice-activated radio and the prototype. The differences in function of the car manufacturers are marginal, but the linguistic terms (commands) differ considerably.

Table 1: Voice-activated functions for car radios

(x available, + announced)

Function	Audi	BMW	Ford	Mercedes	Prototype
Radio on	x	x	x	x	x
Radio off	x	x	x	x	x
Volume up	x	x	x	x	x
Volume down	x	x	x	x	x
Input frequency	x	x	x	x	x
Input station	x	x	x	x	x
Edit station list	x	x	x	x	x
Traffic news	x	x		x	x
Date, time	+	x			x
Programme type	+				x
Title, item, author etc.	+				x
Weather report	+				x
Press review					x
Generic information					x

It is not simple to draft what an appropriate speech interface for broadcasting devices would look like, nor what users would expect from it [12]. Therefore, a questionnaire was set up by us in order to gather user requirements and expectations into several categories. Additionally, we analysed and compared several instruction sets of current speech control systems and detected gaps of possible applications. In addition to general commands such as *next*, *previous*, *up*, *down*, *skip* etc. one main feature of our solution is the provision of audible topics such as traffic news, weather report or press reviews taken directly from DAB data services; traffic news do not yet come from TPEG services. Titles and authors from programme items such as commentaries, features, music etc. are audible messages converted from Dynamic Label information, see Table 1. Our prototype enlarges the command set with information functions as presented in [2] [3]. These functions are until now complete radio-related information, there is no fundamental obstacle to integrate web-based information sources in the system. But for mobile car-integrated devices with a wireless connection at high velocities are to solve amongst others serious problems of reliability. In the future the system will decide the most efficient way for information reception founded on location-based information. Only if free radio services are not available, the system should switch to expensive wireless connected web services.

AVAILABLE DATA SOURCES IN DIGITAL RADIOS

For a speech-controlled digital radio all data sources should deployed for an improved information offer using an internal data base. Indeed, the digital radio development started basically focused on audio services, but data options have been introduced into the system right from the beginning as well. Data transmissions can either be a service component of an audio service or if there is no relation to a programme service, a data service can be broadcast as a standalone service.

Two general types of information sources are available:

(a) The conventional part of the multiplex signal is the well-known audible signal. This audible information stream includes breaking news, headlines, educational and cultural items, current affairs, discussions etc. Internet audio and podcast files are applicable as well for hybrid applications.

(b) The digital audio broadcasting system is famous for its rich set of data services. They are a significant and versatile part of the broadcasting system. Numerous information is simultaneously transmitted as text messages in a DAB receiver environment, therefore it is obvious to apply text-to-speech (TTS) and ASR solutions for a communication by speech assistance, because displays are often not appropriate. Broadcast Websites (BWS) may contain multifaceted news, press reviews etc. Internet-based text information is applicable for hybrid solutions as well.

Futhermore, other sources of information are Electronic Programme Guide (EPG), Intellitext™, Journaline™, and Dynamic Label Plus (DL). Compared with (a) these information sources are more reliable with respect to structure and content, but less detailed and not always available. Unfortunately broadcasters do often never or not consistently broadcast these useful information.

One aim of the development was to monitor as many as possible information channels simultaneously in the background imperceptible for the user. The scanning of a great number of channels was not a serious matter compared to time consuming processing tasks. Hence, it was indispensable to establish a hierarchical sequence of sources depending on reliability, quality, and convenience of the different information sources. Figure 2 presents a comprehensive view of available information sources in an advanced digital receiver concept in form of a manifold mix of audio and data services.

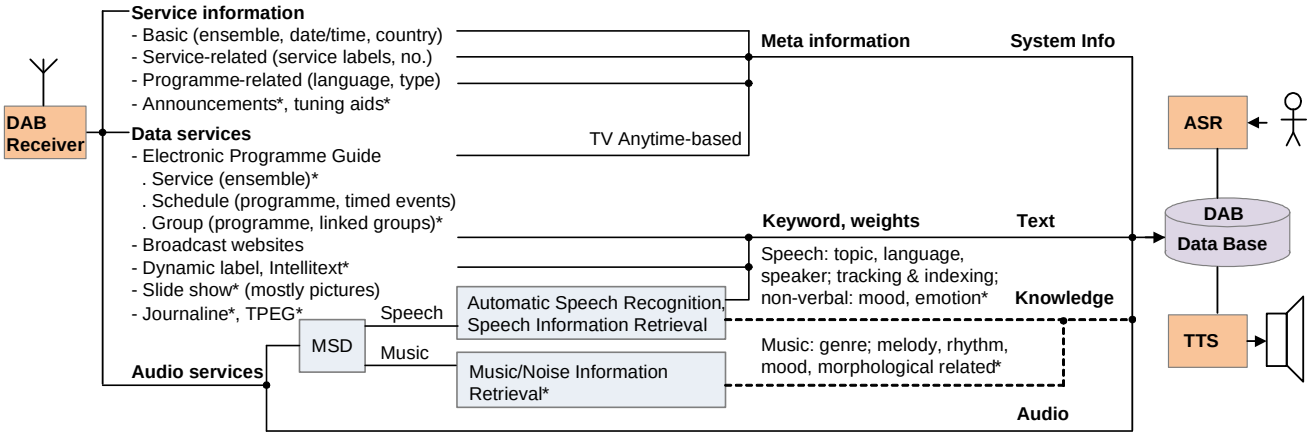


Figure 2: Available data sources for an information and knowledge base for DAB receiver (*not yet in use)

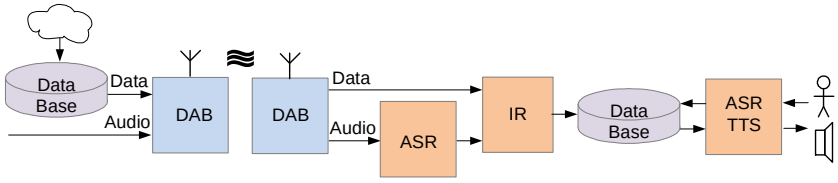
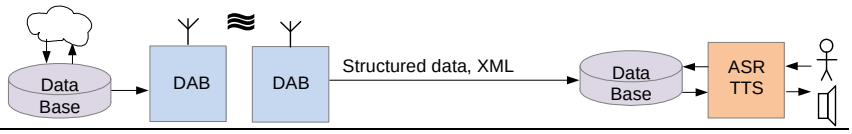
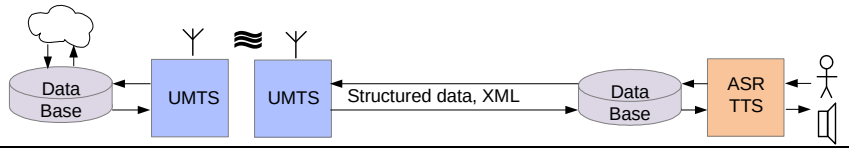
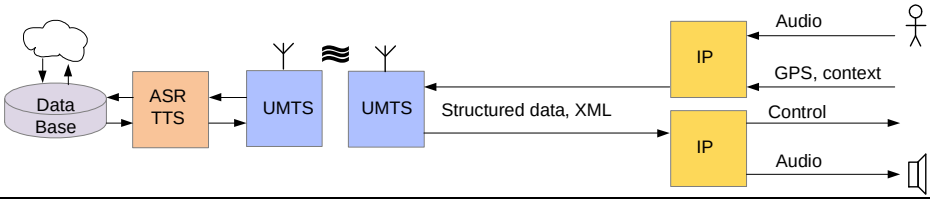
SURVEY OF ARCHITECTURES

The transport mechanism for digital broadcast systems is the so-called carousel, applied for repeatedly delivering data in a continuous cycle by pushing them, see Table 2, lines 1 and 2. The repetition is a simple method of error prevention and allows a random access at an unpredictable time after one cycle period. Thus, the process of content selection is accomplished by a local selection in a database which has been charged by a huge amount of transmitted broadcast content. The information preferences are not defined in advance but at any time of demand.

In order to access the maximum of information of a DAB system we used data and voice information as well [2]. The problem arises that processing load is significant for discrimination and retrieval processes. A second approach is based on sending pre-structured data for a combined voice and a graphic user interface as well [13].

Information access differs considerably dependent on the chosen communication principle: unidirectional broadcasting versus bidirectional network access. The advantages of mobile broadcasting devices such as ubiquity and information availability at no charge can compensate certain disadvantages such as the absence of the return channel. However, the pinpoint content selection as known from Internet services is missing.

Table 2: Architectural concepts of voice-enabled wireless information systems
(ASR: Automatic Speech Recognition, IR: Information Retrieval, IP: Internet Protocol, TTS: Text-to-Speech)

Architecture	Schema of sender-receiver respective server-client relation
Unidirectional DAB connection (data carousel) with information retrieval (IR) for data and audio on the client	
Unidirectional DAB data supply (data carousel) with structured data with speech processing on the client	
Bidirectional phone-based connection with structured data and speech processing on the client	
Bidirectional phone-based connection with speech processing on the server and IP-based structured data transfer	

Therefore several manufacturers try to apply bidirectional mobile channels via UMTS or LTE for information supply with the advantage of high topicality and diversity of available information–with the disadvantage of higher costs for the user. However, an absolutely free access with a voice interface is actually not yet a reality. First systems use a structure of line 3 in Table 2 with a resource consuming voice interface on the client side.

If the signal processing tasks reside on the server side [10], then more sophisticated queries are possible: free semantic search, context dependent answers, location based services etc., see Table 2 line 4. It will be beneficial and convenient to deliver all data and audio signals by IP, for example by [17].

The combination of both principles in a single functional unit will lead to a Hybrid Broadcast Broadband Radio (HbbRadio), see Figure 1. Our prototype uses the classic carousel-based approach for information delivery, trials with speech-prepared data are in progress.

CONCEPT

The W3C Speech Interface Framework is a suite of markup specifications aimed at realising several goals. The most important part is VoiceXML, an established

standard dialogue description language especially for voice applications in the telephony industry. The members of the first working group in the late 1990ies came from this environment, see Figure 3. The current VoiceXML 2.1 is the latest version, discussions for improving the language are ongoing and should be finished temporarily with version 3.0 in 2011 [13].

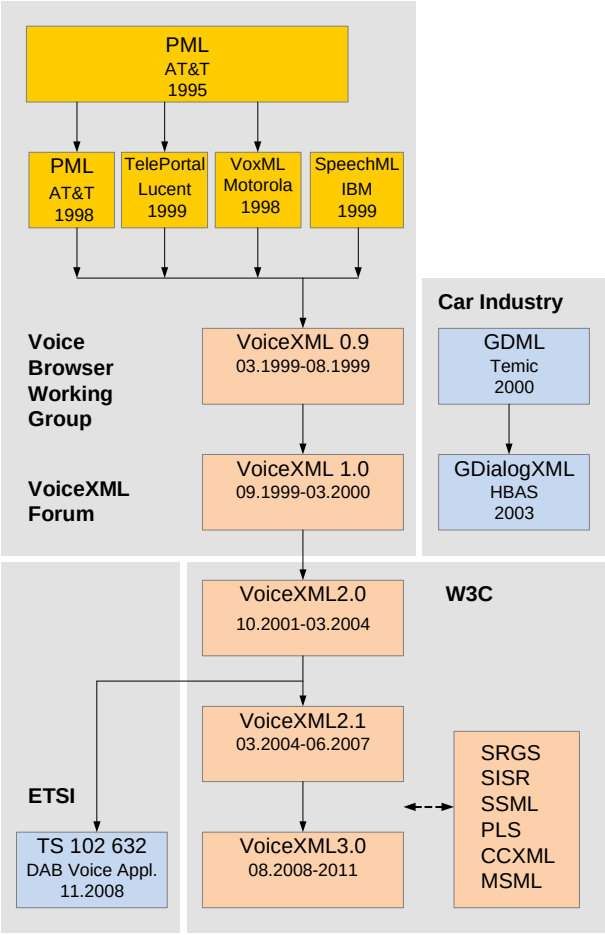


Figure 3: Genealogy of XML-based speech systems

Only a while later after the initial standard VoiceXML 0.9, first approaches of a markup language for speech-based automotive applications (GDML) appeared in 2000 [14]. The European Telecommunications Standards Institute (ETSI) published a technical specification for DAB voice applications based on a markup language in 2008 [15], see Figure 3 for a genealogy of XML-based speech systems for general and radio specific (light blue boxes) applications.

VoiceXML covers voice dialogues, speech recognition and synthesis, and other requirements for interactive voice response applications such as broadcasting devices including radios. VoiceXML is a programming language for scripting voice interactions between a computer and a person. It holds the promise of making voice interfaces easy to build, deploy, and use as other interfaces. VoiceXML is designed for creating audio dialogues that feature synthesised speech, digitised audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations. The basic element of interaction is a spoken dialogue in which the computer produces spoken promts to evoke spoken answers from the user. VoiceXML prompts may be recorded or generated using TTS synthesis. Spoken user responses are processed using speech recognition and grammars defined in the VoiceXML programme. But VoiceXML is not the universal tool for dialogue design, the resource requirements of its interpretation and a certain complexity have so far limited the distribution basically to application areas other than telephony-based services.

Usual entertainment functions have the primary task to control on-board external devices such as radio, CD/DVD, telephone, navigation system etc. They shall satisfy a wish immediately without the need of negotiations and deep specifications. These tasks demand an easy to use, intuitive and flexible user interface to manage devices by speech control and to a lesser extent a real speech dialogue. Such closed systems do not need access data dynamically from communication channels or data bases. For advanced features with the need of a higher degree of refinement and sophistication of information respective knowledge supply is the communication with a data base essential. This dynamic creation of VoiceXML documents makes the processing procedures resource demanding.

The current VoiceXML standard allows two methods to access external resources. The first are CGI scripts, which need a HTTP server. The second method bases on ECMAScript, but requires an ECMAScript interpreter. This leads for both methods to additional requirements for the respective device, because extra memory is needed and the existence of a file system is necessary.

IMPLEMENTATION

The runtime environment was implemented on the base of several free available components connected by own scripts and adaptations. The main tasks respective key requirements are: interpretation of several scripting formats using very small amount of memory, a flexible interface able to communicate with several external components, audio output, preferably a language independent dialogue in German and English as well. All parts of the system were selected to obtain a diminished consumption of resources such as memory and processor load, but reserves remain. Figure 4 shows the components of the system.

Central part of the solution is a JVoiceXML interpreter, a free VoiceXML interpreter for Java with an open architecture for custom extensions. JVoiceXML is an implementation of VoiceXML 2.1, the Voice Extensible Markup Language, specified at www.w3.org. Demo implementation platforms are supporting Java APIs such as JSAPI and JTAPI. The interpreter is able to process

several scripts such as SSML, JSGF, SRGS, and VoiceXML as well. The subsystems od TTS and ASR are connected via JSAPI, see Figure 4. For the implementation of a TTS are FreeTTS and MBROLA possible. The ASR can be realised with CMU Sphinx or other programmes. Those free modules have not a very high quality standard but for test purposes they seem to be sufficient. With the help of a wrapper such as TalkingJava or the Chant SpeechKit are other solutions with a SAPI conformity and higher speech quality obtainable.

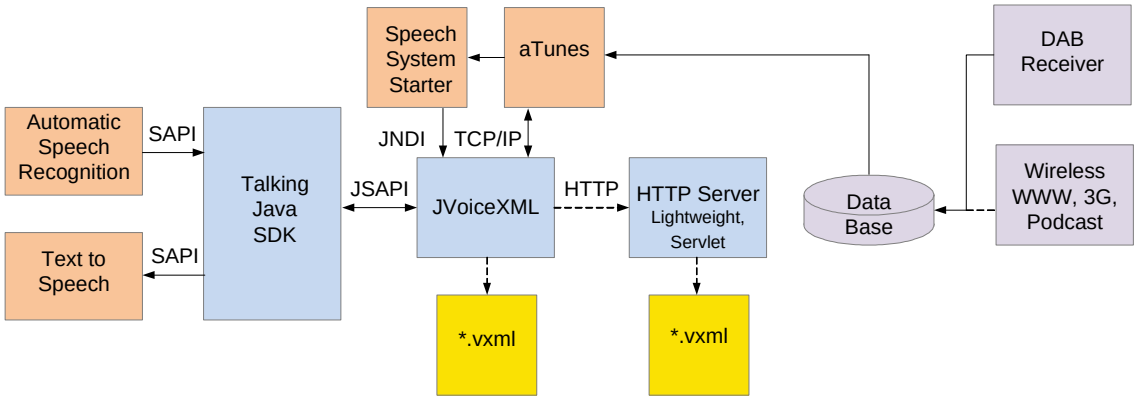


Figure 4: Runtime environment

The VoiceXML documents reside in a HTTP server and are accessed via this protocol, a servlet container is possible as Java counterpart to dynamic technologies such as CGI. The client is connected with the JVoiceXML by using the Java Name and Directory Interface. Each implementation platform can add supplementary libraries of the user. The system is not always stable, in [16] was indicated that the JSAPI interface of Sphinx4 is not yet free of bugs. Therefore we used OS specific speech converters via SAPI additionally as an alternative solution.

Concerning the memory consumption we have to note for FreeTTS and Sphinx4 (TTS and ASR) 19 MB. Embedded speech technologies would allow an optimisation of memory demand of around 2 MB. JVoiceXML and Tomcat require each with a minimum about 10 MB. Further improvements are possible by using lightweight web servers. They have been written as endeavours to create low memory requirements, the sizes of the executable files can be considerable smaller then 1 MB. Thus, at present a VoiceXML system (50…100 MB) for smart phones is not yet realistic but for tiny embedded systems (< 5…10 MB) it is a feasible alternative in the near future. Further studies and optimisations are planned. The DAB receiver is a tried USB-connected DR1 box (Terratec) with a standard laptop (Core2Duo; 1.8 GHz; 4 GB DDR3-RAM; 500 GB HDD) and operates quite mobile.

FIRST EXPERIENCES

The main constraint of our approach remains the presumption of a total predictability of all dialogue steps concerning a communication act. This design hinders free conversational or mixed-initiative dialogues. One limitation is that markup languages do not allow dynamic applications and loop handling in the easy way. However, this is not a special problem of the chosen platfom. With form filling a higher degree of design freedom is obtainable. The risks are growing with complexity, the application of a dialogue manager is recommend. Furthermore, there are not yet well suited tools for multimodal dialogue support (mixed voice and graphical interface). But for hands- and eyes-free environments this represents only a minor problem.

Although this technology points out efficient ways to improved user interfaces for audio managers, it still has several limitations:

- the complexity of dialogue structures is very limited,
- the possibility of dynamic variations, shortcuts and information requests is restricted,
- the resource consumption is not yet minimised for tiny and weak environments.

VoiceXML is an easy understandable but light language, because more significant properties have to be provided by using CGI script or ECMAScript. Some functionalities are underdeveloped such as several types of loops, lists and array handling. It would be good to extend VoiceXML to allow more complex and flexible dialogues, which would appear to be more natural. Especially needed is the introduction of loops, which would allow more complex and flexible dialogues. The main advantage is a well-structured software design process with flexibility for application development, reusable software solutions and limited efforts for initial training.

CONCLUSIONS AND FURTHER WORK

The proposed solution consists of a structured dialogue development system and is applicable in many situations for digital broadcasting receivers with their abundant data services. This approach opens a new perspective for VoiceXML since it then could not only be used for describing traditional telephone-based applications but additionally embedded systems such as broadcast receivers, mobile phones, organisers and media players and supports reusable software development.

For further developments we would implement and evaluate mixed initiative dialogues because they are becoming more reliable and more attractive for the user. Speech systems should integrate more shortcuts for frequent users, therefore more dynamic dialogues are appropriate. The main problems remain from ASR while mapping the spoken inputs to the VoiceXML grammar. Improving full sentence recognition must be a major aim for the next years [3]. Questions of mispronunciation of foreign languages, abbrevations, names, accentuations are to solve in the near future.

The work presented in this paper opens an interesting perspective on a number of future directions. First of all, the coverage of VoiceXML elements has to be extended to new classes of applications such as broadcast-based information systems.

ABBREVIATIONS

ASR	Automatic Speech Recognition
CCXML	Call Control XML
ECMA	European Computer Manufacturers Association
DTMF	Dual-Tone Multi-Frequency signaling
GDML	Generic Dialog Modelling Language
IP	Internet Protocol
IR	Information Retrieval
JNDI	Java Naming and Directory Interface
JSAPI	Java Speech Application Interface
JSML	Java Speech Markup Language
LTE	Long Term Evolution
ML	Markup Language
MSD	Music Speech Discrimination
MSML	Media Server Markup Language
PLS	Pronunciation Lexicon Specification
PML	Phone Markup Language
SCXML	State Chart XML
SISR	Semantic Interpretation for Speech Recognition
SRGS	Speech Recognition Grammar Specification
SSML	Speech Synthesis Markup Language
TPEG	Transport Protocol Experts Group
TTS	Text-to-Speeech
UMTS	Universal Mobile Telecommunications System
VoiceXML	Voice Extensible Markup Language
VXI	VoiceXML Interpreter
XML	Extensible Markup Language

REFERENCES

[1] Sonus-1XT (2006, discontinued product) http://www.pure.com/support/Manuals/VL-60751/SONUS-1_XT_Owners_Manual_-_488KB_PDF.pdf (acc. 2010/07/15)

[2] G. Schatter; A. Eiselt; B. Zeller: A multichannel monitoring Digital Radio DAB utilising a memory function and verbal queries to search for audio and data content. IEEE Transactions on Consumer Electronics 54(2008)3, August, p. 1082-1090.

[3] G. Schatter; A. Eiselt; B. Zeller: Digital Radio as an Adaptive Search Engine. Verbal Communication with a Digital Audio Broadcasting Receiver. Proceedings Int. Conf. on Signal Processing and Multimedia Applications SIGMAP2009. July 7-10, 2009, Milan, Italy, p. 157-164.

[4] L. Mowatt: Nuance Voice Control for Automotive. Enabling a Single Consistent Voice User Interface to Connected Car Services. Nuance Communications February 2009. http://www.nuance.com/industries/automotive/whitepapers/AutomotiveConnectedCarWP.pdf (acc. 2010/07/15)

[5] H. Kim; E. Hwang: VoiceEPG: Speech Interface for Electronic Program Guide. In: Proc. of the IASTED Conf. on Internet and Multimedia Systems and Applications. Honolulu, Hawai, August 14-16, 2003.

[6] H. Shinjo et. al.: Intelligent User Interface based on Multimodal Dialog Control for Audio-visual systems. Hitachi Review March 2006.

[7] Y. Wang; St. Hamerich; M. Hennecke; V. Schubert: Speech-controlled Media File Selection on Embedded Systems. Proc. of the 6th SIGdial Workshop on Discourse and Dialogue, 2005. http://www.sigdial.org/workshops/workshop6/proceedings/pdf/51-mp3-demo.pdf (acc. 2010/07/15)

[8] N. Sawhnwy; C. Schmandt: Nomadic Radio: Speech and Audio Interaction for Contextual Messaging in Nomadic Environments. ACM Transactions on Computer-Human Interaction 7(2000)3, September 2000, p. 353-383.

[9] D. Bühler; St. Hamerich: Towards VoiceXML Compilation for Portable Embedded Applications in Ubiquitous Environments. Proc. of the European Conf. on Speech Communication and Technology, Lisbon, Portugal, 2005. p. 3397-3400. http://www.martens-hamerich.de/stefan/pub/vxml-euro05.pdf (acc. 2010/07/15)

[10] D. Zaykovskiy: Survey of the Speech Recognition Techniques for Mobile Devices. 11th Int. Conf. on Speech and Computer SPECOM'2006. St. Petersburg, June 2006.

[11] A. Schmitt; D. Zaykovskiy; W. Minker: Speech Recognition for Mobile Devices. International Journal of Speech Technology 11(2009)2, p. 63-72.

[12] Maix: Automotive Voice UI Usability Study User Survey. Attitude, Experience, Motivation and Key Issues. Market Research & Consulting Maix Aachen, 2009. http://www.nuance.com/industries/automotive/whitepapers/AutomotiveUsabilityStudy_Final.pdf (acc. 2010/07/15)

[13] Voice Extensible Markup Language (VoiceXML) 3.0. W3C Working Draft 17 June 2010. http://www.w3.org/TR/voicexml30/ (acc. 2010/07/15)

[14] M. E. Hennecke; G. Hanrieder: Easy Configuration of Natural Language Understanding Systems. Proceedings Voice Operated Telecom Services. Coopération européenne dans le domaine de la recherche scientifique et technique (COST) 249. Gent, Belgium, 2000, p. 87-90.

[15] European Telecommunications Standards Institute: Digital Audio Broadcasting (DAB); Technical Specification Voice Applications. ETSI TS 102 632 V1.1.1 (2008-11)

[16] D. Schnelle-Walka: JVoiceXML - The Open Source VoiceXML Interpreter, 2010. http://jvoicexml.sourceforge.net/ (acc. 2010/07/15)

[17] European Broadcasting Union: Audio contribution over IP. Requirements for interoperability. Technical spec. 3326, revision 3. Geneva, 2008. http://tech.ebu.ch/docs/tech/tech3326.pdf. (acc. 2010/07/15)