STANDARDS-BASED MULTIMODAL INTERFACE FOR MOBILE DEVICES

Carsten Günther, Markus Klehr, Siegfried Kunzmann

EMEA Voice Technology Development – IBM Deutschland Entwicklung GmbH
Gottlieb-Daimler-Str.12, 68165 Mannheim, Germany
gcarsten@de.ibm.com

Abstract: To support the interoperability of voice driven applications across different devices a standardized programming model is necessary. This paper outlines a new standard, XHTML + Voice (X+V), for the design of multimodal interfaces for the evolving variety of mobile devices. This new standard is based on standards for the well established XHTML-based visual interface design and the evolving VoiceXML-based voice interface design.

1 Introduction

Given the growing usage of mobile devices it is increasingly becoming common practice to access information and transaction services via mobile phones or Personal/Mobile Digital Assistants (PDA/MDA) from anywhere at anytime. These devices support different input/output channels (voice only, visual only, voice&visual) and as these devices become smaller and more complex the speech interface is a very important part of the overall system design.

To support the interoperability of voice driven applications across different devices a standardized programming model is necessary. This paper outlines a new standard XHTML + Voice (X+V) [1] for the design of multimodal interfaces for the evolving variety of mobile devices. This new standard brings together both sides, the well established XHTML-based visual interface design and the evolving VoiceXML-based voice interface design.

The standard supports speech input, speech recording, pre-recorded prompt playback, and prompt synthesis. Currently the recognition process is based on grammars and their complexity and efficiency is only limited by the available system resources and network bandwidth. This standard gives an application writer a single authoring environment for the design of a unique application supporting different input/output modes simultaneously. The X+V standard enables the interoperability of multimodal applications, which means that the same code can run on any device with a multimodal browser installed on it. This standard allows a ubiquitous access to web-based information and transaction services.

We will show the benefit of multimodal applications compared to multichannel ones. Whereas a multichannel application allows access to enterprise data from multiple channels (PC, smartphones, MDA) by only one channel at a time, a real multimodal application supports the access via different channels simultaneously within one session. The necessary synchronization of input/output modalities is one of the key elements described in the X+V standard. This difficult task is performed using the XML event handling and introduces some specific X+V elements in order to ensure that spoken and typed input lead to the same result, i.e. the next dialog step. The standard allows the specification of a number of speech recognizer properties such as confidence level support and speed vs. accuracy optimization. To cope with the problems of background noise it is possible to adjust dynamically the sensitivity of the speech recognition engine.

The system architecture implementing this new standard will be sketched in this paper and an application running on an MDA will be demonstrated. The implementation of the standard relies on a client-server infrastructure, although the complete application can run on a client

device only, e.g. a number dialing application on smartphones. The data backend and application server reside on the server, and it is there that the code (incl. grammars, exception pronunciation dictionaries) for the multimodal client is generated. On the client a multimodal browser is running accessing locally installed embedded speech recognition and speech synthesis engines. An overview on IBM's embedded recognition and synthesis engines will be given.

2 XHTML + Voice

2.1 Overview

The XHTML+Voice standard (short X+V) is a combination of the commonly used XHTML standard [2,3], which is an XML based definition of HTML and the VoiceXML standard [4,5], which is an XML based mark-up language for the creation of voice applications.

A multimodal application requires two well designed and synchronized user interfaces: a graphical and a voice user interface. In the X+V standard, the XHTML part is responsible for the definition of the GUI (graphical user interface). Existing HTML documents could in most cases easily be transformed into XHTML compliant documents. For the definition of the VUI (voice user interface) the VoiceXML standard is used This standard supports: recognition and recording of spoken input, speech synthesis, playback of pre-recorded prompts, event handling, dialog flow design, and backend access mechanisms. To support and synchronize these multiple modes of interaction, X+V is using the event handling which is provided by the build-in ECMA script engine of the XHTML browser.

Fig. 1 shows how the content of a single application can be rendered for the access with different devices supporting different user modalities. Web Servers host the Web application. The view of such an application can be rendered towards different output devices. Pure XHTML code can be sent to standard computing devices or handhelds with a running visual browser. Pure VoiceXML code can be interpreted by a VoiceXML Browser running on a Voice Server. There, voice is synthesized or recognized transmitted via a phone channel (landline, cellular). And now, in addition, a device with multimodal capabilities can interpret X+V documents. The multimodal browser interprets the X+V documents and synchronizes between voice, pen or keyboard based user input events.

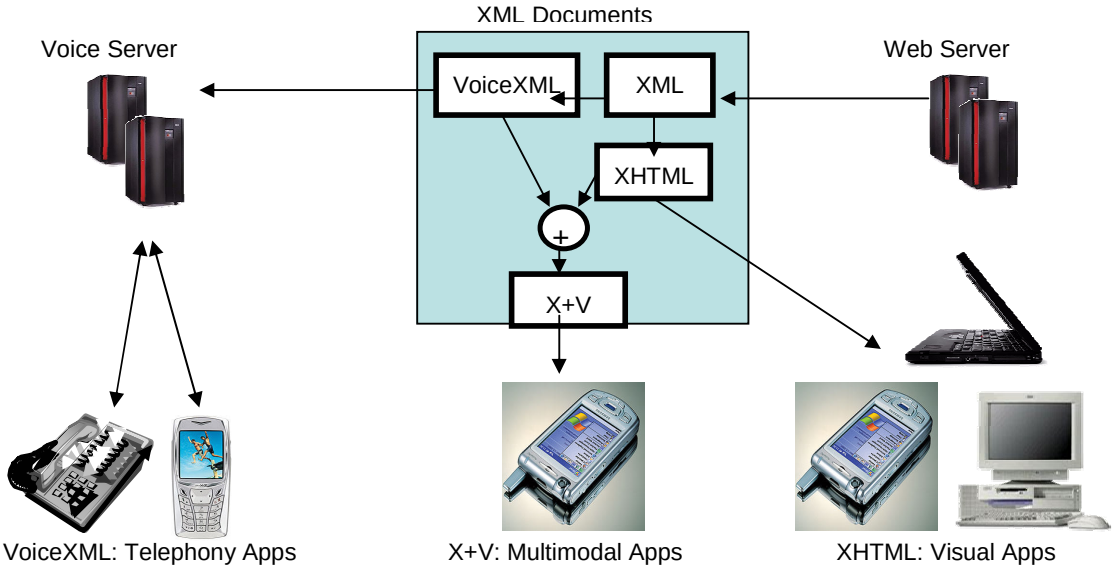

Fig. 1 – Device dependend access of a single application via VoiceXML, X+V, XHTML documents

There are some general design principles defining the relationship between the XHTML and the VoiceXML language part of an X+V document. The host language is XHTML, meaning that X+V extends the basic XHTML language by a modularised subset of VoiceXML. An embedded speech dialog is defined as a VoiceXML form and its processing follows the VoiceXML form interpretation algorithm. A certain user action like an interaction with an XHTML element or an document event (like load page) can generate a XML event that activates a speech dialog form.

In the following sections we will demonstrate the step-by-step integration of multiple interaction modes into an X+V document.

2.2 Plain VoiceXML document

In the following example (Fig. 2) a simple VoiceXML document is shown where the user is asked for the language in which the application is to be started. Based on the given VoiceXML document the following dialogue could be realized.

C: "Welcome to the multimodal stock information system, please choose your language."
User: „German."

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml xmlns="http://www.w3.org/2001/vxml" version="2.0" xml:lang="en-US">
<form id="welcome">
<field name="language">
<grammar>
      <![CDATA[
            #JSGF V1.0 iso-8859-1;
            grammar links;
            public <links> =
            english {$="link1"} | german {$="link2"}|
            deutsch {$="link2"} | englisch {$="link1"} ; ]]>
</grammar>
<prompt>
      Welcome to the multimodal stock information system,
      please choose your language.
</prompt>
<filled>
      <submit next="index.jsp" namelist="language"/>
</filled>
</field>
</form>
</vxml>
```

Fig. 2 – VoiceXML document for the "Welcome" dialog

This document defines a dialog step (<form>-tag) with an initial system prompt (<prompt>-tag), a user-input prescribed by a grammar (<grammar>-tag) and the jump to the next dialog step after a correct user input (<filled>-tag, <submit>-tag). The grammar is here defined in the JSGF format but also the other standard grammar formats are supported (SRGS – Speech Recognition Grammar Specification [5]).

2.3 Plain XHTML document

In accordance to this introductory spoken dialog the visual application start-up is designed. The XHTML version of the intro document shows an application logo with two icons to select the preferred language (Fig. 3). Fig. 4 presents the corresponding XHTML code.

Fig. 3 – Application's start page defined in XHTML format

```
<?xml version="1.0" encoding="iso-8859-1"?>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>Welcome to the Multimodal Stock Information System</title>
</head>
<body id="main_body">
<table width="100%">
  <tr>
    <td width="100%"><h2 align="center">
      <img border="0" src="./images/startbild.jpg"/> </h2> </td>
  </tr>
</table>
<center>
<table width="100%">
      <tr>
      <td width="50%"> <center>
                  <a href="./index.jsm?lang=en-US">
                  <img border="0" src="./images/usa.gif" />
      </a> </center> </td>
      <td width="50%"> <center>
                  <a href="./index.jsm?lang=de-DE">
                  <img border="0" src="./images/brd.gif" />
      </a> </center> </td>
      </tr>
      <tr>
      <td>  <center>
        <a id="link1" href="./index.jsm?lang=en-US"> English/Englisch
        </a> </center> </td>
      <td> <center>
        <a id="link2" href="./index.jsm?lang=de-DE"> German/Deutsch
        </a> </center> </td>
      </tr>
</table> </center>
</body>
</html>
```

Fig. 4 – XHTML document defining the application's start page

2.4 Combined XHTML+Voice document

The given XTML document can easily be speech enabled by integrating the VoiceXML code resulting in a X+V document (Fig. 5, VoiceXML originated parts are shaded).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ev="http://www.w3.org/2001/xml-events"
      xmlns:vxml="http://www.w3.org/2001/vxml"
      xml:lang="de-DE">
<head>
<title>Welcome to the Multimodal Stock Information System</title>

<vxml:form id="welcome">
<vxml:field name="language">
<vxml:grammar>
      <![CDATA[
            #JSGF V1.0 iso-8859-1;
            grammar links;
            public <links> =
            english {$="link1"} | german {$="link2"} |
            deutsch {$="link2"} | englisch {$="link1"} ; ]]>
</vxml:grammar>
<vxml:prompt>
      Welcome to the multimodal stock information system,
      please choose your language.
</vxml:prompt>
<vxml:filled>
      <vxml:assign name="selectLink"
            expr="document.getElementById(language).focus()" />
</vxml:filled>
</vxml:field>
</vxml:form>

</head>

<body id="main_body" ev:event="load" ev:handler="#welcome">
<table width="100%">
  <tr>
    <td width="100%"><h2 align="center">
            <img border="0" src="./images/startbild.jpg"/></h2> </td>
  </tr>
</table>
<center>
<table width="100%">
      <tr>
      <td width="50%"> <center>
            <a href="./index.jsm?lang=en-US">
            <img border="0" src="./images/usa.gif" />
            </a> </center> </td>
      <td width="50%"> <center>
            <a href="./index.jsm?lang=de-DE">
            <img border="0" src="./images/brd.gif" />
      </a> </center> </td>
      </tr>
    <tr>
    <td> <center>
        <a id="link1" href="./index.jsm?lang=en-US"> English/Englisch
        </a> </center> </td>
```

```
    <td> <center>
        <a id="link2" href="./index.jsm?lang=de-DE"> German/Deutsch
        </a> </center> </td>
  </tr>
</table>
</center>
</body>
</html>
```

Fig. 5 – XHTML + Voice (X+V) document defining the application's start page

For the <html>-element the additional namespace attributes need to be defined to support VoiceXML code and XML event handling. Additionally, each VoiceXML element is prefixed with **vxml**. To activate the initial voice dialog (VoiceXML form *welcome*) at the start-up of the document, this form is defined as an event handler for the body of the XHTML document:

```
<body id="main_body" ev:event="load" ev:handler="#welcome">
```

After loading the X+V document into the Multimodal Browser the XHTML content is rendered on the device's screen and speech grammars are activated. If a user uses a pen the corresponding link (e.g. `id="link1" href="./index.jsm?lang=en-US"`) is opened. A new X+V document will be loaded and a new speech grammar will be enabled. But the user has also the option to use speech for selecting the link. The *welcome*-form will be processed and the Java script function `document.getElementById(language).focus()` is executed to trigger the same event as if the link was selected by the pen. For more sophisticated tasks (like input fields) it is possible to access the XHTML elements from the VoiceXML part of the document and vice versa.

As in an X+V application you have usually to use a push-to-talk button to start the speech recognition. The *noinput* event known from VoiceXML will not be thrown if a user interacts with the browser in a visual mode only.

3 Server and Client Infrastructure

A mobile multimodel application infrastructure can be separated into a server and client side. On the server side the Application Server is handling the application logic and renders the content for the different user interfaces. Content stored in a database (possibly in an XML format) is accessed via Servlets. The rendering of the content into the different XML dialects can be done with a Style Sheet Transformer (XSLT). At that stage the actual X+V documents are generated including the speech grammars to be enabled at the client side by the browser in the embedded speech recognition engine.

The major components at the client side are a Multimodal Browser [7] and the respective Speech Recognition and Speech Synthesis Engines. For the recognition task on the client side, for instance, the IBM Embedded ViaVoice recognition [8] and TTS engines can be used. This technology supports a variety of platforms, operating systems, languages, and sampling rates. System requirements are 25 MIPS at minimum (about 175 MIPS for 100k words) and 490K bytes RAM and 790K bytes ROM.

Fig. 6 shows a complex infrastructure where a single server infrastructure is supporting different kinds of client devices via different XML dialects (WML, HTML, VoiceXML, X+V). The web content is generated once but can be accessed via different interaction mod.

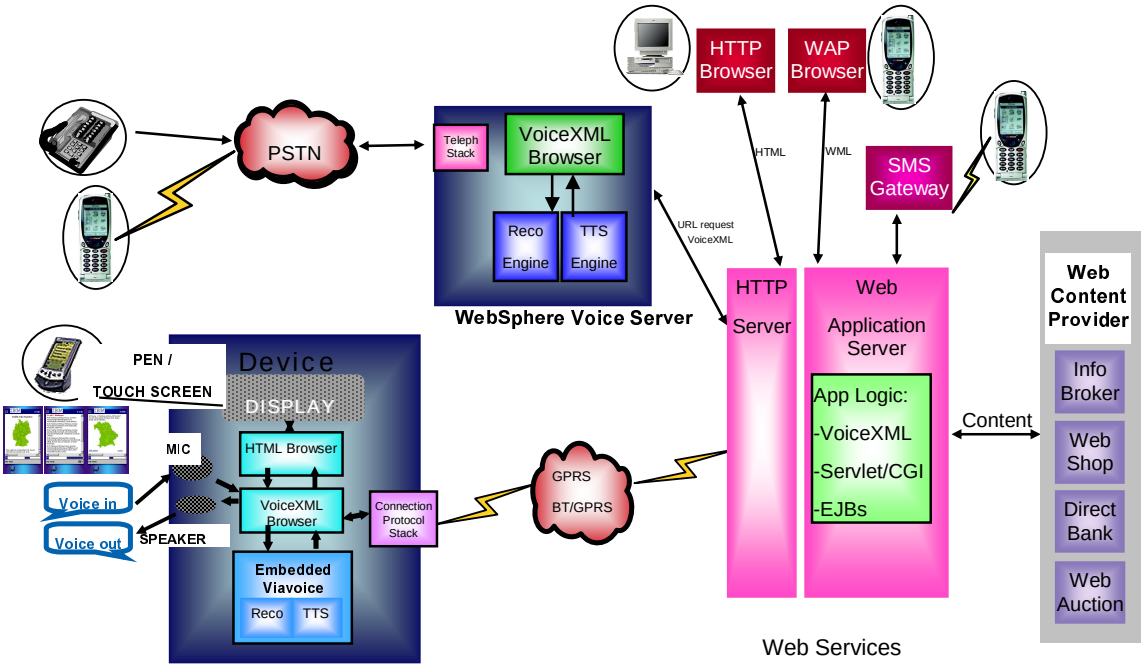

Fig. 6 – Web Application Architecture supporting different XML dialects

4 Sample Application

The sample application which will be shown is a multimodal stock information system. The stock quotes from commonly known stock indexes are stored in the backend database. These stock quotes are regularly being updated to have near real time stock quotes.

At the introduction dialogue the user could choose his preferred language in which the application will continue. In the following dialogue the users have got the opportunity to select the stock index from which they like to get quotes. Currently some European stock indexes, the Dow Jones and the NASDAQ are supported.

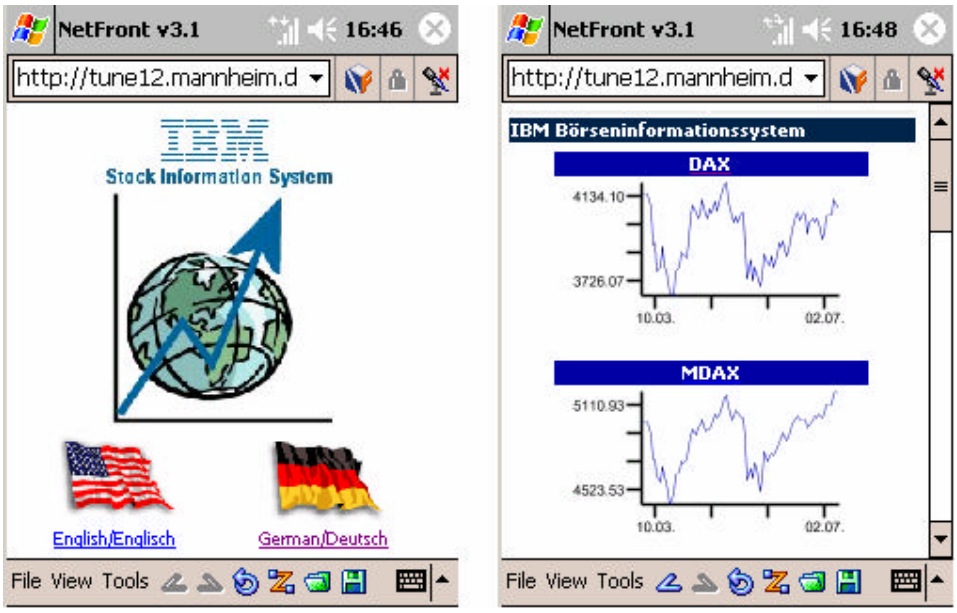

Fig. 7 – Application's views defined in XHTML format but with voice access

After the users have decided for a stock index, they first get a summary. In this summary the current reports are shown, as well as the changes since the day before and the highs and lows of the present day. In this dialogue, the user can now ask for the stock quote of a particular share.

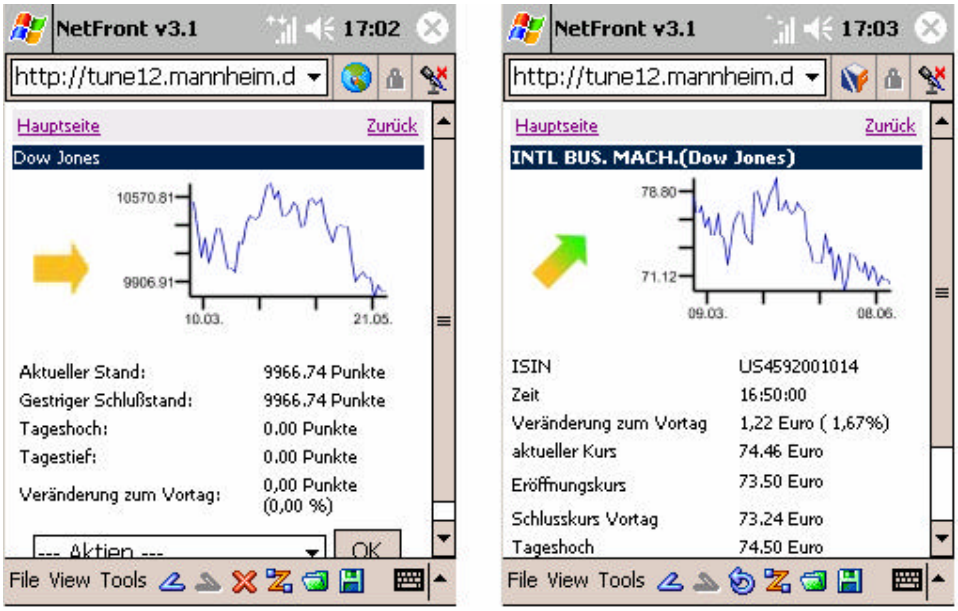

Fig. 7 – Application's detail views defined in XHTML format but with voice access

In the resulting document the user gets a detailed overview of the selected share. Now the user has got the opportunity to ask for a different share from the same stock index, or ask for another stock index.

5 Summary

This paper presented the new X+V programming model for multimodal applications that allows a smooth, standards based transition and enhancement of XHTML based web applications towards multimodal applications by adding spoken interaction. A sample application showed the architectural integration into an existing web infrastructure.

Literatur

[1] XHTML+Voice Profile 1.2: www.voicexml.org/specs/multimodal/x+v/12/spec.html
[2] XHTML1.0: www.w3.org/TR/2000/REC-xhtml1-20000126/
[3] XHTML 1.1 - Modul-based XHTML: www.w3.org/TR/xhtml11/
[4] VoiceXML 2.0: www.w3.org/TR/voicexml20/
[5] Günther, C., Klehr, M.: VoiceXML 2.0. Bonn, mitp-Verlag, 2003.
[6] Speech Recognition Grammar Specification 1.0: www.w3.org/TR/speech-grammar/
[7] IBM Multimodal Homepage: www.ibm.com/software/pervasive/multimodal/
[8] Bergl, V., Fischer, V., Günther, C., Labsky, M., Sedivý, J., Tydlitat, B., Ures, L.: Towards Multi-Modal Interface for Embedded Devices. In: Tagungsband Elektronische Sprachsignalverarbeitung ESSV 2002, Dresden 2002, p. 154-160.