MULTIMODAL MESSAGING USING X+V

A Hands-On Approach to Authoring Multimodal Applications

AUTHORS

SUNIL KUMAR (V-ENABLE)
DR. T.V. RAMAN (IBM)
ATUL SURI (V-ENABLE)
RAFAH A HOSN (IBM)

IBM Business Partner

V-ENABLE
YOUR WAY, ANYWAY™

Preface

This white paper is intended for application developers, content providers and wireless operators interested in developing and deploying MultiModal applications.

The purpose of this white paper is to educate application developers on the merits on using X+V to build multimodal applications. A detailed, hands-on example is provided to better the understanding of the standard and how it applies to developing a mobile multimodal application.

The white paper is based on V-Enable's current experience with X+V and our expertise with all wireless standards, including WAP, SMS, MMS, VoiceXML, xHTML, AURORA and MRCP.

About the Authors

Sunil Kumar, Director, Research & Development, V-Enable Inc.
Mr. Kumar is responsible for driving X+V standards efforts within W3C. Additionally, Kumar leads the innovative Multimodal technology group aimed towards strengthening the V-Enable patent portfolio. Kumar holds a Masters Degree in Computer Science from University of New Hampshire and a B.S. from University of Delhi.

Dr. T.V. Raman, Architect – Conversational and Multimodal, IBM
T. V. Raman works in IBM Research on multimodal user interfaces and is the author of Auditory User Interfaces. He is the editor of XForms 1.0 specification and is an active participant in a number of W3C working groups including XForms, voice browser and XHTML. He is also the author of Xforms ---XML Powered Web Forms (Addison, Wesley, 2003).

Atul Suri, Director, Strategic Marketing, V-Enable Inc.
Mr. Suri is responsible for driving strategic marketing initiatives and working with partners to develop complete end-to-end multimodal solutions. Suri holds an MBA from UCLA Anderson, MS in Electrical Engineering from Clemson University and a B.S. from Delhi Institute of Technology, India.

Rafah A Hosn, Team Lead – Interaction Middleware and Standards, IBM
Miss Hosn works at IBM research on Web programming models and middleware solutions. She is currently leading the design of the next generation frameworks for multimodal and speech applications. Rafah holds an MS in Computer Science and a BS in Computer Engineering from McGill University, Montreal, CA.

V-Enable Inc.
4250 Executive Square, #200
La Jolla, California 92037
USA

Multimodal Messaging Using X+V

A Hands-On approach to authoring Multimodal applications

The Cry for Multimodality

Numerous studies have highlighted that there is a usability challenge with today's wireless data services. According to one recent study, for each additional click that is imposed in the user interface between the user and the desired information, 50% of mobile users will quit their session[1]. The wide proliferation of mobile devices in the current consumer market and the constant growth of wireless communications are driving the need for a new interaction model that is more suited for mobile access. Yesterday's visual only interfaces no longer satisfy the need of users whose device display keeps getting smaller with every new release.

Wireless users want to send text messages, browse information and services, carry out wireless commerce transactions, and access business applications, all from their mobile device and all with the click of a single button. Effective *mobile* computing is about delivering the information users want; on the device they need it on and in the *way (mode)* they want to access that information. Different situations and job functions require different *modes* to access the same information. Also, depending on the type of mobile application, the mode changes. With more available modes of access, the value of the mobile application increases as it becomes accessible under different circumstances.

Multimodality is the next step in the evolution of data and speech services for wireless carriers. Combining the visual with the spoken is a step towards achieving this requirement and the xHTML+Voice W3C proposal is an example of how spoken interaction can be brought to standard web content. Several recent technological developments have contributed to Multimodality "coming of age". Speech recognition capabilities have significantly improved, both in terms of allowable grammars as well as noise filtering technologies. Text-to-speech capabilities have also seen vast enhancement. New algorithms have contributed to formerly robotic outputs being transformed to near natural voice. In addition, advancements in handset processing power and battery life have made it feasible for robust applications to execute on thin clients.

An emerging growth area in wireless is messaging – which includes email, instant messaging, SMS, MMS, chat, conference and other forms of communications across communities, colleagues and family. This paper will provide hands-on, X+V code snippets to demonstrate building a multimodal interface to wireless messaging, with a focus on a mobile email application, using X+V as the authoring tool.

Multimodal Messaging – A closer look at a multimodal, mobile email solution

The existing visual interface for mobile email, allows for menu-based retrieval of email messages, along with a menu interface for other options and features such as setting preferences, signatures, changing password etc…. With menu based browsing of email, the user has to navigate through the whole list of email messages on a mobile device with limited display screen. On the other hand in a voice only email interface the user has to listen to all email messages in the inbox prior to getting to the message that they really wanted to listen to. Further in voice only

[1] Nuance Speech Report, 2001

access to email, the user may loose the comprehension fairly quickly. Additionally in voice interface, search the email message by sender name, date, company name is cumbersome since the results will come back in voice which will be better comprehended if the search results are displayed back on the screen.

A rich, intuitive, multimodal interface to the existing email solution combines the best features of speech based input and visual output to allow the users multiple interface choices in accessing their inbox and other email options! Regardless of the interface being used - Voice, Browser based (xHTML, cHTML), Java/Symbian/Brew – the experience provides full Read/Compose/Delete/ Search functionality, from any device, on any network in a single session!

Usage Scenarios for Multimodal Email Access

So, what would be the ideal multimodal interface for accessing email on the phone? Clearly, the value of multimodality lies in solving the user pain in the inherent interface. Specifically, reading an email, sorting or searching the inbox to find an email that is of interest, forwarding an email (typing the email address), composing a message are all tasks that are painful. In this paper, we will explore a multimodal interface for searching or sorting the email INBOX to retrieve an email based on user name, domain name or subject.

The email application is built on an X+V browser running on a phone over a network agnostic infrastructure (CDMA, GSM, GPRS, 1xRTT, EDGE). The application interface shows the user the number of emails (25 in this case) sorted in the order received. The user is interested in retrieving from the list of 25 emails, the ones from Mark. Figure 1 shows the richness of a user interface built using multimodal technology.

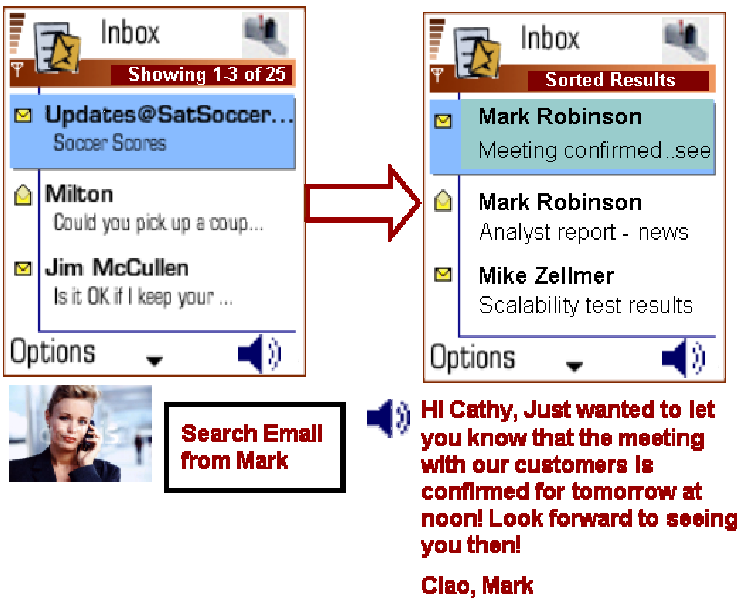

Figure 1: Multimodal User Interface for Email Access

Interaction requirements

The X+V based Multimodal email application running on a mobile phone needs to have a browser running on the phone that can process an X+V document. The browser needs a DOM processor for processing X+V, an audio player and a speech recognition system to process the Voice part of the content. The mobile phone however is a very small device and processing data and speech effectively on a mobile processor is difficult if not impossible. A device only X+V browser will have limited speech recognition and poor text to speech needed for playing back the email in text. In order to provide a true and usable MultiModal application it is required to have a good speech recognition and text to speech systems. With arrival of efficient codec techniques such as EVRC, AMR, DSR etc. which can compress the voice as low as 4.8 kilo bits per second, the voice can easily be processed over the network without experiencing any latency. The MultiModal interaction on a mobile device requires to have a distributed X+V browser where the V part runs on the network and synchronized with the X part running on the phone. In an X+V document the voice dialogue can be initiated with events such as onClick, onFocus etc. However the mobile device may not have those capabilities but instead are provided with soft keys, stylus etc. which are used for navigation. The X+V browser on a mobile device is required to translate the events such as onClick, onFocus etc. into respective device specific keys or stylus. This is certainly not easier than said. The further section will delve into the details of combining voice and visual interaction to present a usable X+V application running on a mobile device.

Authoring Multimodal Messaging Application

This section discusses authoring the above multimodal interaction using X+V for an email application that filters email messages using speech commands. The speech input, in this case, could be either the sender's first or last name.

Authoring xHTML

This section describes the authoring of xHTML which allows users to browse their inbox. An event in the xHTML will trigger the VoiceXML form executing the dialogue which takes voice input that is being used for filtering the emails.

```
<?xml version = "1.0">
<html
xmlns = http://www.w3.org/1999/xhtml
xmlns:vxml=http://www.w3.org/2001/vxml
xmlns:ev=http://www.w3.org/2001/xml-events
xmlns:xv=http://www.voicexml.org/2002/xhtml+voice
>

<head>
      <title>Inbox</title>
</head>
<body>
      <p>this example demonstrates filtering of emails using speech as
input.
      </p>
      <form id="search_inbox" method="post" action = msgsearch.jsp">
            <label id="search"> Please enter name
                  <input name="search_criteria" type="text"/>
            </label>
      </form>
```



```
</body>
</html>
```

Authoring VoiceXML

This section describes the authoring of VoiceXML which allows users to filter their emails by name of the sender. The VoiceXML is generated dynamically from a java script that allows generation of appropriate grammar values of search criteria needed for filtering emails. The grammar is generated by extracting the senders name from the inbox. The grammar can also be generated statically from the address book. This example assumes that there are only a few senders. Once the application has rightly identified the search criteria (name of the person), it uses a script to search into the inbox of the user that returns the list of emails matching the search criteria. The resulted filtered emails are then shown back to the user.

```
<form id="searchInbox">
      <field name="criteria">
            <prompt>
                  Please speak the name of the person
                  <break time="200"/>
                  or company you are looking for.
            </prompt>
            <grammar mode="voice" xml:lang="en-US" version="1.0"
      root="command">
            <rule id="command" scope="public">
                  <ruleref uri="#action"/>
            </rule>
            <rule id="action">
                  <one-of>
                        <item>James</item>
                        <item>David</item>
                        <item>John</item>
                        <item>Michael</item>
                        <item>Karen</item>
                  </one-of>
            </rule>
            </grammar>
            <filled>
                  I think you said <value expr="criteria"/>
            </filled>
      </field>
</form>
```

Combining Voice And Visual Interaction To Enable Multimodality

The Multimodal Messaging provides both voice and visual interfaces to the user for accessing their email. In such a scenario it becomes important to understand how a voice and visual interface can be combined to present a most natural way of accessing information (email in this case).

The X+V is designed to enable a user to interact with voice and visual at the same time. At any particular instant the user may choose either mode for interaction. Or the user may provide input using voice and then want to see results in visual form or vice-versa. That means the input provided using voice can be used in visual and vice-versa. In order to support the mixed initiative

dialogs, which involves both data and voice, interaction the X+V specification provides several methods. The X+V specification provides a <sync> tag which allows the application author to bind the same value to a VoiceXML variable and to an xHTML variable.

In above example of 'filtering the inbox' using a speech input needs a mechanism to pass the 'filtering criteria' to the hosting language (xHTML). The SYNC tag of X+V helps in passing the criteria that is stored in variable 'criteria'. Also a trigger has to be defined in the hosting language (xHTML), which initiates the voice dialogue. The trigger can be initiated through an event such as onFocus, onClick etc. The event is then has to be associated with appropriate voice dialogue (searchInbox) which accepts the spoken input as per the grammar specified. Once the user input is correctly recognized the value of the 'criteria' is available to xHTML in variable 'search_criteria', which then can use a script (msgsearch.jsp) to filter the inbox using the value of 'search_criteria'.

The following is the integrated X+V code that will search the Inbox:

```
<?xml version = "1.0">
<html
xmlns = http://www.w3.org/1999/xhtml
xmlns:vxml=http://www.w3.org/2001/vxml
xmlns:ev=http://www.w3.org/2001/xml-events
xmlns:xv=http://www.voicexml.org/2002/xhtml+voice
>

<head>
      <title>Inbox</title>
      <vxml:form id="searchInbox">
            <vxml:field name="criteria">
            <vxml:prompt>
                  Please speak the name of the person
                  you are looking for.
            </vxml:prompt>
            <vxml:grammar mode="voice" xml:lang="en-US" version="1.0"
            root="command">
            <vxml:rule id="command" scope="public">
                  <vxml:ruleref uri="#action"/>
            </vxml:rule>
            <vxml:rule id="action">
                  <vxml:one-of>
                        <vxml:item>James</vxml:item>
                        <vxml:item>David</vxml:item>
                        <vxml:item>John</vxml:item>
                        <vxml:item>Michael</vxml:item>
                        <vxml:item>Karen</vxml:item>
                  </vxml:one-of>
            </vxml:rule>
            </vxml:grammar>
            <vxml:filled>
                  I think you said <vxml:value expr="criteria"/>
            </vxml:filled>
            </vxml:field>
      </vxml:form>

      <xv:sync input="search_criteria" field="criteria"/>
</head>
```



```
<body>
      <p ev:event="onclick" ev:handler="#searchInbox" >this example
demonstrates filtering of emails using speech as input. Click anywhere
to enter search criteria using voice
      </p>
      <form id="search_inbox" method="post" action = msgsearch.jsp">
            <label id="search"> Please enter name
                  <input name="search_criteria" type="text"/>
                        ev:event="focus" ev:handler="searchInbox"/>
            </label>
      </form>
</body>
</html>
```

Conclusion

The X+V snippet discussed brings out the salient features of implementing a multimodal user interface for an application as universal in its appeal, as multimodal email. Apart from implementing standard email functions such as forward, reply, compose, delete using a similar code implementation, the application can also include messaging interfaces such as sending SMS, MMS from within the interface. Furthermore, as speech interfaces evolve to support NLU (natural language understanding), users can speak complete sentences for commands, rather than just keywords, making the speech entry more natural. Figure 2 shown below presents a "view of the multimodal messaging future" for an email application interface.

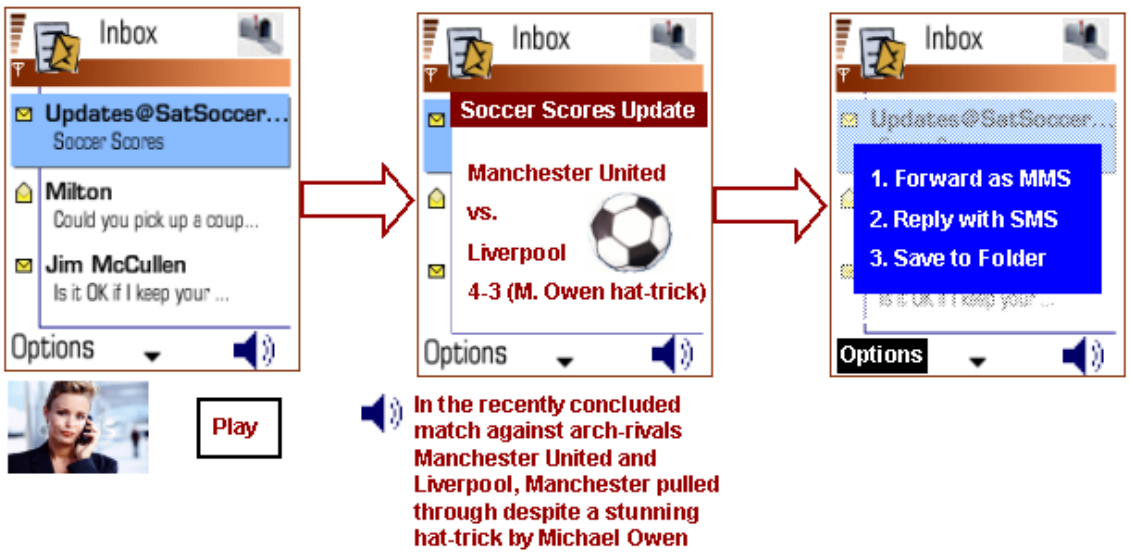

Figure 2 View of Multimodal Messaging