

14 Design methods and software tools for consistency in multi-channel web applications

Charlie Wiecha, IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 10598, USA, wiecha@us.ibm.com

Rahul Akolkar, IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 10598, USA, akolkar@us.ibm.com

Rafah Hosn, IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 10598, USA, rhosn@us.ibm.com

Thomas Ling IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 10598, USA, ling@us.ibm.com

TV Raman, Google Research, raman@google.com

INTRODUCTION

The desire for ubiquitous access to web applications like online shopping, combined with the coming of age of speech technology, creates unique challenges as we design for multi-channel interaction. Usable and cost-effective addition of spoken access to these applications requires:

- Identifying user scenarios that benefit from voice interaction, implying the ability to treat spoken interaction as a first-class user interface modality; experience shows that simply speaking the same content as in the visual interface does not deliver usable interfaces.
- High quality speech components that need not be developed on a per application basis and that deliver a consistent "sound and feel" – often called the speech "persona" – within the voice channel.
- An application design that reuses many of the programming artifacts used by mainstream Web developers in delivering visual access, and hence also offers appropriate consistency across the voice and visual channels.

This paper presents a design and engineering approach to multi-channel applications that supports both intra-channel consistency and cross-channel consistency. The main innovation is the introduction of a UI component model allowing for sharing or overriding design decisions as appropriate to interaction using multiple channels. In this paper we illustrate such an iterative enablement of a transactional web application with voice interaction – using Amazon.com as a test case.

CONSISTENCY IN MULTI-CHANNEL APPLICATIONS

We do not assume that the entirety of the business process is supported on each of the interaction channels. Indeed, an important part of the design process is to determine the desired allocation of function to each user category and to each interaction channel. Our goal is that interaction should be consistent both within and across channels. For example, all voice interaction in an application should collect dates using a common prompt style, recognition vocabulary, and confirmation dialog.

Across channels as different as voice and GUI, even though prompts, grammars, and confirmations may be specific to voice interaction, the navigation rules for collecting such data may reflect the use of a consistent design with the visual interface. Consider a linear wizard for collecting individual data items followed by a confirmation of the aggregate data in one final step. Since Web Application frameworks factor the implementation of such user tasks into controllers and views, they afford us the opportunity to re-use navigation controllers across channels, while appropriately customizing the view layer on a per-modality basis. This leads to cross-channel consistency in the overall user experience.

© 2006 for the individual papers by the papers' authors. Copying permitted for private and scientific purposes. Re-publication of material on this page requires permission by the copyright owners.

SPEECH ENABLING AMAZON.COM

Our goal was to analyze scenarios of widely used online applications to determine specific user tasks that are best performed using spoken interaction, and to support those in a manner consistent with the complementary visual interactions. Amazon.com (<http://www.amazon.com>) was chosen as the target application and offers a set of user tasks accessible through web services interfaces thus readily supporting multi-channel interaction.

To support multi-channel design, we have defined a common programming model for user navigation in both voice and visual displays. This notation for flow is based on the well-known State Chart semantics from UML, and is being extended with an XML notation in the W3C Voice Browser Working Group (SCXML) [2]. State charts avoid some of the common problems of state-transition networks, for example an exploding number of states for global actions, by supporting both sequential and parallel (concurrent) flows. Our team is participating in the definition of the SCXML language, and in addition is contributing an open source implementation of an SCXML interpreter to Apache [3].

WITHIN-CHANNEL CONSISTENCY

Interactions within the speech-enabled sections of the application are under control of components implemented using our library of Reusable Dialog Components (RDCs) in Apache[1]. RDCs encapsulate a recursive MVC design pattern that capture unique aspects of voice interaction, and help better align the data interfaces amongst the different interaction modalities. Each of a number of data-type specific Reusable Dialog Components encapsulates the best practices of prompting and error correction – typically across multiple dialog turns – and corresponds roughly to the unit of content that a single HTML element would gather in an visual page. Associated with the implementation of an RDC is a more detailed interaction flow diagram, represented again in a State Chart XML notation, which drives the dialog management within that component.

An example interaction with the voice-based catalog is:

S: Welcome to the RDC Music Store. Choose a music genre or select one of the V-Store categories.

U: [silence]

S: Say Today's Deals, Top Sellers, New Releases or choose a genre.

U: New Releases

S: The following titles are available. Please select one of: Thelonious Monk Quartet with John Coltrane at Carnegie Hall, Amarantine, Confessions on a Dance Floor, At Folsom Prison, Some Hearts, Born to Run: 30th Anniversary 3-Disc Set, X and Y, Breakaway, Brokeback Mountain, The Breakthrough.

U: Brokeback Mountain

S: Brokeback Mountain is available for \$13.49 You can buy the album, obtain more information or look up similar titles.

U: Describe album

S: Brokeback Mountain has been released by Verve on November 01 2005. You can buy the album or look up similar titles.

U: Buy album

S: Brokeback Mountain has been added to your cart

U: View Cart

S: The following titles are in your cart: Brokeback Mountain, Confessions on a Dance Floor.

S: You can proceed to checkout or continue shopping.

ACROSS-CHANNEL CONSISTENCY

SCXML is the programming model both for voice dialogs in RDCs and for dialog flows across multiple JSP pages in a visual application. The goal is to capture essential navigation patterns and then allow for their refinement as either visual pages in HTML or voice interaction using the RDC component library generating VoiceXML. The detailed voice interaction will certainly be different from the HTML pages –and will typically involve multiple turns of dialog prompting for inputs and correcting for errors, in order to complete what is a single screen of a visual interface. However, the higher level task oriented navigation can often remain the same.

An interesting special case is in mobile GUI devices, where the information content at each step of visual interaction is similar to that observed in voice interaction. The smaller screen size of mobile devices more closely approximates the information a voice user is able to hear and generate per turn of dialog. In that case, we have been able to draw SCXML dialog flows once and reuse them across both visual and voice interaction without further refinement on the voice side – the following screen shots are from the PDA version of the Amazon voice catalog search described above and were generated from the same navigation controller in both versions of the application.



Figure 1: PDA version of Amazon.com catalog shopping

We believe the greatest utility of the above technology may be less to achieve a literal reuse of the same web artifacts (navigation controllers, data models, etc) across multiple channels as it is to allow for a much more rapid exploration of refinements to any one channel without disrupting the overall design of the application. When done with our initial catalog search, for example, we felt that the order fulfillment and tracking tasks ("Where's My Stuff?" in Amazon) might be even more valuable over a speech channel than initial order entry. The effort to have experimented with order entry and then observed its voice characteristics was comparable to that of implementing it over a visual channel – unlike traditional speech development which may be far more costly. We believe, therefore, that techniques such as in our approach can now bring iterative voice development, and hence a practical approach to consistency, within the range of affordable use.

REFERENCES

- [1] Reusable Dialog Components at Apache: <http://jakarta.apache.org/taglibs/doc/rdc-doc/intro.html>
- [2] Working Draft of State Chart XML (SCXML): State Machine Notation for Control Abstraction 1.0: <http://www.w3.org/TR/scxml/>
- [3] Commons SCXML implementation at Apache: <http://jakarta.apache.org/commons/sandbox/scxml/>