Building Context-Aware Multimodal Applications Using W3C Open Standards

Using Cue-me™ Studio

Presented by: Nagesh Kharidi
Technical Director, Openstream
In a nutshell: What is Cue-me™?

Cue-me is a multimodal browser

Dynamically configurable, secure, hybrid-web interaction container based on W3C-MMI Architecture

Enhances applications with modality-and-context-appropriate features

Delivers rich UX

Makes applications future-proof
Cue-me™ multimodal browser

- Integrates with device peripherals and functionality such as Camera, GPS, Microphone & Speaker, Pen & Touch input, Barcode scanner, etc.
- Exposes functionality as “components” – Camera component, Location component, Voice component, Ink component, etc.
- Browser is also a component
Cue-me™ multimodal browser

- Components provide developers ability to:
  - multimodal-enable their webapps
  - access and control device hardware and data
- Component functionality is accessed through SCXML – an XML-based language standardized by the W3C
- Components define the SCXML events that can be sent to and received from them
IM is the linchpin for Multimodal, Context-Aware interaction

Cue-me™ Architecture: Based on W3C MMI

IM

Mobile Application Server /
Distributed Applications

SCXML Document

W3C and OSGi based architecture
http://www.w3.org/TR/mmi-arch

GUI
Voice
Gesture

Modality Components

CoDA Component

Application Components

Application

...
Cue-me™ multimodal browser

- One application – multiple platforms: the same app works on all supported platforms

- Available on following platforms: Android, Apple iOS, BlackBerry, Symbian Series60, Windows Mobile, Windows Phone, Windows Desktop

- Uniform, standards-based markup language to program all components
A Cue-me™ App consists of:

Open Standards based Package of Markup Documents and Scripts

Package executes on the mobile device in the context of Cue-me™ Components @ Run Time

- **Build Time View Consists of:**
  - HTML/5 Markup Documents
  - CSS/3
  - Javascript Libraries
  - Images and other resources (typical of a web application)
  - Interaction Management (IM) Document
Cue-me™ Studio is based on Eclipse

- Embedded Web Server
- Cue-me™ Simulator
- Cue-me™ Application Builder
- Cue-me™ Interactions Editor
- Cue-me™ single authoring Package generator
- Web Tools Platform (Standard Eclipse Package)
Development Methodology

Normal Web Dev Practice

Author HTML(GUI) App with service/data sync stubs

Author and Connect HTML app with services and data sync

Author interaction markup for Cue-me™

Test App in Cue-me™ Simulator

Generate Cue-me™ App Package
With Cue-me™ Studio you can:

Build Open Standards based Multimodal Applications on any desktop...

... leveraging what you already know ...very little training..and tooling is Open (Eclipse)

Run these apps on a variety of devices in the Cue-me™ Multimodal Platform....

Secure and Manage them from a single management platform
SCXML Program

- SCXML is an event-based state machine language (xml-based)
- Each component is addressed using a unique name in the scxml: x-html, x-voice, x-camera, x-ink, etc.
- Developing a multimodal App for Cue-me involves programming the components using SCXML
SCXML Program

- SCXML program contains one or more states
- Each state generally encapsulates functionality to achieve a particular task: Login, Order, Payment, etc.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<x-scxml initialstate="Login">
  <state id="Login">....</state>
  <state id="Order">...</state>
  <state id="Payment">...</state>
</x-scxml>
```
SCXML examples

- Components are programmed by sending SCXML events to them and responding to events from them.

  - Example: Play a voice prompt:
    ```xml
    <send event="playText" to="x-voice" data="Welcome to the conference" />
    ```

  - Example: Invoke Camera component to take a picture:
    ```xml
    <send event="startPictureCapture" to="x-camera" />
    ```
Deposit a Cheque

ENDORSE HERE

DO NOT WRITE, STAMP OR SIGN BELOW THIS LINE
RESERVED FOR FINANCIAL INSTITUTION USE
**Example**: Invoke Annotation component to annotate the camera picture:

```xml
<go on="ImageData" from="x-camera">
   <send event="setBackgroundImageData" to="x-annotate"
        data="event.value" />
</go>
```
Endorse a Cheque
<link rel="stylesheet" media="only screen and (max-device-width: 480px) and (orientation:landscape)"/>
<link rel="stylesheet" media="only screen and (max-device-width: 480px) and (orientation:portrait)"/>
<link rel="stylesheet" media="only screen and (min-device-width: 480px) and (orientation:landscape)"/>
<link rel="stylesheet" media="only screen and (min-device-width: 480px) and (orientation:portrait)"/>

<script type="text/javascript">

</script>

<script type="text/javascript" src="js/lib/cueme.min.js"></script>
<script type="text/javascript" src="js/lib/cuemexmlhttpreq_desktop.min.js"></script>
<script type="text/javascript" src="js/lib/cuemelogger.min.js"></script>
<script type="text/javascript" src="js/lib/Queue.js"></script>
<script type="text/javascript" src="js/lib/hashtable.js"></script>
<script type="text/javascript" src="js/lib/json.js"></script>
<script type="text/javascript" src="js/lib/database.min.js"></script>
<script type="text/javascript" src="js/lib/speech.js"></script>
</head>

<body>
  <div id="container">
    <h1>Welcome to MVC 2013!</h1>
    <div class="clear"></div>
    <div class="buttonDiv"></div>
  </div>
</body>
W3C Resources

- http://www.w3.org/2002/mmi/
- http://www.w3.org/TR/scxml/

To get started, contact me:
nagesh@openstream.com