W3C Multimodal Interaction Activities

Deborah A. Dahl
August 9, 2006
Topics

- Multimodal Architecture
- EMMA
Multimodal Architecture
Multimodal Architecture and Interfaces

- A loosely-coupled, event-based architecture for integrating multiple modalities into applications
- All communication is event-based
- Based on a set of standard life-cycle events
- Components can also expose other events as required
- Encapsulation protects component data
- Encapsulation enhances extensibility to new modalities
- Can be used outside a web environment
MMI Architecture

- Defines five basic components of an MMI system
  - Runtime Framework or Browser: initializes application and runs markup
  - Interaction Manager: coordinates modality components and provides application flow
  - Modality Components: provide modality capabilities such as speech, pen, keyboard, mouse
  - Data Model: handles shared data
  - DCI (Delivery Context and Interfaces): device properties and user preferences
- http://www.w3.org/TR/mmi-arch/
Generic MMI Architecture

Runtime Framework

- Interaction Manager
- Data Model
- DCI

Input
- Speech Interaction
- Keyboard
- Pen Modality
- Pointing
- Speech Input

Output
- Speech Output
- Image Output
- Video Output
- Audio Output
- Graphical Output

Conversational Technologies
MMI Architecture Principles

• Runtime Framework communicates with Modality Components through asynchronous events
• Modality Components don’t communicate directly with each other but indirectly through the Runtime Framework
• Components must implement basic life cycle events, may expose others
• Modality components can be nested (e.g. a Voice Dialog component like a VoiceXML <form>)
• Components need not be markup-based
• EMMA communicates users’ inputs to IM
## Life Cycle Events

<table>
<thead>
<tr>
<th>Event</th>
<th>From</th>
<th>To</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewContextRequest</td>
<td>Modality</td>
<td>Runtime Framework</td>
<td>Request new context</td>
</tr>
<tr>
<td>NewContextResponse</td>
<td>Runtime Framework</td>
<td>Modality</td>
<td>Send new context id</td>
</tr>
<tr>
<td>Prepare</td>
<td>Runtime Framework</td>
<td>Modality</td>
<td>Pre-load markup</td>
</tr>
<tr>
<td>PrepareResponse</td>
<td>Modality</td>
<td>Runtime Framework</td>
<td>Acknowledge Prepare</td>
</tr>
<tr>
<td>Start</td>
<td>Runtime Framework</td>
<td>Modality</td>
<td>Run markup</td>
</tr>
<tr>
<td>StartResponse</td>
<td>Modality</td>
<td>Runtime Framework</td>
<td>Acknowledge Start</td>
</tr>
<tr>
<td>Done</td>
<td>Modality</td>
<td>Runtime Framework</td>
<td>Finished running</td>
</tr>
<tr>
<td>Cancel</td>
<td>Runtime Framework</td>
<td>Modality</td>
<td>Stop processing</td>
</tr>
<tr>
<td>CancelResponse</td>
<td>Modality</td>
<td>Runtime Framework</td>
<td>Acknowledge Cancel</td>
</tr>
<tr>
<td>Pause</td>
<td>Runtime Framework</td>
<td>Modality</td>
<td>Suspend processing</td>
</tr>
<tr>
<td>PauseResponse</td>
<td>Modality</td>
<td>Runtime Framework</td>
<td>Acknowledge Prepare</td>
</tr>
<tr>
<td>Resume</td>
<td>Runtime Framework</td>
<td>Modality</td>
<td>Resume processing</td>
</tr>
<tr>
<td>ResumeResponse</td>
<td>Modality</td>
<td>Runtime Framework</td>
<td>Acknowledge Resume</td>
</tr>
<tr>
<td>Data</td>
<td>either</td>
<td>either</td>
<td>Send data values</td>
</tr>
<tr>
<td>ClearContext</td>
<td>Runtime Framework</td>
<td>Modality</td>
<td>Deactivate context</td>
</tr>
</tbody>
</table>
Example of MMI Architecture

- Runtime Framework
- Speech Recognition Modality
- Face Recognition Modality
- TTS
- View life cycle event traffic
Example Modality Components

Conversational Technologies
Conversational Technologies

Startup

Home Control with MMI Life Cycle Events

Window is open
Thermostat is set to 75 degrees F
Light is off

<table>
<thead>
<tr>
<th>Event</th>
<th>Context</th>
<th>From</th>
<th>To</th>
<th>Status</th>
<th>Errorinfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare</td>
<td>content1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrepareResponse</td>
<td>content1</td>
<td>speech</td>
<td>runtimeFramework</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>content1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>PrepareResponse</td>
<td>content1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td>Success</td>
<td></td>
</tr>
</tbody>
</table>

Start ASR
Pause ASR
Resume ASR
Cancel ASR
Clear Context
Start Face Recognition
Send Pause to Face Recognition
Face Recognition

Home Control with MMI Life Cycle Events

<table>
<thead>
<tr>
<th>Event</th>
<th>Context</th>
<th>From</th>
<th>To</th>
<th>Status</th>
<th>Errorinfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare Response</td>
<td>context1</td>
<td>speech</td>
<td>runtimeFramework</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare Response</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartResponse</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Done</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td>Success</td>
<td></td>
</tr>
</tbody>
</table>

Start ASR
Clear Context
Start Face Recognition
Send Pause to Face Recognition
# Speech Recognition

A demonstration of a home control system using speech recognition technology. The system is shown in a graphical interface with various events and actions related to home control. The interface includes buttons for starting ASR, pausing ASR, resuming ASR, and canceling ASR. The table below shows the events and actions:

<table>
<thead>
<tr>
<th>Event</th>
<th>Context</th>
<th>From</th>
<th>To</th>
<th>Status</th>
<th>ErrorInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrepareResponse</td>
<td>context1</td>
<td>speech</td>
<td>runtimeFramework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrepareResponse</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartResponse</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Done</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td>Success</td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartResponse</td>
<td>context1</td>
<td>speech</td>
<td>runtimeFramework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>context1</td>
<td>speech</td>
<td>runtimeFramework</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The system includes options for managing context and starting face recognition.
Failure Responses

Home Control with MMI Life Cycle Events

- Window is open
- Thermostat is set to 80 F
- Light is off

<table>
<thead>
<tr>
<th>Event</th>
<th>Context</th>
<th>From</th>
<th>To</th>
<th>Status</th>
<th>Errorinfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrepareResponse</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrepareResponse</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartResponse</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pause</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PauseResponse</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>face recognition</td>
<td>Failure</td>
<td>Can'tPause</td>
</tr>
<tr>
<td>Data</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Done</td>
<td>context1</td>
<td>face recognition</td>
<td>runtimeFramework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartResponse</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartResponse</td>
<td>context1</td>
<td>runtimeFramework</td>
<td>speech</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Controls:
- Start ASR
- Pause ASR
- Resume ASR
- Cancel ASR
- Clear Context
- Start Face Recognition
- Send Pause to Face Recognition
EMMA (Extensible MultiModal Annotation)

• XML format
• represents results of processing user input
• includes annotations adding information about the input (confidence, timestamp, tokens, language, etc.)
• can be used for input from speech, ink camera, keyboard…
• Contents of Data property (e.g. of Data or Done event)
• http://www.w3.org/TR/emma
Advantages of EMMA

• Vendor neutral, uniform representation of semantic results across modalities
• Rich annotation of input context
• XML format enables leveraging of XML tools
• Designed to take into account VoiceXML requirements
Applications of EMMA

Represent speech or other modality input for
• dialog manager
  – VoiceXML
  – SCXML
  – scripts
• processing stages after recognition
  – classification
  – reference resolution
• logging and storing in a database
  – OA&M
  – debugging
  – diagnosing dialog problems
Example Modalities

- speech recognition with rule grammar
- speech recognition with dictation grammar
- face recognition
- typed input
EMMA Result for Speech in a Dialog

<emma:emma version="1.0" xmlns:emma="http://www.w3.org/2003/04/emma/">
  <emma:info>
    <application>
      home
    </application>
  </emma:info>
    <action>turn_on</action>
    <object>light</object>
  </emma:interpretation>
</emma:emma>
EMMA Result for Dictation

<emma:emma version="1.0"
    xmlns:emma="http://www.w3.org/2003/04/emma/">
    <emma:one-of emma:start="1152023348282"
        emma:end="1152023349213" emma:dialog-turn="1"
        emma:function="transcription" emma:medium="acoustic"
        emma:mode="speech" emma:verbal="true" id="oneof1" >
        <emma:interpretation id="interp1" emma:tokens="I like to go on a trip ">
            <emma:literal>
                I like to go on a trip
            </emma:literal>
        </emma:interpretation>
    </emma:one-of>
</emma:emma>
EMMA for Face Recognition

```xml
<emma:emma version="1.0" xmlns:emma="http://www.w3.org/2003/04/emma/">
  <emma:info>
    <application>
      home
    </application>
  </emma:info>
  <emma:interpretation id="interp1" emma:confidence="1.0" emma:signal="file://c:\multimodalSoftware\face\users\debbie.jpg" emma:medium="visual" emma:mode="photograph" emma:verbal="false" emma:function="identification" emma:uninterpreted="false" emma:dialog-turn="1">
    <result>debbie</result>
  </emma:interpretation>
</emma:emma>
```
EMMA Result for Text

```xml
<emma:emma version="1.0" xmlns:emma="http://www.w3.org/2003/04/emma/">
  <emma:info>
    <application>
      travel
    </application>
  </emma:info>
  <emma:interpretation id="interp7" emma:duration="7611" emma:confidence="1.0" emma:process="file://Microsoft English (U.S.) v6.1 Recognizer, SAPI5, Microsoft" emma:medium="tactile" emma:no-input="true" emma:mode="keys" emma:verbal="true" emma:uninterpreted="false" emma:function="dialog" emma:dialog-turn="7" emma:end="1152022929520" emma:lang="en-US" emma:tokens="i'd like to go from new york to philadelphia">
    <source>new york</source>
    <destination>philadelphia</destination>
  </emma:interpretation>
</emma:emma>
```
Summary

• MMI Architecture provides a general, clean interface to a wide range of modality components
• EMMA provides a standard and general way of representing user inputs
• Very easy to integrate new modalities
• Loose coupling and lack of access to internal modality data improves security
Empowering Your customers and employees with speech technologies

SpeechTEKI 2006
The Voice Solutions Showcase

Empower

Thank You