



Technologies, Tools, and Standards for Multimodal Application Development

Deborah Dahl

Principal, Conversational Technologies

Mobile Voice Conference 2014

San Francisco



I want to develop a multimodal application!



....Now what?

- What devices do I want my application to run on?
- What technologies does my application need?
- What development tools are available?
- What standards do I need to know about?

Devices: The Olden Days



Devices: Now

- Smartphones: Android, iPhone, Windows Phone, Blackberry
- Wearables (Xowi, Google Glass, Pebble)
- Tablets
- Car
- TV
- Appliances
- Smart Environments (home, office, public places)
- Things

Technologies

Input

- Speech recognition
- Natural language understanding
- Dialog management
- Touch
- Sensors (GPS, accelerometer, proximity sensor)
- Camera
- Handwriting
- Haptics

Output

- One or more displays
- Audio (audio files or TTS)
- Video
- Vibration

Example: Smartphone with speech recognition

Needs

- Speech recognition
- GUI

Other possibilities

- Camera input (recognizing face, objects, gestures...)
- Audio output (TTS or audio files)
- Sensors

Speech Recognition Options

- Google Android
- Google Chrome
- CMU PocketSphinx.js
- Nuance NDEV
- ATT Developer
- OpenEars
- Sphinx
- iSpeech

Consider requirements

- Offline or cloud
- Languages
- What language models are available?
- Are grammars or SLM's needed
- Cost
- Platform-independence
- Development complexity

Natural Language Understanding

- Component technologies for various tasks
 - Translation (Bing, Google)
 - Part of speech tagging
 - Parsing
 - Named entity recognition
 - LingPipe <http://alias-i.com/lingpipe/>
- Open source
 - Stanford NLP tools
 - OpenNLP
- All in one -- text in meaning out
 - WIT (<https://wit.ai/> API for natural language)

Dialog Tools

- OpenDial (open source dialog manger)
 - <https://code.google.com/p/opendial/>
 - Java-based software for developing robust and adaptive dialog applications using XML rules
- Pandorabots (API)
 - <http://www.pandorabots.com/>
 - AIML (Artificial Intelligence Markup Language) based framework for
- State Chart XML (SCXML)

GUI Options

- Native iOS, Android, Windows
- PhoneGap (HTML5)
- Appcelerator Titanium
- Web Browser (HTML5)

Tool Tradeoffs: All-in-one vs. Component Tools

- In a mashup world, it's not so important that all the components of a multimodal application come from the same place
- There are more options and more flexibility if the application can use components from different vendors
- But this makes applications more complex



All in one

- Openstream
 - platform-independent
 - standards-based
 - Cue-me browser and Cue-me Studio
- MIT App Inventor
 - Android
 - Very simple to use
 - Limited capabilities

Example: MIT App Inventor

- A simple app development environment for Android
- Includes speech recognition, typical HTML elements, typical sensors, media capture and playback and many more
- <http://appinventor.mit.edu>

MIT App Inventor Screen

The screenshot displays the MIT App Inventor 2 Beta web interface. At the top, the MIT App Inventor 2 Beta logo is on the left, and navigation links for Project, Connect, Build, and Help are in the center. On the right, there are links for My Projects, Guide, Report an Issue, and the email address ds1@connect.cornell.edu. Below the navigation bar, the interface is divided into four main panels: Palette, Viewer, Components, and Properties.

- Palette:** A vertical list of components categorized into User Interface, Layout, Media, Drawing and Animation, Sensors, Social, Storage, Connectivity, and LEGO MINDSTORMS.
- Viewer:** A central workspace showing a mobile device screen. The screen has a status bar at the top with signal strength, Wi-Fi, and battery icons, and the time 9:46. Below the status bar, the screen is titled "Screen 1" and is currently empty.
- Components:** A panel on the right showing the components currently on the screen. It lists "Screen 1" with a "Rename" and "Delete" button below it. There is also a "Media" section with an "Upload File ..." button.
- Properties:** A panel on the far right showing the properties for the selected "Screen 1" component. The properties include:
 - AboutScreen: A text input field.
 - AlignHorizontal: A dropdown menu set to "left".
 - AlignVertical: A dropdown menu set to "top".
 - BackgroundColor: A color selection tool set to "white".
 - BackgroundImage: A text input field set to "None".
 - CloseScreenAnimation: A dropdown menu set to "Default".
 - Icon: A text input field set to "None".
 - OpenScreenAnimation: A dropdown menu set to "Default".
 - ScreenOrientation: A dropdown menu set to "Unspecified".
 - Scrollable: A checkbox that is checked.
 - ID: A text input field.
 - Title: A text input field set to "Screen 1".
 - VersionCode: A text input field set to "1".
 - VersionName: A text input field set to "1.0".

Standards

- Organizing interaction
 - MMI Architecture
 - SCXML
- Moving information around
 - WebRTC
 - WebAudio
 - WebSockets
 - HTTP
 - Ajax
- Managing Speech
 - SRGS, SISR, SSML
- Representing user input and system output
 - EMMA
 - EmotionML
 - InkML

Many Options

- Must carefully weigh requirements of application
- This area developing very rapidly
- New options are continually becoming available