

NATURAL LANGUAGE INTERACTION WITH THE WEB OF THINGS

Deborah A. Dahl

Conversational Technologies

Mobile Voice 2015

April 20-21

San Jose

Connected things will be everywhere

- Everyday life – home appliances, entertainment, fitness monitor, toys, pets...
- Industry – office, building, industrial robot, parts for manufacturing...
- Public environments – retail store, city services, highway, museum...
- Medicine – sensors, alarms...
- Transportation -- car, train, bus, airplane

Current interaction model

- Vendor provides an SDK for user devices
- Vendor provides an API for accessing the functions of their connected things
- Company and/or Developers create apps for various user devices using provided API's
- Users buy connected things and use GUI apps to interact with things

What's wrong with this approach?

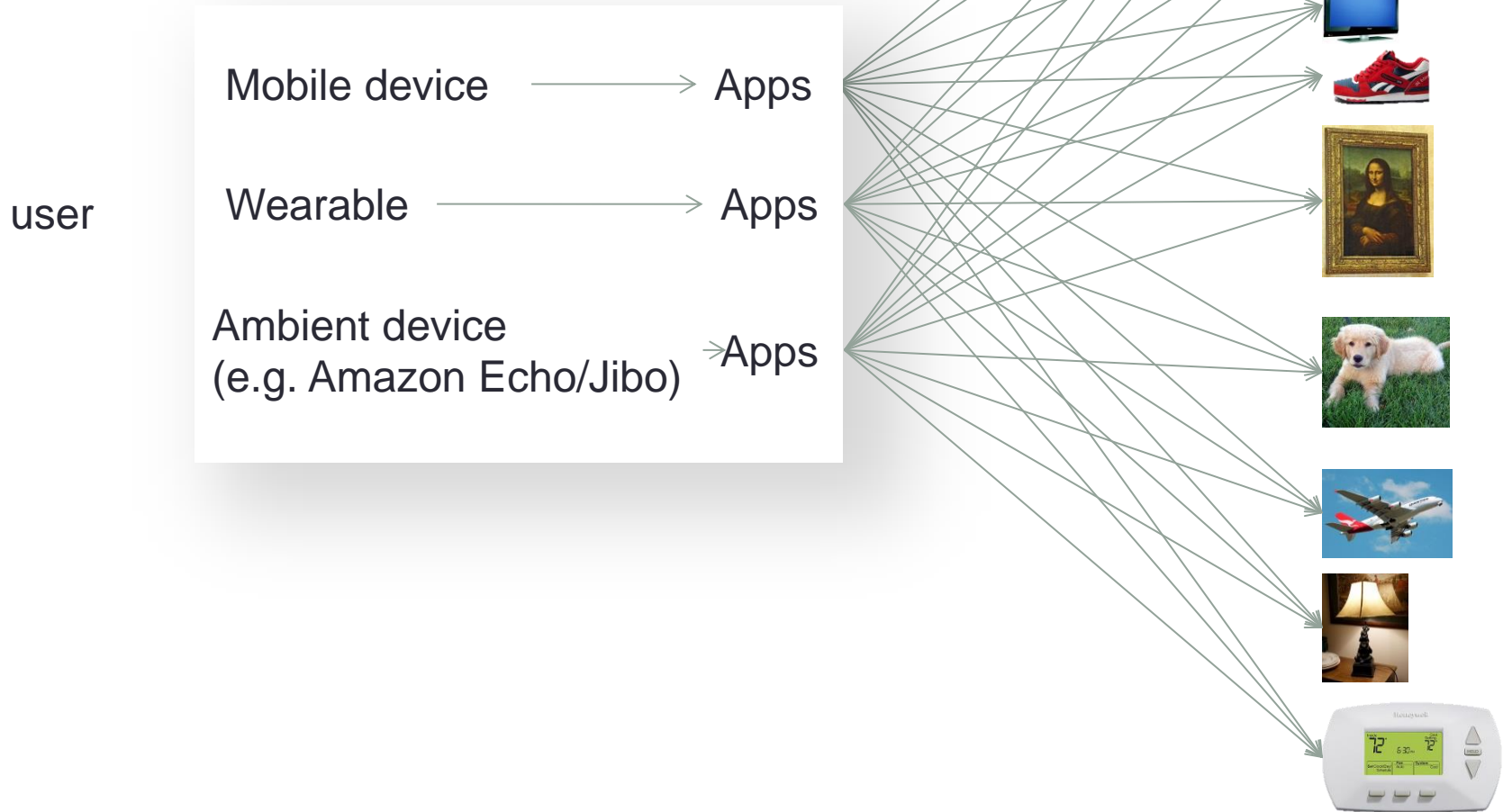
Problem 1: UI's

One Approach to UI's: the basket of remotes



UI Proliferation

Control/Interface



Users need a uniform way to interact

- Don't burden the user with too many interfaces
- No matter how well-designed a GUI interface is, it's different from other GUI's
- User has to learn it
- Harder for users like older adults or people with cognitive disabilities
- UI's can be aggregated into categories (like one UI for the connected home)
- Natural language provides a uniform UI across connected things

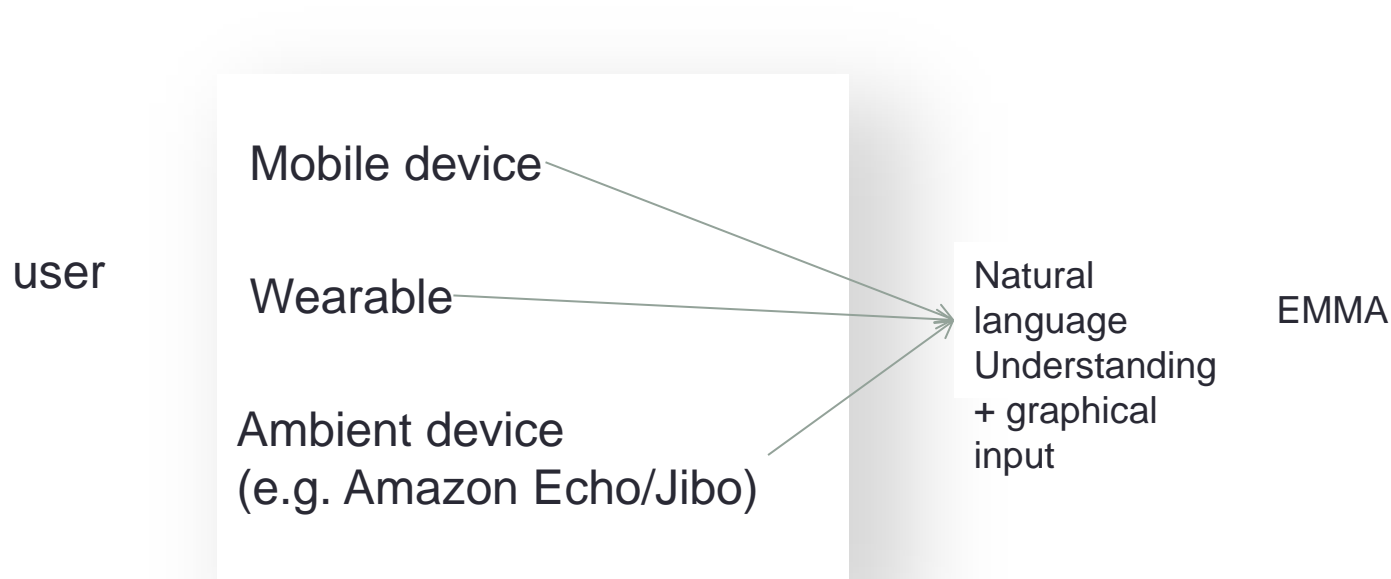
Problem 2: Too many API's

- Many different API's for different things
- Want to hide details of each thing API to simplify development

The Answer: Standards!

- For natural language UI's: W3C EMMA
- For API's: The W3C Multimodal Architecture

EMMA: a standard format for natural language inputs



What's in an EMMA document?

- Standard metadata about utterance
 - Confidence, input tokens, alternative interpretations, process, timing...
- Interpretation of utterance in a structured format
 - Key-value pairs, for example
- Groupings with related utterances
- A standard way to represent natural user inputs

What about API's?

- W3C Multimodal Architecture is generic API for interacting with encapsulated functionality (Modality Components)
- Life-cycle events like “start”, “cancel”, “pause”, “resume”
- Coordinated by an Interaction Manager
- Work on dynamic systems is starting

Putting things together

User

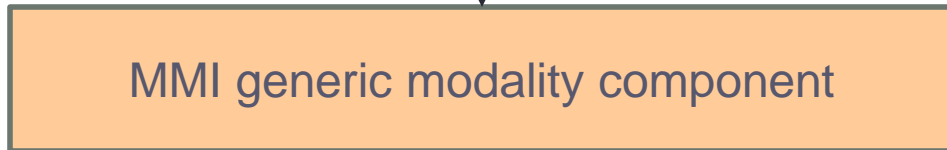


Interpretation
Modality
Components

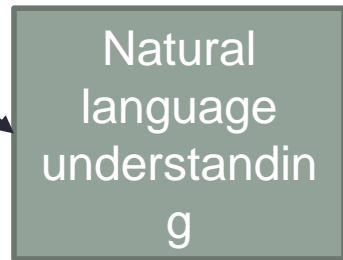
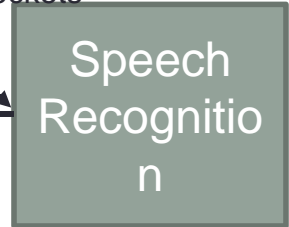
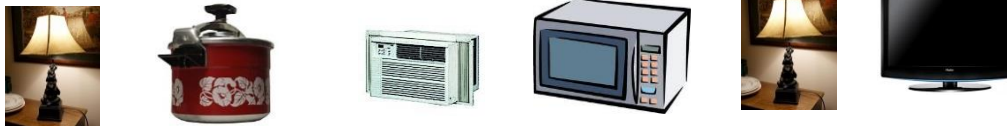
Audio over HTTP or Web Sockets



MMI Life-cycle events



things



EMMA over MMI

EMMA Over MMI

Example: Light bulb

- Philips Hue Light Bulb
- LED light that can change colors
- Controlled through LAN via bridge
- Bridge communicates to light bulbs via Zigbee protocol
- Has own API for developers
- 45 apps on Google Play, 70 on iTunes Store, 3 Windows available



Standards-based Control of Hue Light



```
<emma:emma
  xmlns:emma="http://www.w3.org/2003/04/emma"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.1"
  xsi:schemaLocation="http://www.w3.org/2003/04/emma
    http://www.w3.org/TR/2009/REC-emma-20090210/emma.xsd">
  <emma:interpretation emma:confidence="0.744"
    emma:process="wit.ai" emma:tokens="it's dark in here" id="interp237
    <nIResult>
      <_text>it's dark in here</_text>
      <msg_id>7e5e96f7-4a4d-4f62-9449-43c455a530f7</msg_id>
      <outcomes>
        <e>
          <_text>it's dark in here</_text>
          <confidence>0.744</confidence>
          <entities>
            <on_off>
              <e>
                <value>on</value>
              </e>
            </on_off>
            <thingType>
              <e>
                <value>light</value>
              </e>
            </thingType>
          </entities>
          <intent>changeState</intent>
        </e>
      </outcomes>
    </nIResult>
  </emma:interpretation>
</emma:emma>
```

How to Scale the Web of Things

- Too many UI's?
 - Use natural language
 - Use EMMA for a regular format for natural language results
- Too many API's?
 - Use MMI Architecture to encapsulate things as Modality Components