

# Mastering Multiple Modalities

Presented:

Mobile Voice Conference, San Francisco, CA, April 15, 2013



**Emmett Coin**  
*Industrial Poet, CEO*



MVC, April 15, 2013

# The Problem

Many applications claim to be “multimodal”.

In reality they are usually just “alternately modal”.

- You can **Speak** to it ... or
- You can **Type** to it ... or
  - You can **Touch** it

But they do not support a blended input. You **cannot**:

- Ask for a list (e.g. movies) ... then
  - Touch one ... then
  - Ask “what is the rating?”

The modalities are not **Coordinated**



# The Solution

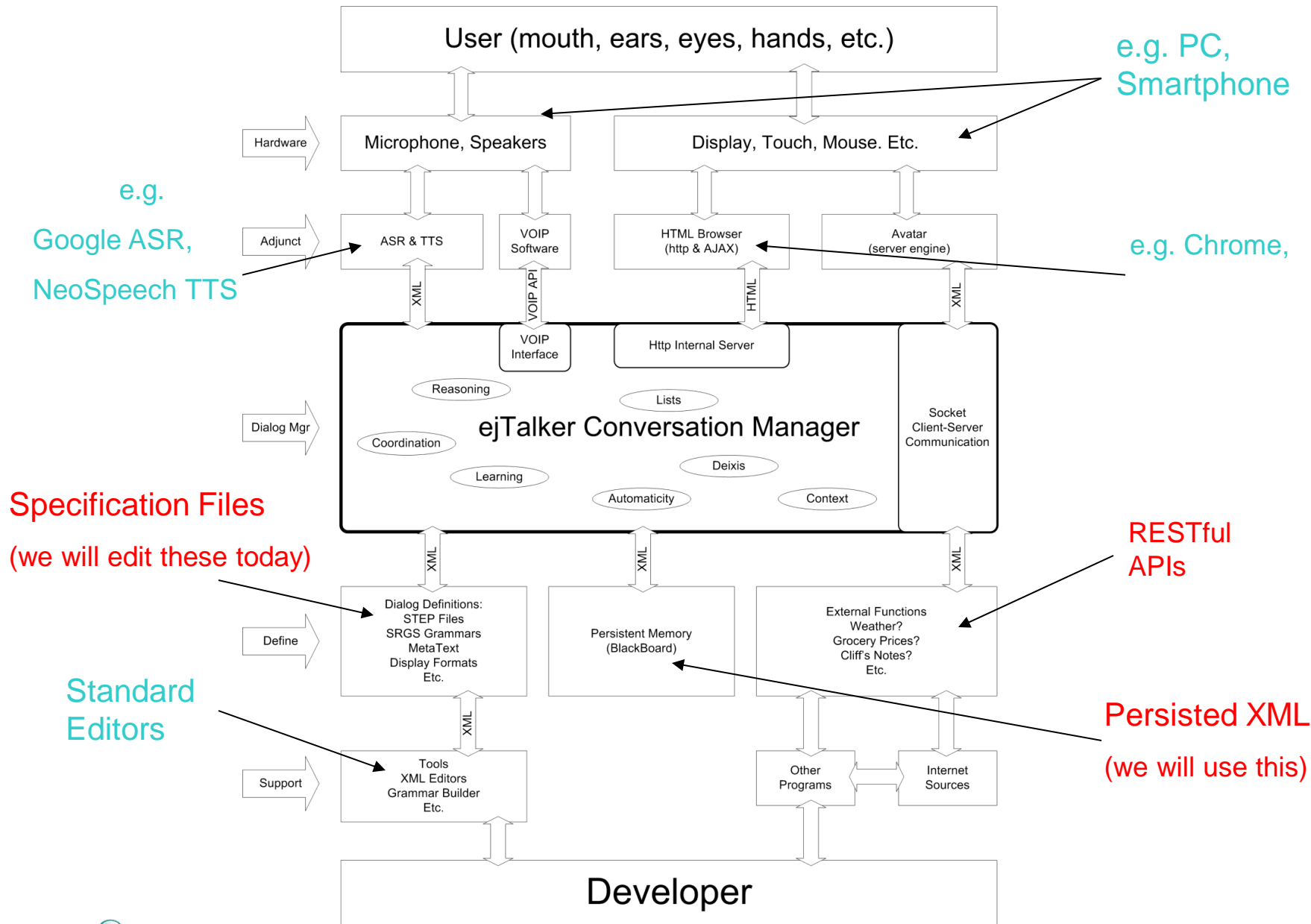
Merge all input modalities into a **single** channel in a **single** dialog manager:

- **Evaluate** all modalities in a common form (text)
- **Share** a common control algorithm (patterns)
- **Scope** all the modality contexts together

**Avoid** modality dependent processing pathways

All inputs are created **EQUAL**





e.g.  
Google ASR,  
NeoSpeech TTS

e.g. PC,  
Smartphone

e.g. Chrome,

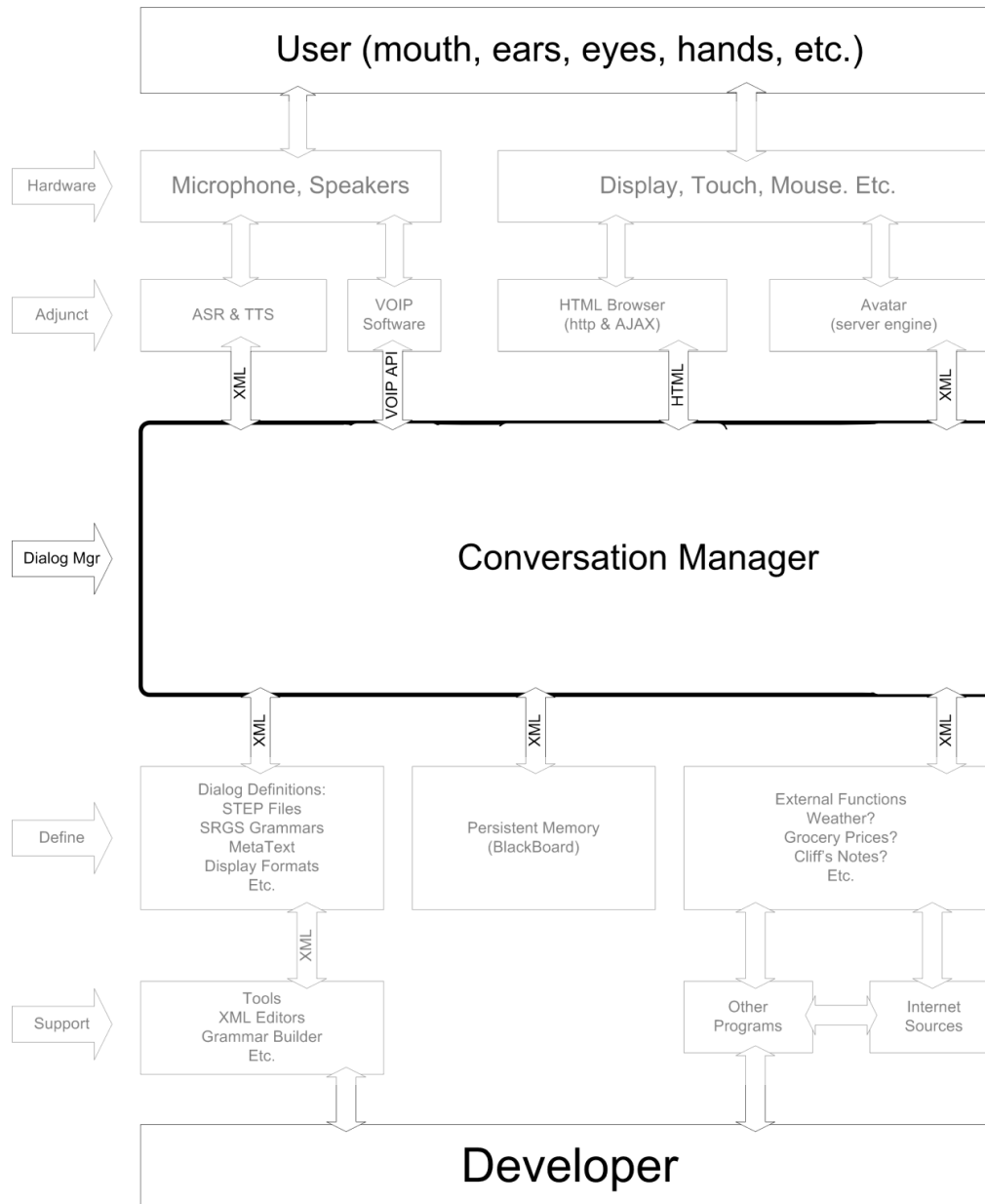
Specification Files  
(we will edit these today)

Standard  
Editors

RESTful  
APIs

Persisted XML  
(we will use this)





# Every Modality “Speaks”

- All input modalities **speak** to the manager
  - Actually it’s more like “texting”
  - A single “consciousness”
- The manager operates **solely** on text input
  - Actually structured text (e.g. XML)
  - It is a “Brain in a Jar”
- Information is shared **across** modalities
  - Context changes continuously
  - What part of each modality is important at a given moment?



# The Example Conversation Manager I will use today

## ejTalker Conversation Manager

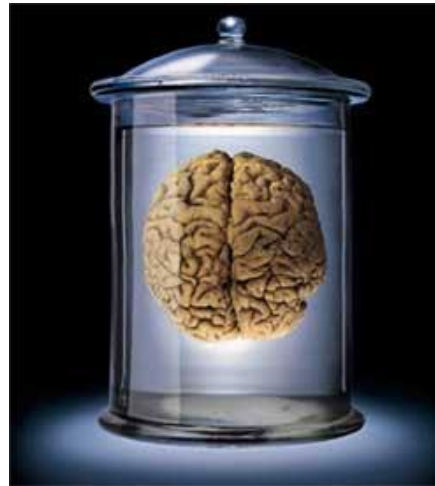
- ejTalker (pronounced: “edge” talker)
- It coordinates all aspects of a conversation
- It has been around for years
- Cloud based or embedded
  - Small executable footprint
  - Low compute loads
- XML formalism
  - For all Input and Output Modalities
  - For all dialog definition files
- Common/shared rule pipeline
  - All inputs test against the same rules
  - Automaticity (automatic adaptation to usage)



Cassandra is based on ejTalker



# The Conversation Engine (Just Thoughts)



(But, ALL of the thoughts!)



# The ejTalker's declarative language.

- **STEP** files
  - Initial instructions for what to say and do
  - Sets of **Rules** that specify how to react.
- **MetaText** files
  - Automatic adaptation specs
    - How often and how recently have we done something
- **ProdGram**
  - Specs for automatic variability
    - Choosing parts of output phrases from semantically equivalent alternatives.
- **setMEM** & **{V:some/path/to/data:default}**
  - How to persist a simple memory: `setMEM`
  - How to access that memory: `{V:a/path/:no memory found}`



# STEP Files: A first look

```
<step>
  <head>
    <name>myMovie.step.xml</name>
  </head>
  <body>
    <introduction>
      <action>
        <presentation>
          <text> Say something about Lincoln. </text>
        </presentation>
      </action>
    </introduction>
    <attention></attention>
    <response>
      <rule name="mentionLincoln">
        <pattern>*lincoln*</pattern>
        <action>
          <presentation>
            <text> Lincoln was a good movie! </text>
          </presentation>
        </action>
      </rule>
    </response>
  </body>
</step>
```



# STEP Files: Rules & Presentation

```
<step>
  <head>
    <name>myMovie.step.xml</name>
  </head>
  <body>
    <introduction>
      <action>
        <presentation>
          <text> Say something about Lincoln. </text>
        </presentation>
      </action>
    </introduction>
    <attention></attention>
    <response>
      <rule name="mentionLincoln">
        <pattern>*lincoln*</pattern>
        <action>
          <presentation>
            <text> Lincoln was a good movie! </text>
          </presentation>
        </action>
      </rule>
    </response>
  </body>
</step>
```



# STEP Files: Rules & Presentation

```
<step>
  <head>
    <name>myMovie.step.xml</name>
  </head>
  <body>
    <introduction>
      <action>
        <presentation>
          <text> Say something about Lincoln. </text>
        </presentation>
      </action>
    </introduction>
    <attention></attention>
    <response>
      <rule name="mentionLincoln">
        <pattern>*lincoln*</pattern>
        <action>
          <presentation>
            <text> Lincoln was a good movie! </text>
          </presentation>
        </action>
      </rule>
    </response>
  </body>
</step>
```



ejTalk Browser Converse ...

localhost:15404/br1/ejConverseGASRejc.html

Speech RC NLP CassCloud Cookies Balloon Webish AI C++ scapableSites hardware

# ejTalk Browser Converse ...



ReLogon End Session [Edit Files](#)

**Human:** click mic and speak

**Understood:** StartUp

**Said:** Say something abo

Speak now

Cancel



# What we just saw

- Speech and GUI modalities
  - Speech recognition
  - Graphic displays
  - Touch screen
- Some lightweight banter
  - Hellos, thanks, quips
- Pronoun resolution
  - Who was “he” and what was “that”
- Queries dependant on context
  - Context began with touch and finished with talk
- **NOT** a split brain
  - Shared data for a shared goal
- **Coordinated** data unification of different modalities



# Merging Modalities (M&Ms?)

- Display, Speech and Touch:
  - Display function generated HTML
    - “onClick” text embedded automatically in the HTML
  - “Text” input to the conversation engine
    - Speak after mic click
      - Sends “... did he direct 2001 ...”
    - Touch on a movie
      - Sends “(TABLE List:simpleMovie,Record:5,Field:poster)”
      - sets focus on this movie
- Rules manage context:
  - the “directorName” by Touch and a Speech query.
  - The “movieName” by Touching a movie poster.



# MM Display & Speech (Rule)

```
<rule name="showMovieList">
  <pattern input="{CK:ejMovie:ejExist},{CK:ejShow:ejExist}">TRUE,TRUE</pattern>
  <examplePattern>
    <ex>show me the movies you know about</ex>
    <ex>let's look at some movie info</ex>
  </examplePattern>
  <action>
    <presentation>
      <text>Here are the ones I know about.</text>
      <semantic>Show Movies List</semantic>
    </presentation>
    <ejScript name="movieExtractAndDisplay.script.xml">
      <paramNode>
        <diplayFormatSpec>movieBasicListWS.xml</diplayFormatSpec>
      </paramNode>
    </ejScript>
  </action>
</rule>
```





# MM Display (Rule)

```
<rule name="showMovieList">
  <pattern input="{CK:ejMovie:ejExist},{CK:ejShow:ejExist}">TRUE,TRUE</pattern>
  <examplePattern>
    <ex>show me the movies you know about</ex>
    <ex>let's look at some movie info</ex>
  </examplePattern>
  <action>
    <presentation>
      <text>Here are the ones I know about.</text>
      <semantic>Show Movies List</semantic>
    </presentation>
    <ejScript name="movieExtractAndDisplay.script.xml">
      <paramNode>
        <diplayFormatSpec>movieBasicListWS.xml</diplayFormatSpec>
      </paramNode>
    </ejScript>
  </action>
</rule>
```



# MM Display & Speech (Rule)

```
<rule name="showMovieList">
  <pattern input="{CK:ejMovie:ejExist},{CK:ejShow:ejExist}">TRUE,TRUE</pattern>
  <examplePattern>
    <ex>show me the movies you know about</ex>
    <ex>let's look at some movie info</ex>
  </examplePattern>
  <action>
    <presentation>
      <text>Here are the ones I know about.</text>
      <semantic>Show Movies List</semantic>
    </presentation>
    <ejScript name="movieExtractAndDisplay.script.xml">
      <paramNode>
        <diplayFormatSpec>movieBasicListWS.xml</diplayFormatSpec>
      </paramNode>
    </ejScript>
  </action>
</rule>
```



# MM Touch (Rule)

```
<rule name="MM_Poster">
  <pattern>(ejMM) (TABLE List:simpleMovie,Record:[W:recordIndex],Field:poster)</pattern>
  <examplePattern>
    <ex>(ejMM) (TABLE List:simpleMovie,Record:5,Field:poster)</ex>
  </examplePattern>
  <action>
    <setMEM>
      <v>movie/mmTouch/record={WILD:recordIndex:}</v>
      <v>movie/mmTouch/field=poster</v>
    </setMEM>

    <function>
      <AFS function="list.getValue" saveStatus="movie/listFunction/status">
        <paramNode>
          ...
          <recordFocus>{V:movie/mmTouch/record:1}</recordFocus>
          <fieldClicked>{V:movie/mmTouch/field:poster}</fieldClicked>
          <fieldSpoken name="title"/>
        </paramNode>
        <resultNode>movie</resultNode>
      </AFS>
    </function>

    <setMEM>
      <v>movie/mmTouch/title={V:movie/fieldSpoken:}</v>
    </setMEM>

    <presentation>
      <speak>{V:movie/fieldSpoken:interesting choice}!</speak>
      <text>You picked {V:movie/fieldSpoken:an interesting one}!</text>
      <semantic>Clicked on: {V:movie/fieldSpoken:something}</semantic>
    </presentation>
  </action>
</rule>
```



# MM Touch (Rule)

```
<rule name="MM_Poster">
  <pattern>(ejMM) (TABLE List:simpleMovie,Record:[W:recordIndex],Field:poster)</pattern>
  <examplePattern>
    <ex>(ejMM) (TABLE List:simpleMovie,Record:5,Field:poster)</ex>
  </examplePattern>
  <action>
    <setMEM>
      <v>movie/mmTouch/record={WILD:recordIndex:}</v>
      <v>movie/mmTouch/field=poster</v>
    </setMEM>

    <function>
      <AFS function="list.getValue" saveStatus="movie/listFunction/status">
        <paramNode>
          ...
          <recordFocus>{V:movie/mmTouch/record:1}</recordFocus>
          <fieldClicked>{V:movie/mmTouch/field:poster}</fieldClicked>
          <fieldSpoken name="title"/>
        </paramNode>
        <resultNode>movie</resultNode>
      </AFS>
    </function>

    <setMEM>
      <v>movie/mmTouch/title={V:movie/fieldSpoken:}</v>
    </setMEM>

    <presentation>
      <speak>{V:movie/fieldSpoken:interesting choice}!</speak>
      <text>You picked {V:movie/fieldSpoken:an interesting one}!</text>
      <semantic>Clicked on: {V:movie/fieldSpoken:something}</semantic>
    </presentation>
  </action>
</rule>
```

what rule responds to

what "touch" might send



# MM Touch (Rule)

```
<rule name="MM_Poster">
  <pattern>(ejMM) (TABLE List:simpleMovie,Record:[W:recordIndex],Field:poster)</pattern>
  <examplePattern>
    <ex>(ejMM) (TABLE List:simpleMovie,Record:5,Field:poster)</ex>
  </examplePattern>
  <action>
    <setMEM>
      <v>movie/mmTouch/record={WILD:recordIndex:</v>
      <v>movie/mmTouch/field=poster</v>
    </setMEM>

    <function>
      <AFS function="list.getValue" saveStatus="movie/listFunction/status">
        <paramNode>
          ...
          <recordFocus>{V:movie/mmTouch/record:1}</recordFocus>
          <fieldClicked>{V:movie/mmTouch/field:poster}</fieldClicked>
          <fieldSpoken name="title"/>
        </paramNode>
        <resultNode>movie</resultNode>
      </AFS>
    </function>

    <setMEM>
      <v>movie/mmTouch/title={V:movie/fieldSpoken:</v>
    </setMEM>

    <presentation>
      <speak>{V:movie/fieldSpoken:interesting choice}!</speak>
      <text>You picked {V:movie/fieldSpoken:an interesting one}!</text>
      <semantic>Clicked on: {V:movie/fieldSpoken:something}</semantic>
    </presentation>
  </action>
</rule>
```

what rule responds to

what "touch" might send

Context manager "remembers"



# MM Touch (Rule)

```
<rule name="MM_Poster">
  <pattern>(ejMM) (TABLE List:simpleMovie,Record:[W:recordIndex],Field:poster)</pattern>
  <examplePattern>
    <ex>(ejMM) (TABLE List:simpleMovie,Record:5,Field:poster)</ex>
  </examplePattern>
  <action>
    <setMEM>
      <v>movie/mmTouch/record={WILD:recordIndex:</v>
      <v>movie/mmTouch/field=poster</v>
    </setMEM>

    <function>
      <AFS function="list.getValue" saveStatus="movie/listFunction/status">
        <paramNode>
          ...
          <recordFocus>{V:movie/mmTouch/record:1}</recordFocus>
          <fieldClicked>{V:movie/mmTouch/field:poster}</fieldClicked>
          <fieldSpoken name="title"/>
        </paramNode>
        <resultNode>movie</resultNode>
      </AFS>
    </function>

    <setMEM>
      <v>movie/mmTouch/title={V:movie/fieldSpoken:</v>
    </setMEM>

    <presentation>
      <speak>{V:movie/fieldSpoken:interesting choice}!</speak>
      <text>You picked {V:movie/fieldSpoken:an interesting one}!</text>
      <semantic>Clicked on: {V:movie/fieldSpoken:something}</semantic>
    </presentation>
  </action>
</rule>
```

what rule responds to

what "touch" might send

Context manager "remembers"



# MM Touch (Rule)

```
<rule name="MM_Postter">
  <pattern>(ejMM) (TABLE List:simpleMovie,Record:[W:recordIndex],Field:poster)</pattern>
  <examplePattern>
    <ex>(ejMM) (TABLE List:simpleMovie,Record:5,Field:poster)</ex>
  </examplePattern>
  <action>
    <setMEM>
      <v>movie/mmTouch/record={WILD:recordIndex:</v>
      <v>movie/mmTouch/field=poster</v>
    </setMEM>

    <function>
      <AFS function="list.getValue" saveStatus="movie/listFunction/status">
        <paramNode>
          ...
          <recordFocus>{V:movie/mmTouch/record:1}</recordFocus>
          <fieldClicked>{V:movie/mmTouch/field:poster}</fieldClicked>
          <fieldSpoken name="title"/>
        </paramNode>
        <resultNode>movie</resultNode>
      </AFS>
    </function>

    <setMEM>
      <v>movie/mmTouch/title={V:movie/fieldSpoken:</v>
    </setMEM>

    <presentation>
      <speak>{V:movie/fieldSpoken:interesting choice}!</speak>
      <text>You picked {V:movie/fieldSpoken:an interesting one}!</text>
      <semantic>Clicked on: {V:movie/fieldSpoken:something}</semantic>
    </presentation>
  </action>
</rule>
```

what rule responds to

what "touch" might send

Context manager "remembers"

Get some other data from the "touched" record



# MM Touch (Rule)

```
<rule name="MM_Poster">
  <pattern>(ejMM) (TABLE List:simpleMovie,Record:[W:recordIndex],Field:poster)</pattern>
  <examplePattern>
    <ex>(ejMM) (TABLE List:simpleMovie,Record:5,Field:poster)</ex>
  </examplePattern>
  <action>
    <setMEM>
      <v>movie/mmTouch/record={WILD:recordIndex}</v>
      <v>movie/mmTouch/field=poster</v>
    </setMEM>

    <function>
      <AFS function="list.getValue" saveStatus="movie/listFunction/status">
        <paramNode>
          ...
          <recordFocus>{V:movie/mmTouch/record:1}</recordFocus>
          <fieldClicked>{V:movie/mmTouch/field:poster}</fieldClicked>
          <fieldSpoken name="title"/>
        </paramNode>
        <resultNode>movie</resultNode>
      </AFS>
    </function>

    <setMEM>
      <v>movie/mmTouch/title={V:movie/fieldSpoken}</v>
    </setMEM>

    <presentation>
      <speak>{V:movie/fieldSpoken:interesting choice}!</speak>
      <text>You picked {V:movie/fieldSpoken:an interesting one}!</text>
      <semantic>Clicked on: {V:movie/fieldSpoken:something}</semantic>
    </presentation>
  </action>
</rule>
```

what rule responds to

what "touch" might send

Context manager "remembers"

Get some other data from the "touched" record

Present this to the user







# Thank you



Emmett Coin  
ejTalk, Inc  
[emmett@ejTalk.com](mailto:emmett@ejTalk.com)



MVC, April 15, 2013

[www.ejTalk.com](http://www.ejTalk.com)