Modularization of Multimodal Interaction Specification

Matthias Denecke, Kohji Dohsaka, Mikio Nakano

NTT Communication Science Laboratories
Kyoto, Japan
denecke@cslab.kecl.ntt.co.jp
1 Introduction

Modularization of dialogue systems
  • Necessary as complexity increases

Advantages
  • Encapsulation of Knowledge
    System resources: Reusability of components
    Human resources: Divide development by discipline
  • Structured system development
    Explicit integration points
1 Introduction

Problems:
- Dialogue Management not well-defined task
- No generally agreed-upon architecture

Consequence:
- An attempt to encapsulate a dialogue manager in an API will be difficult!

♫ So, let’s try something else…
1 Scope of this presentation

Figure 1: The W3C Multimodal Interaction Framework

Modular specification of interaction management
2 Modularity in Dialogue Systems

Dialogue Objects

- Prepackaged dialogue subsystems
- Reusability of application components

Disadvantages:

- Black box does not address crosscutting concerns
- Difficult to express dialogue strategies across several components
2 Modularity in Dialogue Systems

What we would like to have is…
2 Modularity in Web pages

HTML and Cascading Style Sheets

Separate:

- What is presented (HTML)
- How it is presented (CSS)

Interface:

- Tag names, class labels

Style sheets cut across multiple web pages
2 HTML and CSS Example

```html
<div class="main">
  <h1 class="header1">W3C Workshop</h1>
  The <a class="link1" href="http://www.w3c.org">W3C</a> workshop takes place on <em class="em1">July 19 and 20</em> in <em class="em2">Sophia Antipolis</em>.
</div>

```

```css
div main {
  font-size: large
}
a.link1:link {
  color: #333399
}...
```

```css
div main {
  position: absolute;
  left: 10; top: 300;
}
a.link1:link {
  color: #333399;
}...
```
2 HTML and CSS Example

W3C Workshop

The W3C workshop takes place on July 19 and 20 in Sophia Antipolis.

W3C Workshop

The W3C workshop takes place on July 19 and 20 in Sophia Antipolis.
3 How about Dialogue Systems?

In multimodal dialogue systems:
Can we separate, similar to HTML and CSS,

5. What we talk about
   • Credit card
   • Date,…

from

9. How we talk about it?
   • What is the date?
   • Please enter the date on the number pad
3 Examples

If one slot has been prompted twice, and remains unfilled or with low confidence, abort the dialogue.

If the last two times speech was used a problem occurred, actively suggest to use a different input channel.

If the user asks for help more than twice, switch modes.
Three things needed:

2. Content specification (~ HTML)
   Assuming: something like RDFS + RDF

3. Interface declaration (~ Tags + classes)
   Introduce vocabulary 1. + 3. can use
   Schema-like document

4. Interaction specification (~ CSS)
   Specify dialogue management
3 Content Representation (~HTML)

RDFS: modularized vocabulary

- Common upper ontology
- Domain specific concepts

Annotated with facets (Denecke & Yang 2000)

- ~EMMA+abstraction, partial order
- Numeric intervals and symbols

act_getinfo, high, once
ARG obj_flight, high, once, sp + gst
DEP date, high, once
DAY 17th, low, twice, sp
MON Oct, low, twice, gst

Confidence
# times prompted
Input channels
3 Interface Declaration (~Class labels)

Introduce shared vocabulary containing

1. Facets
2. Common Upper Ontology
3. Abstract dialogue state (Denecke 2000)

Abstract Dialogue State

- Collection of features describing dialogue state
- Aggregate information in facets, content
  1) Over time
  2) Over location in representations
3 Abstract Dialogue State (~Classes)

Example:

- # slots w/ low confidence in this turn
- # slots w/ low confidence up until now
- # times speech used
- # times handwriting used
- # corrections in speech channel
- # corrections in handwriting channel
3 Interaction Specification (~CSS)

Concrete representations hidden

- Use ADS, facets, Common Ontology only
- Proprietary implementations encapsulated

Express interaction management

- In terms of vocabulary defined in interface
  - Interface spec encourages reusability, but
  - Designer determines degree of domain dependence
  - Overcomes difficulties of API approach
3 Comparison

Content / HTML

Interface / Tags + Classes

Rendering

Output

Style / CSS

Content / RDF(S)

Interface / Facets + ADS

Interaction Mgr

Output

Style/Interact. Spec
3 Multimodal Interaction Framework

1 Content
   \[\text{Input Components}\]
   \[\text{Input}\]
   \[\text{State}\]
   \[\text{Output}\]

2 Interface
   \[\text{Abstraction}\]
   \[\text{Facets}\]
   \[\text{ADS}\]

3 Interaction
   \[\text{Interaction Manager}\]

\[\begin{array}{c}
\text{Input Components} \\
\text{Content}
\end{array}\]
\[\begin{array}{c}
\text{State} \\
\text{Abstraction}
\end{array}\]
\[\begin{array}{c}
\text{Output Components} \\
\text{Output}
\end{array}\]
\[\begin{array}{c}
\text{Content Selection}
\end{array}\]
4 Implementation of Interaction Mgr

IM can be seen as

- \( f: ADS \times Input \not\subseteq Output \)

Two ways:

1. Fix \( f \), specify parameters
   \( f_{\langle Parms \rangle}: ADS \times Input \not\subseteq Output \)

2. \( f \) becomes parameter to Interpretation Mgr
   Provide API or scripting language to access facets, ADS, ontology
4 Interaction Implementation Way 1

Generic multimodal algorithm \( f_{\text{Parms}} \)
- Parametrized by domain specific information
- Cf VoiceXML

Features:
- Control over application specification
  Given by parameters
- Closed system
- Tool support easy, but too limited?
4 Interaction Implementation Way 2

No generic algorithm
- Provide access to ADS, facets
- Implement own IM

Features:
- No control over application specification
  Can be anything: rule based, learned,…
- Open system
- More complex
5 Example 1

If one slot has been prompted twice, and remains unfilled or with low confidence, abort the dialogue

If (exists path(p) :

#prompts(p) == 2 &&

(confidence(p) == low ||

filler(p) == nil)

Then

abort();
5 Example 2

If the confidence of the last utterance is low, and the used channel is unreliable, suggest another channel

\[
\text{Confidence}($lastUtterance) == \text{low}
\]

\[
\text{ChannelRel}($lastChannel) \ni \text{unreliable}
\]

red : ADS variables
5 Applications: Channel Management

Observations:

1. Initial use establishes suboptimal patterns (Bhavnani 2000)
2. Multiple input channels:
   - Compensate for imperfect input
   - Quality of input component hidden

Input Channel Management necessary

1. Control interaction (vocabulary size)
2. Suggest alternative input channels
5 Applications: Affective Interfaces

Affective Interfaces (Picard 1997)

♦ React to users’ changing emotions
  ♦ Encapsulate appropriate reactions

♦ Areas:
  ♦ Telemarketing
  ♦ Health care
  ♦ User interfaces…

Empathic avatar
(Lisetti et al, 2003)
5 Applications: Virtual Personalities

Specify character in Interaction Manager

Applications:
- Education / Tutoring systems
  Didactic vs socratic teaching
  (Fiedler 2003)
- Games
- Marketing

www.yellostrom.de
6 What has been done?

Some ideas implemented

- Unimodal systems
- Facets, ADS work together with reinforcement learning (Denecke et al 2004)
- Facets, ADS allow encapsulation of rule-based dialogue strategies (Denecke et al 2003)
- Open source system www.opendialog.org
6 What is missing?

Examples require increasingly complex abstractions

- Can they be found?
- Can they be expressed in the interface declaration?
- Do they capture necessary information?

Abstractions needed for input and output
Summary

- Need for modularization in interaction mgmt
- Existing approaches insufficient
- Proposal motivated by HTML + CSS
  Allows cross cutting across application
  Requires appropriate abstractions
Thank you!

denecke@cslab.kecl.ntt.co.jp