Approaches to Dialogue
System Design

Ajay Juneja
11-716 Dialog Seminar

akj@andrew.cmu.edu
Papers covered

• An Inference-based Approach to Dialogue System Design
  – Johan Bos and Tetsushi Oka, University of Edinburgh

• The Queen’s Communicator: An Object-Orientated Dialogue Manager
  – Ian O’Neill, Philip Hanna, Xingkun Liu, Michael McTear, Queen’s University and University of Ulster, N. Ireland.

• Rapid Prototyping for Spoken Dialogue Systems
  – Matthias Denecke, CMU
The Queen’s Communicator

- Object-oriented approach
- Keeps track of negated or modified information from the user
- Allows for rule constraints and relaxing of constraints
Figure 1: High level UML class diagram for the Java dialogue manager
An Inference-based approach to Dialogue System Design

- Utilizes principles of first order logic
- Two main goals:
  - The representation language should have a level of expressiveness capable of capturing a substantial amount of natural language meaning
  - The logic should be supported by many different inference engines to provide enough capacity to support different dialogue behavior
Figure 1: Example DRS paraphrasing the utterance "Switch every light in the kitchen on".

\[ D=\{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\} \]
\[ F(\text{possible-world})=\{d_1, d_2, d_3\} \]
\[ F(\text{system})=\{ (d_1, d_4), (d_2, d_4), (d_3, d_4) \} \]
\[ F(\text{kitchen})=\{ (d_1, d_5), (d_2, d_5), (d_3, d_5) \} \]
\[ F(\text{action})=\{ (d_1, d_2, d_3) \} \]
\[ F(\text{light})=\{ (d_1, d_8), (d_2, d_8), (d_3, d_8), \]
\[ (d_1, d_7), (d_2, d_7), (d_3, d_7) \} \]
\[ F(\text{in})=\{ (d_1, d_6, d_5), (d_2, d_6, d_5), (d_3, d_6, d_5), \]
\[ (d_1, d_7, d_5), (d_2, d_7, d_5), (d_3, d_7, d_5) \} \]
Inference based approach

Syntax of DRS-conditions:

1. If $R$ is a relation symbol for an $n$-place predicate and $x_1 \ldots x_m$ are discourse referents then $R(x_1, \ldots, x_m)$ is a DRS-condition;
2. If $x_1$ and $x_2$ are discourse referents, then $x_1 = x_2$ is a DRS-condition;
3. If $B$ is a DRS, then $\neg B$, $\square B$, $\Diamond B$, $\exists B$ are DRS-conditions;
4. If $B_1$ and $B_2$ are DRSs, then $B_1 \lor B_2$, $B_1 \Rightarrow B_2$, $B_1 ? B_2$ are DRS-conditions;
5. If $x$ is a discourse referent and $B$ a DRS, then $x:B$ is a DRS-condition;
6. If $A$ is a DRS-action-term, and $B$ a DRS, then $[A]B$ and $(A)B$ are DRS-conditions;
7. If $A$ is a DRS-action-term, then $!A$ is a DRS-condition.

Syntax of DRS-action-terms:

1. If $B$ is a DRS, then $\delta B$ is a DRS-action-term;
2. If $A_1$ and $A_2$ are DRS-action-terms, then so are $(A_1;A_2)$ and $(A_1|A_2)$.
Inference based approach

- Theory of using model-builders to construct finite minimal first-order models
- Utilize Discourse Representation Theory (DRT)
- Why utilize this approach? Because there are many model builders, out of the box inference tools for first order logic.
Implementation of Interface-based approach

Figure 3: Architectural overview of the inference-based dialogue system. Boxes represent OAA-agents (see Section 5). The application shown here is dialogue in home automation (see Section 6).
Rapid prototyping for Spoken Dialogue Systems

- Also an object-orientated, class based approach.
- Implemented in Ariadne
- Utilizes Soup or Phoenix parsers.
- Dialogue state is represented by a set of six features:
  - CURRENTQUALITY, OVERALLQUALITY, CURRENTSPEECHACT, REFERENCE, REFERRINGEXPRESSIONS, INTENTION
Rapid prototyping for Spoken Dialogue Systems

- Interaction patterns are classified by the following:
  - Question, Undo, Correction, State

- Dialogues are rapidly created utilizing a class-based, hierarchal approach that does not require the creator to have a vast knowledge of dialogue systems
Comparisons: Queen

e.g. IF (failed search was to find accommodation name [e.g. Hilton, Holiday Inn, etc.]
AND constraints were location Belfast and class four-star and accommodation type hotel)
THEN relax constraint class four-star and re-do search

String userFocussedRule1 = "
[RULE]
  { AccoName UNSPECIFIED }
  { AccoType UNSPECIFIED }
[ACTION]
  { INTENTION AccoType SPECIFY }
[RULE-END]"

while (2) above appears as

String dbFocussedRule1 = "
[RULE]
  { AccoName TARGET }
  { AccoType CONSTRAINING }
  { Location CONSTRAINING }
  { AccoClass CONSTRAINING }
[ACTION]
  { RELAX " AccoClass " }
[RULE-END]"
Comparisons: Ariadne

```plaintext
enumqst {
  state:(determined = lookup),
  text:  "There are $objs.num employees with the last name $sem[EMPLOYEE|LastName]
called $objs.first[EMPLOYEE|FirstName], $objs.middle[EMPLOYEE|FirstName]
and $objs.last[EMPLOYEE|FirstName]. What is the first name of the person
you would like to lookup?"
  options: "$objs[EMPLOYEE|FirstName]"
  commands: 'undo' 'repeat' 'start over'
  location: [EMPLOYEE] <obj_employee:NP:->
};

infoqst {
  state:(determined = lookup),
  text:  "What is the first name of the person you would like to lookup?"
  options: "$db.DirectoryDB.Employees.FirstName"
  commands: 'undo' 'repeat' 'start over' 'i dont know it'
  location: [EMPLOYEE] <obj_employee:NP:->
};
```

Figure 5: Templates. The syntax has been slightly simplified for expository reasons.
Comparisons

- Ariadne and The Queen’s communicator utilize similar approaches. Both allow constraint changing, and are object oriented approaches.
- The Inference-based approach is very logic focused.
  - First order logic allows verification and automated theorem provers to be utilized for assuring dialogue consistency.
  - Object Oriented approaches are likely to run faster, since there is little known about the performance of theorem provers in linguistic problems.
  - Theorem proving and model-building approaches fail for larger dialogues and less restricted domains – takes too long.