Finite State Models

- Behavior of system described as a set of nodes and transitions
- Can also allow arbitrary code at each node
- Tedious to add common functionality
- Repeat that clarifications, help

Overview

- Introduction
- Theory
- Comparison
- EDIS System
- GoDIS
- TrindiK
- Tridi
Example: Car Buying System

• System prompts for values of a number of slots: year, make, et cetera
• Might want to give information to user: 'I have 5 cars that match that'
• Might want to order prompts based on which past questions the user has cared about: alternatives might all be automatic, so don't ask transmission type
• Might want to order prompts based on which past questions can best narrow down the alternatives
• Might want to give information to user: 'I have 5 cars that match that'

Form-Filling Dialogue Manager

• Goal of system: have user fill in values of a (hierarchical) form
• Behavior of system is declared in data structures associated with forms and slots that need to be filled in

• Alternatives make it easier to reason about it
• Big issue in AI is declarative versus procedural knowledge
• Behavior of system is declared in data structures associated with the forms and slots that need to be filled in
• Goal of system: have user fill in values of a (hierarchical) form

- If user does not care about power windows, might not care about power
- Alternatives make it easier to reason about it
- Alternatives make it easier to reason about it
- Alternatives make it easier to reason about it

- Alternatives make it easier to reason about it
What is 'Information State'?

- Definition:
  - Represents the information necessary to distinguish a dialogue from other dialogues.
  - Represents the information necessary to distinguish a dialogue from others.
  - Information that a participant has at a particular point in the dialogue.
  - Information that a participant has at a particular point in the dialogue.
  - Information that a participant has at a particular point in the dialogue.
  - Consists of:
    - Information that a participant has at a particular point in the dialogue.
    - Information that a participant has at a particular point in the dialogue.
    - Information that a participant has at a particular point in the dialogue.
    - Also referred to as conversational score, discourse context, mental state.
    - Example: statements generally add propositional information; questions generally provide motivation for others to provide specific statements.
    - In the dialogue, and motivating future action.

Can also talk of information state of each participant:
- Information that a participant has at a particular point in the dialogue.
- Information that a participant has at a particular point in the dialogue.
- Information that a participant has at a particular point in the dialogue.
- Also referred to as conversational score, discourse context, mental state.
- Example: statements generally add propositional information; questions generally provide motivation for others to provide specific statements.
- In the dialogue, and motivating future action.

What is 'Information State'?

- Represent system behavior as a set of if-then rules:
  - When it now thinks is true.
  - What it says next.
  - Need to characterize:
    - If certain condition holds, then do a certain action.
    - Represent system behavior as a set of if-then rules.
Key Issue: Update of Information State

- Information state holds everything that system should use in deciding what to do next.
- How what the user said is incorporated into information state
- How what the system does is decided from the information state

State Update

Information State

System Says Something

User Says Something
Part 1: Informational Components

- Model just the history of what has happened
- Model the beliefs, intentions, and obligations of the system

Mental State versus Dialogue History

- Private: knowledge that just the system itself has
- Public: knowledge that is assumed to be known by both participants

Private versus Public:

- Might want to include static information, which will not change
- Might want to include dynamic information, which will change

Static versus Dynamic:

What information do we want to include in the system?

Information State Theory of Dialogue Modeling

- Formal representation of the above components
- Formal representation of the above components (e.g., as first-order logic)
- Informal representations of the above components
- Description of what information will be used (e.g., participant's beliefs, facts, preferences, ...)

an update strategy for deciding which rule(s) to select at a given point in the dialog

- a set of update rules governing how information state is updated given performed dialogue moves. Also includes selection rules that specify what rule to apply
- the set of applicable ones. Which be as simple as pick the first rule that causes the first rule that

analyze previous moves. Assume semantic interpretation already done

- a set of rules for detecting when information state is updated given performed dialogue moves

the set of applicable ones. Which be as simple as pick the first rule that
Part 2: Formal Representations

• How should we model the information components?
  - What should be the semantic representation language?
    - frames, FOPC, ad-hoc
  - How should they be stored?
    - Set, list, queue, stack?

Example:
- Set, list, queue, stack?
- How should they be stored?
  - Frames, FOPC, ad-hoc
- What should be the semantic representation language?
- How should we model the information components?

Example:
- Set, list, queue, stack?
- How should they be stored?
  - Frames, FOPC, ad-hoc
- What should be the semantic representation language?
- How should we model the information components?

Example:
- Set, list, queue, stack?
- How should they be stored?
  - Frames, FOPC, ad-hoc
- What should be the semantic representation language?
- How should we model the information components?
Part 4: Update Rules

Update rules are meant to encapsulate coherent bundles of change to the information state, given a particular theory of dialogue. Rules consist of a set of applicability conditions and a set of effects.

- Effects are changes that will be made to the information state when the rule is applicable, for the rule to be applicable.
- Applicability conditions specify aspects of the information state that must be present for the rule to be appropriate.

Each rule consists of a set of applicability conditions and a set of effects.

Update rules formalize the way that information state is changed.

Dialogue

Change to the information state, given a particular theory of dialogue can be expressed in terms of the current state of the dialogue. Semantics are abstracted from the dialogue moves to express what the system needs to accomplish.

Complex issues:

- Sometimes, a single utterance has multiple functions:
  - Inform
  - Ask
  - Request
  - Acknowledge

- What distinctions semantic interpretation can reliably make:
  - When distinctions change, system needs to accomplish the task

- How speech acts should be used, depending on:
  - Information requested, acknowledged
  - When speech acts are used
  - Semantic information

- Abstraction of what the user is saying
Other examples

- Remove question from QUD once it has been answered
- Answer top question on QUD

Example of Update Rules

- Rule for adding question to QUD if ask move has been performed
  - User asks a question that system wanted to be raised
  - Make question the center of conversation
  - Rule has two conditions:
    - that the latest move was of type ask
    - the top of the agenda was the action of raising a question
  - Effects:
    - push the question that is the content of both the raise agenda item and the ask
    - pop this item from the agenda

+ push the question that is the content of both the raise agenda item and the ask
+ pop this item from the agenda

+ push the question that is the content of both the raise agenda item and the ask
+ pop this item from the agenda
Discussion

• Choices for one component influence choices for another

• Choices for formal representation influence rules

• Choices for update rules depend on update strategy

Examples:

Part 5: Update Strategy

• Need strategy to decide which of the applicable rules should be applied
• May have to redesign rules and information state to make update strategy work in the desired way
• Do not want to apply all rules, as might be contradictory

Possible strategies:

1. Take the first rule that applies (iteratively until no rules apply)
2. Apply each rule (if applicable) in sequence
3. Apply rules according to class
4. Choose among applicable rules using probabilistic information
5. Present choices to user to decide (for development modes)
Overview

Example

Sys: Where do you want to go?

Usr: Malvern

Before exchange, system has an agenda item to raise the question about the user’s destination.

After exchange, system has an agenda item to raise the...

Update rule with this condition has effect of selecting an ask...

After answer, rule for integrating user answers will check that answer matches top question on QUD, pop the question off QUD, and move question from agenda to QUD.

After system utterance, update rule will move question from QUD.

Update rule with this condition has effect of selecting an ask.

Sys: Where do you want to go?
Information-state approach lends itself to implementation of a Dialogue Move Engine (DME) as its main functions are updating information state based on observance of moves and selecting moves to be performed.

- Call implementation a Dialogue Move Engine (DME)

Information-state approach lends itself to implementation of a Dialogue Move Engine (DME)
TrindiKit also provides...

- Default modules for input, interpretation, generation, and output
- Algorithms for definition and control modules
- Methods for accessing IS
- Language for specifying update rules
- Language for selecting update rules
- Also provides definitions of datatypes

Multiple DMEs

- System might contain multiple DMEs
  - One for update (update information state after last observed)
  - One for selection (determine next move)
  - One for control module how to arbitrate or interleave

Some parts of information state might actually be external to IS

- IS uses abstract data types, each permitting a specific set of coherent ways to inspect and change the information in other modules by blocks of update rules, which can be used by queries to inspect the type and operations to change it. These rules are the building blocks of update rules, which can be used by other modules to inspect and change the information in IS.
GoDiS System

- Experimental dialogue system built using TrindiKit
- Uses fairly simple algorithms for control, update and selection
- Keywords, keyword-based interpretation and template-based generation
- Able to handle simple grounding phenomena
- Allows users to answer unasked but salient questions
- Distinguishes 8 dialogue move types
- Distinguishes 8 dialogue move types
- Distinguishes 8 dialogue move types

Overview

• Introduction
• Theory
• TrindiKit
• GODIS System
• Comparison
• EDIS System

© P. Heeman, 2003
Grounding

- Grounded: explicitly by user repetition or implicitly by user making next relevant move (e.g., question).

- Grounded: explicitly by user repetition or implicitly by user making next relevant move (e.g., question).

- As system says information, just goes into Tmp, until it is grounded.

- Alternate version: As system says information, just goes into Tmp.

Grounded: explicitly by user repetition or implicitly by user making next relevant move (e.g., question).

Private versus Shared

- Shared: information that has been explicitly established during conversation.

- Set of propositions which user accepts and are currently under discussion (not yet answered).

- Information that has been explicitly established during conversation.

- Stack of questions under discussion.

- System's short term intentions for the next turn.

- Information that has not yet been grounded.

- Tmp: mirrors shared fields.

- Can be changed during the course of conversation.

- Plan: list of actions that are long-term dialogue goals.

- Agenda: system's short term intentions for the next turn.

- Shared: propositions system holds to be true.

- System's short term intentions for the next turn.

- Information that has not yet been grounded.

- Tmp: mirrors shared fields.

- Can be changed during the course of conversation.

- Plan: list of actions that are long-term dialogue goals.

- Agenda: system's short term intentions for the next turn.

- Shared: propositions system holds to be true.

- System's short term intentions for the next turn.

- Information that has not yet been grounded.

- Tmp: mirrors shared fields.

- Can be changed during the course of conversation.

- Plan: list of actions that are long-term dialogue goals.

- Agenda: system's short term intentions for the next turn.

- Shared: propositions system holds to be true.

- System's short term intentions for the next turn.

- Information that has not yet been grounded.

- Tmp: mirrors shared fields.

- Can be changed during the course of conversation.

- Plan: list of actions that are long-term dialogue goals.

- Agenda: system's short term intentions for the next turn.

- Shared: propositions system holds to be true.
Accommodation II

• Usually, user states goal "I want to take a trip to Miami"
  - System then adopts a plan to help user with information and then updates agenda

• But, user might just say "any hotels in Miami?"
  - System then adopts a plan to help user with information and then updates agenda

Accommodation

• Accommodation: what to do when user's response does not match structures on QUD
  - When user addresses questions that have not been explicitly raised in the dialogue

J: What month do you want to go?
P: Well around 3rd April / some time there

P: as cheap as possible
P: Well around 3rd April / some time there
P: What month do you want to go?

• "Cheap as possible" not on QUD
  - Update rule adds plan action to QUD, so normal QUD rules can interpret

• "Cheap as possible" not on QUD
  - But, system will have in its plan action to raise question: what price
  - Update rule adds plan action to QUD, so normal QUD rules can interpret

Accommodation: what to do when user's response does not match structures on QUD
Example

Sys Welcome to the travel agency.

Usr flights to paris

Sys What city do you want to go from.

User utterance answers two questions: how and to

But no question was asked

User utterance answers two questions: how and to

Sys What city do you want to go from.

User utterance answers two questions: how and to

Sys Welcome to the travel agency.

Sys What city do you want to go from.

User utterance answers two questions: how and to

Sys Welcome to the travel agency.

Sys What city do you want to go from.

User utterance answers two questions: how and to

Sys Welcome to the travel agency.

Sys What city do you want to go from.

User utterance answers two questions: how and to

Sys Welcome to the travel agency.
Update Algorithm

- Store: stores current shared in prev/curr (1 rule)
- chuglause: performs database search (7 rules)
- accommodate: handles question (and task) accommodation (3 rules)
- integrate: integrates the effects of the latest move (12 rules)
- accommodate: handles accommodation (7 rules)
- chuglause: handles grounding (7 rules)
- refill: puts new actions on the agenda (6 rules)

There are currently 6 rule types and 28 rules

God's uses an update algorithm where different types of rules are applied at different stages of the update process.

Update Rule for Accommodating Questions

After plan has been accommodated, apply the following to accommodate unspecified question:

1. Rule: accommodateQuestion
2. Rule: answerPlan
3. Rule: plan
4. Rule: answerPlan
5. Rule: plan
6. Rule: answerPlan
7. Rule: plan
8. Rule: answerPlan
9. Rule: plan
10. Rule: answerPlan
11. Rule: plan
12. Rule: answerPlan
13. Rule: plan
14. Rule: answerPlan
15. Rule: plan
16. Rule: answerPlan
17. Rule: plan
18. Rule: answerPlan
19. Rule: plan
20. Rule: answerPlan
21. Rule: plan
22. Rule: answerPlan
23. Rule: plan
24. Rule: answerPlan
25. Rule: plan
26. Rule: answerPlan
27. Rule: plan
28. Rule: answerPlan
 Algorithm

• Overview:
  + Selection algorithm simply picks the first applicable rule.
  + Calls each module in turn in a serial fashion.

  Control algorithm
  + Control algorithm ensures that the current shared field is stored in private.tmp in case the next
    optimistic assumption should prove to be wrong.
  + Otherwise, the current shared field is stored in private.tmp in case the next
    optimistic assumption should prove to be wrong.
  + Finally, if the latest move was performed by the user, the agenda is filled.
  + If necessary, database searches are performed.
  + After grounding, the effects of the move are integrated into the information.

  + Selection algorithm selects the next rule which enters the latest move into
    the agenda.
  + Whenever a grounding rule is applied which enters the latest move into
    the agenda, the agenda is filled.
  + If the interpreter fails on the latest utterance, the agenda is filled.
  + If necessary, database searches are performed.

  + Control algorithm ensures that the current shared field is stored in private.tmp in case the next
    optimistic assumption should prove to be wrong.

 Code

```c
{
    if (latest_moves == failed) {
        repeat {refill}
    } else {
        repeat {integrate, accommodate, integrate, repeat {refill}
            if (latest_speaker==usr) {
                repeat {refill}
            } else {
                store
            }
        }
    }
}
```
EDIS System

- Formulation of IS based on work by Poesio and Traum

EDIS System

Comparison

EDIS System ≡

GoDiS

Trindi

Theory

Introduction

Overview
EDIS uses the same general pipelining of modules as GoDiS, however the update algorithm is a bit different. Whenever a set of dialogue acts are placed in the dialog history, the following algorithm is applied:

1. Create a new DU and push it on top of UDUs (and set CDU to each step).
2. Perform updates on the basis of backwards grounding acts, such as merging PDU.C with G for an acknowledgment.
3. If any other type of act is observed, record it in the dialogue history in CDU and apply the update rules for this kind of act.
4. Apply update rules to all parts of the IS which contain newly added acts.

Formal Representation:

```
+-----------------+-----------------+-----------------+
| (start)         | (end)           | (end)           |
| (start)         | (end)           | (end)           |
| (start)         | (end)           | (end)           |
| (start)         | (end)           | (end)           |
+-----------------+-----------------+-----------------+
```

CDU (which are not grounded)

UDUs is a list of the DUs (which may include the ones identified in PDU and/or CDU for current and PDU, the previous one

Some-public includes:

Common info shown separately for exposition purposes.
Overview

Control (Cont.)

Following deliberation, dialogue acts are selected to fulfill any obligations understood to result from conditions in the COND field of GCDU and to perform complex acts.

Also a deliberation step, applied for each system turn, is the next moves Interface variable, for intentions and placed in the system.
Information State versus Form-Filling

- Form-filling: keeps track of which slots are filled in.
- Can turn it into an information state approach.

Information State versus Dialogue State

- Dialogue State (structured dialogues)
  - Dialogue is a set of states with transitions between them.
  - Which transition is followed depends on what the user says.
  - Amount of information is very limited.

- Information State approach: potentially unlimited information
  - Can turn a dialogue state system into an information state one.
  - Information state: dialogue state, and global variables
  - Update rules: transitions from dialogue state approach.
  - Update strategy: pick the single transition that applies.