ICT Dialogue Manager Tutorial: Session 5: Implementation in Soar (II)
Outline

- **Review**
  - Soar basics
  - Dialogue Information State
  - Dialogue Processing Cycles
  - Code overview

- Content and Dialogue Act Representations
- Stepping through the system
- Code viewing
- Q&A
Implementation: SOAR (7.3)

- **Information state (Working Memory)**
- **Production rules**
  - Elaborate state rules
  - Operator Proposal rules
  - Operator application rules
- **Processing cycles**
  - Elaboration cycle
    - Invoke all rules that apply, change WM preferences
    - Calculate WM changes based on preference arbitration
  - Deliberation cycle
    - Choose operator
    - Input processing
    - Elaboration cycle+ (until no more rules apply)
    - Output processing
Example Elaboration Rule
(inference.soar)

sp {top-ps*elaborate*task*belief*intend*true

  (state <s> ^problem-space.name top-ps
       ^agent-name <me>
       ^plan <task>)

  (<task> ^intend true
       ^responsibility <me>
       ^authorized yes)

  -->

  (<task> ^belief true)

}
Example Proposal Rule (nlu.soar)

sp {top-state*propose*operator*update-dialogue-state

(state <s> ^name top-state
    ^speech-event-history <ss>)
(<ss> ^speech-input <si>)
(<si> ^processed-by understand-speech
   ^processed-by update-dialogue-state)
-->
(<s> ^operator <o> + =) ;# "indifferent" preference
(<o> ^name update-dialogue-state
    ^speech-input <si>
    ^priority-class listen)
Example Operator Application Rule (nlu.soar)

sp {top-state*apply*operator*understand-speech*gesture-processed

(state <s> ^name top-state

  ^operator <o>)

(<o> ^name understand-speech

  ^speech-input <si>

  ^gesture <new>)

(<si> ^processed-by understand-speech)

(<new> ^type gesture

  ^unprocessed yes)

-->

(<new> ^unprocessed yes -)

}
Main Aspects of Dialogue Context

- **Persistent State**
  - Social State
  - Lexicon
  - Ontology
  - Speech-event-history

- **Transient**
  - Conversation(s)
  - Social Planning
  - Participants
  - Task model
    - Causal-history, current-state
    - Plan,next-step
    - Task focus
  - Emotion Model
    - Emotional state
    - Coping strategies
Dialogue Processing Cycles

- **Dialogue Inference**
  - Elaborate-state rules

- **Language Interp Stages**
  - ASR: 4 message types (speech in, timing/text out)
  - NLU: semantic interpretation(s)
  - Perception: integrated un(der) interpreted speech
  - SOAR Understand Speech Operator
  - SOAR Update Dialogue State Operator
  - SOAR Coping Focus Operator

- **Language Production Stages**
  - Output Speech proposal (desire)
  - pre NLG Output Speech operator (intention)
    - NLG Sub-state Operators
      - External NLG
      - String look-up
      - XMLify/vrExpress call
      - NVBG
        - Beavin
      - SBM
      - TTS
        - Festival
      - GameEngine
      - Post-NLG message passing/callbacks
Communication Paths

- **Language Interpretation**
  - **ASR:**
  - Message format (human messages):
    - `vrSpeech start <id> <speaker>`
    - `vrSpeech interp <id> <interp-id> <conf> <intonation> <surface>`
    - `vrSpeech finished-speaking <id>`
    - `vrSpeech asr-complete <id>`
Understand Speech Operator

- **Triggered by complete speech input event**
  - Human (ASR,NLU)
  - Agent (vrSpeech,vrNLU)
  - Self (autonomous “event” from generation)

- **Main Purposes:**
  - Adjust semantic interpretation using soar-internal context (task, language and situation specific)
  - Reference resolution (pick out acts, entities, concepts mentioned)
  - Detect uninterpretable or ambiguous/underspecified content
  - Recognize dialogue acts
Update Dialogue State Operator

- Triggered by Interpreted Speech Input
- Purpose:
  - Update context by calculating effects of dialogue acts
Example: Soar Update rule

```xml
sp {top-state*apply*operator*update-dialogue-state*csa*order
   (state <s> ^name top-state ^operator <o>)
   (<o> ^name update-dialogue-state ^speech-input <si>)
   (<si> ^speaker <speaker> ^interpretation <i>)
   (<i> ^conversation <c> ^speech-act <csa>)
   (<c> ^grounding <cgu>)
   (<cgu> ^dialogue-history <csa>)
   (<csa> ^action order ^content <sem> ^addressee <addr>)

   -->

   (<cgu> ^obligation <obl> + &)
   (<obl> ^type obligation ^holder <addr>

   ^obligated-to <speaker> ^deadline asap
   ^sanction order ^action <sem>)
```
NLG Approach

- Asynchronous Communicative Goal Proposal
- Selection of Goal
- Content planning
- Realization: Hybrid Approach
  - External Generator (GNLG, RNLG, ANLG, DNLG)
  - Hand-crafted prompts
    - Rapid-prototyping
  - Selection of detailed sentences plans
    - Emotional impact, natural expression
  - Generic case frames
    - Broad domain coverage
  - Template generation (discussion of emotions)
Selecting Acts to Perform

- **Considerations:**
  - The turn
  - Obligations to ground
  - Obligations to repair
  - Degree of understanding of prior utterances
  - (potential) obligations to address info-request
  - Beliefs about true answers
Output Speech Operator

- Triggered by (successful) desire to speak

- Main Purposes
  - Deliberation over how to achieve communication goal
  - Content planning
    - Sentence planning
  - Realization
  - Selection
  - Produce speech & wait for callback
Example Output Speech Rule

sp {top-state*propose*operator*output-speech*accept-obligation-to-act
  (state <s> ^name top-state ^agent-name <me> ^conversation <c>
    ^social-planning <sp>)
  (<sp> ^my-potential-obligation <obl>)
  (<obl> ^obligated-to <other> ^action <a>
    ^dialogue-state needs-discussion ^relevant-party <me>
    ^plan-state << good considered-good not-in-coa >>)
  (<c> -^turn <other> ^grounding <cgu1>
    ^participant <me> ^participant <other>)
  (<cgu1> ^dialogue-history <order>)
  (<order> ^action << order request >> ^actor <other>
    ^addressee <me> ^content <a>)
  -->
  (<s> ^operator <o> + =)
  (<o> ^name output-speech ^priority-class respond
    ^conversation <c> ^goal <b>)
  (<b> ^action accept
    ^type backward
    ^addressee <other>
    ^speaker <me>
    ^content <order>)
}
Example Output Speech Rule: take initiative

sp {top-state*propose*operator*output-speech*take-initiative-agenda-no-strategy

(state <s> ^name top-state ^agent-name <me>
  ^social-planning <sp> ^conversation <c>
  ^social-state <ss> -^plan-status update-needed)
-(<c> -^turn <me> -^turn *none*)
(<sp> ^agenda <ag> ^take-initiative yes)
(<ag> -^strategy.strategy << delay negative >> ^next <item> )
-->

(<s> ^operator <o> + = <)
(<o> ^name output-speech ^priority-class advance-task ^agenda <item>))
Task Model: Basic Types

- **States**
  - Object-id
  - Attribute
  - Value
  - Polarity
  
  E.g.: `:object-id boy :attribute health-status :value critical-injuries :polarity positive`

- **Tasks**
  - Pre, Add, Delete (states)
  - Case roles (event, agent, patient, location, source, destination, instrument, path)
  
Task Model: Plans

- **Task Model:** Set of steps and relations

- **Step**
  - Primitive or abstract
    - Subtasks, decomposition
  - Roles (for group task)
    - Responsibility
    - Authority

- **Links**
  - Ordering constraints
  - Causal links
  - Threat relations

- **Indirect action**

- **Partial Order Planning Algorithm**
  - Updated as world changes
Alternative Courses of Action

- Hierarchical tasks representing mutually exclusive recipes
- Relevant vs intended
  - All subtasks in COA marked
  - Alternative COAs not intended
- No threat from elements of alternative COA
- Identification of salient positive and negative consequences (side effects)
- Preferences based on utility
  - best, better than, worst
Proposition and Questions

- Duran is in the landing zone.
  - (P1 ^attribute location ^object-id 3s1 ^polarity positive ^time present ^type state ^value lz)

- Is Duran in the landing zone?
  - (Q1 ^q-slot polarity ^prop (P2 attribute location ^object-id 3s1 ^time present ^type state ^value lz) ^type question)

- Where is Duran?
  - (Q2 ^q-slot value ^prop (P3 attribute location ^object-id 3s1 ^polarity positive ^time present ^type state) ^type question)

- Who is in the landing zone?
  - (Q3 ^q-slot object-id ^prop (P4 attribute location ^polarity positive ^time present ^type state ^value lz) ^type question)
Dialogue Acts (1):
Core Speech acts (CSA): Forward

- (A1 \textit{^action} info-req \textit{^actor <speaker>} \textit{^addressee <adr>*} \textit{^content <Q> ^type csa})
- (A2 \textit{^action assert} \textit{^actor <speaker>} \textit{^addressee <adr>*} \textit{^content <P> ^type csa})
- (A2 \textit{^action order} \textit{^actor <speaker>} \textit{^addressee <adr>*} \textit{^content <P> ^type csa})
- (A2 \textit{^action request} \textit{^actor <speaker>} \textit{^addressee <adr>*} \textit{^content <P> ^type csa})
- (A2 \textit{^action thank} \textit{^actor <speaker>} \textit{^addressee <adr>*} \textit{^type csa})
- (A2 \textit{^action greeting} \textit{^actor <speaker>} \textit{^addressee <adr>*} \textit{^type csa})
- (A2 \textit{^action closing} \textit{^actor <speaker>} \textit{^addressee <adr>*} \textit{^type csa})
Dialogue Acts (2):
Core Speech acts (CSA): Backward

- (A3 ^action answer ^actor <speaker> ^addressee <adr>* ^answer <CSA-or-param-or-obj> ^question <Q> ^type backward)
- (A3 ^action divert ^actor <speaker> ^addressee <adr>* ^response <CSA> ^question <Q> ^type backward)
- (A3 ^action accept ^actor <speaker> ^addressee <adr>* ^content <CSA> ^style ^param > ?^manner <param> ^type backward)
- (A3 ^action hold ^actor <speaker> ^addressee <adr>* ^content <p-or-csa> ^style <param> ^type backward)
- (A3 ^action check ^actor <speaker> ^addressee <adr>* ^context <CSA> ^num-candidates <integer> ^type backward)
- (A3 ^action reject ^actor <speaker> ^addressee <adr>* ^content <CSA> ^reason <obj-or-param> ^type backward)
- (A3 ^action counterpropose ^actor <speaker> ^addressee <adr>* ^content <CSA> ^conversation <c> ^reason <param> ^counterproposals <p> ^type backward)
- (A3 ^action express ^actor <speaker> ^addressee <adr>* ^content <CSA> ^express <param> ^act ?^role < role> ?^issue <param> ^type backward)
- (A3 ^action redirect ^actor <speaker> ^addressee <adr>* ^content <CSA> ^relevant-party <agent> ^type backward)
- (A4 ^action clarify-parameter ^cand <cand>* ^context <SA> ^num-candidates <integer> ^parameter <slot> ^candidate <val>* ^type backward)
Dialogue Acts (3): Grounding

- **Grounding**
  - (A8 ^type initiate ^actor <speaker> ^cgu <cgu> ^content <SA>* ^conversation <CON>* )
  - (A8 ^type continue ^actor <speaker> ^cgu <cgu> ^content <SA>* ^conversation <CON>* )
  - (A7 ^type acknowledge ^actor <speaker> ^cgu <cgu> ^content <SA>* ^conversation <CON> )
  - (A5 ^type repair ^actor <speaker> ^cgu <cgu> ^content <SA>* ^context <SA2>* ^conversation <CON> ^parameter <slot> ^value <filler> ?^confirm <sa3> ?^remove <sa4>)
  - (A6 ^type request-repair ^actor <speaker> ^cgu <cgu> ^content <SA2>* ^conversation <CON> )
  - (A7 ^type cancel ^actor <speaker> ^cgu <cgu> ^conversation <CON> )
Dialogue Acts (4): Conversation and Turn-taking

**Conversation**
- (A8 \(^\text{action start-conversation}\) \(^\text{speaker <speaker>}\) \(^\text{addressee <addr>}\) \(^\text{mode <param>}\) \(^\text{conversation <CON>}\))
- (A8 \(^\text{type confirm-start}\) \(^\text{actor <speaker>}\) \(^\text{conversation <CON>}\))
- (A8 \(^\text{type deny-start}\) \(^\text{actor <speaker>}\) \(^\text{conversation <CON>}\))
- (A8 \(^\text{type identify-topic}\) \(^\text{actor <speaker>}\) \(^\text{conversation <CON>}\) \(^\text{addressee <addr>}\) \(^\text{topic <param>}\))
- (see also closing in forward acts)

**Turn-taking**
- (A8 \(^\text{type take-turn}\) \(^\text{actor <speaker>}\) \(^\text{conversation <CON>}\))
- (A8 \(^\text{type hold-turn}\) \(^\text{actor <speaker>}\) \(^\text{conversation <CON>}\))
- (A7 \(^\text{type assign-turn}\) \(^\text{actor <speaker>}\) \(^\text{assignee <agent>}\) \(^\text{conversation <CON>}\))
- (A5 \(^\text{type release-turn}\) \(^\text{actor <speaker>}\) \(^\text{conversation <CON>}\))
- (A6 \(^\text{type request-repair}\) \(^\text{actor <speaker>}\) \(^\text{cgu <cgu>}\) \(^\text{content <SA2>}\) \(^\text{conversation <CON>}\))
- (A7 \(^\text{type cancel}\) \(^\text{actor <speaker>}\) \(^\text{cgu <cgu>}\) \(^\text{conversation <CON>}\))
Team Negotiation  (Traum et al AAMAS 2003)

- IS: task (&CGU) annotated with negotiation objects
  - Components: Agent, Action, Stance, audience, reason
    - Stances: Committed, endorsed, mentioned, not mentioned, disparaged, rejected

- Action effects:
  - Suggestion: mentioned
  - command, promise, request, or acceptance: committed
  - Rejection: rejected
  - Counterproposal: disparaged
  - Justification: endorsed or disparaged (depending on direction)
  - Offer: mention (conditional commitment)
  - Retract stance

- Factors:
  - Relevant Party: Authorizing or Responsible Agent
  - Dialogue State: who has discussed
  - Plan State: how do I feel about it
Dialogue Code Files Overview: Austin/DIALOGUE.config

- **coresteve**
  - General Dialogue
    - dialog-init.soar
    - inference.soar
  - Understanding & update
    - nlu.soar
    - reference.soar
    - ref-candidate.soar
    - Expectations.soar
    - feedback.soar
  - Generation
    - nlg.soar
    - output-strings.soar
    - Initiative.soar
    - test-nlg.soar
  - Conversations and Dialogue acts (understand and update)
    - conversation.soar (also nlg)
    - csas.soar
    - grounding.soar
    - turn-taking.soar
    - negotiation.soar

- **Saso-EN-doctor-perez**
  - Domain-specific
    - Saso-en-dialogue.soar
    - Lexicon.soar
    - Saso-en-output-strings{-negative/positive}.soar
Dialogue Code Files (1)
Overview: Austin/DIALOGUE.config

- **coresteve**
  - dialog-init.soar (general initializations)
    - Ontology object, social-state object, social-planning, speech-event history, nlp-flags, things-i-said
  - reference.soar, ref-candidate.soar
    - perform (task-related) reference resolution
  - inference.soar
    - (general inference rules, e.g. relating location of group to individual)
  - csas.soar
  - grounding.soar
  - turn-taking.soar
  - negotiation.soar
    - Collaborative negotiation
Reference Resolution
Taking Initiative

- What to communicate
  - Task model
  - Emotion model
  - Special domain-specific rules

- When to communicate
  - Response only
  - Too much silence
  - Too much misunderstanding
  - Too much irrelevance
  - Directed by other

- How to communicate
  - Questions
  - Hints
  - Suggestions
  - Performances