Planning Dialog Actions

Mark Steedman    Ron Petrick

School of Informatics
University of Edinburgh
Edinburgh, Scotland, UK

Presented at SIGdial 2007 (2 September 2007)

Antwerp, Belgium
Motivation

• An agent operating in a dynamic world often needs to do so with incomplete information about its environment, by
  – Making decisions based on what it knows or believes
  – Reasoning about the effects of its actions
  – Sensing the world to gather information

• Example: dialling an unknown telephone number
  – Read the number from a piece of paper (Action Planning)
  – Ask another agent for the number (Dialogue Management)

• “Reading” and “asking/telling” are instances of sensing or knowledge-producing actions

• Planning dialogue actions is an instance of the general problem of planning with incomplete information and sensing
Previous approaches

• Treating dialogue management as planning is not a new idea

• Recent work has tended to separate action planning and dialogue planning and focused on specialized approaches
  – Finite state-transition machines
  – Information state
  – Rule-based approaches to speech act theories, dialogue games, textual coherence, etc.

• Planning is often rejected on complexity grounds
Our approach

- Adapt recent techniques from the knowledge representation and planning communities to mixed-initiative dialogue
  - Represent sensing actions that return large or infinite sets of possible outcomes using knowledge-level variables
  - Restrict complexity for tractable reasoning
- Extend the **Linear Dynamic Event Calculus** (LDEC) as a formal language for representing dialogue domains (Steedman 1997, 2002)
- Use intuitions from the **PKS planner** (Planning with Knowledge and Sensing) (Petrick & Bacchus 2002, 2004)
- Goal: PKS as a target platform for generating dialogue plans
Linear Dynamic Event Calculus (LDEC)

- Logical formalism based on Event Calculus, Situation Calculus, STRIPS planner, Dynamic and Linear Logics (Steedman 1997, 2002)
- World properties are modelled by logical fluents, e.g., temperature, doorOpen, objLoc(x), ...
- Actions provide the sole means of change in the world and are defined by a deterministic necessity modality $[\alpha]$

Examples

$$[\alpha] F$$

"$F$ must be true after performing $\alpha$."

$$[\alpha_1; \alpha_2] F_1 \land \neg F_2 \land F_3$$

"$F_1$ must be true, $F_2$ must be false, and $F_3$ must be true after performing $\alpha_1$ followed by $\alpha_2$."

Mark Steedman, Ron Petrick
Planning Dialog Actions / Presented at SIGdial 2007 / 2007-09-02
LDEC and dialogue modelling

- Modalities for representing multi-agent dialogue

\[
\begin{align*}
[S] & \quad \text{Speaker supposition} \\
[H] & \quad \text{Hearer supposition} \\
[X],[Y],... & \quad \text{Other participant/agent suppositions} \\
[C_{XY}] & \quad \text{Common ground between X and Y}
\end{align*}
\]

Examples

\[
\begin{align*}
[S]p & \quad \text{“The speaker supposes } p.\text{”} \\
[S][H]p & \quad \text{“The speaker supposes the hearer supposes } p.\text{”} \\
[H][C_{SH}][S]p & \quad \text{“The hearer supposes it’s common ground between the speaker and hearer that the speaker supposes } p.\text{”}
\end{align*}
\]
LDEC domains

- **LDEC domains** are formally described by 3 sets of axioms
  - Action precondition axioms
    \[ F_1 \land F_2 \land \ldots \land F_k \Rightarrow \text{affords}(\alpha) \]
  - Effect axioms
    \[ \{\text{affords}(\alpha)\} \land F_1 \land F_2 \land \ldots \land F_m \Rightarrow o[\alpha]F'_1 \land F'_2 \land \ldots \land F'_n \]
  - Initial situation axioms
    \[ F_1 \land F_2 \land \ldots \land F_l \]

- **STRIPS-style updates** are built directly into the effect axioms
- Linear implication \(o\) provides a solution to the **frame problem** (McCarthy & Hayes 1969) without explicit frame axioms
LDEC and knowledge

- Conversational acts like *ask/tell* can be viewed as instances of sensing or knowledge-producing actions.
- Original version of LDEC does not model knowledge which is required to represent and reason about sensing actions.
- Our solution: apply the intuitions of PKS for building plans with incomplete knowledge and sensing actions.
• A “knowledge-level” planner that builds plans based on what an agent knows (Petrick & Bacchus 2002, 2004)
• Based on an extension of the STRIPS representation
• Planner’s knowledge state is represented by 5 databases, each of which models a different type of knowledge
• Knowledge is restricted for tractable reasoning (cf. world-level planners with general representations)
• Efficiency has been demonstrated on traditional benchmarks and novel scenarios
Restricted knowledge in PKS databases

- $K_f$: knowledge of positive and negative facts
  
  \[ p \in K_f : \text{the agent knows } p \]

- $K_w$: knowledge of binary sensing effects
  
  \[ \phi \in K_w : \text{the agent knows } \phi \text{ or knows } \neg \phi \]

- $K_v$: knowledge of function values, multi-valued sensing effects
  
  \[ f \in K_v : \text{the agent knows the value of } f \]
### PKS sensing actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Preconditions</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>readPaper</code></td>
<td><code>KhavePaper</code></td>
<td><code>add(K_v, phoneNumber)</code></td>
</tr>
<tr>
<td><code>dial</code></td>
<td><code>K_v phoneNumber</code></td>
<td><code>add(K_f, dialledOk)</code>&lt;br&gt;<code>add(K_w, connected)</code></td>
</tr>
</tbody>
</table>

- $K_v$ and $K_w$ model the information returned from sensing actions<br>$\Rightarrow$ compact representation, tractable reasoning
• Introduce PKS-style knowledge assertions into LDEC using **knowledge fluents** (Demolombe & Pozos Parra 2000)

$$K_p \quad \text{“Know } p\text{”}$$

$$K_v t \quad \text{“Know the value of } t\text{”}$$

$$K_w p \quad \text{“Know whether } p\text{”}$$

• Knowledge fluents are treated as ordinary fluents that have particular meaning with respect to the knowledge state.

• $K_v$ and $K_w$ act as “placeholders” for indefinite information ⇒ sensed values.

**Examples**

$$[X] K_v phoneNumber \Rightarrow \text{affords}(dial(X))$$

$$\{\text{affords}(dial(X))\} \rightarrow o [dial(X)] [X] K_w connected$$
We also require a set of general purpose rules for reasoning about speaker-hearer and common ground modalities

A1. \([X] p \Rightarrow p\)  
Supposition Veridicality

A2. \([X] \neg p \Rightarrow \neg [X] p\)  
Supposition Consistency

A3. \(\neg [X] p \Rightarrow [X] \neg [X] p\)  
Negative Introspection

A4. \([C] p \Leftrightarrow ([S] [C] p \land [H] [C] p)\)  
Common Ground

A5. \([X] [C] p \Rightarrow [X] p\)  
Common Ground Veridicality

We do not require axioms for reasoning within the scope of the knowledge fluents \(K/K_v/K_w\)
Example: catching a train
Initial facts

F1. “If I know what time it is then I know what train I will catch.”

\[[S] K_v \text{time} \Rightarrow [S] K_v \text{train}\]

F2. “I don’t know what train I will catch.”

\[[S] \neg K_v \text{train}\]

F3. “I suppose you know what time it is.”

\[[S] [H] K_v \text{time}\]

F4. “I suppose it’s not common ground I don’t know what time it is.”

\[[S] \neg [C_{SH}] \neg [S] K_v \text{time}\]
Actions: \(\text{ask}(X, Y, p)\) and \(\text{tell}(X, Y, p)\)

R1. “If \(X\) doesn’t know \(p\) and \(X\) supposes \(Y\) does, \(X\) can ask \(Y\) about it.”

\[
\neg [X] p \land [X] [Y] p \Rightarrow \text{affords}(\text{ask}(X, Y, p))
\]

R2. “If \(X\) asks \(Y\) about \(p\), it makes it common ground \(X\) doesn’t know it.”

\[
\text{affords}(\text{ask}(X, Y, p)) \rightarrow (\text{ask}(X, Y, p)) \land [C_{XY}] \neg [X] p
\]

R3. “If \(X\) supposes \(p\), and \(X\) supposes \(p\) is not common ground, \(X\) can tell \(Y\) \(p\).”

\[
[X] p \land [X] \neg [C_{XY}] p \Rightarrow \text{affords}(\text{tell}(X, Y, p))
\]

R4. “If \(X\) tells \(Y\) \(p\), \(Y\) stops not knowing it and starts to know it.”

\[
\{\text{affords}(\text{tell}(X, Y, p))\} \land \neg [Y] p \rightarrow (\text{tell}(X, Y, p)) [Y] p
\]
Planning a direct speech act

Goal: I need to know which train I will catch

(D1) ⇒ [H] K_v time                                      (F3),(A1)
(D2) ⇒ ¬ [S] K_v time                                    (F2),(A2),(F1)
(D3) ⇒ affords(ask(S, H, K_v time))                      (D2),(F3),(R1)
   * ask(S, H, K_v time)
(D4) ⇒ [C_{SH}] ¬ [S] K_v time                           (D3),(R2)
(D5) ⇒ affords(tell(H, S, K_v time))                     (D1),(D4),(A4),(A5),(R3)
   * tell(H, S, K_v time)
(D6) ⇒ [S] K_v time                                      (D5),(D2),(R4)
(D7) ⇒ [S] K_v train                                     (D6),(F1)

- After the speaker asks the hearer for the time the hearer tells the speaker the time.
Planning an indirect speech act

Goal: I need to know which train I will catch

(D1) ⇒ [S] ¬ [S] \(K_v time\)  (F2),(A2),(F1),(A3)
(D2) ⇒ [S] ¬ [C_{SH}] ¬ [S] \(K_v time\)  (F4)
(D3) ⇒ affords(\textit{tell}(S, H, ¬ [S] \(K_v time\)))  (D1),(D2),(R3)
  * \textit{tell}(S, H, ¬ [S] \(K_v time\))
(D4) ⇒ [C_{SH}] ¬ [S] \(K_v time\)  (R2)
  ⇒ ...
  * \textit{tell}(H, S, \(K_v time\))
  ⇒ ...

- After the speaker says “I don’t know what time it is” the hearer tells the speaker the time.
Observations about generated plans

- Plan generation in the space of multi-participant plans
  - No reasoning about other participants’ goals
  - Cannot guarantee other participants’ actions
- Actions *ask* and *tell* are treated as sensing actions in our plans
- Conclusions follow from direct rule applications
- Both direct and indirect speech acts result from the machinery for reasoning about knowledge and common ground
- All speech acts are indirect in the sense of involving inference
Conclusions

- Recent techniques from the KR and planning communities are applied to mixed-initiative dialogue management.
- Approach is driven by the knowledge state without specific conversational rules (except common ground consistency).
- Current rule set plus STRIPS-style updates are sufficient (in certain scenarios).
- Illocutionary acts like questioning and requesting emerge from general rules manipulating common ground and knowledge.
- This work offers a way to generate plans directly from first principles and provides a basis for automation.
Future work: dialogue planning with PKS

• Use PKS as a target platform for dialogue planning
• Syntactically compile LDEC axioms into PKS actions
• Implement inference rules required to support speaker/hearer modalities in PKS
• Manage common ground as an instance of local closed world (LCW) information (Etzioni et al. 1994)
• Evaluation
  • Efficiency and effectiveness have been demonstrated on traditional planning benchmarks
  • We believe performance on dialogue problems will be similar
• We are looking for challenging problems!


References...(2)


