xperience.org stamina-robot.eu

Combining Plan Recognition, Goal Reasoning, and Planning for Cooperative Task Behaviour

Christopher Geib
Drexel University, Philadelphia, PA, USA

Bart Craenen
Newcastle University, Newcastle upon Tyne, UK

Ron Petrick
Heriot-Watt University, Edinburgh, UK

IJCAI Workshop on Goal Reasoning (GRW 2016)
NYC NY, USA
9 July 2016

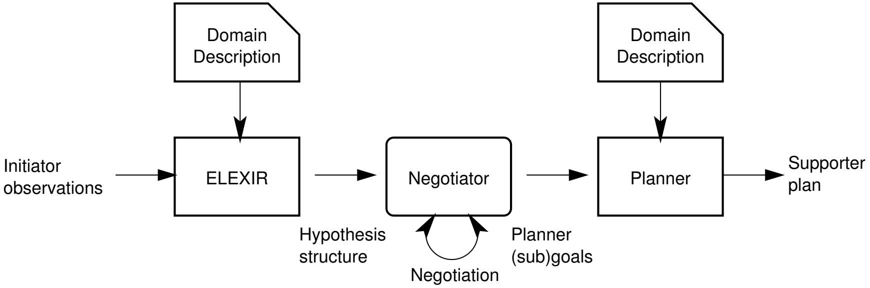
Motivation

- The ability of an agent to **help** another agent is a critical attribute when designing artificial entitles that must operate together with humans.
- The task of deciding how to help can be difficult:
  - Must recognise the goals or intentions of other agents,
  - Reason about potential opportunities to contribute,
  - Generate appropriate actions, and
  - Potentially communicate such information to the agents involved.
- Computational cost can be impractical if we consider reasoning over the joint space of all agents, actions, and goals.
- However, there are also scenarios where the reasoning can be simplified. E.g., setting a table for dinner by placing the plates, knives, and spoons.

This work: restricted case of collaborative behaviour

- An agent called the **supporter** wants to helps another agent called the **initiator** achieve its goals.
- Initiator: could be an artificial or human agent.
- Supporter: use **plan recognition** + (lightweight) **negotiation** + **planning**
  - Plan recognition is used to recognise the initiator's overall goal and subgoals that haven't yet been performed.
  - Appropriate subgoals are chosen and proposed to the initiator.
  - Agreed upon subgoals are passed to a planner which constructs a plan to execute to help the initiator.
- No centralised planning, no joint actions, independent subgoals.

Framework and operation



1. Supporter observes action by initiator
2. Observations are fed into ELEXIR, producing hypotheses structure
3. Hypotheses structure is used in negotiation process
4. Negotiation uses directed search, producing goals for the planner
5. The planner uses goals in attempt to generate supporter plan

Plan recognition

- Plan recognition attempts to identify the goal being pursued by another agent.
- We require knowledge of the plan being executed, including:
  - which subgoals have been achieved, and
  - those future subgoals that have yet to be started.
- We are not performing activity recognition. Activity recognition just identifies what one is doing now. It says nothing about future subgoals.
- We use the Engine for LEXicalized Intent Recognition (ELEXIR)
  (Geib and Steedman 2007; Geib 2009; Geib and Goldman 2011)

ELEXIR

- ELEXIR views the problem as probabilistic parsing.
  - Parse a sequence of observed actions using a probabilistic plan grammar based on Combinatory Categorial Grammars (CCG) (Steedman 2000) into tree structured explanations of the plans.
  - Build the complete and covering set of such explanations consistent with the grammar.
  - Use the probabilities of each explanation to compute the conditional probability of goals.
- ELEXIR:
  - recognises partially ordered plans, multiple concurrent and interleaved plans, and plans with looping constructs,
  - supports differential encoding of high criticality plans for early recognition, and
  - has state of the art runtimes.

ELEXIR: CCG action grammar

$$\begin{aligned}
\textbf{set-forks} :=& SetTable/\{SetKnives, SetSpoons, SetPlates, SetGlasses\} \mid \\
& (CleanForks/\{PutAwayForks\})/\{WashForks\}.\\
\textbf{set-knives} :=& SetKnives. \;\; \textbf{set-spoons} := SetSpoons.\\
\textbf{set-plates} :=& SetPlates. \;\; \textbf{store-forks} := PutAwayForks.
\end{aligned}$$

ELEXIR: explanations

- Because ELEXIR is able to support hypotheses that an agent is engaged in multiple plans at the same time, we represent an individual recognised plan explanation as a tuple with the following form:

$$(P, [\{G_i : \{\mathsf{sg}_1, ..., \mathsf{sg}_n\}^*\}^+])$$

- Where:
  - P is the probability of the explanation,
  - G_i is the top level goal of the plan,
  - each set $\{\mathsf{sg}_1, ..., \mathsf{sg}_n\}$ represents a set of subgoals that are unordered with respect to each other, but must be solved before those subgoals in any subsequent set.
- Example:

$$(0.95, [\{\mathsf{SetTable}:\{\mathsf{SetKnives, SetSpoons, SetPlates}\}\}]),$$
$$(0.045, [\{\mathsf{CleanForks}:\{\mathsf{WashForks}\}\{\mathsf{PutAwayForks}\}\}),$$
$$(0.005, [\{\mathsf{CountingForks}:\{\}\}]).$$

Negotiation and subgoal identification

- Plan recognition produces a goal/subgoal hypotheses structure.
- Negotiation first tries to confirm goal of initiator's high-level plan
  - Saves time wasted on suggesting all achievable subgoals
  - Use hypothesis structure by ranking goals using probabilities
  - Supporter verifies initiator's high-level plan by simply query
- Then attempt to identify subgoals shared with identified goal
  - Example hypothesis:

$$(0.95, [\{\text{SetTable:}\{\text{SetKnives, SetSpoons, SetPlates}\}\}])$$

  - supporter could suggest SetKnives, SetSpoons, and/or SetPlates
- Use of the ELEXIR hypothesis structures in negotiation is a directed search: first goals, then (recursively) subgoals.

Plan generation

- Once negotiation has produced a set of subgoals, the supporter must generate a plan to execute.
- For the current work, any off-the-shelf planner can be used. We use PKS (Planning with Knowledge and Sensing) (Petrick and Bacchus 2002, 2004) to build knowledge-level plans.
- Planning domain is defined by: initial state, actions, goals. Actions are predefined and the initial state is taken from the plan recognizer's context.
- Goals are generated by a syntactic compilation process based on the agreed upon subgoals.

Actions and plans

```
action grasp(?h : hand, ?l : loc, ?o : obj)
    preconds: K(graspable(?o, ?h)) &
              K(objectAt(?o, ?l)) &
              K(holding(?h) = nil)
    effects:  add(Kf, holding(?h) = ?o),
              del(Kf, objectAt(?o, ?l))

action putdown(?h : hand, ?l : loc, ?o : obj)
    preconds: K(holding(?h) = ?o)
    effects:  add(Kf, objectAt(?o, ?l)),
              add(Kf, holding(?h) = nil)
```

```
Example plan: grasp(left,drawer,fork1),
              putdown(left,table_pos1,fork1),
              grasp(left,drawer,fork2),
              putdown(left,table_pos2,fork2)
```

Scenario 1

- **Objective**: Base case: correct goal and subgoal identification
- **Hypothesis**: $(0.8, [\{\text{SetTable:}\{\text{SetPlates}, \text{SetSpoons}, \text{SetGlasses}\}\}])$
- **Negotiation**:

	Supporter	Initiator
1.	Are you setting the table?	Yes.
2.	Do you want me to set the plates?	Yes.
3.	Do you want me to set the spoons?	Yes.
4.	Do you want me to set the glasses?	Yes.

- **Partial plan** SetPlates:

```
grasp(left,sidetable,plate1),
grasp(right,sidetable,plate2),
putdown(left,table_pos1,plate1),
putdown(right,table_pos2,plate2).
```

- **Outcome**: Correct (partial) plans for goals and subgoals

Scenario 2

- **Objective**: Incorrect goal identification
- **Incorrect Hypothesis**:
  (0.8, [{CleanForks:{WashForks}{PutAwayForks}}])
- **Correct Hypothesis**:
  (0.8, [{SetTable:{SetPlates, SetSpoons, SetGlasses}}])
- **Negotiation**:

	Supporter	Initiator
1.	Are you cleaning the forks?	No.
2.	Are you setting the table?	Yes.
3.	Do you want me to set the plates?	Yes.
4.	Do you want me to set the spoons?	Yes.
5.	Do you want me to set the glasses?	Yes.

- **Outcome**: Framework able to recover from incorrect goal identification

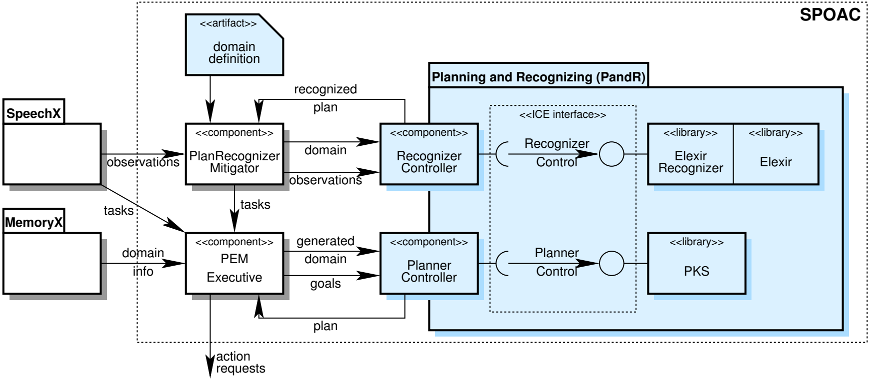
Scenario 3

- **Objective**: Incorrect subgoal identification
- **Hypothesis**: $(0.8, [\{\text{SetTable:}\{\text{SetPlates, SetSpoons, SetGlasses}\}\}])$
- **Negotiation**:

	Supporter	Initiator
1.	Are you setting the table?	Yes.
2.	Do you want me to set the plates?	No.
3.	Do you want me to set the spoons?	Yes.
4.	Do you want me to set the glasses?	Yes.

- **Outcome**: Framework able to recover from incorrect subgoal identification

Robot integration and testing



Discussion

- Motivated by the supporter being **proactive**. No centralised planning.
- Approach currently involves alignment between plan recognition domain and planning domain
- Larger role for the planner in the future
  - Reasoning about subgoal achievability
  - Reasoning about multiagent knowledge
  - Communicative actions involving other agents
  - E.g., setting wine glasses for people who want wine.
- Connections with multiple research areas: multiagent systems, multiagent planning, decomposition methods, hybrid architectures, dialogue-based coordination, ...

Conclusion and future work

- Approach combined plan recognition and planning, with successful operation relying on appropriate subgoal identification by the plan recogniser plus negotiation through a lightweight process.
- System is built with the planner and plan recogniser as C++ libraries and APIs exposed through ZeroC's Internet Communication Engine (ICE), a modern middleware for distributed computing.
- Approach has been implemented on a real robot platform. We are currently extending it to more complex robot scenarios.
- We are also exploring the notion of "help" in a formal framework, with a definition based on planning graph structures.
  - Reasoning about multiagent knowledge
  - Mapping to recursive language constructs (agent A helps agent B help agent C achieve G)

Thanks!

References

Geib, C. and Steedman, M. (2007). On natural language processing and plan recognition. In *Proceedings of IJCAI 2007*.

Geib, C. W. (2009). Delaying commitment in probabilistic plan recognition using combinatory categorial grammars. In *Proc. of IJCAI 2009*, pages 1702–1707.

Geib, C. W. and Goldman, R. P. (2011). Recognizing plans with loops represented in a lexicalized grammar. In *AAAI*.

Petrick, R. P. A. and Bacchus, F. (2002). A knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of the International Conference on Artificial Intelligence Planning and Scheduling (AIPS 2002)*, pages 212–221.

Petrick, R. P. A. and Bacchus, F. (2004). Extending the knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 2004)*, pages 2–11.

Steedman, M. (2000). *The Syntactic Process*. MIT Press.