Using General-Purpose Planning for Action Selection in Human-Robot Interaction

Ron Petrick
Department of Computer Science / Edinburgh Centre for Robotics
Heriot-Watt University, Edinburgh, UK
http://petrick.uk/

Mary Ellen Foster
School of Computing Science
University of Glasgow, Glasgow, UK
http://maryellenfoster.uk/

AAAI 2016 Fall Symposium on Artificial Intelligence for Human-Robot Interaction (AI-HRI)
Arlington, VA, USA
17 November 2016
Motivation

**Action selection** is a fundamental task in both the research areas of **AI planning** and **interactive systems**.

Numerous **interactive systems toolkits exist** — but they mix representation, reasoning, and technical infrastructure.

In AI planning, there has been a focus on **general purpose representation languages** and **multiple planning engines**.
Automated planning for HRI

**Action**: ask-drink(?a : agent)
- **Preconditions**: $K(\text{inTrans} = ?a) \land \neg K(\text{ordered}(?a)) \land 
  \neg K(\text{otherAttentionRequests}) \land 
  \neg K(\text{badASR}(?a))$
- **Effects**: $\text{add(Kf,ordered(?a))}, \text{add(Kv,\text{request}(?a))}$

**Action**: serve(?a : agent, ?d : drink)
- **Preconditions**: $K(\text{inTrans} = ?a) \land K(\text{ordered}(?a)) \land 
  K(\text{request}(?a)) \land K(\text{request}(?a) = ?d) \land 
  \neg K(\text{otherAttentionRequests}) \land 
  \neg K(\text{badASR}(?a))$
- **Effects**: $\text{add(Kf,\text{served}(?a))}$

<table>
<thead>
<tr>
<th>Plan step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>greet(a1), ask-drink(a1), ack-order(a1), serve(a1, request(a1)), bye(a1).</td>
<td>Greet agent a1, Ask a1 for a drink order, Acknowledge a1’s order, Serve a1’s ordered drink, End the interaction</td>
</tr>
</tbody>
</table>
Case study: a robot bartender

http://james-project.eu/  photo: fortiss GmbH
Using General-Purpose Planning for Action Selection in Human-Robot Interaction

**Ronald P. A. Petrick**, Heriot-Watt University  
http://petrick.uk/  
**Mary Ellen Foster**, University of Glasgow  
http://maryellenfoster.uk/

**Action selection** is a fundamental task in both the research areas of AI planning and interactive systems.

**Numerous interactive systems toolkits exist** – but they mix representation, reasoning, and technical infrastructure.

In AI planning, there has been a focus on general purpose representation languages and multiple planning engines.

We aim to revisit the use of techniques from automated planning in the context of interactive systems, especially human-robot interaction. We believe that the time is right for the HRI community to benefit from recent advances in automated planning.

**Case study: AI planning for interaction management in JAMES (james-project.eu)**

- Interaction management performed by PKS domain-independent planner
- State and action representations defined separately from reasoning tools and infrastructure
- Planner could be swapped out for comparison

**Plan step** | **Description**
--- | ---
greet(a1), ask-drink(a1), ack-order(a1), serve(a1, request(a1)), bye(a1). | Greet agent a1, Ask a1 for a drink order, Acknowledge a1’s order, Serve a1’s ordered drink, End the interaction

**Multi-agent knowledge extensions for PKS**

- Information modelled with nested belief operators (e.g., “agent A believes agent B believes P”)
- Challenge: Provide a solution that is expressive enough to model a variety of domains, and efficient enough to support practical plan generation
- Solution: restrict form of representation: keep reasoning language simple (see Steedman and Petrick, SIGdial 2007)

**Action** ask-drink(7a : agent)  
**Preconditions:** (inTrans = 7a) ^ ¬K(ordered(7a)) ^ ¬K(otherAttentionRequests) ^ ¬K(badASR(7a))  
**Effects:** add(Kf, ordered(7a)), add(Kv, request(7a))

**Action** serve(7a : agent, 7d : drink)  
**Preconditions:** (inTrans = 7a) ^ K(ordered(7a)) ^ Kv(request(7a)) ^ K(request(7a) = 7d) ^ ¬K(otherAttentionRequests) ^ ¬K(badASR(7a))  
**Effects:** add(Kf, served(7a))

**Action** ask(?x,?y,?p)  
**Preconditions:** ¬K(\(?x\)?p ^ K(\(x\)?p)  
**Effects:** add(Kf, K(\(x\)?p))

**Action** tell(?x,?y,?p)  
**Preconditions:** K(?x)?p ^ K(?x)¬K(?p)  
**Effects:** add(Kf, K(?x)K(?y)?p)