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

Ronald P. A. Petrick
Department of Computer Science
Heriot-Watt University
Edinburgh EH14 4AS, Scotland, UK
R.Petrick@hw.ac.uk

Mary Ellen Foster
School of Computing Science
University of Glasgow
Glasgow G12 8RZ, Scotland, UK
MaryEllen.Foster@glasgow.ac.uk

Abstract

A central problem in designing and implementing interactive systems—action selection—is also a core research topic in automated planning. While numerous toolkits are available for building end-to-end interactive systems, the tight coupling of representation, reasoning, and technical frameworks found in these toolkits often makes it difficult to compare or change the underlying domain models. In contrast, the automated planning community provides general-purpose representation languages and multiple planning engines that support these languages. We describe our recent work on automated planning for task-based social interaction, using a robot that must interact with multiple humans in a bartending domain.

Introduction

A fundamental component of any interactive dialogue system, such as a robot that is able to converse with a human using natural language, is the *interaction manager* (Bui 2006), whose core task is to carry out *action selection*: that is, based on the current state of the interaction and of the world, the interaction manager makes a high-level decision as to which spoken, non-verbal, and task-based actions should be taken next by the system as a whole. In contrast to more formal, descriptive accounts of dialogue (Asher and Lascarides 2003), which aim to model the full generality of language use, work on interaction management has concentrated primarily on developing end-to-end systems and on evaluating them through interaction with human users (Jokinen and McTear 2009; McTear, Callejas, and Griol 2016).

An important direction in interactive systems research has been the development of toolkits that support their construction (Rich and Sidner 1998; Larsson and Traum 2000; Bohus and Rudnicky 2009; Lison 2015). Such toolkits generally incorporate three main features. First, they provide a representational formalism for specifying states and actions. Second, the state/action representations are usually tightly linked to the reasoning strategy used to carry out action selection. Finally, most toolkits include infrastructure building tools to support modular system development. While these features clearly simplify the task of implementing end-to-end systems, the fact that the features are so tightly connected complicates the task of comparing or changing the underlying representational formalisms or reasoning strategies: in general, doing so often requires re-implementing the entire system (Peltason and Wrede 2011).

However, the problem of selecting high-level actions is not unique to dialogue systems, but is also a problem addressed in a variety of research communities including *automated planning*. In planning, the emphasis is on applying problem-solving techniques to find an ordered sequence of actions (a *plan*) that, when chained together, transform an initial state into a state where a set of specified goal objectives are achieved. The general planning problem is usually divided into two parts: a description of the planning domain and a definition of a planning problem instance to be achieved within that domain (Ghallab, Nau, and Traverso 2004). A *planning domain* provides a definition of the symbols and actions used by the planner. Symbols specify the objects, properties, states, and knowledge that make up the planning agent's operating environment, often defined in a logic-like language. Actions are typically specified in terms of the state properties that must be true to execute that action (its *preconditions*) and the changes that the action makes to the state when it is executed (its *effects*). A *planning problem* provides a definition of the initial state the planner begins its operation in, and a description of the goals to be achieved. A central goal of planning research is to build *general purpose* or *domain-independent* planning systems that are able to solve a range of planning problems in a variety of domains, rather than just a single problem in a single domain.

While the link between natural language processing and automated planning has a long tradition (Perrault and Allen 1980; Appelt 1985; Hovy 1988; Cohen and Levesque 1990; Young and Moore 1994), the approach has for the most part been largely overlooked more recently (with some notable exceptions (Koller and Stone 2007; Steedman and Petrick 2007; Benotti 2008; Brenner and Kruijff-Korbayová 2008)). In this paper, we highlight our recent work on using general-purpose automated planning for action section in human-robot interaction, building on our prior work from the JAMES (Joint Action for Multimodal Embodied Social Systems) robot bartender project (Foster et al. 2012).[1]

[1] http://james-project.eu/

Figure 1: The JAMES robot bartender.

```
action ask-drink(?a : agent)
    preconds: K(inTrans = ?a) &
              !K(ordered(?a)) &
              !K(otherAttentionRequests) &
              !K(badASR(?a))
    effects:  add(Kf,ordered(?a)),
              add(Kv,request(?a))

action serve(?a : agent, ?d : drink)
    preconds: K(inTrans = ?a) &
              K(ordered(?a)) &
              Kv(request(?a)) &
              K(request(?a) = ?d) &
              !K(otherAttentionRequests) &
              !K(badASR(?a))
    effects:  add(Kf,served(?a))
```

Figure 2: Example PKS actions in the bartender domain.

Action	Description
`greet(?a)`	Greet agent `?a`
`ask-drink(?a)`	Ask agent `?a` for a drink order
`serve(?a,?d)`	Serve drink `?d` to agent `?a`
`bye(?a)`	End interaction with agent `?a`
`wait(?a)`	Tell agent `?a` to wait
`ack-order(?a)`	Acknowledge agent `?a`'s order
`ack-wait(?a)`	Thank agent `?a` for waiting
`ack-thanks(?a)`	Acknowledge agent `?a`'s thanks

Table 1: A partial list of actions in the bartender domain.

Planning in a Robot Bartender Domain

The JAMES robot bartender (Figure 1) has the goal of supporting socially appropriate multi-party interaction in a bartending scenario. Based on (uncertain) observations of the human users in the scene, provided by the vision and speech recognition components, the system maintains a model of the social context and task state, and decides on the appropriate responses that are required to respond to users.

In this system, high-level action selection is performed by a domain-independent planner which manages the interactions with customers, tracks multiple drink orders, and gathers additional information as needed with follow-up questions (Petrick and Foster 2013). In particular, the task of interacting with human customers is mixed with the physical task of ensuring that the correct drinks are delivered to the correct customers. Plans are generated using PKS (Planning with Knowledge and Sensing) (Petrick and Bacchus 2002; 2004), a planner that works with incomplete information and sensing actions. Figure 2 shows an example of two actions from the robot domain, defined in the PKS representation. Here, `ask-drink` models an information-gathering dialogue action that asks a customer for their drink order, while `serve` is a physical robot action for serving a drink to a customer. The complete planning domain description also includes actions such as `greet(?a)` (a purely social action to greet customer `?a`), `wait(?a)` (tell `?a` to wait, e.g., by nodding), and `bye(?a)` (end an interaction with `?a`), among others. A partial list of the available actions in the bartender domain is given in Table 1. The planner uses these actions to form plans by chaining together ground actions instances to achieve the goals of a planning problem. For instance, Table 2 shows a plan for interacting with a single customer `a1`. If unexpected situations arise during the execution of a plan, the current interaction plan is discarded and a new plan is generated based on the new social context and task state.

An important design decision for the robot bartender was to define the state and action representations separately from the tools used to reason about them, and also from the infrastructure needed for the planner to communicate with the rest of the system (using the Ice object middleware (Henning 2004)). In addition to supporting the modular, distributed development of the system, this also permitted the PKS planner to be exchanged with a completely separate interaction manager based on Markov Decision Processes (Keizer et al. 2013), with no other changes needed to the system. In terms of our particular planning approach, PKS's representation language can be compiled into a variant of PDDL (McDermott et al. 1998), a standard planning language, enabling the bartender domain to be tested with other planning systems. Similarly, our approach could be easily integrated into other interactive systems using its existing application programming interface (Petrick and Gaschler 2014).

The system was tested in a series of studies involving human users ordering drinks from the robot bartender (Foster et al. 2012; Giuliani et al. 2013). Offline testing was also performed with other planners in the bartender domain, to study the efficiency of plan generation and the quality of the generated plans when the number of agents or the number of subdialogues was increased (Sharma 2012). The user studies showed high success rates for successful drink order-

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

Table 2: A plan for interacting with a single customer.

ing (e.g., a 95% success rate for 31 users), while the offline experiments indicated little trouble scaling the planning approach (e.g., plans for 20 users are built in under 2 seconds).

Extending the Planning Approach

More recently, we have extended the basic planning approach from JAMES in two main areas. First, we have explored the modelling and scalability of *common interaction patterns* that arise in the bartender domain. One area of investigation is the inclusion of clarification questions in the standard interaction plan to resolve ambiguities concerning drink requests. For instance, consider the interaction fragment responding to a user's drink order:

```
clarify-drink(lemonade,a1),
if (request(a1) = blue-lemonade) then
    ack-order(a1),
    serve(a1,blue-lemonade).
else if (request(a1) = pink-lemonade) then
    ack-order(a1),
    serve(a1,pink-lemonade).
else
    unknown-drink(a1).
```

Here, the `clarify-drink(lemonade,a1)` action is used as an instance of an information-gathering or *sensing action* to ask a customer to disambiguate the type of drink they ordered ("What kind of lemonade do you want?"). In this case, the planner performs a type of *contingent planning* (Hoffmann and Brafman 2005) to build different context-dependent plan branches (a conditional plan), each of which considers a different possible execution path (i.e., the customer ordered blue lemonade versus pink lemonade). Unexpected replies, such as unknown drinks (`unknown-drink(a1)`) can also be accommodated. A more complex example of this is the interaction:

```
if (request(a1) = beer) then
    ack-order(a1),
    subdialogue1,
    serve(a1,beer).
else if (request(a1) = water) then
    ack-order(a1),
    subdialogue2,
    serve(a1,water).
```

In this case, *subdialogue1* and *subdialogue2* could themselves contain additional clarification questions of the previous form (e.g., a request to clarify the type of beer, or a question to ask if ice should be put in the water). In general, the planner can automatically build interactions of the form:

```
sense(P1)
if (P1 = v1) then
    sense(P2)
    if (P2 = u1) then
        sense(P3)
        ...
    ...
else if (P1 = v2) then
    ...
```

which we can use to model a variety of interactive contexts in our planning domain. An important consequence of this work is that it also enables easy scalability testing in order to measure the computational overhead of generating plans with large numbers of clarifications, building on the prior JAMES work (Sharma 2012; Petrick and Foster 2013) for generating interaction plans with an increasing number of customers in the bar. We are also modelling common interaction patterns for scenarios involving group ordering and for managing low-confidence automatic speech recognition.

The second area of work we are exploring is to extend the representation language of the PKS planner to support *reasoning with multiagent beliefs*—information modelled using nested belief operators (e.g., "agent A believes agent B believes P") (Fagin et al. 1995). Currently, PKS's representation language does not support such information; any reference to agents and their intentions, beliefs, or goals must be "shoehorned" into the existing language provided by the planner (a strategy which is commonly used in planning and other logic-based systems). However, including native models of multiagent beliefs enables more realistic actions to be represented in more comprehensive contexts.

While formal models of such reasoning exist for dialogue and interaction (Asher and Lascarides 2003), the main problem for automated planning is to provide a solution that is both expressive enough to model a variety of domains and efficient enough to be implemented in a manner that does not negatively affect the plan generation process. In our case, we build on the approach of (Steedman and Petrick 2007) by restricting the form of the representation used by the planner and keeping the reasoning language simple. This enables us to write extended action operators in certain contexts. An example of actions encoded in this way is given below:

```
action ask(?x,?y,?p)
    preconds: ¬K[?x]?p & K[?x]K[?y]?p
    effects:  add(Kf,K[?y]¬K[?x]?p)

action tell(?x,?y,?p)
    preconds: K[?x]?p & K[?x]¬K[?y]?p
    effects:  add(Kf,K[?y]?p)
```

Here, the definitions of `ask(?x,?y,?p)` (agent `?x` asks agent `?y` about `?p`) and `tell(?x,?y,?y)` (agent `?x` tells agent `?y` about `?p`) include references to specific knowledge conditions of particular agents (e.g., the syntax `K[?x]p` denotes the idea that "agent `?x` knows `p`").

While these extensions are not yet complete (only a small set of restricted belief operators are implemented), we are nevertheless exploring applications of their use in the JAMES bartender domain, to identify situations where such models are necessary or helpful, and where our existing approach suffices. Beyond the bartender domain, we are also planning to apply such models in collaborative contexts, where a variety of planning techniques are already being applied (Freedman and Fukunaga 2015; Hayes and Scassellati 2016; Geib, Craenen, and Petrick 2016).

Conclusions

The work outlined in the paper is situated in the context of a larger research programme aimed at revisiting the use of techniques from automated planning in the context of interactive systems, especially human-robot interaction (Foster and Petrick 2016). We believe that the time is right for the HRI community to benefit from recent advances in the area of automated planning. We have already demonstrated that components from the two communities can be successfully combined in the JAMES system, as an alternative to the use of traditional toolkits for implementing interactive systems. In particular, we believe that the adoption of common, formally understood representation languages for states and actions that are separated from reasoning mechanisms and technical infrastructure can facilitate closer links between the two research communities.

Acknowledgements

This research has been partially funded by the EU's 7th Framework Programme under grant no. 270435 (JAMES, `http://james-project.eu/`) and grant no. 610917 (STAMINA, `http://stamina-robot.eu/`), and by the EU's Horizon 2020 programme under grant no. 688147 (MuMMER, `http://mummer-project.eu/`).

References

Appelt, D. 1985. *Planning English Sentences*. Cambridge, England: Cambridge University Press.

Asher, N., and Lascarides, A. 2003. *Logics of Conversation*. Cambridge University Press.

Benotti, L. 2008. Accommodation through tacit sensing. In *Proceedings of LONDIAL 2008*, 75–82.

Bohus, D., and Rudnicky, A. I. 2009. The RavenClaw dialog management framework: Architecture and systems. *Computer Speech & Language* 23(3):332–361.

Brenner, M., and Kruijff-Korbayová, I. 2008. A continual multiagent planning approach to situated dialogue. In *Proceedings of LONDIAL 2008*, 67–74.

Bui, T. H. 2006. Multimodal dialogue management - state of the art. Technical Report 06–01, University of Twente (UT), Enschede, The Netherlands.

Cohen, P., and Levesque, H. 1990. Rational interaction as the basis for communication. In *Intentions in Communication*. Cambridge, MA, USA: MIT Press. 221–255.

Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. Y. 1995. *Reasoning About Knowledge*. Cambridge, MA, USA: MIT Press.

Foster, M. E., and Petrick, R. P. A. 2016. Separating representation, reasoning, and implementation for interaction management. In *Proceedings of IWSDS 2016*.

Foster, M. E.; Gaschler, A.; Giuliani, M.; Isard, A.; Pateraki, M.; and Petrick, R. P. A. 2012. Two people walk into a bar: Dynamic multi-party social interaction with a robot agent. In *Proceedings of ICMI 2012*, 3–10.

Freedman, R. G., and Fukunaga, A. 2015. Integration of planning with plan recognition using classical planners (extended abstract). In *Proceedings of AAAI Fall Symposium on Artificial Intelligence for Human-Robot Interaction (AI-HRI 2015)*, 48–50.

Geib, C.; Craenen, B.; and Petrick, R. 2016. Generating collaborative behaviour through plan recognition and planning. In *Proceedings of the ICAPS 2016 Workshop on Distributed and Multi-Agent Planning (DMAP)*, 98–105.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: Theory and Practice*. Morgan Kaufmann.

Giuliani, M.; Petrick, R. P. A.; Foster, M. E.; Gaschler, A.; Isard, A.; Pateraki, M.; and Sigalas, M. 2013. Comparing task-based and socially intelligent behaviour in a robot bartender. In *Proceedings of ICMI 2013*, 263–270.

Hayes, B., and Scassellati, B. 2016. Autonomously constructing hierarchical task networks for planning and human-robot collaboration. In *Proceedings of ICRA 2016*, 5469–5476.

Henning, M. 2004. A new approach to object-oriented middleware. *IEEE Internet Computing* 8(1):66–75.

Hoffmann, J., and Brafman, R. 2005. Contingent planning via heuristic forward search with implicit belief states. In *Proceedings of ICAPS 2005*, 71–80.

Hovy, E. 1988. *Generating natural language under pragmatic constraints*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates.

Jokinen, K., and McTear, M. 2009. Spoken dialogue systems. *Synthesis Lectures on Human Language Technologies* 2(1):1–151.

Keizer, S.; Foster, M. E.; Lemon, O.; Gaschler, A.; and Giuliani, M. 2013. Training and evaluation of an MDP model for social multi-user human-robot interaction. In *Proceedings of SIGdial 2013*.

Koller, A., and Stone, M. 2007. Sentence generation as planning. In *Proceedings of ACL 2007*, 336–343.

Larsson, S., and Traum, D. R. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering* 6(3–4):323–340.

Lison, P. 2015. A hybrid approach to dialogue management based on probabilistic rules. *Computer Speech & Language*.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL – The Planning Domain Definition Language (Version 1.2). Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.

McTear, M.; Callejas, Z.; and Griol, D. 2016. *The Conversational Interface*. Springer.

Peltason, J., and Wrede, B. 2011. The curious robot as a case-study for comparing dialog systems. *AI Magazine* 32(4):85–99.

Perrault, C. R., and Allen, J. F. 1980. A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics* 6(3–4):167–182.

Petrick, R., and Bacchus, F. 2002. A knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of AIPS 2002*, 212–221.

Petrick, R., and Bacchus, F. 2004. Extending the knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of ICAPS 2004*, 2–11.

Petrick, R., and Foster, M. E. 2013. Planning for social interaction in a robot bartender domain. In *Proceedings of ICAPS 2013*, 389–397.

Petrick, R., and Gaschler, A. 2014. Extending knowledge-level planning with sensing for robot task planning. In *Proceedings of PlanSIG 2014*.

Rich, C., and Sidner, C. L. 1998. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction* 8(3–4):315–350.

Sharma, V. 2012. Automated Planning for Natural Language Robot Dialogue. MSc Project, University of Edinburgh, Edinburgh, Scotland, United Kingdom.

Steedman, M., and Petrick, R. P. A. 2007. Planning dialog actions. In *Proceedings of SIGdial 2007*, 265–272.

Young, R. M., and Moore, J. D. 1994. DPOCL: a principled approach to discourse planning. In *Proceedings of INLG 2004*, 13–20.