I Can Help! Cooperative Task Behaviour through Plan Recognition and Planning

Christopher Geib
College of Computing and Informatics
Drexel University
3141 Chestnut Street
Philadelphia, PA 19104, USA
cwg33@drexel.edu

Bart Craenen and **Ronald P. A. Petrick**
School of Informatics
University of Edinburgh
10 Crichton Street
Edinburgh EH8 9AB, Scotland, UK
{bcraenen,rpetrick}@inf.ed.ac.uk

Abstract

This paper presents a framework for integrated plan recognition and automated planning, to produce cooperative behaviour for one agent to help another agent. By observing an "initiator" agent performing a task, the plan recogniser hypothesises how a "supporter" agent could help the initiator by proposing a set of subgoals to be achieved. A lightweight negotiation process mediates between the two agents to produce a mutually agreeable set of goals for the supporter. The goals are passed to a planner which builds an appropriate sequence of actions for satisfying these goals. The approach is demonstrated in a series of experimental scenarios.

Introduction

The ability of an agent to *help* another agent is a desirable attribute when designing artificial entities, such as robots, that must operate together with humans in real-world environments. Indeed, the idea of building assistive agents that must work alongside humans in a cooperative fashion has been a long-standing goal of artificial intelligence and robotics since its earliest days. However, the task of deciding when and how to help another agent can be difficult. Effective helping involves recognising the goals or intentions of other agents, reasoning about opportunities to contribute to existing plans, generating appropriate actions, and potentially communicating such information to the agents involved.

While the computational cost of reasoning about cooperative action in its most general form may be entirely impractical, constrained forms of reasoning do exist that can be used as the basis for helpful behaviour. For instance, consider the task of two agents setting a table for dinner, where the first agent sets the plates and glasses, and the second agent sets the knives, forks, and spoons. The subgoals pursued by each agent are disjoint but together they contribute to a shared overall goal. Moreover, each action is performed by a single agent, with no action requiring the joint coordination of multiple agents (e.g., two agents lifting a table). Finally, the order of subgoal achievement is independent of the actions of the other agent (e.g., it makes no difference if the knives are put onto the table before the forks or vice versa).

In this paper we consider scenarios of the above form, where one agent, called the *supporter*, must decide how to act to help a second agent, called the *initiator*, achieve its goals. While the supporter is considered to be an artificial agent, no assumption is made about the initiator which may be a human or artificial agent. We consider goals which can be decomposed as in the above example, and plans that can be executed as independent sequences of actions for each agent. While such conditions may appear to be restrictive, they nevertheless characterise a useful collection of problem scenarios whose solution is far from trivial: the goals of the initiator must be identified and suitable subgoals must be appropriately selected for the supporter to achieve.

Our approach combines *plan recognition* with *automated planning*, together with a lightweight negotiation process for ensuring that the supporter's goals are acceptable to both agents. As a result, we focus on the high-level (symbolic) reasoning involved in this task. In particular, the supporter tries to infer the high-level plans of the initiator and identify possible subgoals that contribute to the initiator's plan. Pairs consisting of the initiator's hypothesised high-level goal and a candidate subgoal are then proposed to the initiator as possible helpful subgoals that the supporter could accomplish. Once negotiation is complete, the agreed upon goals are passed to a planner which constructs a sequence of actions for the supporter to execute to help the initiator.

Related Work

The idea of an agent helping another agent has sometimes been viewed as a primary property of a plan, or as implicit in multiagent actions. For example, (Pollack 1990; Lochbaum, Grosz, and Sidner 1990) explicitly reason about coordination and helping in the form of shared plans and mutual beliefs. However, establishing agreement of plans or beliefs has typically relied on shared knowledge which has often been a difficulty in such theories. Similarly, representations for multiagent joint actions (i.e., actions that require two or more agents for their execution) (Brafman and Domshlak 2008; Boutilier and Brafman 2001) could be used to model situations where one agent helps another agent, but such representations do not typically address the case where help is not a consequence of such joint actions.

There has also been significant work on multiagent planning (e.g., (Brafman and Domshlak 2008; Brenner 2003; Crosby, Jonsson, and Rovatsos 2014)) and for the decentralised solving of constraint optimisation problems (Modi et al. 2003). However, such work has not directly addressed the problem of when one agent can help another agent, rather

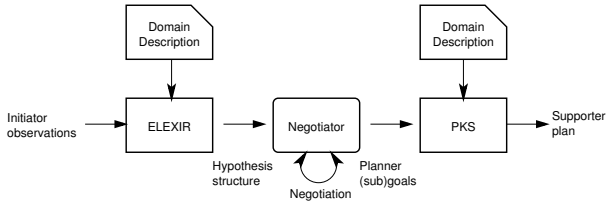

Figure 1: Components and interactions in the framework.

than simply working on a separate goal or plan.

The role of natural language dialogue as an effective means of coordinating actions between a robot and a human has also been studied (Fong, Thorpe, and Baur 2003). Combining natural language with goal inference has also been explored for the task of selecting individual actions to contribute to an ongoing task, or for correcting the action of a human already engaged in the task (Foster et al. 2008; Giuliani et al. 2010). Other approaches have considered the use of multiagent plan recognition (Sukthankar and Sycara 2008) and intention recognition (Han, Pereira, and Santos 2001; Han, Lenaerts, and Pereira 2015) in cooperative behaviour. Finally, the use of hybrid architectures has been a common approach for integrating diverse components with different representational requirements, particularly when a robot must cooperate with a human partner (Hawes et al. 2007; Kennedy et al. 2007; Zender et al. 2007).

A Framework for Collaborative Behaviour

We now present our framework by describing the plan recognition, negotiation, and planning components used in our approach. The general framework is shown in Figure 1.

Plan Recognition with ELEXIR

Plan recognition attempts to identify the goal being pursued by an agent, the subgoals of the plan being executed, and those that are anticipated to still be part of the plan. (This contrasts work in activity and goal recognition (Liao, Fox, and Kautz 2005; Hoogs and Perera 2008; Blaylock and Allen 2003).) Plan recognition systems attempt to produce some form of hierarchical structure that captures the current state of subgoals the observed agent is pursuing (Bui, Venkatesh, and West 2002; Avrahami-Zilberbrand and Kaminka 2005). E.g., given observations of an agent picking and placing forks followed by knives, these systems could identify not only the goal as setting the table, but also that the agent was in the process of setting knives.

For our work, we require a plan recognition algorithm that is able to produce the complete unexecuted frontier of the hierarchical plan being followed (Kautz 1991; Blaylock and Allen 2006; Geib 2009). Such an algorithm not only produces the subgoal stack of the observed plan but also identifies future subgoals that are yet to be executed. E.g., given the observations of pick and place actions for forks and knives, such a system would identify that the goal was to set the table, the current subgoal was to set the knives, and in the future, the agent would also set spoons and plates.

We use ELEXIR (Geib 2009) to perform the kind of plan recognition described above. ELEXIR is a probabilistic plan recogniser that views the problem as an instance of parsing a probabilistic grammar. As such, ELEXIR takes as input a formal probabilistic grammar that specifies the set of plans to be recognised and a set of observed actions. It computes a set of acceptable hierarchical plan structures representing the plans hypothesised as being executed by the agent. It then outputs each hypothesis as the ordered sets of subgoals that must still be executed for the goal to be achieved, and a probability measuring the likelihood of that hypothesis.

ELEXIR also supports the possibility that an agent can be pursuing multiple plans at the same time as well, as the partially ordered plans. Therefore, we will represent a hypothesis produced by ELEXIR as a tuple of the form:

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

where P is the probability of the hypothesis, G_i represents the goal of the hypothesised plan, and sg_j the remaining sets of possibly partially ordered subgoals that must be achieved for G_i to be completed. The sg_j within one set of braces are treated as unordered with respect to each other, but all of the sg_j must be achieved before those in the next set.

E.g., we might capture three hypothesis from the table setting example (after observing just the setting of forks) as:

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

The first tuple captures the hypothesis that with 95% probability the agent is following a plan to set the table and still has the subgoals to set the knives, spoons, and plates. These subgoals are unordered with respect to each other in the plan. The second tuple captures the hypothesis that with 4.5% probability the agent is cleaning the forks and still needs to wash them and put them away, in that order. The third tuple captures the hypothesis that with only 0.5% probability the agent is simply counting the forks and is done with its plan. Thus, each hypothesis provides us with the probability of the plans being executed, the goals they are intended to achieve, and the subgoals in the plan that have yet to be achieved.

Subgoal Identification and Negotiation

In order to negotiate collaboration, a supporter must first confirm that it understands the goal of the initiator's high-level plan. Otherwise, the supporter might waste time suggesting subgoals that do not contribute to the initiator's goal. In the case where a single plan is being pursued by the initiator, using the hypothesis structures built by ELEXIR provides a straightforward way to rank the goals of the plan being pursued. This makes it relatively easy for the supporter to verify the initiator's plan by a simple query to the initiator.

When the initiator's goal is identified, the supporter can then attempt to identify a subgoal that shares the identified goal. For instance, when considering the hypotheses for the setting of forks, the first hypothesis is the most likely:

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

If the initiator confirms that SetTable is in fact the goal of its plan, the supporter could then suggest that it take on the

subgoals of SetKnives, SetSpoons, SetPlates, or some combination thereof. As we will see below, a maximally helpful agent could volunteer to do all three of these subgoals. However, the negotiation could also result in a number of other outcomes whereby the supporter agrees to do some subset of these subgoals, or none of them at all. In effect, the negotiation process is then a directed search: first to identify the goal of the initiator's plan, and then to find appropriate subgoals from the set of known unaccomplished subgoals of the plan the supporter has inferred for the goal.

Automated Planning with PKS

Once negotiation has produced a set of subgoals for helping the initiator, the supporter must generate a plan to execute. To do so, we use PKS (Planning with Knowledge and Sensing) (Petrick and Bacchus 2002; 2004), a knowledge-level (Newell 1982) planner that builds plans using incomplete information and sensing (Petrick and Bacchus 2002; 2004). PKS operates by reasoning about how its knowledge state changes due to action, based on a generalisation of STRIPS (Fikes and Nilsson 1971). In PKS, the planner's knowledge state is represented by a set of databases, each of which models a particular type of knowledge and has a formal interpretation in a modal logic of knowledge. Actions can modify the databases, which has the effect of updating the planner's knowledge. To ensure efficient inference, PKS restricts the type of knowledge it can represent.

Like other planners, a PKS planning domain consists of an initial state, a set of actions, and a set of goals. The initial state is simply the initial knowledge state (databases). Goals specify the knowledge conditions that the planner is trying to achieve, formed from the supporter's subgoals through a syntactic compilation process which transforms them into PKS goals. Actions in PKS are modelled by their knowledge preconditions and effects on the planner's databases. Plans are constructed using a forward-chaining heuristic search.

Figure 2 shows two PKS actions taken from our experimental domains (see below). A precondition `K`(ϕ) queries PKS's knowledge to determine if the planner knows ϕ, while an effect that references `Kf` updates PKS's database of known world facts. Using these actions, a plan such as:

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

might be built in support of a goal to put forks on the table.

Integration and Operation

Both ELEXIR and PKS are implemented as C++ libraries and expose user interfaces for integration through ZeroC's Internet Communication Engine (ICE), a modern distributed computing platform (Henning 2004). Operation proceeds with the supporter observing actions performed by the initiator. These observations are fed into ELEXIR, which produces a structure of hypotheses about the initiator's high-level plan, as goal/subgoal pairs. This structure is then used in the negotiation process between the supporter and initiator. Negotiation applies directed search to the hypothesis structure to produce a set of goal for the planner. Finally, PKS uses these goals to attempt to build the supporter's plan.

```
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)
```

Figure 2: PKS actions from the experimental domain.

Experiments

We demonstrate our framework by considering three scenarios that lead to different subgoals. The underlying domains remain the same: an initiator has begun setting a table for a dinner for two people. Each place is to be set with a knife, fork, spoon, plate, and glass. The aim of the supporter is to help the initiator complete its goal. The observations provided to ELEXIR are also the same: one by one the initiator picks up two forks and two knives, and places them in their appropriate positions on the table. However, the scenarios differ in the way ELEXIR interprets these observations.

Correctness depends on two factors: first, whether the plan recogniser interprets the observation correctly and, second, whether the planner produces the correct plans. We note that the computational requirements for these scenarios are minimal: both plan recognition and planning take minimal time, and the cost of the negotiation process, excluding the time taken for the negotiation exchange, is negligible.

Scenario 1: In this scenario, the plan recogniser correctly identifies the initiator's goal, as well as the subgoals that the supporter could fulfil. The first hypothesis is:

$$(0.8, [\{\text{SetTable:}\{\text{SetPlates}, \text{SetSpoons}, \text{SetGlasses}\}\}]).$$

In this scenario, there is no need for a directed search of the hypothesis structure, and negotiation takes the form:

```
   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.
```

Once completed, the *SetPlates*, *SetSpoons*, and *SetGlasses* subgoals are syntactically translated into PKS goals and a plan is generated. E.g., the partial plan for *SetPlates* may be:

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

(The plans for the other two subgoals will be similar.)

Scenario 2: This scenario extends the first scenario, and is designed to test the use of directed search to correctly identify the initiator's goal from the hypothesis structure supplied by ELEXIR. The search focuses on high-level goal

identification during negotiation, with the initiator rejecting the hypothesis initially presented by the supporter.

In the first iteration of the negotiation process, the supporter presents the initiator with the hypothesis:

$$(0.8, [\{\text{CleanForks:}\{\text{WashForks}\}\{\text{PutAwayForks}\}\}]).$$

This hypothesis incorrectly identifies the initiator's goal to be that of cleaning the forks. After the initiator rejects this hypothesis, the supporter moves to the next most probable hypothesis, thus iteratively negotiating until the correct goal is found. (The number of negotiation iterations can be reduced by additional reasoning about the hypothesis.) For brevity, we assume the next hypothesis correctly identifies the initiator's goal so further negotiation is unnecessary. The correct hypothesis is then the same as in Scenario 1:

$$(0.8, [\{\text{SetTable:}\{\text{SetPlates}, \text{SetSpoons}, \text{SetGlasses}\}\}]).$$

Negotiation would then take the following form:

```
   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.
```

The remainder of the process then follows the one given in Scenario 1: subgoals are translated for use by PKS; the planner builds a plan for setting the plates, spoons, and glasses; and the supporter performs the plan. This scenario shows that by considering all hypotheses, the framework can recover from an incorrect identification of the goal through negotiation and directed search of the hypothesis structure.

Scenario 3: The final scenario is designed to test the framework when dealing with the situation in which the goal of the plan pursued by the initiator is correctly identified, but one (or more) of the hypothesised subgoals is not, and is thus rejected by the initiator. If this happens, the supporter, using directed search of the hypothesis structure, will iteratively negotiate with the initiator until it finds an acceptable subgoal. If none of the remaining subgoals in the hypothesis are acceptable to the initiator, it is possible for the supporter to run out of subgoals. If this occurs, the supporter can then revert back to the hypothesis structure to find another hypothesis with the same goal, and continue negotiation to see if its (other) subgoals are acceptable. This case is not examined here for space reasons; instead, this scenario considers the same hypothesis as in the first scenario:

$$(0.8, [\{\text{SetTable:}\{\text{SetPlates}, \text{SetSpoons}, \text{SetGlasses}\}\}]).$$

In this case, negotiation takes the form:

```
   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.
```

The remainder of this process differs from the other scenarios since the rejected subgoal is not passed to the planner. Instead, only a plan for setting the spoons and glasses is generated. Thus, by utilising the hypothesis structure, the initiator is not limited to accepting all subgoals in a hypothesis: the framework is flexible enough for the initiator to decide how, and in which way, he wants to be helped, without elaborate reasoning or goal decomposition on the part of the supporter.

Discussion

While the experimental scenarios demonstrate that our approach successfully generates cooperative plans, the framework also relies on certain assumptions concerning the knowledge of the initiator and supporter. For instance, the plan inferred by the supporter is never shared with the initiator in this process, and this approach does not generate plans with joint actions, where multiple agents must coordinate to perform the same task (e.g., lifting a table). Instead, we only generate independent action sequences for the supporter once there is agreement as to the supporter's subgoals.

Another representational issue to be addressed involves aligning the domains used by the plan recogniser and the planner, which may differ. However, since the plan recogniser and planner must exist within the same reasoning framework, the onus is currently placed on the domain designer to ensure that the domains interoperate correctly. One area of future work is to explore common representations, or to automatically induce one representation from the other.

Finally, in this early stage of our work we have focused primarily on plan recognition, rather than planning. However, in future work we plan to extend our approach to more complex domains, such as those involving incomplete information and uncertainty, where we can use PKS's ability to use sensing actions, including communicative actions (Petrick and Foster 2013), to gather information or take into account the involvement of other agents. For instance, if the supporter agreed to place wine glasses on the table, it may first need to query the initiator as to which people want wine (and which type of wine) to ensure the table is properly set. One way to do this is by constructing a contingent plan with information-gathering communicative actions, in order to obtain the necessary information from the initiator.

Conclusion

This paper presented a framework that combined plan recognition and planning to produce cooperative behaviour between a pair of agents. Successful integration of the reasoning components centred around appropriate subgoal identification by the plan recogniser, combined with a lightweight negotiation process which generated goals to be used by the planner for constructing appropriate plans. A set of experiments demonstrated the potential of our approach, and helped motivate our ongoing and future work to extend these techniques to more complex real-world situations.

Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme under grant no. 270273 (XPERIENCE, `xperience.org`) and grant no. 610917 (STAMINA, `stamina-robot.eu`).

References

Avrahami-Zilberbrand, D., and Kaminka, G. A. 2005. Fast and complete symbolic plan recognition. In *Proc. of IJCAI 2005*, 653–658.

Blaylock, N., and Allen, J. 2003. Corpus-based statistical goal recognition. In *Proc. of IJCAI 2003*, 1303–1308.

Blaylock, N., and Allen, J. 2006. Fast hierarchical goal schema recognition. In *Proc. of AAAI 2006*, 796–801.

Boutilier, C., and Brafman, R. 2001. Partial-order planning with concurrent interacting actions. *JAIR* 14:105–136.

Brafman, R., and Domshlak, C. 2008. From one to many: Planning for loosely coupled multi-agent systems. In *Proc. of ICAPS 2008*, 28–35.

Brenner, M. 2003. A Multiagent Planning Language. In *Proceedings of the Workshop on PDDL at ICAPS 2003*.

Bui, H. H.; Venkatesh, S.; and West, G. 2002. Policy recognition in the Abstract Hidden Markov Model. *JAIR* 17:451–499.

Crosby, M.; Jonsson, A.; and Rovatsos, M. 2014. A single-agent approach to multiagent planning. In *Proc. of ECAI 2014*, 237–242.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

Fong, T.; Thorpe, C.; and Baur, C. 2003. Collaboration, dialogue, and human-robot interaction. In *Robotics Research, Volume 6 of Springer Tracts in Advanced Robotics*. Springer. 255–266.

Foster, M. E.; Giuliani, M.; Müller, T.; Rickert, M.; Knoll, A.; Erlhagen, W.; Bicho, E.; Hipólito, N.; and Louro, L. 2008. Combining goal inference and natural-language dialogue for human-robot joint action. In *ECAI Workshop on Combinations of Intelligent Methods and Applications*.

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

Giuliani, M.; Foster, M. E.; Isard, A.; Matheson, C.; Oberlander, J.; and Knoll, A. 2010. Situated reference in a hybrid human-robot interaction system. In *Proc. of INLG 2010*, 67–75.

Han, T. A.; Lenaerts, T.; and Pereira, L. M. 2015. Synergy between intention recognition and commitments in cooperation dilemmas. *Scientific Reports* 5(9312). doi:10.1038/srep09312.

Han, T. A.; Pereira, L. M.; and Santos, F. C. 2001. The role of intention recognition in the evolution of cooperative behavior. In *Proc. of IJCAI 2011*, 1684–1689.

Hawes, N.; Sloman, A.; Wyatt, J.; Zillich, M.; Jacobsson, H.; Kruijff, G.-J. M.; Brenner, M.; Berginc, G.; and Skočaj, D. 2007. Towards an integrated robot with multiple cognitive functions. In *Proc. of AAAI 2007*, 1548–1553.

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

Hoogs, A., and Perera, A. A. 2008. Video activity recognition in the real world. In *Proc. of AAAI 2008*, 1551–1554.

Kautz, H. A. 1991. A formal theory of plan recognition and its implementation. In Allen, J. F.; Kautz, H. A.; Pelavin, R. N.; and Tenenberg, J. D., eds., *Reasoning About Plans*. Morgan Kaufmann. 69–126.

Kennedy, W. G.; Bugajska, M. D.; Marge, M.; Adams, W.; Fransen, B. R.; Perzanowski, D.; Schultz, A. C.; and Trafton, J. G. 2007. Spatial representation and reasoning for human-robot collaboration. In *Proc. of AAAI 2007*, 1554–1559.

Liao, L.; Fox, D.; and Kautz, H. A. 2005. Location-based activity recognition using relational Markov networks. In *Proc. of IJCAI 2005*, 773–778.

Lochbaum, K.; Grosz, B.; and Sidner, C. 1990. Intentions in communication. In *Proc. of AAAI 1990*, 485–490.

Modi, P. J.; Shen, W.-M.; Tambe, M.; and Yokoo, M. 2003. An asynchronous complete method for distributed constraint optimization. In *Proc. of AAMAS 2003*, 161–176.

Newell, A. 1982. The Knowledge Level. *Artificial Intelligence* 18(1):87–127.

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

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

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

Pollack, M. 1990. Plans as complex mental attitudes. In Cohen, P.; Morgan, J.; and Pollack, M., eds., *Intentions in Communication*. MIT Press. 77–103.

Sukthankar, G., and Sycara, K. 2008. Robust and efficient plan recognition for dynamic multi-agent teams (short paper). In *Proc. of AAMAS 2008*.

Zender, H.; Jensfelt, P.; Óscar Martínez Mozos; Kruijff, G.-J. M.; and Burgard, W. 2007. An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Proc. of AAAI 2007*, 1584–1589.