Integrating a Closed World Planner and an Open World Robot

Kartik Talamadupula
Department of Computer Science
Arizona State University
Tempe, AZ 85287 USA
krt@asu.edu
Advised by: Subbarao Kambhampati
Joint work with: J. Benton, Paul Schermerhorn, Matthias Scheutz

Abstract

In this paper, we present an integrated planning system that actively directs an agent engaged in an urban search and rescue (USAR) scenario. We describe three salient features that comprise the planning component of this system, (1) the ability to plan in a world open with respect to objects, (2) execution monitoring and replanning abilities, and (3) handling soft goals, and detail the interaction of these parts in representing and solving the USAR scenario at hand. We show that though insufficient in an individual capacity, the integration of this trio of features is sufficient to solve the scenario that we present. We test our system with an example problem that involves soft and hard goals, as well as goal deadlines and action costs, and show via an included video that the planner is capable of incorporating sensing actions and execution monitoring in order to produce goal-fulfilling plans that maximize the net benefit accrued.

Introduction

Consider the following problem: a human-robot team is actively engaged in an *urban search and rescue* (USAR) scenario inside a building of interest. The robot is placed inside this building, at the beginning of a long corridor. The human team member has intimate knowledge of the building's layout, but is removed from the scene and can only interact with the robot via on-board wireless audio communication. The corridor in which the robot is located has doors leading off from either side into rooms, a fact known to the robot. However, unknown to the robot (and the human team member) is the possibility that these rooms may contain injured humans (victims). The robot is initially given a hard goal of reaching the end of the corridor by a given deadline based on wall-clock time. As the robot executes a plan to achieve that goal, the team is given the (additional) information regarding victims being in rooms. Also specified with this information is a new soft goal, to report the location of victims.

It is natural to encode this soft goal as a quantified goal, since it is expected that the planner will report the location of as many victims as it can find given its time and cost constraints. The planner must reason about the net benefit of attempting to find a victim, since it is a soft goal and can be ignored if it is not worth the pursuit; it must then direct the robot to sense for the information that it needs in order to determine the presence of a victim in a particular location. This can be modeled as a partial satisfaction planning (PSP) problem, and solved using planners that can handle PSP problems.

Unfortunately, the dynamic nature of the domain coupled with the partial observability of the world precludes complete a priori specification of the domain, and forces the robot and its planner to handle incomplete and evolving domain models (Kambhampati 2007). This fact, coupled with the fallibility of human experts in completely specifying the information that is relevant to the given problem and goals up-front, makes it quite likely that information crucial to achieving some soft goals may be specified at some later stage *during* the planning process. In our USAR scenario, for example, the knowledge that victims are in rooms may be relayed to the planner while it is engaged in planning for the executing robot. In order to handle the specification of such statements in the midst of an active planning process, and enable the use of knowledge thus specified, we need to relax two other crucial invariants that most modern planners rely on. The first is the closed world assumption with respect to the constants (objects) in the problem—the planner can no longer assume that the only objects in the scenario are those that are mentioned in the initial state. The other modification requires that we interleave planning with execution monitoring and, if required, replanning in order to account for the new information.

In this paper, we explore the issues involved in engineering an automated planner to guide a robot towards maximizing net benefit accompanied with goal achievement in such scenarios. We will start by noting that the planning problem that we face here involves partial satisfaction (in that the robot has to weigh the rewards of the soft goals against the cost of achieving them). It also requires replanning ability (in that the robot has to modify its current plan based on new goals that are added). An additional (perhaps more severe) complication is that the planner needs to handle goals involving objects whose existence is not known in the initial state (e.g., the location of the humans to be rescued in our scenario). The system described in this paper was used to guide a Pioneer P3-AT robot as it navigated the scenario

presented previously in order to achieve the hard goal of getting to the end of the corridor, while trying to accrue the maximum net benefit possible from the additional soft goal of reporting the location of injured humans (Talamadupula et al. 2009). A video of the robot performing these tasks can be viewed via the following link:
`http://hri.cogs.indiana.edu/videos/USAR.avi`

Planning in an Open World

The planning for the robot is performed by *SapaReplan*, a forward state-space planner based on *SapaPS* (Do and Kambhampati 2004). *SapaReplan* adds the ability to handle updates to the state through the use of a monitor process. We additionally introduce a novel goal construct called an *open world quantified goal* (OWQG) that combines information about the open world and partial satisfaction aspects of the problem.

Goals in an Open World

Our approach seeks to open the world by allowing statements, called open world quantified goals, that label sections of the domain as open with respect to objects. Using these, the domain expert can furnish details about when new objects may be encountered through sensing and include goals that relate directly to the sensed objects. This can be seen as a complementary approach to handling open world environments using *local closed world* (LCW) information produced by sensing actions (Etzioni, Golden, and Weld 1997).

An open world quantified goal (OWQG) is defined as a tuple $\mathcal{Q} = \langle F, \mathcal{S}, \mathcal{P}, \mathcal{C}, \mathcal{G} \rangle$. Here, F and $\mathcal{S}$ are typed variables that are part of the problem Π, where F belongs to the object type that $\mathcal{Q}$ is quantified over, and $\mathcal{S}$ belongs to the object type about which information is to be sensed. $\mathcal{P}$ is a predicate which ensures sensing closure for every pair $\langle f, s \rangle$ such that f is of type F and s is of type $\mathcal{S}$, and both f and s belong to the set of objects in the problem, $\mathcal{O} \in \Pi$; for this reason, we term $\mathcal{P}$ a *closure condition*. $\mathcal{C} = \bigwedge_i c_i$ is a conjunctive first-order formula where each c_i is a statement about the openness of the world with respect to the variable S. For example, `c = (in ?hu - human ?z - zone)` with `S = ?hu - human` means that c will hold for new objects of the type 'human' that are sensed. Finally $\mathcal{G}$ is a quantified goal on $\mathcal{S}$.

Newly discovered objects may enable the achievement of goals, granting the opportunity to pursue reward. For example, detecting a victim in a room will allow the robot to report the location of the victim (where reporting gives reward). Note that reward in our case is for each reported injured person. As such, there exists a quantified goal that must be allowed partial satisfaction. In other words, the universal base (Weld 1994), or total grounding of the quantified goal on the real world, may remain unsatisfied while its component terms may be satisfied. To handle this, we use partial satisfaction planning (PSP) (van den Briel et al. 2004), where the objective is to maximize the difference between the reward given to goals, and the cost of actions. Reward is given for each term $g \in \mathcal{G}$ satisfied, $u(\mathcal{G})$. Additionally each term g is considered soft in that it may be "skipped over" and remain unachieved.

As an example, we present an illustration from our scenario: the robot is directed to "report the location of all victims". This goal can be classified as open world, since it references objects that do not exist yet in the planner's object database $\mathcal{O}$; and it is quantified, since the robot's objective is to report *all* victims that it can find. In our syntax:

```
1 (:open
2   (forall ?z - zone
3     (sense ?hu - human
4      (looked_for ?hu ?z)
5       (and (has_property ?hu injured)
6            (in ?hu ?z))
7   (:goal (reported ?hu injured ?z)
8            [100] - soft))))
```

In the example above, line 2 denotes F, the typed variable that the goal is quantified over; line 3 contains the typed variable $\mathcal{S}$ about which information is to be sensed. Line 4 is the unground predicate $\mathcal{P}$ known as the closure condition (defined earlier). Lines 5 and 6 together describe the formula $\mathcal{C}$ that will hold for all objects of type $\mathcal{S}$ that are sensed. The quantified goal over $\mathcal{S}$ is defined in line 7, and line 8 indicates that it is a soft goal and has an associated reward of 100 units.

Of the components that make up an open world quantified goal $\mathcal{Q}$, $\mathcal{P}$ is required[1] and F and S must be non-empty, while the others may be empty. If $\mathcal{G}$ is empty, i.e., there is no new goal to work on, the OWQG $\mathcal{Q}$ can be seen simply as additional knowledge that might help in reasoning about other goals.

Interleaving Planning and Execution

For most of the sensors on the robot, it is too expensive to sense at every step, so knowing exactly when to engage in perceptual monitoring is of critical importance. Low-level sensing for navigation is handled through action scripts on the robot's end, but for more expensive, high-level operations we use OWQGs. Planning through an open world introduces the possibility of dangerous faults or nonsensical actions. While in some sense, this can be quantified with a risk measure (see (Garland and Lesh 2002), for example), indicating the risk of a plan does nothing to address those risks. A more robust approach in an online scenario involves *planning to sense* in a goal-directed manner. When plans are output to the `ADE` *goal manager*, they include all actions up to and including any action that would result in closure (as specified by the *closure condition*).

Problem Updates and Replanning Regardless of the originating source, the monitoring process receives updates from the robot and correspondingly modifies the planner's representation of the problem. Updates can include new objects, timed events (i.e., an addition or deletion of a fact at a particular time, or a change in a numeric value such as action cost), the addition or modification (on the deadline or

[1]If $\mathcal{P}$ were allowed to be empty, the planner could not gain closure over the information it is sensing for, which will result in it directing the robot to re-sense for information that has already been sensed for.

reward) of a goal, and a time point to plan from. As discussed in (Cushing and Kambhampati 2005), providing for updates to the planning problem allows us to look at unexpected events in the open world as new information rather than faults to be corrected. In our setup, problem updates cause the monitor process to immediately stop the planner (if it is running) and update its internal problem representation. The planner is then signaled to replan on the new problem. In the presence of reward and action cost[2], the replanning process also allows the planner to exploit new opportunities, potentially finding plans that may achieve better net benefit than previous ones.

Implementation

To handle open world quantified goals, we developed a methodology that grounds the problem into the closed world using a process similar to Skolemization. More specifically, we generate *runtime objects* from the sensed variable $\mathcal{S}$ that explicitly represent the potential existence of an object to be sensed. These objects are represented with a suffixed exclamation mark on the object type, followed by a number (e.g., `human!1`). One can look at $\mathcal{S}$ as a Skolem function of F, and runtime objects as Skolem entities that substitute for the function. Runtime objects are then added to the problem and ground into the closure condition $\mathcal{P}$, the conjunctive formula $\mathcal{C}$, and the open world quantified goal $\mathcal{G}$. In other words, runtime objects substitute for the existence of S dependent upon the variable F. The facts generated by following this process over $\mathcal{C}$ are included in the set of facts in the problem through the problem update process. The goals generated by $\mathcal{G}$ are similarly added. This process is repeated for every new object that F may instantiate.

We treat $\mathcal{P}$ as an *optimistic closure condition*, meaning a particular state is considered closed once the ground closure condition is true. On every update the ground closure conditions are checked and if true the facts in the corresponding ground values from $\mathcal{C}$ and $\mathcal{G}$ are removed from the problem.

Consider the scenario at hand and its open world quantified goal. Given a known zone, `zone1`, the process would generate a runtime object `human!1`. Subsequently, the facts `(has_property human!1 injured)` and `(in human!1 zone1)` and the goal `(report human!1 injured zone1)` (with reward 100) would be generated and added to the problem. A closure condition `(looked_for human!1 zone1)` would also be created. When the planning system receives an update including `(looked_for human!1 zone1)`, it will update the problem by deleting the facts `(has_property human!1 zone1)` and `(in human!1 zone1)` and the goal `(report human!1 injured zone1)` at the apprioriate time point. Recall that the planner must only output a plan up to (and including) an action that will make the closure condition true. The idea behind the closure condition is that after it becomes true we can expect closure on certain aspects of the world. Once the condition becomes true, the truth values of the facts in $\mathcal{C}$ are known.

[2]In our scenario, the action costs are determined by a combination of the temporal and resource costs incurred by the robot.

Discussion

It has been a regrettable reality in planning ranks that as the time required to generate a complete plan has decreased, so too has the ability to encode interesting details about the world. This has led to a situation where state-of-the-art planners can only deal with a subset of the features necessary to encode any domain of interest with a real world perspective. To be sure, there do exist planners that can handle the expressivity required to model some or all of the problems that delineate our USAR scenario from existing planning benchmarks (Penberthy and Weld 1992; Golden, Etzioni, and Weld 1994); unfortunately, none of these planners combine all the features necessary to solve our problem in the real world. Planning technologies and systems have been analyzed previously (Smith 2003) in order to move these techniques closer to being able to step up and perform in the real world (albeit other-worldly) domains used at NASA.

We took a similar approach towards our problem—we first considered the assumptions we would need to relax in order for a state-of-the-art planner to be able to reason about and solve our problem. In doing so, we found that these assumptions—about the closed world, the separation of planning and execution stages, and all goals being hard—had a direct correspondence to the features that we wished to model. We also discovered the lack of a planner that combines solutions to these problems in one integrated system.

The first (and in some ways most important) assumption we had to relax was one that most modern day planners have come to take for granted—the assumption that the world is closed with respect to facts, objects and operators. Since our scenario involved the specification of new knowledge concerning the world at any stage during the robot's progress, we had to allow for the possibility that there may be objects in the world that are not specified to the robot initially. We retained the closed nature of the world with respect to operator templates, since it is only reasonable that the robot is made aware of its capabilities initially and does not gain any additional powers on the way.

In order to deal with objects that the robot may either discover or *attempt* to discover to achieve some reward, it is essential that the planner not close its possibilities with regard to objects in the world. To enable this, we introduced the open world quantified goals defined in the previous section and equipped the planner with a mechanism to parse and use the information specified within these goals.

The issue of planning with an open versus closed world representation has been dealt with before, notably in the work of Etzioni et al. (Etzioni, Golden, and Weld 1997) via the specification of *local closed world* (LCW) statements. However, there exists at least one major difference between their work and this attempt. We note that the representation used in that work, of closing a world that is open otherwise via the LCW statements, is complementary to our representation. Since our interest in providing support for open world quantified goals (OWQGs) is to relax our planner's assumption of a world closed with respect to object creation, we are *opening* parts of a completely closed world with the aid of OWQGs.

However, the ability to recognize and represent new objects that appear during execution in the world means nothing if the planner cannot actively use these objects in order to output new plans that improve the quality metric (be it time, cost or net-benefit). To support this requirement, we had to endow our planning system with the capability to interleave planning and execution monitoring, so that changes to the world could be transmitted to the planner in the form of updates and subsequently parsed into the planner's database, as outlined in the previous section. This kind of *online planning* seems to be a case-restrictive yet simple solution to the complex problem of dealing with sensing actions. Specifically, this approach seems to be restricted to problems that contain simple reward models where it is reasonable to take a greedy approach.

It may be argued that XII, the planning system used by Etzioni et al. (Etzioni, Golden, and Weld 1997) handles the twin problems of an open world and interleaving planning and execution monitoring described above. However, these two features alone are not sufficient to model the USAR scenario. As described in earlier sections, we wanted the robot (and the planner that creates plans for it to execute) to look at tasks like reporting the presence of victims in rooms as *opportunities*, rather than as hard goals that *must* be achieved. That is to say, we wanted the robot to do its utmost to satisfy the quantification implied in the open world quantified goal, and to report the location of all victims; however, we did not want the pursuit of this objective to cloud the primary goal, which remained getting to the end of the corridor.

To handle this problem, we needed to consider a third relaxation—one that allowed for some goals to be specified as *soft* goals. We used ideas from the field of Partial Satisfaction Planning (PSP) (Smith 2004) and the planner SapaPS (Benton, Do, and Kambhampati 2009) to include support for reasoning about soft goals and net-benefit, as specified in the previous section. Enabling the usage of soft goals mitigates some of the more difficult problems in fully open worlds—in the USAR scenario, when certain rooms are completely undiscoverable, it is infeasible to expect complete satisfaction of certain quantified goals.

Conclusion

In this paper, we presented a novel approach to reconcile a planner's closed world representation with the open world that a robot has to typically execute it. To enable this approach, we presented the integration of techniques which when combined are sufficient to represent and solve the scenario described. We showed that we could handle information about new objects in the world using open world quantified goals, and that our replanning and execution monitoring system is able to handle the new information specified by these goals in order to produce plans that achieve a higher net benefit. We also detailed that our system could support soft goals, thus ensuring that opportunities retain their "bonus" nature, and do not metamorphise into additional hard goals that may constrain existing hard goals. We implemented all novel techniques and algorithms on a robot, and provided a video to demonstrate the robot's new-found abilities. We rounded off with a discussion of work related to our approach.

Acknowledgements

The author wishes to thank J. Benton, Paul Schermerhorn, Subbarao Kambhampati and Matthias Scheutz for their collaboration in this work. Thanks also go to William Cushing for helpful discussions and the generation of the planner *SapaReplan*.

References

Benton, J.; Do, M.; and Kambhampati, S. 2009. Anytime heuristic search for partial satisfaction planning. *AIJ* 178(5-6).

Cushing, W., and Kambhampati, S. 2005. Replanning: A new perspective. In *Proceedings of ICAPS*.

Do, M., and Kambhampati, S. 2004. Partial satisfaction (over-subscription) planning as heuristic search. *Proceedings of KBCS-04*.

Etzioni, O.; Golden, K.; and Weld, D. S. 1997. Sound and efficient closed-world reasoning for planning. *AIJ* 89(1-2):113–148.

Garland, A., and Lesh, N. 2002. Plan evaluation with incomplete action descriptions. In *Proceedings of the National Conference on Artificial Intelligence*, 461–467. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Golden, K.; Etzioni, O.; and Weld, D. 1994. XII: Planning with universal quantification and incomplete information. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning, KR*, volume 94. Citeseer.

Kambhampati, S. 2007. Model-lite planning for the web age masses: The challenges of planning with incomplete and evolving domain theories. *Proceedings of AAAI 2007*.

Penberthy, J., and Weld, D. 1992. UCPOP: A sound, complete, partial order planner for ADL.

Smith, D. 2003. The Next Challenges for AI Planning. Lecture given at the Planet International Summer School.

Smith, D. 2004. Choosing objectives in over-subscription planning. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling*, 393–401.

Talamadupula, K.; Benton, J.; Schermerhorn, P.; Kambhampati, S.; and Scheutz, M. 2009. Integrating a Closed World Planner with an Open World Robot: A Case Study. In *ICAPS 2009 Workshop of Bridging the Gap between Task and Motion Planning (BTAMP).*

van den Briel, M.; Sanchez, R.; Do, M.; and Kambhampati, S. 2004. Effective approaches for partial satisfaction (over-subscription) planning. In *Proceedings of the National Conference on Artificial Intelligence*, 562–569. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Weld, D. 1994. An introduction to least commitment planning. *AI magazine* 15(4):27–61.