Planning for a Mobile Robot to Attend a Conference

Eric Beaudry, Froduald Kabanza, and Francois Michaud

Univeristé de Sherbrooke
Sherbrooke, Canada
{eric.beaudry|froduald.kabanza|francois.michaud}@usherbrooke.ca

Abstract. The AAAI Mobile Robot Challenge requires robots to start from the entrance of the conference site, find their own way to the registration desk, socially interact with people and perform volunteer duties as required, then report at a prescribed time in a conference hall to give a talk and answer questions. These specifications convey some interesting planning problems that appear to be too complex for some of the most efficient AI planning systems that we analyzed. Based on this analysis, we present a new planning approach that we are developing to meet the challenge. Preliminary results show that our approach performs much better on robot conference planning problems than any of the other AI planning systems we tested.

1 Introduction

The AAAI Mobile Robot Challenge, introduced in 1999, is to have a robot start at the entrance of the conference site, find the registration desk, register, perform volunteer duties and give a presentation [6]. The long-term objective is to have robots receive no more information than usually given to human participants attending the conference. These specifications imply that robots must be able to plan tasks, such as registering, navigating to the presentation room and making a presentation, while at the same time interacting with people, ensuring its energetic autonomy and accepting duties.

We plan to participate in the 2005 AAAI Challenge. For this endeavour, we have been developing a planning system to be integrated with the basic robot behavior-producing modules to recommend tasks the robot should be working on at a given point of execution. Planning problems we are dealing with involve metric time constraints (e.g., the robot has to be on time for it presentation), resource constraints (e.g., complex robot processes such as navigation, map registration and planning consume more battery power), safety goals generated reactively at unpredicted times (e.g., charging the battery whenever it becomes low), preferences among goals (e.g., it is more critical to charge the battery than doing anything else; or presenting the paper on time has priority over helping other attendees) and uncertainty in plan execution (e.g., the time it takes to navigate from one point to another depends on the accuracy of the robot's navigation behaviour).

The decisions about which features are to be handled using an automated planning system and which ones are to be managed by the robot behavior-producing modules are ultimately a matter of design choice, depending on the capabilities of the planning system being used. We may choose to ignore uncertainty during the planning phase, and handle it at the level of the robot architecture, by generating deterministic plans that are sequences of actions, and monitoring failures to re-invoke the planner whenever necessary, as is done with Xavier [4]. It is also conceivable not to involve any automated planning at all and still manage to make the robot accomplish complex tasks, as did many teams participating in previous AAAI Challenges [8].

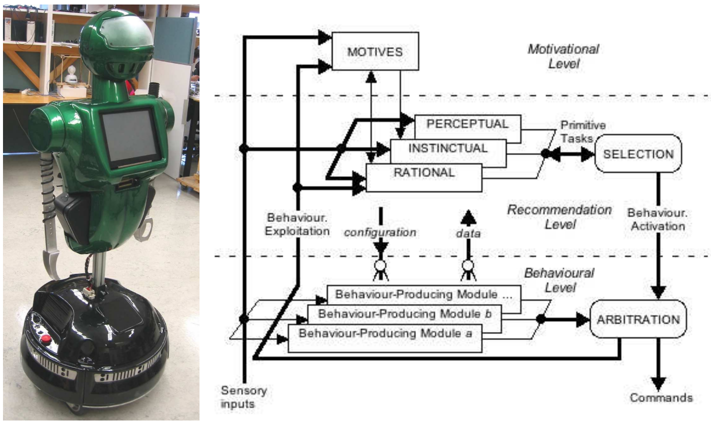
In our case, we have decided to include some form of planning to increase modularity by having automatically behaviors. This should facilitate reconfigurations of the robot onsite as well as adaptation of the robot to new applications that are similar to existing ones. Experimenting with existing probabilistic and nondeterministic planners such as SGP [12] and MBP [1] to handle some degree of uncertainty, we observed that they do not scale up to the complexity of problems in the AAAI Challenge. Using deterministic planners (SAPA [2], Metric-FF [5] and SHOP2 [9], which are among today's most efficient planning systems) and assuming an interleaving of execution monitoring and re-planning for failure situations, the performance observed revealed also to be insufficient.

A planning problem is in general both an action selection problem (i.e., selecting actions relevant to the goal) and a scheduling problem (i.e., assigning resources to actions, include time resource) [11]. The problem of planning to attend a conference is more biased towards scheduling then towards action selection, whereas the planners we tested are biased towards action selection or task decomposition. Based on these observations we designed a planner that has a task scheduling bias (i.e., it more often tries to assign tasks to time windows than examining alternate task decomposition methods). It combines HTN decomposition from SHOP2 [9], with plan post-processing ideas from SAPA [2], to obtain plans with time window execution flexibilities. Preliminary results show that our algorithm performs much better than the others, with a sufficient level of performance that will complement other decision modules embedded in the robot's decison-making architecture.

2 Robot Platform

The robot platform we intend to use is the U2S robot (see Figure 1a), developed at Université de Sherbrooke. It will be equipped with autonomous navigation capabilities using a laser SICK range finder, visual recognition of badges, faces and people, audio sound source localization and separation capabilities, and a graphical tactile interface. The decision-making architecture is based on EMIB [7] (see Figure 1b). EMIB's behavior-producing modules (BPM) are essentially processes that produce commands that directly control the robot's actuators. Primitive tasks are conceptually similar to abstract primitive tasks [9] or actions [2] in AI planning, with the key difference that in EMIB they are not directly executable by the robot. Rather, they are interpreted as recommendations in determining what BPM are desirable and which ones are not, and these recommendations compete with recommendations from other recommendation modules, using an emotion/intentional arbitration metaphore, to produce the actual robot commands.

From a planning application point of view, many robot architectures have already integrated task planning and robot execution, including the Procedural Reasoning Sys-



(a) U2S robot (b) EMIB Architecture

Fig. 1. Our robotics platform and decision mechanism

tem (PRS) [3] and Xavier [4]. Our approach differs with these approaches by having the planner more tightly coupled with the underlying behaviour-based architecture. Our planner competes with the other robot components that recommend its basic actions, hence preserving the design principles of behaviour-based robot programming. This in line with the proposal in [10], but our approach is more general in that EMIB decouples behavior activation conditions from the corresponding actuators. Therefore, the planner is not a central component of the architecture, of which all the decisions will be based. This limitation of the scope of intervention of the planner allows us to use other mechanisms more suitable to handle unpredictable events. Integrating a planner into EMIB is one issue, and designing the planner itself is another. So far, efforts in the development have been on implementing the planner. We tested it using a simplified simulator in which EMIB's primitive behaviors are directly implemented without going through the emotion/intentional arbitration.

3 Reactive Planning Process

To deal with uncertainty, a deterministic planning algorithm is invoked in a reactive loop that combines a plan monitor that monitors the execution of a plan, to update the current plan by removing completed tasks, inserting new tasks, removing existing tasks in order to accommodate tasks with a higher priority, (e.g., it changes its mission or the task turns out to be infeasible), or repairing its plan when failure occurs (e.g., by inserting new tasks to re-enable a failed task precondition, taking into account its time execution window). Tasks priorities are handled essentially via plan merging by accommodating first tasks with a higher priority. For instance, once the robot has its plan for the mandatory tasks, it should accept volunteer duties depending on time availability. This is done by first trying to merge the duties with existing tasks using heuristics. If unsuccessful, a systematic merger is attempted by planning for a conjunction of mandatory and volunteer tasks. If still unsuccessful, the volunteer duties are refused.

Plans are generated with safety constraints such as maintaining a minimum battery level. However, at execution time, a plan may violate them if something unexpectedly goes wrong (e.g., going to the conference room may take longer than expected, overusing the battery in the process). This situation can be detected by the plan monitoring process, causing the introduction of battery-recharge task at an appropriate time.

4 Conference Planner

Our planner (ConfPlan) is a HTN planner but includes a plan post-processing procedure and a built-in time variable in the state representation of which the planner can take advantage. Methods that decompose tasks into smaller tasks can add metric time constraints. With these new constraints, some partial orders of tasks may be discovered. Since our domain is biased towards scheduling, these partial orders help to reduce the search space. Like SHOP2, when primitive tasks are introduced in the resulting plan, we apply effects on the current state. The current-time variable is increased as in SAPA. Once a valid plan is generated, it is passed to the post-processing phase. This procedure takes as input a total ordered plan and generates a partial ordered plan for more flexibility at execution time.

Figure 2 compares ConfPlan to SAPA [2], Metric-FF [5] and SHOP2 [9], on planning problems of different complexities in the conference domain. The results show that ConfPlan solves more problems in the set than any of the three other planners. These problems involve goals of registering at the registration desk, making (or assisting to) one or more presentations at specified periods, presenting posters and taking picture of interesting posters. SAPA has timed initial literals of which we took advantage. For Metric-FF and SHOP2, which do not support them, we introduced a metric time variable. Because we need a fast reactive planning engine, plan solutions taking more than two minutes were rejected.

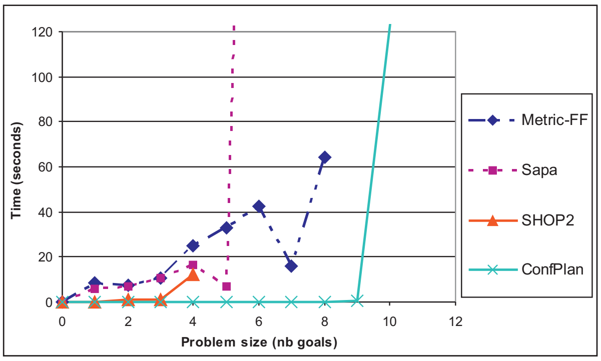


Fig. 2. Performance comparaison

5 Conclusion

From a general AI planning perspective, our planner improves SHOP2 algorithm by adding time constraints and integrating SAPA like post-processing to obtain flexibility. This flexibility will be crucial when repairing a plan or merging in new tasks. As mentioned before, the development of ConfPlan is primarily motivated by the need of a planning system into the EMIB robot architecture with the aim of participating at the next 2005 AAAI Challenge. We have just iniitiated the integration work. It remains interesting to see how the planner will behave on problems different to the conference domain: we are currently in the process of assessing this.

Acknowledgment

This research is supported by the Natural Sciences and Engineering Research Council of Canada and the Canada Research Chair program. We would also like to thank anonymous reviewers, Sylvain Clavette and Ian Bailey for helpful comments.

References

1. A. Cimatti, M. Pistore, M. Roveri, and P. Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 147(1-2):35–84, 2003.
2. M.B. Do and S. Kambhampati. Sapa: A scalable multi-objective metric temporal planner. *Journal of Artificial Intelligence Research*, 20:155–194, 2003.
3. Ingrand F, F and O. Despouys. Extending procedural reasoning toward robot actions planning. In *ICRA*, pages 9–14, 2001.
4. K.Z. Haigh and M. Veloso. Interleaving planning and robot execution for asynchronous user requests. *Autonomous Robots*, March 1998.
5. J. Hoffmann. The Metric-FF planning system: Translating ”ignoring delete lists” to numeric state variables. *Journal of Artificial Intelligence*, 20:291–341, 2003.
6. B.A. Maxwell, W. Smart, A. Jacoff, J. Casper, B. Weiss, J. Scholtz, H. Yanco, M. Micire, A. Stroupe, D. Stormont, and T. Lauwers. 2003 AAAI robot competition and exhibition. *AI Magazine*, 25(2):68–80, Summer 2004.
7. F. Michaud. EMIB - computational architecture based on emotion and motivation for intentional selection and configuration of behaviour-producing modules. *Cognitive Science Quarterly*, pages 340–361, 2002.
8. F. Michaud, J. Audet, D. Létourneau, L. Lussier, C. Théberge-Turmel, and S. Caron. Experiences with an autonomous robot attending the AAAI conference. *IEEE Intelligent Systems*, 16(5):23–29, 2001.
9. D.S. Nau, T.C. Au, O. Ilghami, U. Kuter, J.W. Murdock, D. Wu, and F. Yaman. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, 20:379–404, 2003.
10. M. Nicolescu and M. J. Mataric. Deriving and using abstract representation in behavior-based systems. In *National Conference on Artificial Intelligence ’00*, page 1087, 2000.
11. D.E. Smith, J. Frank, and A.R. Jonsson. Bridging the gap between planning and scheduling. *Knowledge Engineering Review*, 15(1), 2000.
12. D.S. Weld, C.R. Anderson, and D.E. Smith. Extending graphplan to handle uncertainty & sensing actions. In *National Conference on Artificial Intelligence ’98*, pages 897–904, 1998.