Planning domain for robot controll

Jindřich Vodrážka and Roman Barták (supervisor)
Charles University in Prague, Faculty of Mathematics and Physics
Malostranské nám. 25, 118 00 Praha 1, Czech Republic

Research problem

Significant progress of robotics brings new possibilities in development of robots for tasks that require autonomy.

The controll system for a robot can be designed manually by combining various primitive sensing and acting components together with additional software which processes inputs and generates output or takes actions by controlling available actuators.

From a certain point of view, these components can be seen as building blocks. At the hardware level there are simple components such as sensors or motors controlled by their respective drivers. By grouping these low level components together we get more complicated modules capable of performing more abstract tasks (e.g. a module that controlls all four motors in quadrotor drone can move the drone in 3D space). The top level component can be defined by the outer[1] interface of whole robot. Since the robot is embedded in its environment we can model also the interface between the robot and the environment itself. In this manner we would end up with hierarchy of components defined by their respective interfaces.

Every component in such a system can interac in different manner. Some components may be active all the time (e.g. energy monitoring) and other may be switched on and off (e.g. reactive control module providing hover capability of quadrotor drone). Since the interfaces of the components has to be formally defined, it should be possible to describe it as a planning domain.

The research question is: How to design a planing domain that would provide a generic frame for robotic components and interactions between them? Once the domain is available it can be used to describe a complete system or some of its components. This approach will yield a model-based way of control over the described components set. This could be a big step forward towards closer integration of automated planning in robotics.

State of the art

Results from the area of automated planning are already used in robotics to solve various subtasks e.g. path planning, configuration planning and perception planning. There are also systems that use automated planning for complete controll over the robot but these are still quite rare and specialized in certain areas like the space research (Estlin et al. 2005) where there is no other option than full autonomy. Further application of the automated planning in robotics is still acive and important topic.

There are three main branches in domain independent automated planning that can be considered when selecting a planner for the integration into the controll system of the robot.

Firstly there are planners that use state space representations (e.g. (Hoffmann and Nebel 2001), (Coles et al. 2012)). The state space representation is not very flexible ((Boddy 2003),(Vodrážka and Barták 2014)). However the PDDL formalism (Fox and Long 2003), which defines the representation, is a lingua franca in planning community.

Secondly there are planners that define their search space through timelines (e.g. (Barreiro et al. 2012)). Among the strongest advantages of the timeline-based approach is the possibility of straightforward representation of extended goals and reasoning about time and resources.

Thirdly we can identify another group of planners that is based on HTN decomposition (e.g. (Nau et al. 2003)). This approach is very different from the previous two because it does not try to compose solution from elements (actions in state space representation or tokens in the timeline-based approach) but relies on decomposition of predefined methods instead.

While offline planners are quite common, there are only few systems that integrate planning and execution (e.g. (Dvořák et al. 2014),(Ingrand et al. 1996),(Beetz and McDermott 1994))

The PDDL formalism is the most widespread formalism used in automated planning. However its usage outside of the IPC competition is rather scarce.

It seems that real-world planning domains are better described with formalisms that are based on the notion of timelines (Barreiro et al. 2012) or state variables ((Smith, Frank, and Cushing 2008),(Vodrážka and Barták 2012)).

Diversity of hardware architectures has motivated development of various middleware platforms (Quigley et al. 2009; Gerkey, Vaughan, and Howard 2003; Soetens 2006; Fitzpatrick, Metta, and Lorenzo 2008). Not only these platforms enable modular design of robots but they also con-

[1] Sensors and actuators directly interacting with the environment.

tribute greatly to knowledge sharing, reuse and integration.

There are complex systems (Beetz, Mosenlechner, and Tenorth 2010; Tenorth and Beetz 2009) built on previously mentioned platforms.

Recent work (Buehler and Pagnucco 2014) is demonstrating a hardware independent way to model capabilities of a mobile robot as a planning domain in PDDL. This allows users to define simple tasks (e.g. pick up a ball from table) as a planning problem. Then a special planner is used to solve it. The planner can handle:

- object generation during planning,
- concurency requirements in actions,
- external procedure calls.

In order to do enable this features it was necessary to extend the PDDL specification.

Objectives outline

Firstly the main objective of my research is to integrate the automated planning in robotics by describing a general purpose planning domain for description of robotic components.

Secondly I hope to make it easier for a user to program a robot capable of autonomous operation. Such robot would be easily reprogrammed only through change in its goal specification.

Methodology and expected outcome

In the early stage there is a question how to describe an interface for a robotic component. Since there are working implementations already used in middleware platforms (ROS,Orocos,Player,YARP) the main problem to overcome will be the design of a formal model based on these implementations.

As a next step a formal model for various types of interactions will be necessary. According to preliminary research there are following common types:

- *data stream* - sensor data are usually communicated as a stream of data packages transmited one by one at certain frequency.
- *service* - various components can be designed to provide some service (e.g. camera can be switched on/off). The requests can be also parametrized. Respond to requested service is instantaneous.
- *action* - purpose of some components is to perform continuous action (e.g. move 1m forward). Unlike services, actions takes time to execute and it is possible to monitor them and controll their execution.
- *parameters* - each component can has parameters which controll its behavior. These parameters can be read and set by other components.

All these interactions need to be described in order to reason about the whole system of interconnected components.

In the next phase the resulting model will be tested in simple scenarios using various combinations of modules described with previously designed formal representation. At this point a planner will be needed to test the resulting domain. As the requirements on the planner are unknown there are following ways how to proceed:

- use some available planners with simplified problem description,
- extend an existing planner with missing features,
- implement a new planner.

The main objective here is not the developement of the planner per se but a stable interface provided through the resulting domain model.

As a test of scalability of this approach to robot controll a robotic simulator will be used.

In order to verify the idea in field test a quadrotor drone will be used to demonstrate autonomous behavior specified only as a planning problem described within the resulting planning domain.

In the work mentioned earlier (Buehler and Pagnucco 2014) the robot was able to asses its capabilities, compute a plan and execute it as a sequence of commands. Authors caried out experiments in order to measure performance of the system (execution time was compared with predictions based on the capability model) while executing solution plans for a few scenarios (e.g. Pick up the ball from the table).

We intend to focus on the ability of the robot to get the task done and to change plan during execution if necessary. For a quadrotor drone we expect to run tests with tasks simmilar to: Follow object X as long as possible and return safely back to starting position.

Expected outcome of the research is a working methodology for design of an robot/agent capable of autonomous operation in given environment with ability to pursue predefined objective.

References

Barreiro, J.; Boyce, M.; Do, M.; Frank, J.; Iatauro, M.; Kichkaylo, T.; Morris, P.; Ong, J.; Remolina, E.; Smith, T.; et al. 2012. Europa: A platform for ai planning, scheduling, constraint programming, and optimization. *Proceedings of the 22nd International Conference on Automated Planning & Scheduling (ICAPS-12)–The 4th International Competition on Knowledge Engineering for Planning and Scheduling*.

Beetz, M., and McDermott, D. 1994. Improving robot plans during their execution. In *Second International Conference on AI Planning Systems*, 3–12. Morgan Kaufmann.

Beetz, M.; Mosenlechner, L.; and Tenorth, M. 2010. Cram–a cognitive robot abstract machine for everyday manipulation in human environments. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 1012–1017. IEEE.

Boddy, M. 2003. Imperfect match: PDDL 2.1 and real applications. *Journal Of Artificial Intelligence Research* 20.

Buehler, J., and Pagnucco, M. 2014. Planning and execution of robot tasks based on a platform-independent model of robot capabilities. In *Frontiers in Artificial Intelligence and Applications, Volume 263: ECAI 2014*, 171–176.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2012. Colin: Planning with continuous linear numeric change. *Journal of Artificial Intelligence Research* 44:1–96.

Dvořák, F.; Bit-Monnot, A.; Ingrand, F.; and Ghallab, M. 2014. A flexible anml actor and planner in robotics. In *PlanRob Workshop Proceedings, 24th International Conference on Automated Planning and Scheduling*, volume 2, 12–19.

Estlin, T.; Gaines, D.; Chouinard, C.; Fisher, F.; Castano, R.; Judd, M.; Anderson, R.; and Nesnas, I. 2005. Enabling autonomous rover science through dynamic planning and scheduling. In *Aerospace Conference, 2005 IEEE*, 385–396.

Fitzpatrick, P.; Metta, G.; and Lorenzo, N. 2008. Towards long-lived robot genes. *Robotics and Autonomous Systems* 56(1):29 – 45. Human Technologies: Know-how.

Fox, M., and Long, D. 2003. Pddl2. 1: An extension to pddl for expressing temporal planning domains. *J. Artif. Intell. Res.(JAIR)* 20:61–124.

Gerkey, B.; Vaughan, R.; and Howard, A. 2003. The player/stage project: Tools for multi-robot and distributed sensor systems. In *11th International Conference on Advanced Robotics (ICAR 2003)*.

Hoffmann, J., and Nebel, B. 2001. The ff planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 253–302.

Ingrand, F. F.; Chatila, R.; Alami, R.; and Robert, F. 1996. Prs: A high level supervision and control language for autonomous mobile robots. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, 43–49. IEEE.

Nau, D.; Au, T.-C.; Ilghami, O.; Kuter, U.; Murdock, J. W.; Wu, D.; and Yaman, F. 2003. Shop2: An htn planning system. *J. Artif. Intell. Res.(JAIR)* 20:379–404.

Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. Y. 2009. Ros: an open-source robot operating system. *ICRA workshop on open source software* 3(3.2):5.

Smith, D. E.; Frank, J.; and Cushing, W. 2008. The anml language. *In ICAPS Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*.

Soetens, P. 2006. *A Software Framework for Real-Time and Distributed Robot and Machine Control*. Ph.D. Dissertation, Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium.

Tenorth, M., and Beetz, M. 2009. Knowrob–knowledge processing for autonomous personal robots. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 4261–4266. IEEE.

Vodrážka, J., and Barták, R. 2012. A novel framework for modeling planning domains with state variables. *30th Workshop of the UK Planning And Scheduling Special Interest Group (PlanSIG)*.

Vodrážka, J., and Barták, R. 2014. On modeling formalisms for automated planning. In *16th International Conference, AIMSA 2014. Proceedings*, volume 8722, 270–277.