Hybrid Planning in Self-Adaptive Systems

Ashutosh Pandey
ashutosp@cs.cmu.edu
Ph.D. Advisor: David Garlan
Carnegie Mellon University, Pittsburgh, PA

***Abstract*—Self-adaptive software systems make decisions at run time that seek to change their behavior in response to faults, changing environments and attacks. Therefore, having an appropriate planning approach to find an adaptation plan is critical to successful self-adaptation. For many realistic systems, ideally one would like to have a planning approach that is both quick and finds an optimal adaptation plan. However, due to the fundamental trade-off between quality and timeliness of planning, often designers have to compromise between an approach that is quick to find a plan and an approach that is slow but finds an optimal plan. To deal with this trade-off, this work proposes a hybrid planning approach that combines more than one planning approach to bring their benefits together.**

I. Introduction

A typical control loop in many self-adaptive systems has four fundamental computational components: Monitoring-Analysis-Planning-Execution (MAPE), where planning is responsible for determining a plan for self-adaptation [11].

For many self-adaptive systems, quality and timeliness of planning are two particularly important requirements to instantiate the planning component. The quality of planning, generally, relates to the likelihood of a plan meeting the adaptation goals under the assumption that the plan is available instantaneously, when required. A plan with a higher quality is more likely to meet the adaptation goals. For domains, such as safety-critical systems, quality of planning is important since a bad plan could lead a system to an irreparable failure state.

Besides quality, finding an adaptation plan in a timely manner is another important requirement for planning [17] [18]. Timeliness is important particularly for systems operating in real-time domains. For instance, after a security threat detection, if a system fails to determine a defense plan in a timely manner, there is a risk that the system might be compromised resulting in failure to meet the goal of self-protection.

To instantiate the planning component various off-the-shelf approaches, such as case-based reasoning [13], fuzzy-logic [16] and automated planning [10], have been suggested by the research community to determine an adaptation plan. Unfortunately, for a planning approach, quality and timeliness are potentially conflicting requirements. Planning, in essence, is a search/optimization process performed over the space of possible plans – more complete searches provide better quality guarantees, but require more time to complete. As a consequence, while choosing an off-the-shelf planning approach, self-adaptive systems today must compromise between one of the two requirements leading to systems that typically can either respond quickly, or provide a high-quality adaptation but not both.

However, there are many systems such as Amazon Web Services [4] and Netflix [5], that need quick planning under urgent circumstances, but over the long term the performance should be optimized with deliberative planning. Ideally, for such systems, a planning approach is needed that can find optimal adaptation plans in a timely manner.

One direction, explored by the artificial intelligence (AI) community, is to develop customized solutions applicable to a particular domain or a set of planning problems. While there are successful instances of such customized solutions [3] [2] [1], developing them is a non-trivial task for software engineers since it requires deep experience in both: planning technology and the domain. Moreover, developing such solutions is time-consuming, and hence a costly process. Furthermore, the success of such solutions varies from one problem to another [6] [7] [8].

In this thesis, we propose a novel idea of hybrid planning for self-adaptive systems that combines more than one off-the-shelf approaches to deal with the trade-off between quality and timeliness of planning. The key idea is to use a reactive planning approach that provides a quick (but potentially a sub-optimal) response to an emergency problem, but simultaneously use a deliberative approach providing a close to optimal plan. Once a better plan is ready, it takes over execution from the sub-optimal plan to provide a higher utility thereafter.

The hybrid planning has a number of potential advantages over custom planning solutions. Instead of going through the non-trivial process of developing a new algorithms/heuristics, hybrid planning efficiently combines off-the-shelf planners: using off-the-shelf planners shields software engineers from dealing with the complexity of developing new planners. Moreover, by combining multiple planning approaches, hybrid planning raises the level of generality at which a planning problem is solved [9]; depending on domain/problem, hybrid planning chooses an appropriate planner or a combination of planners from a given set of planners.

Our preliminary work has already demonstrated the potential of hybrid planning in a constrained context [14]. We went a step further to formally define the hybrid planning that helped us understand the general nature of this problem [15]. As a result, we found that a practical solution to realize the hybrid planning in self-adaptive systems requires dealing with two key sub-problems:

- *The Planner Selection Problem (PSP)*: when reactive and deliberative planners are combined together to instantiate a hybrid planner, the first challenge is to figure out which planner(s) should be triggered for a given planning problem; this decision depends on the features of a problem and the planners.
- *The Planner Coordination Problem (PCP):* to find a balance between quality and timeliness, hybrid planning relies on transitioning from (possibly) a low-quality plan determined by the reactive planner to a higher quality plan determined by the deliberative planner. However,

guaranteeing a seamless transition between two plans is a challenge.

II. Objectives

To deal with the trade-off between quality and timeliness of planning, we propose a hybrid planning approach that, we claim, will improve the following three qualities of planning in self-adaptive systems:

- **Effectiveness:** For a self-adaptive system, generally speaking, effectiveness is a measure of its ability to meet the adaptation goals: this ability is influenced by the quality and the timeliness of planning. The hybrid planning will improve the effectiveness of self-adaptation over current planning approaches used in self-adaptive systems.
- **Generality:** The hybrid planning will be general enough to be applied to self-adaptive systems operating in different domains.
- **Flexibility:** The AI community has developed numerous planning approaches that could be used to instantiate reactive/deliberative planners. The hybrid planning will be flexible enough to be applied to different instantiations of a deliberative and a reactive planner.

The proposed research will make contributions to both the theory and the practice of hybrid planning for self-adaptive systems.

The contributions to theory are:

- a formal framework to define the problem of hybrid planning;
- a practical approach to solve the problem of hybrid planning under certain assumptions/restrictions that nonetheless apply to many self-adaptive systems.

The contributions to practice are:

- demonstrate effectiveness, generality, and flexibility of hybrid planning for self-adaptive systems using the proposed solution approach;
- suggest different combinations of deliberative and reactive planners that could be effective in the context of self-adaptive systems;
- develop methods/tools to apply hybrid planning to self-adaptive systems;
- concrete examples bridging the gap between theory and practice.

III. Methodology

Even though our theoretical formalization helped us understand the general nature of hybrid planning problem that combines any number of reactive and deliberative planners, we realized that solving this problem in it's general form is intractable [15]; therefore, we need to find an approximate solution for the two sub-problems noted above. To this end, we make some simplifying assumptions such as restricting the number of planners to a deliberative and a reactive planner.

Now I briefly explain our methodology to solve the two sub-problems.

A. PSP

In the proposed hybrid planning approach, if the deliberative planner is ready with a plan, it is preferred over the reactive planner. In other words, if the deliberative planner would have been instantaneous, a system would use it all the time. However, since the deliberative planner is likely to be slow in determining a plan; this delay could cause a drop in system's utility. Therefore, while deliberative planning is done in the background, the reactive planner is used to provide a quick response, thereby likely improve the system's utility.

Unfortunately, a low-quality plan by the reactive planner could also lead a system to a permanent failure state, which could be a problem, particularly for safety critical systems. Therefore, for a given planning problem, a mechanism is needed to decide whether to use the reactive planner or not. We propose two mechanisms to meet this goal.

- Constraint based invocation: In this approach, the reactive planner is only invoked when a constraint is violated; the idea is to use the reactive planner to provide a quick response to the constraint violation. We validated this approach for a self-adaptive cloud system [14], however, another validation on a different domain is pending.
- Utility based invocation: Compared to constraint-based invocation, utility based invocation is a more general solution to PSP, since invocation of the reactive planner does not depend only on a constraint violation; instead, it covers broad categories of situations to invoke the planner. For utility based invocation, we propose a case-based reasoning approach where, for a given planning problem at run time, the decision to invoke the reactive planner depends on its performance on the similar problems seen in the past. Developing an effective metric to compare the similarity between two planning problems, and a priori approximation of the performance of a planner for a given problem is still an open research question for us.

B. PCP

The formal model ensures a seamless transition between plans in two ways: (a) defining the plan structure as universal plan [12] i.e., state-action pairs for all the reachable states for a given planning problem, where each pair suggests the action to be executed for the corresponding state; and (b) assuming the domain to be Markovian [19]. These two elements ensure that if the reactive and the deliberative planning have the same initial state, once a deliberative plan is ready, it can take over a plan execution from a reactive plan in any of the future states, ensuring the optimal plan execution thereafter.

IV. Research Plan

Following table summarizes my research plans.

Task	Status	Est. Time
Formalization of the hybrid planning problem	Completed	
First validation of constraint-based invocation	Completed	
Second validation of constraint-based invocation	Pending	2 months
First validation of utility-based invocation	Pending	6 months
Second validation of utility-based invocation	Pending	6 months
Dissertation writing	Pending	4 months

V. Conclusion

This paper argues for the need to directly deal with the fundamental trade-off between quality and timeliness of planning. This is a long-standing problem, yet no general solution exists to this problem. My thesis will address this fundamental problem of self-adaptive systems in achieving both through a new approach based on hybrid planning.

Acknowledgments

This material is based on research sponsored by AFRL and DARPA under agreement number FA8750-16-2-0042. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFRL, DARPA or the U.S. Government.

References

[1] J. Hoffmann and B. Nebell, "The FF Planning System: Fast Plan Generation Through Heuristic Search", Journal Artificial Intelligence Research, vol. 14, 2001, pp. 253 - 302.

[2] Blai Bonet and Hector Geffner, "Planning as heuristic search", Artificial Intelligence, vol. 129, 2001, pp. 5-33.

[3] Sungwook Yoon et al., "FF-Replan: A Baseline for Probabilistic Planning", Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling, ICAPS 2007, Providence, Rhode Island, USA, September 22-26, 2007.

[4] https://aws.amazon.com/ec2/sla/

[5] http://techblog.netflix.com/2010/12/5-lessons-weve-learned-using-aws.html

[6] David H. Wolpert and William G. Macready, "No free lunch theorems for search". Technical Report SFI-TR-95-02-010, 1995, Sante Fe, NM, USA

[7] David H. Wolpert and William G. Macready, "No free lunch theorems for optimization". IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, April 1997

[8] D. H. Wolpert and W. G. Macready. Coevolutionary free lunches. IEEE Transactions on Evolutionary Computation,, vol. 9, no. 6, pp. 721-735, 2005

[9] Edmund Burke, Graham Kendall and Jim Newall. Hyper-Heuristics: An Emerging Direction in Modern Search Technology. Handbook of Metaheuristics, Vol. 57, July 2003, Pages 457-474

[10] Daniel Sykes et al., "Plan-Directed Architectural Change For Autonomous Systems", Sixth International Workshop on Specification and Verification of Component-Based Systems (SAVCBS 2007), September 3-4, 2007, Cavtat near Dubrovnik, Croatia, 2007©ACM, ISBN 978-1-59593-721-6/07/0009.

[11] J. O. Kephart and D. M. Chess, "The vision of autonomic computing", Computer, vol 36, issue 1, Jan 2003. doi:[10.1109/MC.2003.1160055]

[12] Malik Ghallab et al., "Automated Planning: Theory and Practice", Morgan Kaufmann, May 2004.

[13] Shang-Wen Cheng et al., "Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure", 1st International Conference on Autonomic Computing (ICAC 2004), pp. 276-277. doi:[10.1109/MC.2004.175]

[14] A. Pandey et al., "Hybrid Planning for Decision Making in Self-adaptive Systems", in International Conference on Self-Adaptive and Self-Organizing Systems, ser. SASO 2016, 2016, pp. 12-16.

[15] A. Pandey et al., "Towards a Formal Framework for Hybrid Planning in Self-Adaptation", in 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS-17

[16] Jamshidi et al. Autonomic resource provisioning for cloud-based software. in 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS-14

[17] Mazeiar Salehie and Ladan Tahvildari. Self-Adaptive Software:Landscape and Research Challenges. ACM Transactions on Autonomous and Adaptive Systems, Vol. V, No. N, March 2009

[18] Frank D. Macías-Escrivá, Rodolfo Haber, Raul del Toro, Vicente Hernandez. Self-adaptive systems: A survey of current approaches, research challenges and applications. Expert Systems with Applications 40 (2013) 7267-7279

[19] Mausam and Andrey Kolobov, "Planning with Markov Decision Processes: An AI Perspective", Synthesis Lectures On Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, June 2012, ISBN:9781608458875

[20] Andrew Berns and Sukumar Ghosh, "Dissecting Self-* Properties", Proceedings of the 2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO '09), Pages 10-19, September 14 - 18, 2009