CARPooL: Collective Adaptation using concuRrent PLanning

Demonstration

Daniel Furelos-Blanco
Universitat Pompeu Fabra
Barcelona, Spain
daniel.furelos@upf.edu

Antonio Bucchiarone
Fondazione Bruno Kessler
Trento, Italy
bucchiarone@fbk.eu

Anders Jonsson
Universitat Pompeu Fabra
Barcelona, Spain
anders.jonsson@upf.edu

ABSTRACT

In this paper we present the CARPooL demonstrator, an implementation of a Collective Adaptation Engine (CAE) that addresses the challenge of collective adaptation in the smart mobility domain. CARPooL resolves adaptation issues via concurrent planning techniques. It also allows to interact with the provided solutions by adding new issues or analyzing the actions done by each agent.

KEYWORDS

Socio-Technical Systems; Collective Adaptation; Ensembles; Sustainable Urban Mobility; Concurrent Planning

ACM Reference Format:

Daniel Furelos-Blanco, Antonio Bucchiarone, and Anders Jonsson. 2018. CARPooL: Collective Adaptation using concuRrent PLanning. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

Collective Adaptive Systems (CASs) [1] consist of diverse heterogeneous agents composing a socio-technical system [1, 16]. Individual agents 'opportunistically' enter a system and self-adapt in order to leverage other agents' resources and capabilities to perform tasks more efficiently or effectively. Self-adaptation within a collaborative system is a challenging task [15]. Changes in the behavior of one agent may break the consistency of the whole collaboration or have negative repercussions on other agents. Therefore, self-adaptation of an individual agent does not only aim at achieving its own goals but also the emerging goals of dynamically formed sub-systems.

A promising application area for CASs is the *mobility domain*. Organizing and managing the mobility services within a city, meeting travelers expectations and properly exploiting the available transport resources is becoming a more complex task. The inadequacy of traditional transportation models is proven by the growth of alternative and social initiatives aiming at a more flexible, customized and collective way of transport. To be collective, a mobility service should offer a way to organize teams of citizens that need to reach equal or closed destinations starting from different locations.

Carpooling is an example of collective service: it provides procedures that allow users to offer resources (i.e., cars) and to ask for them (i.e., searching a ride). Coordination between different participants of a team is necessary to reach each destination, preferably in time. Indeed, depending on their location, their route and additional activities (e.g., refueling), they coordinate their departure times by communicating with each other. By sharing a resource, people save gas and money, reduce auto emissions, pollution, etc. Although carpooling looks very promising and more sustainable, it has limits on how the resolution of unwanted situations is managed. For instance, suppose that a road on the car route becomes blocked because of an accident. In this case, the car driver may need to find an alternative route, which can be longer than the current one, thus causing the passengers on board to be late to their destinations. This situation shows how an unwanted event, occurring daily in big cities, can cause a chain of adaptation issues that must be managed.

Previous studies attempted to compute joint plans for multiple agents in navigation scenarios using *concurrent planning* [10, 12]. However, they usually focused on satisfying certain constraints such as not colliding rather than fostering collaboration. *Numeric planning* [14] allows the identification of the optimal choice based on costs and resources during navigation. The key disadvantages of numeric planners include they are usually more complex, and unable to plan simultaneously for more than one agent.

In the given context, we present the CARPooL demonstrator, which addresses the challenge of collective adaptation in the smart mobility domain via concurrent planning.

2 METHODOLOGY

In this section we explain the theoretical framework underlying the CARPooL demonstrator.

The term *ensemble* denotes large-scale systems of systems that may present substantial socio-technical embedding [9, 16]. Ensembles typify systems with complex design, engineering and management, whose level of complexity comes specifically from gathering and combining in the same operating environment many heterogeneous and autonomous components, systems and users, with specific concerns. Ensembles must self-adapt to sustain the variations induced by their socio-technical nature as well as the high degree of unpredictability and dynamism of their operating environments.

Our approach addresses the challenge of collective adaptation by proposing a new notion of ensembles that enables systems with collective adaptability to be built as emergent aggregations of autonomous and self-adaptive agents. Each agent is defined by a set of *roles* (e.g. carpool driver or passenger). A role is determined by its collaborations with other roles. Collaboration involves taking *actions* and generating *issues*, i.e. formation of critical situations. In our context, issues could be blocked streets that force an agent to change its planned route. When an issue arises, a role can choose to handle the issue using one of its *solvers*.

Key properties of our approach include (i) the emphasis on collaboration towards fulfillment of individual, diverse goals, and (ii)

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

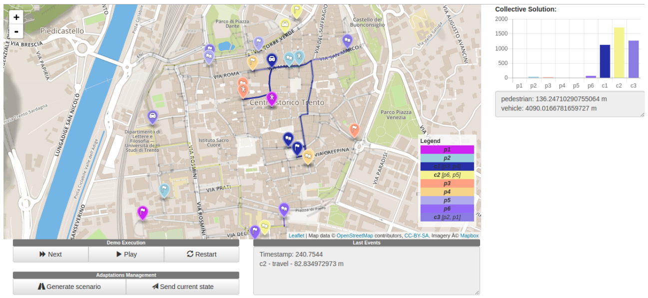

Figure 1: Screenshot of CARPooL's Scenario Viewer.

the heterogeneous nature of an ensemble with respect to the roles, behaviors and goals of its participants. These properties distinguish our approach from other types of ensemble models, such as swarms, multiagent systems, and agent-based organizations. All elements in a swarm exhibit a uniform behavior, and global shared goal [5, 11]. In contrast, those within a multiagent system and agent-based organization may display several distinct roles and behaviors, but the differentiation is still limited and often pre-designed [7].

We use concurrent planning to model decision problems involving multiple agents. An integral part of our system is the ability to generate and solve concurrent planning problems on-the-fly. We use an algorithm called TPSHE [6] to solve temporal planning [8, 13] problems. The reason behind selecting this algorithm is it performs well in the IPC (International Planning Competition) domains that require concurrent actions. Since one criterion of smart mobility is to bring each agent to their goal in the shortest time possible, we want to exploit the ability of temporal planning to express concurrent plans. We remark that TPSHE is *not* an optimal algorithm, although it attempts to minimize the total duration (i.e. *makespan*) of the temporal plan. Besides, it is a centralized algorithm, i.e., the actions of all agents are managed by the same controller.

3 THE CARPOOL DEMONSTRATOR

The CARPooL demonstrator[1] mainly consists in the execution of a webpage. We call this module *Scenario Viewer*, and it allows to:

- Create scenarios.
- Visualize a solution step-by-step for a given scenario.
- Introduce adaptation issues interactively (i.e., blocked streets).

The *Scenario Builder* creates a carpooling problem given:

- The number of passengers and the number of carpools.
- A map obtained from OpenStreetMap (OSM)[2]. OSM maps contain a list of the locations in the map and the links between them. The latitude and longitude are given for each location. Each link between two locations has a maximum speed limit and a list of intermediate locations.
- Two latitude-longitude pairs to form the boundaries of the map area.
- The minimum and maximum walking ranges of the passengers. These express the preference for how far a passenger is willing to walk from/to its origin/target positions.

[1]The software is available at https://github.com/aig-upf/smart-carpooling-demo.
[2]OpenStreetMap website: http://www.openstreetmap.org.

An OSM map of Trento is used. Random initial and target positions (inside the defined boundaries) are assigned to passengers and carpools. Moreover, each passenger has a random walking range between the specified minimum and maximum ranges.

To build an initial random scenario, an *OSM parser* is required to parse the input OSM map. It eases the access to the information of the map by other modules. For example, it allows to get the set of nodes inside an area bounded by two latitude-longitude pairs.

The resulting initial state is processed by the *Collective Adaptation Engine* [2, 3] to create the first set of ensembles (i.e., carpool rides). The instantiation of each ensemble is done using the *Concurrent Planner*, which is responsible for finding a plan for all involved agents in the various ensembles. The resulting plan is then visualized using the *Scenario Viewer*. More specifically, the Concurrent Planner module performs the following steps:

(1) It converts the input problem into a planning problem expressed in PDDL (Planning Domain Definition Language) [8].
(2) It runs the TPSHE temporal planner to compute a plan.
(3) The solution is converted into GeoJSON [4] format, which is used by the *Scenario Viewer*.

The *Scenario Viewer* provides a graphical representation of the solutions returned by the *Concurrent Planner* (see Figure 1). The overall plan can be navigated step by step, i.e., it shows where each agent is at a given intermediate time point. Besides, there is a chart showing the distances traveled by each agent.

Finally, the *Scenario Viewer* allows to block and unblock streets by clicking on them. The current state of the map (including these adaptation issues) can be sent to the resolution algorithm, which will follow the same three steps enumerated before. When a solution is found, the viewer refreshes automatically and shows the new plan assigned to each agent, as well as their updated traveled distances.

A video showing our demo is available at: https://youtu.be/omWu3FpZNsI.

4 CONCLUSIONS

In this paper we have presented CARPooL, a demonstrator that solves adaptation issues collectively in the mobility domain using a concurrent planning algorithm.

ACKNOWLEDGMENTS

This work has been partially supported by the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

REFERENCES

[1] 2016. FoCAS Manifesto – A roadmap to the future of Collective Adaptive Systems. (2016). http://www.focas.eu/focas-manifesto.pdf.
[2] D. Bozhinoski, A. Bucchiarone, I. Malavolta, A. Marconi, and P. Pelliccione. 2016. Leveraging Collective Run-Time Adaptation for UAV-Based Systems. In *SEAA'16*. 214–221.
[3] A. Bucchiarone, N. Dulay, A. Lavygina, A. Marconi, H. Raik, and A. Russo. 2015. An Approach for Collective Adaptation in Socio-Technical Systems. In *IEEE SASO Workshops*. 43–48.
[4] Howard Butler, Martin Daly, Allan Doyle, Sean Gillies, Stefan Hagen, and Tim Schaub. 2016. The GeoJSON Format. *RFC* 7946 (2016), 1–28. https://doi.org/10.17487/RFC7946
[5] C. Pinciroli et al. 2011. ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics. In *IROS*. 5027–5034.
[6] S. Jiménez Celorrio, A. Jonsson, and Héctor Palacios. 2015. Temporal Planning With Required Concurrency Using Classical Planning. In *ICAPS'15*. 129–137.

[7] B. H. Far, T. Wanyama, and S. O. Soueina. 2006. A negotiation model for large scale multi-agent systems. In *IRI*. 589–594.

[8] M. Fox and D. Long. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *J. Artif. Intell. Res. (JAIR)* 20 (2003), 61–124.

[9] M. Holzl, A. Rauschmayer, and M. Wirsing. 2008. Engineering of software-intensive systems. In *Software- Intensive Systems and New Computing Paradigms*. LNCS, Vol. 5380. Springer, 1–44.

[10] W. Hönig, S. Kumar, L. Cohen, H. Ma, H. Xu, N. Ayanian, and S. Koenig. 2016. Multi-Agent Path Finding with Kinematic Constraints. In *ICAPS'16*. 477–485.

[11] P. Levi and S. Kernbach. 2010. *Symbiotic-Robot Organisms: Reliability, Adaptability, Evolution*. Vol. 7. Springer Verlag.

[12] H. Ma, C. Tovey, G. Sharon, S. Kumar, and S. Koenig. 2016. Multi-Agent Path Finding with Payload Transfers and the Package-Exchange Robot-Routing Problem. In *AAAI'16*. 3166–3173.

[13] J. Rintanen. 2007. Complexity of Concurrent Temporal Planning. In *ICAPS'07*. 280–287.

[14] E. Scala, M. Ramírez, P. Haslum, and S. Thiébaux. 2016. Numeric Planning with Disjunctive Global Constraints via SMT. In *ICAPS'16*. 276–284.

[15] D. Weyns and J. Andersson. 2013. On the challenges of self-adaptation in systems of systems. In *SESoS@ECOOP, 2013*. 47–51.

[16] F. Zambonelli, N. Bicocchi, G. Cabri, L. Leonardi, and M. Puviani. 2011. On Self-Adaptation, Self-Expression, and Self-Awareness in Autonomic Service Component Ensembles. In *SASOW*. 108–113.