Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Behavior Composition Optimization*

Nitin Yadav
School of Computer Science and Information Technology
RMIT University, Melbourne, Australia
nitin.yadav@rmit.edu.au

Abstract

The behavior composition problem involves the automatic synthesis of a controller that is able to "realize" (i.e., implement) a desired target behavior specification by suitably coordinating a set of already available behaviors. While the problem has been thoroughly studied, one open issue has resisted a principled solution: *if the target specification is not fully realizable, is there a way to realize it "at best"?* In this doctoral work, we look at quantitative and qualitative ways to address this question.

1 Introduction

With computers now present in everyday devices such as mobile devices, credit cards, cars and planes or places like homes, offices and factories, the trend is to build embedded complex systems from a collection of simple components. For example, a complex surveillance system for a smart house can be "realised" (i.e., implemented) by suitably coordinating the behaviour (i.e., the operational logic) of hundreds (or thousands) of simple devices and artifacts—lights, phones, blinds, vacuum cleaners, video cameras. Such embedded systems can provide services that range from simple tasks, such as "turn on the lights in the bedroom," to more complex ones, such as "bring me a cup of coffee" or "handle house intruder" (by tracking and taking pictures of the intruder, toggling lights rapidly, and alerting the owner by email or phone). Other domains of applicability include composition of web services, factory robotic settings, and agent programming, among others.

The problem of automatically synthesising, that is, building, such an embedded controller-coordinator for a given desired, though not readily available, complex target system is called the *behaviour composition* problem and is the focus of this doctoral work. From an AI perspective, a behavior refers to the abstract operational model of a device or program, and is generally represented as a nondeterministic transition system. The classical behavior composition task has been extensively investigated in the recent literature (see [De Giacomo *et al.*, 2013] for an extensive review). However, one open issue has resisted principled solutions: *if the target behavior specification is not fully realizable, is there a way to realize it "at best"?* The fact is that, many (if not most) realistic problem instances will have *no* complete realization and a (merely) "no solution" outcome may turn out extremely unsatisfactory.

In this doctoral work, we answer such question positively. Concretely, we develop qualitative as well as quantitative principled extensions of the classical composition framework in which problem instances that are otherwise unsolvable in the classical sense, do have meaningful solution concepts beyond the highly unsatisfactory "no solution" answer. Importantly, we the proposed extensions strictly subsume the classical framework. Moreover, effective techniques for computing the new solution concepts are developed.

2 The Classical Composition Framework

The behavior composition problem amounts to synthesising a *controller* that is able to "realize" (i.e., implement) a desired, but nonexistent, *target behavior* module, by suitably coordinating a set of *available behaviors* [De Giacomo *et al.*, 2013].

Technically, a ***behavior*** is a tuple $\mathcal{B} = \langle B, A, b_0, \delta \rangle$, where *(i)* B is the finite set of states; *(ii)* A is the set of actions; *(iii)* $b_0 \in B$ is the initial state; and *(iv)* $\delta \subseteq B \times A \times B$ is the behavior's (nondeterministic) transition relation: $\langle b, \sigma, b' \rangle \in \delta$ denotes that action σ executed in behavior state b may lead the behavior to successor state b'. An (available) ***system*** is a tuple $\mathcal{S} = \langle \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$, where each $\mathcal{B}_i = \langle B_i, A, b_{0i}, \delta_i \rangle$ is referred to as an ***available behavior*** in $\mathcal{S}$ (over shared actions A). The ***target behavior*** is just a deterministic behavior. Informally, the behavior composition task is stated as follows:

> *Given a system $\mathcal{S}$ and a target behavior $\mathcal{T}$, is it possible to (partially) control the available behaviors in $\mathcal{S}$ in a step-by-step manner—by instructing them on which action to execute next and observing, afterwards, their evolution—so as to "realize" the desired target behavior, that is, it appears as if one was actually executing the target module $\mathcal{T}$.*

Though technically involved, one can formally define when a so-called *controller*, a function taking a history (i.e., a run) of the system and the next action request and outputting the index of the available behavior where the action is being delegated, *realizes the target behavior*. Such controllers are

*This work was partially funded by the Australian Research Council (DP1094627 & DP120100332).

called <u>*exact compositions*</u>, solutions to the composition problem guaranteeing the complete realization of the target in the system. It was shown that, while decidable, checking the existence of exact compositions is a demanding task, namely, EXPTIME-complete [De Giacomo *et al.*, 2013].

All the literature on behaviour composition has only dealt with exact compositions. This poses a major limitation in that for many, if not most, problem instances there will be *no* composition at all. For such cases, a (merely) "no solution" outcome is extremely unsatisfactory.

3 Quantitative Optimization

In order to better deal with non-solvable composition instances, a framework in which *non-exact* controllers can be compared is required. To that end, in [Yadav and Sardina, 2011], we proposed an extension of the classical composition problem that goes beyond strict uncertainty and judging how well a controller is expected to perform.

The outcome of this work is a *decision-theoretic* behavior composition framework, in which the task is to maximize the so-called <u>*expected realizabability*</u> of a target behavior $\mathcal{T}$ in an available system $\mathcal{S}$. To facilitate that, the two sources of uncertainty ought to be quantified. First, the uncertainty on the nondeterminism in available behaviors needs to be measured. Second, the relative importance of each potential target request at a given state is specified. Note these are reasonable assumptions in many settings, in which such information is readily available to the modeller. For example, the failure rate of a gripper arm when grasping an object may be available to the domain expert, who may also encode that, in a gardening maintenance target module, watering plants is a more frequent and important action request than collecting fruits.

With the uncertainty represented into the model, we defined the notion of <u>*optimal composition controllers*</u> based on the notion of "expected target realizability." Clearly, the new framework is able to output "best" controllers even for problems that do not accept exact controllers. Importantly, we characterized when a best controller is also an exact composition, thus subsuming the classical setting.

Second, we provided a translation of a decision theoretic behavior composition problem into a Markov decision process (MDP), and show that finding an optimal policy for such MDP amounts to finding an optimal composition.

4 Qualitative Optimization

When the uncertainty cannot be quantified, an account of composition optimization in a purely strict uncertainty setting is required. In [Yadav and Sardina, 2012] we developed a *qualitative* composition theory for *"supremal realizablilty."* Intuitively, the overarching idea here is to *look for those parts of the target module that can be realized with the available modules*, and provide this as an (approximate) solution.

So, relying on the well-established formal notion of simulation [Milner, 1971] in computer science, we characterize the notion for the closest target module $\tilde{\mathcal{T}}$ (to the original target $\mathcal{T}$) that admits a complete implementation with the behaviors at hand in system $\mathcal{S}$: $\tilde{\mathcal{T}}$ is the <u>*supremal realizable fragment*</u> (or *optimal approximation*) of $\mathcal{T}$.

Of course, it is expected that such alternative target $\tilde{\mathcal{T}}$ will generally provide less functionalities than the original one. More concretely, some execution paths may be impossible to generate with the new target (e.g., it may no more be feasible to play video games when listening to music). Moreover, the alternative target may accommodate less "freedom" of choices in executions (e.g., when requesting to watch a movie, one may now need to commit to whether one will be playing a video game or listening to radio afterwards). Nonetheless, all requests as per $\tilde{\mathcal{T}}$ will be always fulfilled.

Importantly, we proved that the supremal realizable target is indeed unique. Moreover, we showed the soundness of the supremal characterization by showing that supremals amount to synthesising *maximal* controllers, the best controllers that one could hope for. Finally, we studied effective techniques to compute supremal fragments using model checking techniques. However, so far, tight complexity bounds were identified for the special case of deterministic systems only.

5 Conclusion

The overarching objective of this doctoral project is to develop behavior composition frameworks, and techniques for the actual computation of solutions, that will accommodate non-solvable problem instances. The need for dealing with approximate solution concepts was first recognized by Stroeder and Pagnucco [2009], but has resisted principled approaches. Since those instances are arguably common ones in many realistic scenarios, this will make the behaviour composition problem applicable to a much wider range of cases.

To achieve this, we propose quantitative and qualitative frameworks for behavior composition *optimization*. To our knowledge, this is the first account that is able to accommodate instances with no complete solutions, while strictly subsuming the standard problem formulation. Remaining work in the topic includes a hybrid account integrating both approaches, finding tight bounds for the general qualitative setting, and adapting our recent planning technique [Miguel *et al.*, 2013] to supremal realizations.

References

[De Giacomo *et al.*, 2013] Giuseppe De Giacomo, Fabio Patrizi, and Sebastian Sardina. Automatic behavior composition synthesis. *Artificial Intelligence Journal*, 196:106–142, 2013.

[Miguel *et al.*, 2013] Ramirez Miguel, Nitin Yadav, and Sebastian Sardina. Behavior composition as fully observable non-deterministic planning. In *Proc. of ICAPS*, 2013. To appear.

[Milner, 1971] Robin Milner. An algebraic definition of simulation between programs. In *Proc. of IJCAI*, pages 481–489, 1971.

[Stroeder and Pagnucco, 2009] Thomas Stroeder and Maurice Pagnucco. Realising deterministic behaviour from multiple non-deterministic behaviours. In *Proc. of IJCAI*, pages 936–941, 2009.

[Yadav and Sardina, 2011] Nitin Yadav and Sebastian Sardina. Decision theoretic behavior composition. In *Proc. of AAMAS*, pages 575–582, 2011.

[Yadav and Sardina, 2012] Nitin Yadav and Sebastian Sardina. Qualitative approximate behavior composition. In *Proc. of JELIA*, pages 450–462, September 2012.