Runtime Compositional Analysis of Track-based Traffic Control Systems [*]

Maryam Bagheri
Sharif University of Technology, Tehran, Iran
mbagheri@ce.sharif.edu

Ehsan Khamespanah
University of Tehran, Tehran, Iran
Reykjavik University, Reykjavik, Iceland
e.khamespanah@ut.ac.ir

Marjan Sirjani
Malardalen University, Västeras, Sweden
Reykjavik University, Reykjavik, Iceland
marjan@ru.is

Ali Movaghar
Sharif University of Technology, Tehran, Iran
movaghar@sharif.edu

Edward A.Lee
University of California at Berkeley, California, U.S.A
eal@berkeley.edu

ABSTRACT

In this paper we address the development of dependable self-adaptive systems focusing on the specific domain of track-based traffic control systems where timing issues are critical.

1. INTRODUCTION

Self-adaptive systems are able to autonomously adjust their behavior in accordance with their perception of the environment and the system itself. These systems are usually realized through the MAPE-K feedback loops where a loop consists of Monitor, Analyze, Plan, Execute, and the Knowledge component. Guaranteeing the correctness and quality of self-adaptive systems at runtime is an important and complicated issue. Therefore, building a formal model of the system and environment as the model@runtime, and providing efficient verification, validation, and performance evaluation techniques for analyzing the model@runtime are the most important research topics in the self-adaptation domain [2, 5].

In this paper we explore our current work-in-progress, which is a continuation of our earlier work [1] for developing dependable self-adaptive track-based traffic control systems (TTCSs). In [1], we proposed the coordinated actor model to build the MAPE-K feedback loop along with the model@runtime in the domain of TTCSs. TTCSs are a kind of transportation system in which traffic is passed through the pre-specified tracks and is coordinated by a controller. For any unexpected changes in the environment, the controller observes and adapts the system to guarantee the safe movement of the traffic objects and satisfy the performance criteria given in the form of the quantitative properties. Hence, in TTCSs, adaptation can take place by rerouting the moving objects or switching between different rerouting algorithms. To build the model@runtime, an actor is associated to each track, source, and destination of the traffic and traffic objects are modeled as messages passing among the actors. Moreover, the controller is modeled as a coordinator that besides governing the message passing between the actors, encompasses the Analyze and Plan components of the MAPE-K loop. We proposed to use the Ptolemy [3] framework as the implementation platform.

Our current work-in-progress, for developing *reliable large-scale* self-adaptive systems, encompasses providing efficient analysis (formal verification and performance evaluation) techniques benefiting from compositional approaches. In the next section we briefly argue this issue.

2. PROPOSED RESEARCH

The current work-in-progress has two parts: we provide a modular and hierarchical model for building large-scale self-adaptive TTCSs; and we exploit the characteristics of the actor model to design the compositional analysis techniques, tuned to react to the changes in a timely manner.

Architecture of TTCSs: Since the traffic network is usually huge, it can be divided into smaller networks, where each network has its own traffic controller. In other words, in a large-scale traffic control system, all adaptations cannot be handled by a single MAPE-K loop, and developing multiple MAPE-K loops to capture the decentralized nature of the control system is necessary. Toward this aim, as shown in Figure. 1, we have been focusing on developing multiple coordinated actor models (multiple MAPE-K loops) and establishing the patterns for their interactions. Analysis can be efficiently accomplished and benefits from the presented modular and hierarchical architecture.

Compositional approach for analyzing self-adaptive TTCSs: The *autonomous response* to internal or environmental *changes* is an essential character of self-adaptive systems. Following an adaptation, a sequence of changes may happen in the system. For instance, in TTCSs, a set of traffic objects are rerouted if an adversarial environment condition appears in a part of the assigned route to the traffic objects. To avoid conflict, as a consequence of rerouting the traffic objects, other traffic objects may need to be rerouted. This sequence of changes must be propagated in the traffic network. It is notable that to detect or predict any requirement violation, the behavior of the system should be analyzed per each change.

Due to the large dimensions of the traffic network, employing time sensitive and scalable approaches for analysis purposes is an important issue. Therefore, we will use

[*]Copyright retained by the authors.

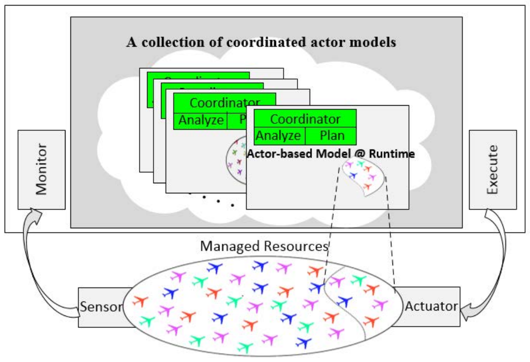
A collection of coordinated actor models
Coordinator
Analyze
Plan
Actor-based Model @ Runtime
Monitor
Execute
Managed Resources
Sensor
Actuator

Figure 1: Modeling self-adaptive track-based traffic control systems with multiple coordinated actor models

our modular and hierarchical architecture to maintain the models@runtime and employ our experiences in compositional verification of the actor-based models in Rebeca[4] to provide the theoretical foundation and tools for analysis purposes. The asynchronous communication mechanism enables actor-based models to be decomposed into independent modules. Benefiting from this feature, as shown in [4], compositional approaches can be applied on the actor-based models. The theoretical foundation of compositional analysis will be elaborated considering the coordination mechanism and timing (and probabilistic) features, and the Ptolemy tool will be extended to support it. As in [4], we think of each coordinated actor model as a component, which is an open system and the rest as the environment that makes the overall system closed. For compositional analysis, we concentrate on a component and abstract the environment by only considering external messages sent to the component. These messages are extracted using the developed interaction patterns between the MAPE-K feedback loops. The coordinator of each component is responsible for analyzing the maintained model@runtime in that component.

We call our approach for runtime compositional analysis *Magnifier*. To clarify the approach, imagine a magnifier that moves upon the model along with any change that happens in the system. When a change happens in a component, the magnifier zooms in on the component. Consequently, the component's coordinator analyzes its model@runtime. By propagating the changes to other components, magnifier zooms out and zooms in on the next components. Toward this aim, three questions should be answered.

- How is a change detected? The Monitor component of each MAPE-K feedback loop (coordinated actor model) updates the model@runtime in the Knowledge component and uses a set of pre-specified rules to detect a change in a component of the decomposed model.

- How is a component separately analyzed? To use the compositional analysis techniques, providing an abstract model of the component's environment is needed. For this purpose, the overall model, which consists of multiple MAPE-K feedback loops is executed such that the incoming messages to the magnified component are extracted, while the magnified component is analyzed through its coordinator. By occurrence of any further changes propagated to other components, the affected components are detected and subsequently are analyzed one by one.

- To what extent is the magnifier moved upon the model? Since the propagation domain of the changes is huge, we define a safety space or time margin to restrict the analysis over a set of components. In other words, we guarantee that by occurrence of a change in the system, the requirement violations in a time margin (over the components in a space margin) are predicted and no violation outside of this margin will threaten the system. For instance, using the experience in the transportation systems, it is estimated that an adversarial condition remains in the environment of a system for two hours. Therefore, all the components that are affected by the changes during this time margin are explored and analyzed by the Magnifier technique one by one.

We will apply our proposed approaches to air traffic control systems and railway control systems as two real world large scale cyber-physical systems.

3. CONCLUSION

The work outlined in this paper addresses developing dependable self-adaptive TTCSs. Toward this aim, at the first step we will provide a modular and hierarchical architecture based on coordinated actor models to capture the decentralized nature of the TTCSs. Then upon the provided architecture, we will provide new theories and tools for runtime verification and performance evaluation of TTCSs using the compositional approaches.

Acknowledgment

The work on this paper has been supported in part by the project "Self-Adaptive Actors: SEADA" (nr. 163205-051) of the Icelandic Research Fund.

4. REFERENCES

[1] M. Bagheri, I. Akkaya, E. Khamespanah, N. Khakpour, M. Sirjani, A. Movaghar, and E. A. Lee. Coordinated actors for reliable self-adaptive systems. 2016. Proceedings of the 13th International Conference on Formal Aspects of Component Software (FACS 2016).

[2] B. H. C. Cheng, K. I. Eder, M. Gogolla, L. Grunske, M. Litoiu, H. A. Müller, P. Pelliccione, A. Perini, N. A. Qureshi, B. Rumpe, D. Schneider, F. Trollmann, and N. M. Villegas. *Using models at runtime to address assurance for self-adaptive systems*, pages 101–136. Springer International Publishing, Cham, 2014.

[3] C. Ptolemaeus. *System Design, Modeling, and Simulation: Using Ptolemy II.* Ptolemy. org Berkeley, CA, USA, 2014.

[4] M. Sirjani, A. Movaghar, A. Shali, and F. S. de Boer. Modeling and verification of reactive systems using rebeca. *Fundam. Inf.*, 63(4):385–410, June 2004.

[5] G. Tamura, N. M. Villegas, H. A. Müller, J. P. Sousa, B. Becker, G. Karsai, S. Mankovskii, M. Pezzè, W. Schäfer, L. Tahvildari, and K. Wong. *Towards practical runtime verification and validation of self-adaptive software systems*, pages 108–132. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.