Distributed and Heterogeneous Event-based Monitoring in Smart Cyber-Physical Systems*

László Balogh[1], István Dávid[2], István Ráth[3], Dániel Varró[1], András Vörös[1]

[1]Budapest University of Technology and Economics, Hungary
laszlo.balogh@inf.mit.bme.hu, {varro, vori}@mit.bme.hu

[2]University of Antwerp, Belgium
istvan.david@uantwerpen.be

[3]IncQuery Labs Ltd
rath@incquerylabs.com

Abstract Runtime monitoring plays a key role in the assurance of smart cyber-physical systems (CPS) by checking execution traces against formal specifications. This paper presents a synthesis approach for distributed event-based monitors from high-level specification languages deployed over a heterogeneous execution platform.

1 Introduction

Smart CPS typically integrate multiple distributed nodes with a loose coupling that reaches through many layers of abstraction. This allows heterogeneity in terms of execution platforms and available resources (such as processing capacity and operative memory). Rushby [1] observes that *a priori* verification of such systems is too complex and proposes using monitoring techniques for runtime verification.

Efficiency of monitoring techniques can be enhanced by leveraging the computational autonomy of CPS nodes and *distribute* the monitoring logic over them. As nodes can solve partial monitoring tasks, a distributed approach promises increased efficiency in terms of performance and response time. Furthermore, by considering the execution platform of the node, and its available resources, the most appropriate deployment configuration can be selected, resulting in a *heterogeneous* configuration of monitoring units.

We approach monitoring with *event-based* semantics. Talcott [2] identifies event-based semantics as the appropriate means to describe various aspects of CPS, due to the support for high-level specification and reasoning. By investigating enterprise systems, Luckham [3] observes that events of (business) relevant phenomena usually constitute *complex* patterns, i.e. structured events augmented with timing and temporal information. We argue, that the same holds for CPS systems as well and therefore, we propose an approach for monitoring smart CPS based on complex event processing (CEP).

In our paper, we propose (1) a conceptual architecture along with a domain-specific modeling language for top-down monitor design methodology; and (2) a semantic integration technique for bottom-up synthesis to ensure the correct behavior over a set of different monitoring formalisms (languages).

2 Overview

Figure 1 presents our conceptual architecture and typical workflow.

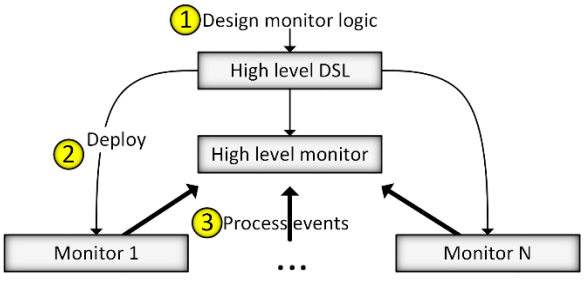

Figure 1: Conceptual architecture and workflow.

In *Step 1*, the monitoring rules are specified using a high level CEP language. In our prototype tool, we used the VIATRA-CEP Event Processing Language (VEPL). [4][5] In *Step 2*, deployable monitor logic is generated. This requires the distribution of the

* Work to be presented both as a poster and a talk.

fragments of the monitoring rules over the nodes, possibly for different execution platforms. During execution, in *Step 3*, events of the environment are processed by monitors. To enable monitoring rules evaluated over multiple nodes, higher level monitors are able to process events of lower level monitors too.

3 Semantic integration of monitor behavior

Orchestrating the interplay among the monitors of heterogeneous execution platforms raises the need for a unified execution model. We address the problem by *semantic integration*, i.e. by providing an execution model for event processing that various CEP formalisms can be translated to and evaluated in a uniform way. The feasibility of a similar approach has been shown by Hinze and Voisard [6]. Our approach is shown in Figure 2.

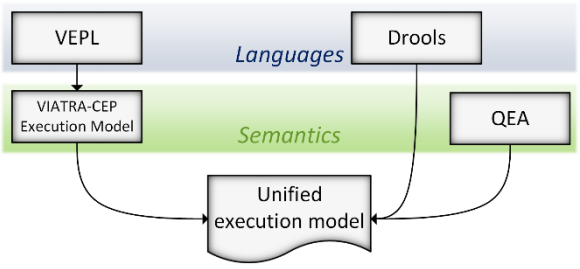

Figure 2: Semantic integration by the AL.

The *Unified execution model* is based on regular languages. This provides a wide range of possible CEP languages and semantics to be translated onto it. Most notably, we formalized translation rules of deterministic finite automata (in resource constrained embedded environments), the highly expressive Quantified Event Automata [7], and the Timed Automaton formalism [8]. The unified execution model is explicit about various aspects of execution semantics, which additionally enables verification and validation of distributed monitoring patterns.

4 Conclusions and future work

In this paper, we outlined the concepts of a distributed and heterogeneous monitoring framework for smart CPS, built on the techniques of complex event processing. The approach supports top-down specification of monitors and efficient bottom-up synthesis of different event processing formalisms. We evaluated our findings using a case study on model-based development and monitoring of safety-critical embedded systems in the railway domain. [9]

As the primary future work, we plan to investigate how the unified execution model can support the verification and validation of distributed event patterns used for monitoring.
From a development point of view, we plan to integrate our ideas into the VIATRA-CEP open-source Eclipse project [5] by integrating the unified execution model with the current execution model of the VEPL language. Additionally, we plan to support out-of-the-box integration with multiple CEP platforms.

References

1. Rushby, J.: Runtime Certification. Runtime Verification: 8th International Workshop. Selected Papers. pp 21-35. Springer, 2008
2. Talcott, C.: Cyber-Physical Systems and Events. Software-Intensive Systems and New Computing Paradigms, LNCS, 5380, 101-115. Springer, 2008
3. Luckham, D.C.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Longman Pub. Co., Boston, MA, USA, 2001
4. Dávid, I., Ráth, I., Varró, D.: Streaming Model Transformations By Complex Event Processing. Model-Driven Engineering Languages and Systems, LNCS, 8767, pp 68-83. Springer, 2014
5. VIATRA-CEP Documentation. https://wiki.eclipse.org/VIATRA/CEP
6. Hinze, A. M., Voisard, A.: EVA: An event algebra supporting complex event specification. Information Systems, 48, 1-25, 2015
7. Reger, G., Cruz, H. C., Rydeheard, D.: MarQ: Monitoring at Runtime with QEA. ETAPS 2015, LNCS, 9035, pp 596-610. Springer, 2015
8. Alur, R., Dill, D.L.: A Theory of Timed Automata. Theoretical Computer Science, vol 126, pp 183-235, 1994
9. Model-based Demonstrator for Smart and Safe Systems Project Page. http://inf.mit.bme.hu/en/research/projects/modes3