



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://oatao.univ-toulouse.fr/>
Eprints ID: 6873

To link to this article: DOI:10.4230/DagRep.2.7.30
URL: <http://dx.doi.org/10.4230/DagRep.2.7.30>

To cite this version: Feiler, Peter and Hugues, Jérôme and Sokolsky, Oleg
Architecture-Driven Semantic Analysis of Embedded Systems (Eds)
Dagstuhl Seminar 12272. (2012) Dagstuhl Report, vol. 2 (n° 7). pp. 30-55.
ISSN 2192-5283

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@inp-toulouse.fr

Report from Dagstuhl Seminar 12272

Architecture-Driven Semantic Analysis of Embedded Systems

Edited by

Peter Feiler¹, Jérôme Hugues², and Oleg Sokolsky³

1 Carnegie Mellon University – Pittsburgh, US, phf@sei.cmu.edu

2 ISAE – Toulouse, FR, jerome.hugues@isae.fr

3 University of Pennsylvania, Philadelphia, US, sokolsky@cis.upenn.edu

Abstract

Architectural modeling of complex embedded systems is gaining prominence in recent years, both in academia and in industry. An architectural model represents components in a distributed system as boxes with well-defined interfaces, connections between ports on component interfaces, and specifies component properties that can be used in analytical reasoning about the model. Models are hierarchically organized, so that each box can contain another system inside, with its own set of boxes and connections between them.

The goal of Dagstuhl Seminar 12272 “Architecture-Driven Semantic Analysis of Embedded Systems” is to bring together researchers who are interested in defining precise semantics of an architecture description language and using it for building tools that generate analytical models from architectural ones, as well as generate code and configuration scripts for the system.

This report documents the program and the outcomes of the presentations and working groups held during the seminar.

1 Executive Summary

Peter Feiler

Jérôme Hugues

Oleg Sokolsky

Architectural modeling of complex embedded systems is gaining prominence in recent years, both in academia and in industry. An architectural model represents components in a distributed system as boxes with well-defined interfaces, connections between ports on component interfaces, and specifies component properties that can be used in analytical reasoning about the model. Models are hierarchically organized, so that each box can contain another system inside, with its own set of boxes and connections between them. An architecture description language for embedded systems, for which timing and resource availability form an important part of the requirements, must describe resources of the system

platform, such as processors, memories, communication links, etc. Several architectural modeling languages for embedded systems have emerged in recent years, including AADL, SysML, EAST-ADL, and the MARTE profile for UML.

In the context of model-based engineering (MBE) architectural modeling serves several important purposes:

An architectural model allows us to break the system into manageable parts and establish clear interfaces between these parts. In this way, we can manage complexity of the system by hiding the details that are unimportant at a given level of consideration; Clear interfaces between the components allow us to avoid integration problems at the implementation phase. Connections between components, which specify how components affect each other, help propagate the effects of change in one component to the affected components. Most importantly, an architectural model can be seen as a repository of the knowledge about the system, represented as requirements, design, and implementation artifacts, held together by the architecture. Such a repository enables automatic generation of analytical models for different aspects of the system, such as timing, reliability, security, performance, etc. Since all the models are generated from the same source, ensuring consistency of assumptions and abstractions used in different analyses becomes easier. The first two uses of architectural modeling have been studied in the research literature for a number of years. However, the coordination role of architectural modeling in MBE is just currently emerging. We expect this role to gain importance in the coming years. It is clear that realizing this vision of "single-source" MBE with an architectural model at its core is impossible without having first a clear semantics of the architecture description language.

The goal of the seminar is to bring together researchers who are interested in defining precise semantics of an architecture description language and using it for building tools that generate analytical models from architectural ones, as well as generate code and configuration scripts for the system. Despite recent research activity in this area to use semantic interpretation of architectural models for analytical model generation, we observe a significant gap between current state of the art and the practical need to handle complex models. In practice, most approaches cover a limited subset of the language and target a small number of modeling patterns. A more general approach would most likely require an interpretation of the semantics of the language by the tool, instead of hard-coding of the semantics and patterns into the model generator.

2 Table of Contents

Executive Summary

<i>Peter Feiler, Jérôme Hugues, and Oleg Sokolsky</i>	30
---	----

Overview of Talks

EAST-ADL – An Architecture-centric Approach to the Design, Analysis, Verification and Validation of Complex Embedded Systems <i>De-Jiu Chen</i>	34
Model-Checking Support for AADL <i>Silvano Dal Zilio</i>	34
Model-Based/ Platform-Based/Architecture-Driven Design of Cyber-Physical Systems <i>Patricia Derler</i>	35
On the mechanization of AADL subsets <i>Mamoun Filali-Amine</i>	35
Extended Literate Programming. Introducing the \square (SquareCup) Language <i>Laurent Fournier</i>	36
Software Component Architecture Model Analysis and Executable Generation using Semantic Language Layering <i>Serban Gheorghe</i>	36
Architecture Evaluation @ Run-time: Problems, Challenges and Solutions <i>Lars Grunske</i>	37
Embedded System Architecture for Software Health Management <i>Gabor Karsai</i>	37
Experience of Using Architecture Models in Civil Aviation Domain <i>Alexey Khoroshilov</i>	38
Hierarchy is Good For Discrete Time: a Compositional Approach to Discrete Time Verification <i>Fabrice Kordon</i>	38
Formal Semantics of AADL Component Behavior to Prove Conformance to Specification <i>Brian Larson</i>	39
Software Architecture Modeling by Reuse, Composition and Customization <i>Ivano Malvolta</i>	40
Architecture-Driven analysis with MARTE/CCSL <i>Frédéric Mallet</i>	41
Approximating physics in the design of technical systems <i>Pieter J. Mosterman</i>	41
Satellite Platform Case Study with SLIM and COMPASS <i>Viet Yen Nguyen</i>	41

Correctness, Safety and Fault Tolerance in Aerospace Systems: The ESA COMPASS Project <i>Thomas Noll</i>	42
Synchronous AADL: From Single-Rate to Multirate <i>Peter Csaba Ólveczky</i>	42
Semantic anchoring of industrial architectural description languages <i>Paul Petterson</i>	43
Integration of AADL models into the TTEthernet toolchain; Towards a model-driven analysis of TTEthernet networks <i>Ramon Serna Oliver</i>	44
Architecture Modeling and Analysis for Automotive Control System Development <i>Shin'ichi Shirashi</i>	44
About architecture description languages and scheduling analysis <i>Frank Singhoff</i>	44
Co-modeling, simulation and validation of embedded software architectures using Polychrony <i>Jean-Pierre Talpin</i>	45
Compositional Analysis of Architecture models <i>Michael W. Whalen</i>	45
Working Groups	
Attaching semantics to a modeling framework	46
Expressive Power of Architectural Models	47
Analysis of architectural systems	48
Multi-point view analysis and combination of analysis result	49
Run-time Architectural Analysis	49
Notion of Time: Physical vs. Real-Time vs Discrete vs Logical time	50
Patterns for (de)composing and analysing systems	51
Summary and Open Challenges	52
Bibliography	53
Participants	55