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
Peter Feiler, Jérôme Hugues, and Oleg Sokolsky ........................................ 30

Overview of Talks
EAST-ADL – An Architecture-centric Approach to the Design, Analysis, Verification
and Validation of Complex Embedded Systems
De-Jiu Chen ................................................................. 34
Model-Checking Support for AADL
Silvano Dal Zilio .......................................................... 34
Model-Based/ Platform-Based/Architecture-Driven Design of Cyber-Physical Sys-
tems
Patricia Derler ............................................................ 35
On the mechanization of AADL subsets
Mamoun Filali-Amine ....................................................... 35
Extended Literate Programming. Introducing the $\mathbb{R}$ (SquareCup) Language
Laurent Fournier .......................................................... 36
Software Component Architecture Model Analysis and Executable Generation using
Semantic Language Layering
Serban Gheorghe .......................................................... 36
Architecture Evaluation @ Run-time: Problems, Challenges and Solutions
Lars Grunske ................................................................. 37
Embedded System Architecture for Software Health Management
Gabor Karsai ................................................................. 37
Experience of Using Architecture Models in Civil Aviation Domain
Alexey Khoroshilov ......................................................... 38
Hierarchy is Good For Discrete Time: a Compositional Approach to Discrete Time
Verification
Fabrice Kordon .............................................................. 38
Formal Semantics of AADL Component Behavior to Prove Conformance to Spec-
cification
Brian Larson ................................................................. 39
Software Architecture Modeling by Reuse, Composition and Customization
Ivano Malvolta .............................................................. 40
Architecture-Driven analysis with MARTE/CCSL
Frédéric Mallet ............................................................ 41
Approximating physics in the design of technical systems
Pieter J. Mosterman ......................................................... 41
Satellite Platform Case Study with SLIM and COMPASS
Viet Yen Nguyen .......................................................... 41
Correctness, Safety and Fault Tolerance in Aerospace Systems: The ESA COMPASS Project
Thomas Noll ......................................................... 42
Synchronous AADL: From Single-Rate to Multirate
Peter Csaba Ölveczky ............................................. 42
Semantic anchoring of industrial architectural description languages
Paul Petterson ....................................................... 43
Integration of AADL models into the TTEthernet toolchain; Towards a model-driven analysis of TTEthernet networks
Ramon Serna Oliver ........................................... 44
Architecture Modeling and Analysis for Automotive Control System Development
Shin’ichi Shirashi ................................................. 44
About architecture description languages and scheduling analysis
Frank Singhoff ...................................................... 44
Co-modeling, simulation and validation of embedded software architectures using Polychrony
Jean-Pierre Talpin ............................................... 45
Compositional Analysis of Architecture models
Michael W. Whalen .............................................. 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