Embedded Systems Architecture: Evaluation and Analysis

Bastian Florentz and Michaela Huhn

Technical University of Braunschweig, 38106 Braunschweig, Germany
{florentz, huhn}@ips.cs.tu-bs.de
http://www.ips.cs.tu-bs.de

Abstract. Short innovation cycles in software and hardware make architecture design a key issue in future development processes for embedded systems. The basis for architectural design decisions is a transparent architecture evaluation.

Our model-based approach supports a uniform representation of hierarchies of quality attributes and an integration of different architecture evaluation techniques and methods. We present a metamodel for architecture evaluation as a basis for the precise description of the quality attribute structure and the evaluation methodology. By modelling architecture evaluation, the relationships between architectural elements and quality attributes and interdependencies between quality attributes can be represented and investigated. Thereby, the architecture exploration process with its evaluations, decisions, and optimizations is made explicit, traceable, and analyzable.

1 Introduction

The evaluation of architecture has become an important part of software and system design. The early stage of design in which architecture development takes place shows its impact to the whole design process as well as the software or system at its life time. A lot of effort is invested in making architectural decisions to ensure high quality architectures. An accurate and target-oriented definition of the architecture is just one step in the right direction. The evaluation of architectures to ensure conformance to design goals is another.

We define a metamodel for the description of architectures in the embedded system domain as well as for the definition of architecture evaluation. The common basis of components and connectors allows for analysis of the architecture regarding quality attributes and associated evaluation techniques. This explicit representation and documentation of the architecture evaluation leads to increased traceability and comprehension of architectural decisions. These are essential not only for the justification of certain decisions but in particular for their traceability and reusability.

Short time-to-market requires the reuse of existing and reliable components and architectures to quickly derive and compose new depenable software and system architectures. With their long life cycles, embedded systems can profit

C. Hofmeister et al. (Eds.): QoSA 2006, LNCS 4214, pp. 145–162, 2006.

from high quality architectures and their comprehensible, traceable evaluation. Additionally, with explicit requirements and restrictions already available in the early stages of design, embedded systems are predestined for a precise description of their architecture; and architecture evaluation has high potential to lead to concrete architecture design advices. Such advices of how to understand, build, or change an architecture, and actually the whole system design, are based on analysis of relations between architecture and evaluation elements. The investigation of analysis results with the goal to derive architectural decisions and uncover optimization potential will be one main topic of our research.

In Section 2, the terminology used in this paper is described. Section 3 introduces the metamodels of architecture, evaluation, methodology, and analysis. The approach is applied to an embedded system cut-out in the case study of Section 4. The benefits of using the metamodel especially with respect to analysis results are presented in Section 5. Section 6 concludes.

2 Terminology

As our field of interest is slightly more general than pure software systems, a brief review of the most relevant terms follows.

Architecture evaluation is directed to software, as well as to system architecture in this paper. Embedded systems, consisting of hardware short of resources and software realizing the system's functionality, build the center of interest.

A *quality attribute* (called quality characteristics in ISO/IEC 9126 [1]) is a quality goal that the system under consideration at least meets the quality level given in the requirements. The *quality attribute DAG* (or *QADAG* for short) represents a hierarchical structuring of the quality attributes as a **d**irected **a**cyclic **g**raph. In our setting, a quality attribute (1) may be decomposed into (sub-)attributes or (2) consist of an evaluation technique in combination with scenarios and constraints. An example of a quality attribute is the maximal bus load that arises from a particular application in a distributed controller network. The bus load may be a subattribute of the system performance.

An *evaluation technique* associated with a quality attribute describes exactly how to evaluate the architecture regarding the quality attribute while considering of the attached scenarios and constraints.

A *scenario* represents specific examples of current and future uses of a system that are relevant for architectural design [2]. A constraint is a restriction on the design space.

An *evaluation result* is generated by applying an evaluation technique. The result of an arbitrary unit can be assigned by an *interpretation* to a quality rate.

The *quality rate* (or quality rating in [1]) is an interpreted evaluation result to a value between 0 and 100% representing the ratio of meeting the requirements represented by a quality attribute. The quality rate is also known as *utility* in the economic view of CBAM [2]. A quality rate of 100% represents the best ratio of meeting the requirements, where as a quality rate of 0% represents