Applying the Component Paradigm to AUTOSAR Basic Software

Dietmar Schreiner

Vienna University of Technology
Institute of Computer Languages, Compilers and Languages Group
Argentinierstrasse 8 / 185-1, A-1040 Vienna, Austria
schreiner@complang.tuwien.ac.at

1 Motivation

Current trends in embedded systems software for the automotive domain aim at an increase of reusability, exchangeability and maintainability, and thus at a significant reduction of time- and costs-to-market. One way to reach these goals is the adaption of Component Based Software Engineering (CBSE) for resource constrained embedded systems. The Automotive Open System Architecture (AUTOSAR), an upcoming industry standard within the automotive domain, reflects this fact by constituting CBSE as development paradigm for automotive applications: Application concerns are covered by software components, while infrastructural ones are handled within layered component middleware—the AUTOSAR Runtime Environment (RTE) and the Basic Software (BSW).

However, the AUTOSAR Basic Software itself is specified as layered architecture that is only customizable on a coarse-grained level, and thus tends to be heavy-weight and less flexible.

2 Solution

This paper contributes by applying the component paradigm to AUTOSAR BSW, to improve the capabilities of AUTOSAR compliant software systems, as conceptually depicted in Figure 1: The redesigned BSW externally provides all interfaces to the RTE prescribed by the AUTOSAR standard, whereas the BSW's internal architecture is fully component based.

To find an appropriate partitioning of the standardized software architecture, we first analyzed an implementation of the layered AUTOSAR BSW to identify basic groups of functionality that could serve as base-line for BSW components: All functions, specified by AUTOSAR, which are either coupled via function calls, or which are coupled via shared memory accesses, have been marked as candidates for the same BSW component class. In addition, functions that are semantically related to each other, but are not directly coupled, have manually been assigned to the appropriate BSW components by domain experts. In that way, we identified a set of BSW components that completely resemble the functionality and all external interfaces of the standard's layered BSW.

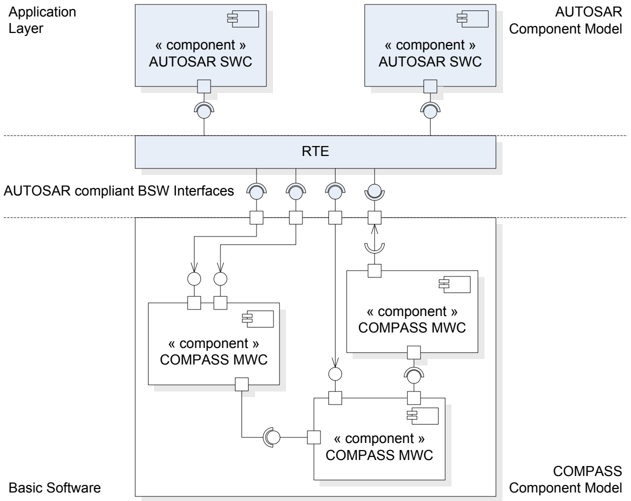
Application Layer
« component » AUTOSAR SWC
« component » AUTOSAR SWC
AUTOSAR Component Model
RTE
AUTOSAR compliant BSW Interfaces
« component » COMPASS MWC
« component » COMPASS MWC
« component » COMPASS MWC
Basic Software
COMPASS Component Model

Figure 1. Component based Basic Software

In a second step, we developed a component model—namely the COMPASS component model—for the AUTOSAR BSW. This component model specifies the BSW component classes, defines their minimal interfaces, and prescribes all means of composition and interaction for BSW components.

As a result, it is now possible to build a component based AUTOSAR BSW, that on the one hand provides a fine-grained, function-based partitioning, enabling the creation of custom-tailored BSW, and that on the other hand highly supports reuse and exchangebility of BSW components.

3 Acknowledgments

This work has been partially funded by the European Community under the FP7 IST project All-Times, contract 215068.