Developing Dependable Automotive Embedded Systems using the EAST-ADL

DeJiu Chen\textsuperscript{1}, Rolf Johansson\textsuperscript{II}, Henrik Lönn\textsuperscript{III}, Martin Törngren\textsuperscript{1}

\textsuperscript{I}: KTH, \textsuperscript{II}: Mentor Graphics, \textsuperscript{III}: Volvo Technology

\{chen, martin\}@md.kth.se, Rolf.Johansson@mentor.com, Henrik.Lonn@volvo.com

Abstract

The complexity of embedded automotive systems calls for a more rigorous approach to system development compared to current state of practice. A critical issue is the management of the engineering information that defines the embedded system. The EAST-ADL\textsuperscript{1} is an architecture description language for automotive embedded systems. The language emphasizes information management as a basis for systematic design and verification. It is currently being refined in the ATESST\textsuperscript{2} project [1].

1. Introduction

Development time, cost efficiency, quality and dependability all benefit from appropriate information management. The approach taken in the ATESST project is to enrich this information structure to support a number of analysis techniques including analysis of nominal and abnormal behavior, at different levels of abstraction. In this short abstract we emphasize the support for safety analysis.

From the viewpoint of dependable systems design, this approach provides several advantages. System modeling based on an architecture description language is a way to keep the engineering information within one structure. The EAST-ADL approach encompasses requirements, functional abstractions all the way down to implementation, and their relations. This supports consistency management, traceability, change management and impact analysis. Systems’ modeling also makes it possible to provide explicit descriptions of faults in functions, software and hardware, and the mechanisms by which they can propagate. Such descriptions in turn facilitate safety analysis techniques like FTA and FMEA.

Current development trends in automotive software feature increasing standardization of the embedded software structure. The need to integrate software from different suppliers, dependable real-time execution, and managing changes, all call for an integrated approach for software-based vehicle systems.

2. Overview of the EAST-ADL

The EAST-ADL architecture description language is intended to support the development of automotive embedded software by capturing all the related engineering information. The scope is the embedded system (hardware and software) and its environment. On top of the formal description of the elements, the language defines several abstraction levels that reflect different views and details of the architecture. They implicitly reflect different stages of an engineering process, but the actual process is company specific.

The EAST-ADL language constructs support: 1. Vehicle feature modeling including variability concepts for product families; 2. Vehicle environment modeling to define context and perform validation; 3. Structural and behavioral modeling of software and hardware entities supporting refinement to code and binaries in the context of distributed systems; 4. Requirements modeling and tracing with all modeling entities; and 5. other information, such as a definition of component timing and failure modes, necessary for design space exploration and system verification purposes. The language is structured in five abstraction layers, each with corresponding system representation, see Figure 1.

\textsuperscript{1} EAST-EAA project (www.east-eea.net) 
\textsuperscript{2} ATESST project (www.atesst.org)

Figure 1. EAST-ADL language abstractions.
Note that the environment model spans all abstraction levels, and that requirements and variability constructs apply to all modeling elements. AUTOSAR modeling constructs are integrated on Implementation level, while the other levels contain EAST ADL constructs based on UML2 and SysML [1].

As a proof-of-concept, an Eclipse based tool with supporting analysis features is being implemented, and used as a basis for case-studies.

2.1. Standardization

The area of model based development of embedded systems is evolving. This is seen in standardization efforts such as UML2.0, UML profiles such as SysML and MARTE, AADL, and AUTOSAR [2].

Safety through modeling techniques can only be achieved if the approach can be effectively used, which in turn relies on tools and mature concepts, where standardization is important. The EAST-ADL language is implemented as a UML profile and incorporates relevant aspects from SysML and MARTE. MARTE is an ongoing effort to define a UML profile for Modeling and Analysis of Real-Time and Embedded systems, initiated to overcome the UML limitations for such systems. The EAST-ADL is also being harmonized with the new automotive domain standardization AUTOSAR. AUTOSAR focuses mainly on the implementation level of abstraction, whereas the EAST-ADL supports the overall systems modeling. Inspiration in the development of the EAST-ADL is also gathered from the SAE Architecture and Analysis Description Language (AADL) [1,2] and safety standards such as the ISO 26262 Functional Safety (committed draft planned for beginning 2008).

3. Managing complexity and safety analysis

In order to better support the development of complex automotive systems, the EAST-ADL does not only include means to create analysis and design models of a system but also language means to specify the required properties, to trace requirements between levels of refinement and decomposition, to refine the specification of requirements by behavioral models, and to manage information related to V&V activities.

The support for Safety requirements and analysis is specifically addressed. Safety requirements in the refined EAST-ADL have attributes and related entities to define the requirement and the hazard it mitigates. Hazards or hazardous events are part of the environment model and are characterized by attributes for severity, exposure and controllability. The hazardous event may be further detailed by e.g. use cases, sequence or activity diagrams. Safety requirement attributes includes safety integrity level, operation state, fault time span, emergency operation times, safety state, and functional redundancy to record dependability characteristics. A requirement can be traced from the abstract vehicle model all the way to its derived requirements allocated to the final hardware and software components. Depending on abstraction level, some or all of these attributes are applicable.

EAST-ADL offers detailed means to explicitly model central artifacts of verification and validation activities and to relate these artifacts to requirements. This allows for explicitly and continuously planning, tracking, updating and managing important V&V-activities and their impact on the system in parallel to the development of the system.

Safety and reliability analysis is supported through the definition of the failure semantics of architectural entities. Such failure specifications, together with environmental conditions, are used to identify hazards and define safety requirements or correct specification errors. A specification can be based on logical or temporal expressions, depending on the analysis techniques and tools of interest. The relationships of local error behaviors are captured and controlled through an explicit propagation construct. Through this artifact, the EAST-ADL supports advanced properties of errors, such as the logical and temporal relationships of source and target errors, the conditions of propagations, and the synchronizations of multiple propagation paths. At present, a proof-of-concept tool integration with the HiP-HOPS method (Hierarchically Performed Hazard Origin and Propagation Studies) [3] is being developed. This method integrates several classical analysis techniques, e.g., FFA, FTA, and FMEA. The analysis include fault trees from functional failures to their causes in software and hardware, minimal cut-sets, FMEA tables for component errors and their effects on the behaviors and reliability of entire system.

4. References


[2] ATESST project Deliverable D6.1.1 - Elicitation of overall needs and requirements on the ADL – Part II Edited by Martin Törngren (available from www.atesst.org)