

# Chapter 23

## EAST-ADL: An Architecture Description Language for Automotive Software-Intensive Systems

**Hans Blom**

*Volvo Technology, Sweden*

**Henrik Lönn**

*Volvo Technology, Sweden*

**Frank Hagl**

*Continental Automotive GmbH, Germany*

**Yiannis Papadopoulos**

*University of Hull, UK*

**Mark-Oliver Reiser**

*Technische Universität Berlin, Germany*

**Carl-Johan Sjöstedt**

*KTH Royal Institute of Technology, Sweden*

**De-Jiu Chen**

*KTH Royal Institute of Technology, Sweden*

**Fulvio Tagliabò**

*Centro Ricerche Fiat, Italy*

**Sandra Torchiaro**

*Centro Ricerche Fiat, Italy*

**Sara Tucci**

*CEA LIST DILS, France*

**Ramin Tavakoli Kolagari**

*Nuremberg Institute of Technology, Germany*

### ABSTRACT

*EAST-ADL is an Architecture Description Language (ADL) initially defined in several European-funded research projects and subsequently refined and aligned with the more recent AUTOSAR automotive standard. It provides a comprehensive approach for defining automotive electronic systems through an information model that captures engineering information in a standardized form. Aspects covered include vehicle features, requirements, analysis functions, software and hardware components, and communication. The representation of the system's implementation is not defined in EAST-ADL itself but by AUTOSAR. However, traceability is supported from EAST-ADL's lower abstraction levels to the implementation level elements in AUTOSAR. In this chapter, the authors describe EAST-ADL in detail, show how it relates to AUTOSAR as well as other significant automotive standards, and present current research work on using EAST-ADL in the context of fully-electric vehicles, the functional safety standard ISO 26262, and for multi-objective optimization.*

DOI: 10.4018/978-1-4666-3922-5.ch023

## INTRODUCTION

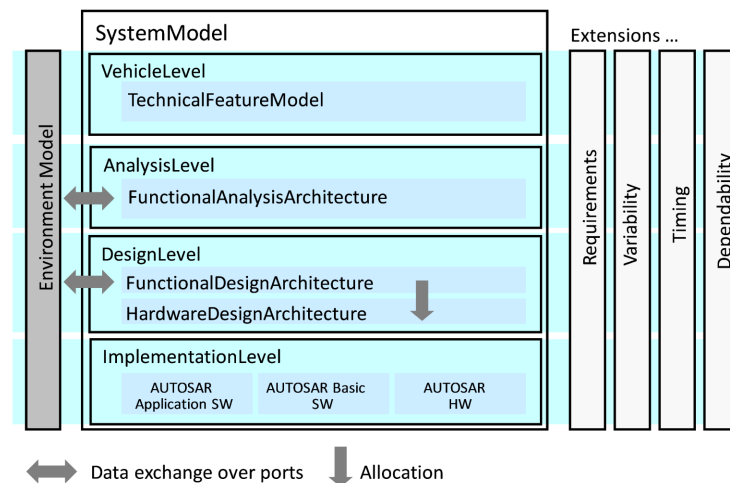
EAST-ADL is an Architecture Description Language (ADL) initially defined in the European ITEA EAST-EEA project and subsequently refined and aligned with the more recent AUTOSAR automotive standard (AUTOSAR Development Partnership, 2012) in the European FP6 and FP7 ATESS2 projects (ATESS2 Consortium, 2010). Currently, it is maintained by the EAST-ADL Association (EAST-ADL Association, 2012). It is an approach for describing automotive electronic systems through an information model that captures engineering information in a standardized form. The language provides a wide range of modeling entities, including vehicle features, functions, requirements, variability, software components, hardware components and communication.

EAST-ADL clearly defines several abstraction levels (see Figure 1) and at each of these levels, the software- and electronics-based functionality of the vehicle is modelled with a different level of detail. The proposed abstraction levels and the contained modeling elements provide a separation of concerns and an implicit style for using the la-

nugage elements. The embedded system is defined completely on each abstraction level, and identical parts of the model are linked across abstraction levels with various traceability relations. This makes it possible to trace an entity from feature down to components in hardware and software.

The features in the “TechnicalFeatureModel” at the vehicle level represent the content and properties of the vehicle from top-level perspective without exposing the realization. It is possible to manage the content of each vehicle and entire product lines in a systematic manner. A complete representation of the electronic functionality in an abstract form is modeled in the Functional Analysis Architecture (FAA). One or more entities (analysis functions) of the FAA can be combined and reused to realize features. The FAA captures the principal interfaces and behavior of the vehicle’s subsystems. It allows validation and verification of the integrated system or its subsystems on a high level of abstraction. Critical issues for understanding or analysis can thus be considered, without the risk of them being obscured by implementation details.

Figure 1. The EAST-ADL’s breakdown in abstraction levels (vertically) and in core, environment and extensions (horizontally)



13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/east-adl-architecture-description-language/76970](http://www.igi-global.com/chapter/east-adl-architecture-description-language/76970)

## Related Content

---

### Semi-Quantitative Risk Assessment of Technical Systems on European Railways

Jens Braband (2012). *Railway Safety, Reliability, and Security: Technologies and Systems Engineering* (pp. 54-64).

[www.irma-international.org/chapter/semi-quantitative-risk-assessment-technical/66667](http://www.irma-international.org/chapter/semi-quantitative-risk-assessment-technical/66667)

### FusionFlow: A Software Development Approach in the Era of Design Thinking and AI

A. D. N. Sarma (2025). *Innovative Design Thinking Approaches in Software Engineering* (pp. 53-84).

[www.irma-international.org/chapter/fusionflow/382580](http://www.irma-international.org/chapter/fusionflow/382580)

### A Comparative Study of Bayesian and Fuzzy Inference Approach to Assess Quality of the Software Using Activity-Based Quality Model

Kawal Jeet, Nitin Bhatiaand Renu Dhir (2013). *Designing, Engineering, and Analyzing Reliable and Efficient Software* (pp. 96-111).

[www.irma-international.org/chapter/comparative-study-bayesian-fuzzy-inference/74876](http://www.irma-international.org/chapter/comparative-study-bayesian-fuzzy-inference/74876)

### Performance-Aware Approach for Software Risk Management Using Random Forest Algorithm

Alankrita Aggarwal, Kanwalvir Singh Dhindsaand P. K. Suri (2021). *International Journal of Software Innovation* (pp. 12-19).

[www.irma-international.org/article/performance-aware-approach-for-software-risk-management-using-random-forest-algorithm/266279](http://www.irma-international.org/article/performance-aware-approach-for-software-risk-management-using-random-forest-algorithm/266279)

### Self-Modifying Code: A Provable Technique for Enhancing Program Obfuscation

Chandan Kumar Beheraand D. Lalitha Bhaskari (2017). *International Journal of Secure Software Engineering* (pp. 24-41).

[www.irma-international.org/article/self-modifying-code/201214](http://www.irma-international.org/article/self-modifying-code/201214)