Evolution Mechanisms of Automotive Architecture Description Languages

Yanja Dajsuren, M. G. J. van den Brand, and Alexander Serebrenik

Eindhoven University of Technology
5600 MB, Eindhoven, The Netherlands
{Y.Dajsuren | M.G.J.v.d.Brand | A.Serebrenik}@tue.nl

As software becomes more and more important for automotive systems, introducing and adopting existing solutions from software engineering discipline are becoming common practice. One of the approaches being recognized as an important contribution to the automotive industry is Architecture Description Language (ADL) [1]. ADLs are used to describe a system at different phases of its development and to facilitate communication between different parties. Although many general-purpose ADLs exist, ADLs for safety critical systems and specifically for automotive systems have been developed to address the need of expressing quality attributes such as dependability, safety, timing aspects and variability issues. Automotive ADLs like EAST-ADL [2], SAE AADL [3], AML [4], and TADL [5] are introduced to improve the software development process of automotive systems. Since evolution is considered as one of the costliest software development activities [6], ADLs need to provide explicit mechanisms to support it. However, there has been so far no attempt to evaluate the evolution mechanisms in the automotive ADLs.

This paper aims to analyze mechanisms of supporting design-time evolution by two widely researched automotive ADLs, namely EAST-ADL [7] and AADL [8][1]. We used the evolution features defined in the ADL classification framework of Medvidović and Taylor [9]. The framework defines architecture modeling features based on *components*, *connectors* and *architectural configurations*. A component is defined as a unit of computation or a data store with an explicit interface, which is an interaction point with other components and external world. The component evolution is informally defined as the change of a component's properties such as interface, behavior or implementation. Subtyping of component types and refinement of component features are considered as common techniques to support systematic evolution of components. A connector is used to model the interactions between different components and to define the rules that govern the interactions. The connectors may not result in compilation units, but they can be implemented as messages between components for example. The connector evolution is a modification of its properties such as interface, semantics, or connector constraints. Evolution of components and configurations are closely related to connectors thus existing connectors may be modified or refined further by evolution mechanisms like incremental information filtering, subtyping, and refinement. An architectural configuration describes architectural structure by connecting appropriate components. Descriptions of configurations enable analyses of architectures for adherence to design heuristics. The configuration evolution is supported by incremental addition, reconnection, replacement, and removal of components and connectors.

After analyzing EAST-ADL and AADL using the Medvidović and Taylor framework, we conclude that evolution mechanisms for connectors are not explicitly addressed in the definition of automotive ADLs. In EAST-ADL, higher level design models are refined by the lower level components containing more implementation-oriented aspects. In AADL, a component evolution is supported by extensions (by enabling component type to have multiple implementations and by refinement of existing elements of a component). Connectors are not modeled as first-class objects in EAST-ADL and AADL, therefore no explicit evolution mechanisms are provided. However, in AADL ports are declared as features in component types and can be refined into concrete features from abstract definitions. In terms of enabling evolution mechanisms for the architecture configuration, EAST-ADL and AADL provide addition and modification of new components and connectors.

[1] We refer to EAST-ADL 2.0 and AADL 2.0.

References

1. N. Navet and F. Simonot-Lion, *Automotive Embedded Systems Handbook*. Industrial information technology series, CRC Press, 2009.
2. P. Cuenot, P. Frey, R. Johansson, H. Lönn, and et al., "The EAST-ADL Architecture Description Language for Automotive Embedded Software," in *Model-Based Engineering of Embedded Real-Time Systems*, vol. 6100 of *Lecture Notes in Computer Science*, pp. 297–307, Springer Berlin, 2010.
3. SAE International, "Architecture Analysis and Design Language." http://www.aadl.info/.
4. U. Freund, M. von der Beeck, P. Braun, and M. Rappl, "Architecture Centric Modeling of Automotive Control Software," 2003.
5. K. Klobedanz, C. Kuznik, A. Thuy, and W. Mueller, "Timing Modeling and Analysis for AUTOSAR-based Software Development - A Case Study," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 642 –645, 2010.
6. C. Ghezzi, M. Jazayeri, and D. Mandrioli, *Fundamentals of Software Engineering*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1991.
7. The ATESST Consortium, "EAST-ADL 2.0 Specification." http://www.atesst.org/home/liblocal/docs/EAST-ADL-2.0-Specification_2008-02-29.pdf.
8. SAE International, "AADL 2.0 Specification." http://standards.sae.org/as5506a/.
9. N. Medvidovic and R. Taylor, "A Classification and Comparison Framework for Software Architecture Description Languages," *IEEE Transactions on Software Engineering*, vol. 26, no. 1, pp. 70 –93, 2000.