Towards Compact Surface Languages for Specific Modelling Aspects in EAST-ADL

Imad Berrouyne[1], Alessio Bucaioni[1], Federico Ciccozzi[1], and Henrik Lönn[2]

[1] *Mälardalen University, Västerås, Sweden*
[2] *Volvo Group Truck Technology, Gothenburg, Sweden*

Abstract

The EAST-ADL is an Architecture Description Language for automotive embedded systems. It offers a comprehensive modelling solution for an integrated system, addressing diverse aspects including but not limited to variability, timing, and safety. Nevertheless, the challenge lies in the intricate nature of specifying these aspects. Both because the expressiveness adds complexity to syntax and semantics and because they are intertwined with the foundational concepts within the EAST-ADL. In this paper, we propose an approach to inject these aspects using a constraints-based surface language. Such a language offers a compact and optional description layer for annotations of the EAST-ADL.

Keywords: EAST-ADL, Embedded Systems, Automotive Systems, Feature Modelling, Variability

1 Introduction

The EAST-ADL is an established Architecture Description Language that addresses the complexities of embedded components in automotive systems [6]. With built-in feature modelling, it offers a comprehensive solution also integrating concerns related to variability [1]. However, the challenge arises in effectively specifying variability constraints within the EAST-ADL, particularly given its intricate relationship with foundational concepts. Indeed, specifying variability using the core constructs is complex and time-consuming which means that efficiency and correctness is at risk. We propose an approach to inject variability concerns into EAST-ADL models using a constraints-based surface language that operates as a loose, separate, and optional layer, offering an additional level of abstraction for constraining EAST-ADL models. Such a language provides a more flexible and modular means of incorporating variability into system architectures, providing a more compact and tractable syntax for defining basic variability. The interpretation of this surface language is conducted in conjunction with an underlying EAST-ADL core model. This integration ensures that the variability constraints are seamlessly woven into the broader modelling framework, preserving the integrity and coherence of the overall system specification. This contribution represents a first step towards offering various compact surface languages for specific modelling aspects, such as, e.g., timing and safety.

The remainder of this paper is structured as follows. Section 2 presents a running example describing a Windscreen Wiper using a EAST-ADL model along with constraints. Section 3 gives an overview of the related work. Section 4 provides our proof-of-concept solution based on a surface language and an augmented EAST-ADL model. Finally, Section 5 presents the conclusion and future work.

2 Running example

Figure 1 depicts the windscreen wiper functionality that we use as a running example. It includes three windscreen wiper variants *WiperCtrlStd*, *WiperCtrlAutoReturn* and *WiperCtrlAutomatic* based on a *DesignFunctionPrototype* specified within a core EAST-ADL model along with the constraints they must satisfy during variability resolution. As a proof-of-concept, we target three frequently used types of variability constraints: ModelYear, Brand, and Class. For instance, the *WiperCtrlAutomatic* should only be included when ModelYear is *2016*, Brand is **X** and Class is *Heavy Duty*.

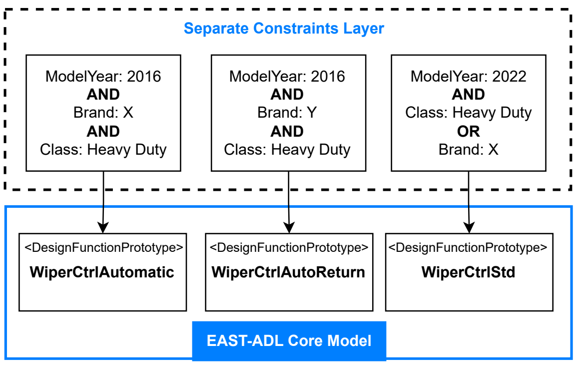

Figure 1: Windscreen wiper running example

Product variability is typically large and follows complex rules. Our goal is to provide a means to specify it in a clearly expressed and separate manner. Throughout this paper, we will show how to specify these three variants and their constraints using a compact surface language, and how they can be injected into an EAST-ADL core model.

3 Related work

Existing modelling languages for automotive systems, such as AUTOSAR and SysML, generally support the description of multiple aspects, such as variability, in addition to the core concepts. AUTOSAR primarily addresses the standardization of software interfaces and communication between different software components in automotive systems [2]. It has means to express variability, but it is for the software-related elements, and it lacks a dedicated and compact syntax for capturing variability. SysML is an extension of the UML that is tailored for system engineering [9]. Variability is expressed through constructs like Block Definition Diagrams and Parametric Diagrams. However, challenges include potential diagrammatic complexity in large-scale systems and a lack of standardized approaches, leading to potential inconsistencies in model interpretation. Some studies addressed the ability to explore various aspects of modelling languages. Zhang et al. [8] introduces a DSL, named EATXT, for the specification of EAST-ADL models using textual format. The approach permits to describe an entire EAST-ADL model but lacks a means to examine the model's aspects using new abstractions. Grönninger et al. [2] suggest using model views to address the inherent complexity of representing SysML variability. These views are meant to focus on specific aspects of the entire model. Other works use model-driven techniques to enhance the management of different aspects of software product lines such as test script generation [3] and configuration files [7].

4 The proposed approach

The motivation for this paper comes from the observation that several concerns that can be addressed in the EAST-ADL are disregarded because their modelling requires a large manual effort. Further, the expressiveness and modularity of the language sometimes hide the meaning of the model. The gains of our approach are separation of concerns by extending the EAST-ADL without altering the standard, and the description of specific modelling aspects in a textual format. We achieve the latter by means of a complementary surface language that offers a limited but compact notation for the EAST-ADL extensions. The complete model still persists in the full-featured standard EAST-ADL model and is generated out of the combined surface language and core model. The enabler for this approach is the inherent separation between core constructs and extensions. The aim is to obtain an EAST-ADL model encapsulating variability through a surface language. We present the architecture of our proposed approach along with its two-step process in the following.

4.1 A two-step process

The proposed approach revolves around a two-step process consisting in 1) specifying a target aspect in a compact surface language and 2) model generation using a model transformation procedure. In the present work, we focus on the aspects of variability as a proof-of-concept. Figure 2 depicts an overview of the process. The green boxes are concepts related to the proposed surface languages, while the blue boxes represent elements strictly related to the EAST-ADL standard. The model transformation procedure results in the integration of surface language aspects into the core EAST-ADL model and leads to the generation of an augmented EAST-ADL model.

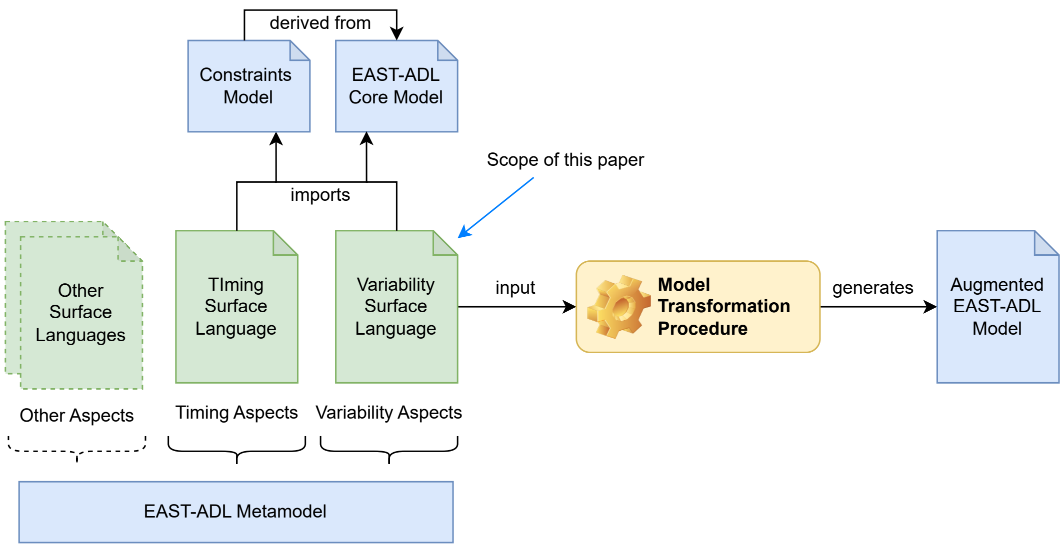

Figure 2: Process overview

In that respect, we aim to effectively disentangle variability concerns from the EAST-ADL, thereby promoting a more lucid approach to engineering such a concern. The initial phase of the proposed process involves the precise delineation of constraints using the surface language while the subsequent stage entails interpreting these constraints bound to the underlying core EAST-ADL model.

4.2 Step 1: Description of modelling aspects

The description of modelling aspects in this case study relies on a constraints-based surface language which works in conjunction with a core EAST-ADL model and a model encompassing the variability features. We assume that these features are derived from the core model *a priori* by engineers. To express the constraints for the present running example, our surface language depends on the metamodel defined in Figure 3. This metamodel consists of the root class **SurfaceModel** which is composed of **Include** instructions. These instructions permits to include an **EAElement** (e.g., WiperCtrlAutomatic– *DesignFunctionPrototype* is also an *EAElement*) in an augmented EAST-ADL model under certain constraints. An **Include** references a EAElement from a core EAST-ADL model and contains a **ConstraintsGroup**. A **ConstraintsGroup** consists of a one or many constraints bound using the logical operators **AND**, **OR** or **XOR**. A **Constraint** references the **Feature** that the augmented model must be constrain with.

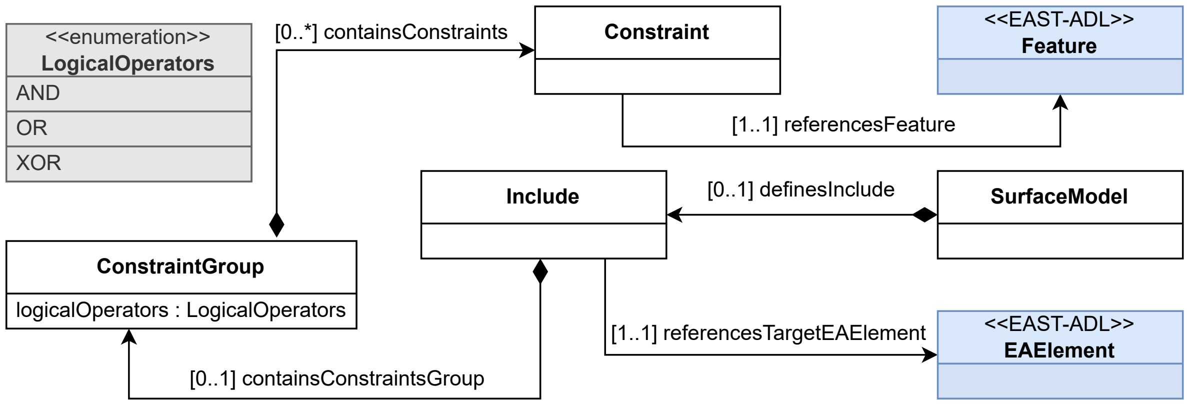

Figure 3: Metamodel of the surface language

Practically, we can describe these modelling aspects using a Domain-Specific Language (DSL) offering a straightforward and decoupled manner to express variability using constraints. In the present example, it targets the specific cases where one wants to ensure that the final model meets certain criteria in terms of variability. However, such a DSL is meant to be compact, which limits its ability to address variability from a wider viewpoint. Thus, it is not intended to replace the built-in feature modeling or product line concepts of EAST-ADL. Listing 1 presents a potential syntax of the DSL that we applied to the running example.

```
INCLUDE WindscreenWipersPackage/WiperCtrlAutomatic IF VariabilityConstraints/ModelYear=”2016”
                    AND VariabilityConstraints/Brand=”X” AND VariabilityConstraints/Class=”Heavy Duty”
INCLUDE WindscreenWipersPackage/WiperCtrlAutoReturn IF VariabilityConstraints/ModelYear=”2016”
                    AND VariabilityConstraints/Brand=”Y” AND VariabilityConstraints/Class=”Heavy Duty”
INCLUDE WindscreenWipersPackage/WiperCtrlStd IF VariabilityConstraints/ModelYear=”2022”
                    AND VariabilityConstraints/Class=”Heavy Duty” OR VariabilityConstraints/Brand=”X”
```

Listing 1: Specification of the running example using the surface language

In this example, we create three **INCLUDE** instructions. The first applies to *WiperCtrlAutomatic*, the second to *WiperCtrlAutoReturn*, and the third *WiperCtrlStd*. For instance, the first **INCLUDE** targets the WiperCtrlAutomatic contained in the WindscreenWipersPackage from the core model. It has three constraints which are all retrieved from the VariabilityConstraints model: ModelYear=”2016”, Brand=”X” and Class=”Heavy Duty”.

4.3 Step 2: Augmented model generation

As this is a preliminary work, this second step is still under development. Nevertheless, we consider that from a conceptual perspective our goal is to populate a detailed variability model according to the content of the surface language. This implementation could be achieved using three methods. The first method could use a model-to-model transformation, consisting of weaving the surface language model with the core EAST-ADL model. The second method could rely on a model-to-text transformation by generating an EAST-ADL model incorporating the target aspects as an EAXML file based on a template. The third method could leverage the idea behind Blended Modelling [5], where we would synchronize the core EAST-ADL model in real time according to the specified aspects. The outcome yields an augmented EAST-ADL model originating from distinct abstractions. This augmentation is achieved through the application of a surface language, allowing for the independent articulation of aspects without necessitating an in-depth exploration of the inherent intricacies of the EAST-ADL. Importantly, this process ensures the preservation of the original model's integrity and adherence to the standard EAST-ADL metamodel.

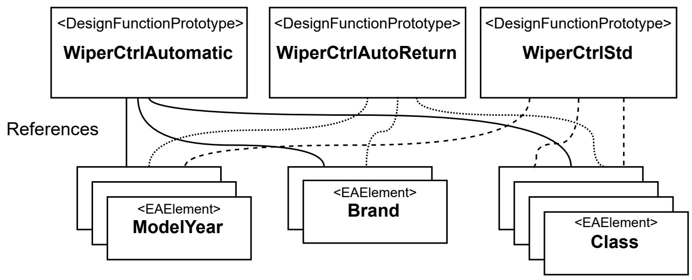

Figure 4: Augmented EAST-ADL model overview; all lines (solid, dashed, dotted) refer to references.

We aim to guarantee the correctness of the transformation using a bottom up approach via two stages. The first stage consists of transforming a set of specific cases and validate them from an EAST-ADL expert. The second stage consists of generalizing such transformations to cover the critical EAST-ADL metamodel elements such as e.g., by guaranteeing the correctness of transformations involving a *DesignFunctionPrototype* based on approaches such as CoqTL[4]. Moreover, we could also consider making the mapping rules bi-directional and thus exploit them for validation of the transformed model. This process will be iterative and aims at targeting most elements of the EAST-ADL metamodel. In the case of our running example, these methods would lead to the model shown in Figure 4. We refer to the relationship between the *DesignFunctionPrototype* in the core EAST-ADL model and the injected variability aspects as a *Reference*. We use such broad terminology as we must investigate thoroughly the kind of reference that would apply and under what conditions. Nevertheless, our preliminary investigation for variability aspects and the feedback of some expert engineers suggests that such a *Reference*, conforming to the semantics of the surface language, would be generally feasible given the inherent versatility of the EAST-ADL.

5 Conclusion

The presented approach introduces a variability surface language as a proof of concept for exploring various aspects of the EAST-ADL using new abstractions. These abstractions, defined separately, give engineers a tailored manner to express their concerns. Surface languages facilitate the identification of additional aspects within the EAST-ADL, thus separating concerns, improving its expressiveness, and broadening its adaptability to meet the diverse requirements of the automotive industry. In future work, we plan to target other aspects such as safety and timing, explore more descriptions constructs and implement a comprehensive transformation procedure for generating the augmented EAST-ADL model.

Acknowledgment

The work in this paper has been supported by the Swedish Knowledge Foundation (KKS) through the Modev project, by the Excellence in Production Research (XPRES) Framework and by the Swedish Governmental Agency for Innovation Systems (VINNOVA) through the iSecure project.

References

[1] Muhammad Waseem Anwar, Alessio Bucaioni, and Federico Ciccozzi. Timing-aware variability resolution in east-adl product line architecture. In *30th Asia-Pacific Software Engineering Conference*, December 2023.

[2] AUTOSAR Consortium. *AUTomotive Open System ARchitecture (AUTOSAR) Specification*. AUTOSAR Consortium, 2022. Release 19-11.

[3] Alessio Bucaioni, Fabio Di Silvestro, Inderjeet Singh, Mehrdad Saadatmand, Henry Muccini, and Thorvaldur Jochumsson. Model-based automation of test script generation across product variants: a railway perspective. In *2021 IEEE/ACM International Conference on Automation of Software Test (AST)*, pages 20–29. IEEE, 2021.

[4] Zheng Cheng, Massimo Tisi, and Rémi Douence. Coqtl: a coq dsl for rule-based model transformation. *Software and Systems Modeling*, 19(2):425–439, 2020.

[5] Federico Ciccozzi, Matthias Tichy, Hans Vangheluwe, and Danny Weyns. Blended modelling-what, why and how. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pages 425–430. IEEE, 2019.

[6] EAST-ADL Association. *EAST-ADL Specification*. EAST-ADL Association, 2023. Available at https://www.east-adl.info/Specification.html, Version 2.2.

[7] Enxhi Ferko, Alessio Bucaioni, Jan Carlson, and Zulqarnain Haider. Automatic generation of configuration files: an experience report from the railway domain. *J. Object Technol.*, 20(3):4–1, 2021.

[8] Jörg Holtmann, Jan-Philipp Steghöfer, and Weixing Zhang. Exploiting meta-model structures in the generation of xtext editors. In *MODELSWARD*, pages 218–225, 2023.

[9] Object Management Group. *Systems Modeling Language (SysML) Specification*. Object Management Group, 2023. Version 2.6.1.