S3O Tool: Fostering Multidisciplinary Collaborative Modeling for Dependable Embedded Systems

Xabier de Carlos, Josune de Sosa, Xabier Mendialdua , Salvador Trujillo

IKERLAN Research Centre. Mondragon, Spain
{xdecarlos, jdesosa, xmendialdua, strujillo}@ikerlan.es

Abstract. The engineering of embedded systems is evolving quickly nowadays to cope with stringent pressures. Multidisciplinary workteams, distributed across several locations worldwide, need to communicate among distinct disciplines. One solution is to specify the system at a higher abstraction level. S3O is a tool suite that addresses some of these issues. S3O provides collaborative and distributed modeling capabilities based on Model Driven Engineering specifically designed for dependable embedded systems. Specified models drive the entire engineering process and embedded software is typically generated automatically. The main benefit is to ease embedded engineering process by including collaborative support and decrease complexity when designing by using abstractions. By using S3O, stakeholders without programming or platform skills can specify embedded systems. Since the code is generated automatically, a significant number of programming errors can be avoided. Additionally, stakeholders are guided through the engineering process and so the multidisciplinary collaborative work for dependable embedded systems is realized.

1 Introduction

Embedded systems are evolving at quick pace nowadays. The engineering of these systems is also evolving to cope with newer pressures (e.g. time-to-market, collaborative and distributed work teams, etc). The embedded software originates a significant part of those challenges nowadays. The key to speed up the engineering process is the intensive use of collaborative modeling tools customized to the application domain.

Among the diversity of existing tool support for embedded, this work concentrates on those for design. Typically, a number of heterogeneous tools are used while designing a dependable embedded system. SysML/UML-based modeling tools, DSL-based modeling tools, and others are combined together to yield a multidisciplinary design. They are closely related, and so far those relationships are typically not explicitly defined.

S3O is a tool we developed to interconnect the assorted tools and individuals involved in the design phase. S3O is a tool suite encompassing several deisng activities (e.g. modeling, collaboration, etc). It is implemented as an Eclipse-based application leveraging on available modeling tooling. The main novelty comes from the primitives for the DSL focused on dependability issues and the collaborative lifecycle management based on Jazz. We begin by introducing a case study and then introduce the tool.

2 Case Study

SD4Rail is in charge of the emergency brake of a railway system. Its mission is to check whether the brake needs to be activated. Most important, the emergency brake must be activated when something goes wrong.

The fundamental functionality of the system is based on a set of inputs from the railtrack, assorted sensors, balises, and so on. Starting from them, it performs some calculations and decides whether the emergency brake needs to be activated. An output signal is sent accordingly.

A proprietary embedded system has been designed to meet stricter safety regulations. In our case, SIL4 level is pursued. To attain this, a number of design techniques from S&D are used, namely, redundancies, votings, diagnostics, secure and safe communications, and so on. A very strict and certified engineering process is followed accordingly.

Votings among redundant computation units is a central design decision of SD4Rail. Three voters are used in our application, where each is actually composed of three (single) voters, each located in a different computation unit. The ultimate goal is to meet dependability regulations. We encounter that these voters were subject to abstract. These and other abstractions were the basis for the S3O modeling tool that is introduced next.

3 S3O

S3O is based on Eclipse plug-ins, namely, Graphical Modeling Framework [1], Epsilon[2], Eclipse Modeling Framework[3] and MOFScript[4]. S3O is composed by different tools for the design of embedded systems to stakeholders of different disciplines. Different views are showed by the tools adapting to the stakeholder skills.

Create and develop custom dependability-oriented S3O models and automatically generate code from S3O models.

S3O Functional is to represent those functional parts of the system that are specific of our case study[5]. The tool can create S3O models. When the system is designed into a S3O model, it can generate executable code automatically (details next). Figure 1a shows an overview of the use of S3O.

S3O Modeling consists of different editors that ease system specification by stakeholders [2]. These tools provide different editors that can create and customize S3O models:

- *Graphical editor*: graphical edition of S3O models. In this editing mode, stakeholders can define an entire system based on provided abstractions. This editor also includes assorted wizards and is geared towards stakeholders with scarce domain skills. Figure 1b shows a graphical representation of our case study with S3O.

[1] http://www.eclipse.org/gmf/

[2] http://www.eclipse.org/gmt/epsilon/

[3] http://www.eclipse.org/emf/

[4] http://www.eclipse.org/gmt/mofscript/

[5] Details are omitted in this paper, but a video demo is available (see below).

- *Textual editor*: textual S3O model edition. It shows the same information as the graphical, but in a textual way. It is intended to stakeholders that are familiar with programming IDEs and are biased towards textual representations.
- *Tree-based editor*: tree format S3O model edition. When the number of elements is large, tree representations produce more concise views.

These editors are synchronized to produce consistent views. They are complemented with validation tools that enables validation while modeling. SD4Rail has been modeled using this tool.

S3O CodeGen. This tool automates code generation for our case study from S3O models. To attain this, code generation capability based on model to text transformations was included. The model to text transformations contains rules and templates for the modeling elements, so that code can be generated. MOFScript was used. Model artifacts represent embedded system designs using abstractions. These abstractions will be transformed into executable code. S3O models conform to the S3O metamodel where the modeling language is defined.

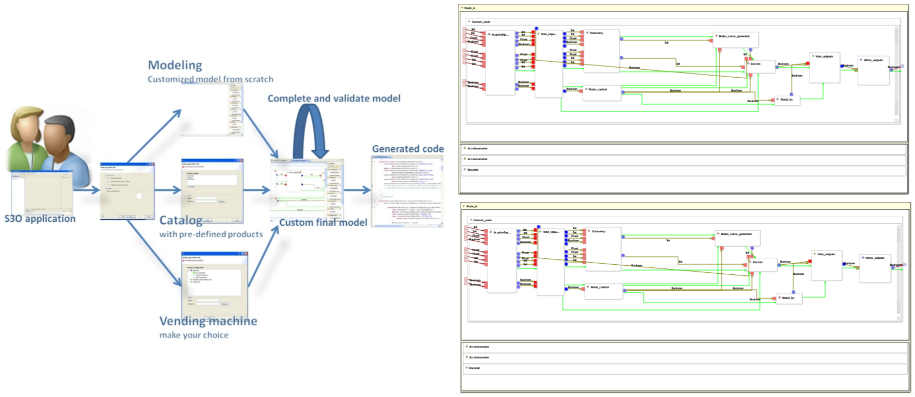

Fig. 1. (a) Wizard of S3O and (b) Designs of Pre-defined products

S3O Product-line. The goal is to produce model variants based on existing pre-defined products or using model customization wizards. When a new system is to be designed, the S3O Product Line provides the capability to create variants based on a catalog or on a feature selection.

The product catalog offers a collection of pre-designed system products. The stakeholder will have to select one product from the catalog. Based on this choice, a variant model will be composed with pre-defined values. It can be later edited to incorporate newer modeling elements.

Based on a conventional product-line feature selection, the tool composes a number of modeling elements to yield a resulting model. CVL tool is used to attain variability for models[1]. Again, this model can be edited later on.

Figure 1a shows the alternative paths to design models by using S3O, namely, new empty model, model based on pre-defined products (catalog) or create model based customization plug-in (feature selection). In Figure 1b shows created new product based on pre-defined tramway product.

S3O Collaborative and Multidisciplinary. S3O eases communication between embedded system development stakeholders. This tool allows to create and view tasks to communicate with other lifecycle phases. Embedded system development is multidisciplinary and collaborative and S3O streamlines the process allowing efficient communication among stakeholders. S3O Collaborative is based on OSLC[6] and uses Rational Team Concert collaborative tool for task managing. S3O has different perspectives. Each perspective is oriented for different discipline stakeholders, showing specific views and tools related to each discipline.

4 Demo & Storyboard

Storyboard. We have prepared a ten minutes storyboard to showcase in detail the capabilities of S3O. We have prepared a short video demo to show the case study in detail with the major capabilities of the tool. The audience will have the opportunity to design systems using the S3O capabilities. We will compare the tool with other commercial tools and open-source tooling.

5 Conclusions

This paper describes the capabilities of the S3O Tool Suite. The major novel feature of this tool is the application of model-driven and product-line engineering for dependable embedded systems. A case study was presented where this tool is being used in practice. The fundamental benefit is that reuse and collaboration are introduced to a relevant field (dependable embedded).

We are currently working at enhancing variability support at system and pattern level, and to improve the model editors, system specification and code generation. Our main goal is to guide system stakeholders throughout the modeling engineering process while automating repetitive tasks.

Acknowledgments. This work is co-supported by the Spanish Ministry of Education under contract TIN2008-06507-C02-02/TIN (VGROVE) and by the European project FP7 TERESA.

References

1. Sintef. Cvl tool. http://www.omgwiki.org/variability.
2. Josune De Sosa, Oscar Diaz, and Salvador Trujillo. Defining dsl expressions collaboratively in multidisciplinary embedded engineering. In *SEAA*, 2011.

[6] http://open-services.net/