OETCS/WP7

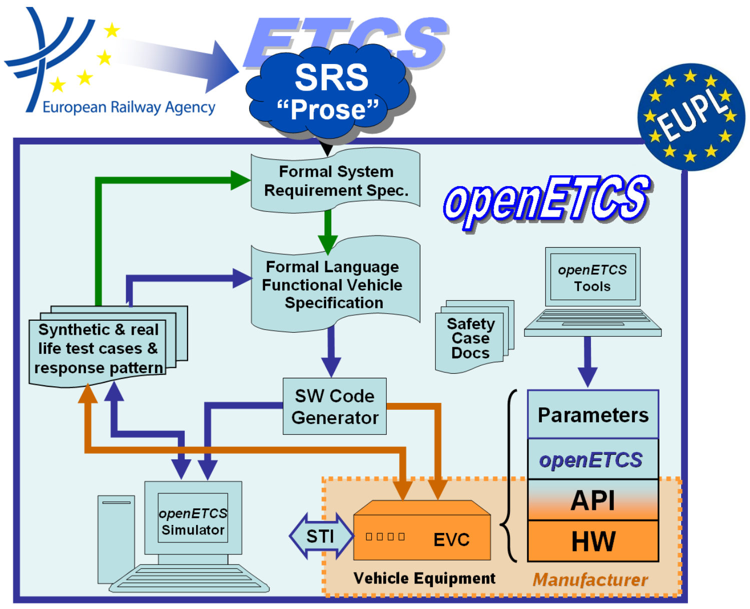
ITEA2 Project
Call 6 11025
2012 – 2015

Work-Package 7: “Tool chain”

openETCS: Migration from Scade to an Open Alternative

Silvano Dal Zilio

October 2015



Funded by:

Federal Ministry of Education and Research

République Française Ministère de l'Enseignement Supérieur et de la Recherche

Investissements d'avenir

Région de Bruxelles-Capitale

Gobierno de España Ministerio de Industria, Energía y Turismo

openETCS

Work-Package 7: “Tool chain”

OETCS/WP7
October 2015

openETCS: Migration from Scade to an Open Alternative

Document approbation

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature Silvano Dal Zilio (LAAS-CNRS)	signature ()	signature ()	signature ()

Silvano Dal Zilio

LAAS-CNRS
7 ave. Colonel Roche
F-31004 Toulouse, France
eMail: dalzilio@laas.fr
WebSite: www.laas.fr

Output Document

Prepared for openETCS@ITEA2 Project

Abstract: This document describes and evaluate a set of scenarios for migrating the openETCS OBU model, currently modeled using the proprietary Scade language, to an open format. After a brief description of the current modeling choices, we describe the different export formats supported by the Scade toolsuite. We base our possible migration scenarios on an analysis of the existing tools that can exploit these output formats.

Table of Contents

Figures and Tables

Figures

Tables

Document Control

Document information	
Work Package Deliverable ID	WP7
Document title Document version Document authors (org.)	openETCS Migration from Scade to an Open Alternative 01

Review information	
Last version reviewed	01
Main reviewers (org.)	

Approbation			
	Name	Role	Date
Written by			October 2015
Approved by	–	–	

Document evolution			
Version	Date	Author(s)	Justification
00.0			creation

1 Introduction

One of the main purpose of the openETCS project is to develop a model of the ETCS using "Open Standards" on all levels. This includes generating open hardware and software specifications, writing interface definition using open languages, and developing open source design tools. One of the most important output of the project is a formal specification of the ETCS onboard unit (OBU) that follows the specification defined by the European Railway Agency (ERA) in its Subset-026 document [1]. Nonetheless, for reasons that we should briefly recall below, the proprietary language Scade—that is a "close-source" solution—was chosen to develop the functional model of the OBU.

Scade, now one of the product of ANSYS (`http://www.ansys.com/`), is a formal modeling language and tool suite targeting safety-critical embedded systems. It has been used for more than 15 years to develop a broad range of control applications in the avionics, rail, and automotive domains. The current Scade model for the OBU is a major output of the project. The architectural model (written in SysML using the Scade system tool) and the functional model (written in Scade using the Scade suite tool) have been developed from scratch—without the use of prior components—and cover all the aspects defined in the ETCS specification. The models are made available on the GitHub of the OpenETCS project (`https://github.com/openETCS/modeling/tree/master/model`) and a complete description is given in deliverable D3.5.3 [2].

Several natural questions arise if we want to meet the "open philosophy" of the OpenETCS project. For instance, would it be possible to develop a model equivalent to what has been achieved with Scade using an Open Source modeling language and Open Source tooling? More interestingly, is it possible to capitalize on the substantial modeling effort already invested in the Scade model and to migrate this model into an open source format? We concentrate on this last question in this report.

Solving this migration problem raises several issues, for instance concerning the simplicity of the migration process or the coherence of the resulting models. Concerning the first point, *simplicity*, we are of course interested by methods that are fully automatic and, if possible, at least partially available (implemented). Concerning the validity of the (target) models—and this is certainly the critical stumbling block raised in this report—we need to evaluate if all the hypotheses used to check the validity of the model (that is to check how close the model is to the intent carried in the informal specification of Subset-026) are safely "carried-over" during the migration process. In the remainder of this text we use the term of *semantics preserving* migration to refer to this issue. This issue is quite delicate. Indeed, Scade relies on a strong implementation hypothesis, the so-called "synchronous approach", and it is not clear how the properties of a model developed with a synchronous approach can be preserved when transferred to another setting, say for example the more "hybrid approach" favored by languages such as Simulink. We give an example of the kind of "semantic" problem that can arise in the next section.

The rest of the report is as follows. In the next sections, we briefly describe the Scade language and the particularity of the OBU Scade model. We close the chapter with a short description of alternative specification languages that could have been used for the model. In chapter 2, we list the criteria that should be taken into account when evaluating the possible migration scenarios. Each migration scenario can be described by a transformation from (one concrete syntax for)

Scade to another language. This is why, in chapter 3, we describe the existing output formats supported by Scade suite. Before concluding, in chapter 4, we define possible scenarios based on the tools that can exploit these different output formats and list the strengths and weaknesses of each approach.

1.1 Overview of Scade and Motivations Behind its Adoption in the Project

Scade is a formal modeling language that provides both a graphical (diagrammatic) and textual concrete syntax. Formal here means that the intended meaning of a model can be defined unambiguously using a mathematical framework. It is therefore possible to reason on models and to prove properties on them with a very high-level of confidence. More precisely, Scade models are synchronously clocked state machines, interacting on infinite data streams, that can be nested and intermixed with each other without limitations. Actually the core of the Scade language is based on Lustre ...XXXX . Another nice property is that Scade is an executable language with a deterministic semantics, meaning that the "execution" of a Scade model is deterministic XXXX????++++. All these characteristics of Scade (formal, concrete, executable and deterministic) made it possible to develop DO-178B and EN50128 certified code generators to C and ADA, which is not a small feat.

The language is supported by a toolbox, the Scade suite, that provides an integrated design and development environment. Scade suite offers several services: simulation and debugging at the model level; test case execution; model test coverage measurement; code verification on models; ... An extension of the tool. called Scade system, provides additional functionalities that are of interest in the context of the openETCS project. Scade System provides systems modeling and model generation based on the SysML standard and the Eclipse Papyrus open source technology. In particular it offers gateway from and to SysML and integration with Eclipse through the use of the Eclipse Modeling Framework (EMF) and its integration with the Papyrus editor. Indeed, Eclipse was selected as the de facto Open Source platform for integrating the tools developed in the project while SysML was selected for modeling the architecture of the system (and the SysML diagrams have been edited using the Papyrus editor, with the participation of CEA)

Scade has been used, in production, on several big projects that covers many application domains: avionics, rail, automotive, . . . It allows the production of rapid prototype as well as of safety related target system software. All these qualities are enough to motivate the choice of Scade for modeling the OBU functions.

But outside these purely factual advantages of Scade upon other formats, there are also important "contextual" reasons. Most importantly there does not exists an Open Source solution with the same degree of maturity. For example, even though the Scade editor requires a moderately high learning curve, it is a solid piece of software that has been tested on many real-size examples. Another is also , since Siemens, one of the major partner of the project, already had a working knowledge of Scade and that some models related to the ETCS (provided by Siemens) were already available. Henceforth, considereing the deadlines of the project, Scade was the most (and the less risky) solution for

We can also list several drawbacks to the choice of Scade. First the choice of formalism means a ... this implies that there is a choice of implementation imposed on the user of the models (namely the choice of a synchronous architecture). This restrict the possible choices when implementing. . The synchronous hypothesis is not XXXXX particularly debilitating XXXX with operating on a single computer since it is not very difficult to have all the components of the OBU to share a common clok. Nonetheless this choice may have some non-trivial consequences if a supplier decide to choose a more distributed architecture. Next we can list the drawbacks related to the choice of a closed-source solution:

- risk of vendor lock-in, ... actually this is at the core of our migration scenario problem since
- difficulty to have special needs taken into account or added to the tool (need of some critical mass in order to have on the);
- risk on the perenity, this is experienced with the recent acquisition of Esterel TEchnologies by ANSYS (and Scade has known multiple "owners" since its debut)

1.2 High-Level Description of the openETCS Scade Model

The Scade model for the OBU is available on the GitHUb of the project (`...`). The A description of the model is given in deliverable XXXX

XXX What were the decision on modeling; what are the choices in modeling that are directly influenced by the choice of Scade ; what was very useful to the modeler in the tooling XXX

For the purpose of this use of a large number of (complicated) types heavy use of state machines, se for example the ManageLevelsAndModes (`https://github.com/openETCS/modeling/tree/master/model/Scade/System/ObuFunctions/ManageLevelsAndModes/Modes`) use of iterators, that are not supported in other flavors of Lustre and other comparable languages, e.g. Simulink

1.3 Existing Open-Source Solutions Around (or Similar to) Scade

link with the Lustre language and existing open-source tooling for Lustre, e.g. lus2lic or prelude, open-source lustre/Scade compiler ; readability of a pure Lustre model when compared to Scade ; licensing issue

possibility to choose another language that can embed Scade (not necessarily synchronous), e.g. a subset of Simulink, for which there is equivalent tooling available and better open-source support

issues concerning the certification of the tools (project requires EN50128)

Choice of a Domain Specific Modeling Language (DSML)

a posteriori evaluation of the use of SysML ; possibility to "reverse engineer" a DSML from the existing Scade code Possible migration path to one or several possible alternatives

SysML would have been a good choice and was explicitly. The drawback is that necessity to choose a "semantics" (different tools have different interpretations of the language, like for examples transitions in State Diagram) and maturity of the tools. With the knowledge accumulated during the modeling effort , but , taken into account the temporal constraints of the project made it impossible to have a backward-forward approach with co-developement. This initial vision of the project for the future ?

2 Overall Goal of the Transformation (What do we Want)

We can try to establish a wish list:

loose as few information as possible (close to round-trip conversion) solution is already existing or at least partially completed available tooling for the format (editor, simulator, formal verification, ...) integration in the OpenETCS toolsuite SysML integration is the translation semantics-preserving ? (use of a discrete time semantics) traceability certification (look at the output of WP2; any reference to a particular deliverable !?) We do not expect to have a working implementation for the end of the project but we can define a roadmap of the possible scenarios for a following project.

Migration raises also several questions related to the life cycle of the model, that is its “evolvability” in the future and its maintainability. Indeed a model is never a finished object, it is evolving and should be updated. The most ambitious goal is to seamlessly replace the use of Scade suite by an open alternative. This will be the most interesting solution. Nonetheless it is necessary to acknowledge the fact that, exactly like software engineering, *model engineering* is essentially a social process and we cannot expect that a transition will not disrupt the work of the modeling team and the quality of its output.

3 Supported Output Formats from Scade

3.1 XSCADE

XML format keeping all editor-specific information (see Sect. 2 of TechnicalManual_SC-TM-R16). Advantage: no information loss Disadvantage: format not very well-documented (related XML Schema not provided !?)

"parsers" are available from OSATE plugin and the S3 tool by Systerel (see below).

3.2 SCADE

SCADE Suite KCG can generate a `kcg_xml_filter_out.scade` file which contains the translation of the graphical SCADE format into a textual format Advantage: no information loss; except for the placement of graphical elements if we want to target a graphical format ; easier to parse Disadvantage: format not very well-documented

"parser" available in the Scade2B project (`https://github.com/cercles/scade2b`)

3.3 KCG

textual format close to Lustre Advantage: Many open-source Lustre implementation are available Disadvantage: we loose the state machines and many information on the underlying "architecture" of the model (hierarchical components are flattened) ; name of variables and ports are quite obfuscated ; KCG is an extension of Lustre

Lustre is not a normalized language and the existing implementations have slightly different syntax. There are some works that have addressed the problem of converting "Scade's version" of Lustre to, for example, VERIMAG Lustre. Nonetheless none of these tools are mature.

3.4 SysML

SCADE System avoids duplication of efforts and inconsistencies between system structural descriptions made of SysML Block Definition Diagram (BDD) and Internal Block Diagram (IBD), and the full software behavioral description designed through SCADE Suite models. Once the system description is completed and checked, the individual software blocks in the system can be refined in the form of models in SCADE Suite or in the form of manually developed source code. Automatic and DO-178B Level A-qualified code generation can then be applied to the SCADE Suite models. Moreover, the SCADE System description can be used as the basis to develop scripts that will automatically integrate the complete application software.

failed attempt of using Scade System during the work on the OpenETCS Data Dictionnary that lists the different values and datagrams used in the ETCS specification together with their datatypes.

4 Possible Scenarios

4.1 Conversion to the AADL

See `http://www.aadl.info/aadl/currentsite/` The OSATE 2 Eclipse plugin provides an importer from XSCADE to AADL Each component in the diagram is mapped into an AADL system component SCADE component connections is mapped using AADL event data port connections SCADE state machines are translated into AADL behavior state machine we need to test the transformation on the current Scade model

4.2 Conversion to HLL

Intermediate format of Systerel S3 tool

the tool is based on a XSCADE parser ; need to extend HLL or adapt the transformation to better take into account (nested) state machines HLL has already been used succesfully on the OpenETCS Scade model (`https://raw.githubusercontent.com/openETCS/validation/master/VnVUserStories/VnVUserStorySysterel/05-Work/S3/Proof_SoM/Modes.hll`)

4.3 Conversion to SysML

with added behavioral information in Lustre

Scade Suite provides an extension mechanism based on the use of TCL scripts. Conversion to Simulink (!?)

5 Conclusion

Solving this migration problem raises several issues, for instance concerning the simplicity of the migration process or the coherence of the resulting models. Concerning the first point, we are of course interested by methods that are fully automatic and, if possible, at least partially available (implemented). It should not come as a surprise that there is no existing solution that satisfies all of our needs. Two main reasons may be identified to explain this situation, that goes beyond the simplistic argument of vendor lock-in: first, there is a relatively small user-base for Scade (and therefore less incentive for third-parties to develop export tools); secondly, there is a lack of compelling scenarios fro trying to export Scade models to another format, except for code generation. Indeed, Scade targets mainly the detailed design phase (the latter design step of the traditional development cycle) and therefore Scade models have no use after the code generation or model verification activities, which are both supposed to be performed using the Scade tools. Nonetheless we provide some plausible scenarios for “escaping” the Scade as uncovered some existing state of the art that gives interesting (and plausible) scenarios.

References

[1] ERA. *System Requirements Specification, SUBSET-026*, March 2012.

[2] Peter Mahlmann et al. *Deliverable D.5.3: openETCS Architecture and Design Specification*, September 2015. `https://github.com/openETCS/modeling/tree/master/openETCS%20ArchitectureAndDesign/D3.5.3`.