ONTOLOGY BASED ASSESSMENT OF DEVELOPMENT PROCESSES

Zoltán SZATMÁRI
Advisor: István MAJZIK

I. Introduction

Our everyday life depends on software to a considerable extent, this way the reduction of the risks of design and implementation faults is of utmost importance. Software development processes are more and more subject to regulations fixed in (general and domain-specific) standards that define criteria for the selection of proper development methods. Accordingly, if software is deployed in a critical environment then an independent assessment is needed to certify that its development process is compliant to the criteria stated in the related standard. The goal of this work is to support the assessment of development processes and toolchains by elaborating a formal verification technique that allows the automated checking of the compliance to standards. On the analogy of classical model checking (that is applied to examine whether a formal design model satisfies some temporal requirements) we represent the development process and tools in a *process model* by means of using ontologies and use a reasoner to check whether the criteria originating in the standard are satisfied.

This vision necessitates the solution of the following tasks. First we formalize the requirements (criteria) in standards that concern the selection of methods and tools. Then we define the modeling techniques to describe the relation and hierarchy of methods, the capabilities of tools, and the construction of (domain-specific) development processes. Finally we elaborate of techniques that check the compliance of concrete development processes (constructed by process designers) to the requirements.

II. Ontology based modeling and model checking

Ontologies are widely used as knowledge management mechanism to capture knowledge about some specific domains. Ontology languages use concepts and relationships between these concepts as a logic model. The ontology languages have a good formal semantics and automatic reasoning algorithms. In order to automate the analysis of an ontology, reasoners are used that are based on the description logic formulation of an ontology.

Web Ontology Language (OWL) is a knowledge representation language specification published by the W3C. This XML based representation of ontologies is supported by the most important tools and is used in this work.

III. Formalization of the requirements

Formalisation is a prerequisite of both formal verification and synthesis support. In this work the focus is on the development processes for safety critical applications, and the EN50128 standard [1] for railway applications is analyzed. This standard defines five safety integrity levels (SIL) for development processes and describes methods that can be applied during the process. For each development step the mandatory (M), highly recommended (HR), recommended (R) and not recommended (NR) methods are described in a tabular form.

The main challenges during the requirement formalisation are the following ones: The development methods are refined hierarchically, i.e., several high level methods are decomposed into alternative

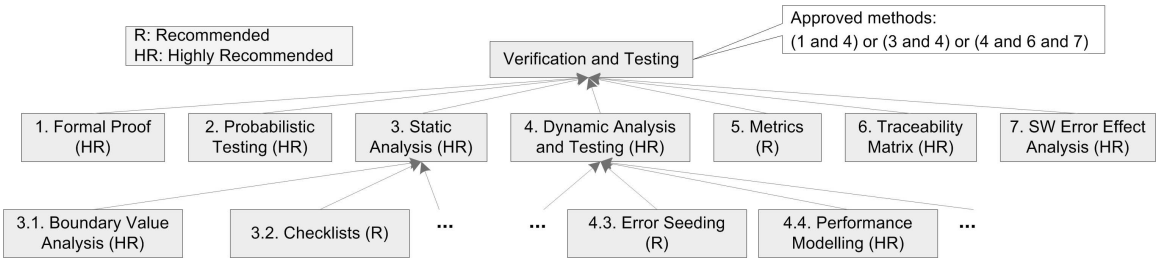

Figure 1: Verification and Testing methods for SIL4 (EN50128)

combinations of lower level ones (Fig. 1). Different requirements are described for each SIL in the standard. Accordingly, this introduces a new dimension into the requirement formalisation. Finally the sufficient conditions for every SIL are formulated using various combinations of the applied methods.

Technique/Method	SIL1	SIL2	SIL3	SIL4
1. Formal Proof	R	R	HR	HR
2. Probabilistic Testing	R	R	HR	HR
3. Static Analysis	HR	HR	HR	HR
4. Dynamic Analysis and Testing	HR	HR	HR	HR
5. Metrics	R	R	R	R
6. Traceability Matrix	R	R	HR	HR
7. SW Error Effect Analysis	R	R	HR	HR

Figure 2: The Verification and Testing methods (EN50128)

In the following, the *Verification and Testing* step of the development process described in the EN50128 standard is presented as a small example in order to demonstrate the mentioned concepts. In Fig. 2 the recommendation level of some methods is shown. The combination of the required methods are expressed as follows: *„For Software Integrity Levels 3 and 4, the approved combinations of techniques shall be (1 and 4) or (3 and 4) or (4, 6 and 7)"*

During the requirement verification step one of the *sufficient conditions* should become true on the process model and none of the *„not recommend"* methods should be used in the process.

These requirements can be represented as Boolean expressions and if these expressions come true on the input process model, then this process model is standard compliant. Because of the challenges in the requirement description (e.g. hierarchical refinement of methods) ontologies are used to characterize the methods and the tasks in the input process models and finally reasoning is used to verify the requirements.

IV. Modeling development processes

The (domain-specific) development process is formalised using a process model which describes the tasks, input and output artifacts, the roles and tools involved in the development process. There are several general purpose process modeling languages (e.g. BPEL, BPML).

In this work the OMG's Software Process Engineering Metamodel (SPEM) specification is used, because the focus of the SPEM is development projects and it is defined as a meta-model as well as a UML 2 Profile. The Eclipse Process Framework supports this specification and is proposed in our environment to model the processes. Using this framework the process designer can construct the specific development process, can assign the available tools to the tasks of the process, or can choose from available toolchain patterns.

In order to support the logical reasoning as model checking, the process is represented using an ontology. W3C's OWL-S ontology supports the description of service composition as well as business

processes. In the following the process description capability of this ontology will be used to describe the development processes [2].

The OWL-S ontology defines the *Process* concept that can be an *Atomic Process* or *Composite Process*. *Atomic processes* correspond to single steps of the development processes and composite processes are decomposable into other processes. Their decomposition can be specified by using control constructs such as *Sequence* and *Choice*.

The tasks of the process implement particular methods that can be classified by the standards, and based on this classification the assessment can be supported. A formal representation of the hierarchical structure of methods can be provided by defining a new ontology. Here concepts refer to the development methods and their relations include the refinement. The OWL-S ontology is specialized in order to support this method classification and this extension is called *methods ontology*. In this extension new concepts are defined as subclasses of the *Atomic Process* concept (e.g. Fault Tree Analysis, Probabilistic Testing.)

A simple example development process is shown in Fig. 3. Note that this development toolchain is compliant to the standard since the combination of Formal Proof (implemented by the SAL model checker) and the Symbolic Execution (which is a Static Analysis method implemented by the PolySpace tool) is a valid combination for SIL3 and SIL4.

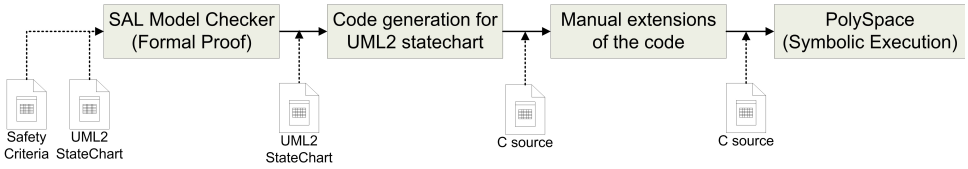

Figure 3: Sample development process

An additional step of the formalisation process is the construction of the tool repository. This repository is a collection of tools that can be used in a given company during the (construction of the) development processes. Each available tool is classified on the basis of the concepts defined in the *methods ontology* constructed in the previous step, i.e., for each tool the supported methods are given.

V. Mapping SPEM to OWL-S

The required SPEM to OWL-S mapping is implemented using the VIATRA model transformation framework [3]. To use VIATRA the metamodel of the SPEM process modeling language and the metamodel of the $\mathcal{SHOIN}(\mathcal{D})$ ontology language is constructed in the VIATRA model space. After that the „SPEM to OWL-S" model-transformation (based on these metamodels) [4], an importer for the SPEM models and an exporter for OWL ontology format is implemented.

The SPEM model is constructed using the SPEM UML profile, so the UML metamodel (extended with UML profile support) is used to represent the input model of the transformation. The OWL metamodel is constructed based on the Ontology Definition Metamodel (ODM) developed by the University of Karlsruhe. [5]. This metamodel fits well into the Meta Object Facitlity (MOF) architecture and can be applied in the VIATRA model transformation framework.

VI. The assessment toolchain

The assessment of development processes is implemented by an *assessment toolchain* in order to support automatic execution of the steps starting with the SPEM model transformation into ontology based models and finishing with the reasoning (Fig. 4)

First the process model is constructed by domain experts using the EPF Composer tool. This input model is transformed into an OWL-S based ontology then the used tools and atomic tasks are classified

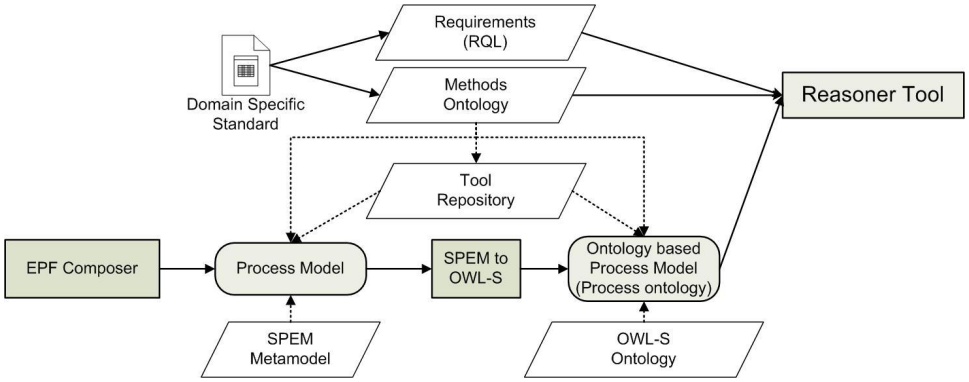

Figure 4: The assessment process

using the *methods ontology*. The output model is a process ontology. Finally, the standard conformance of the development process can be checked using the reasoner tool, that is executed on the process ontology.

According to the approach described above, all of the tasks, the tools and thus the development processes are characterized using the concepts represented in the ontology. Using the concepts defined in the ontology, the sufficient conditions for the selection of methods and the dependency on the safety integrity level should be represented as boolean expressions. These expressions can be described using ontology query language, for example the New RacerPro Query Language (nRQL). The query should check whether the required combination of the methods are included in the process model.

Accordingly, the standard conformance of the selection of methods and their supporting tools in the development process can be checked using an ontology reasoner.

VII. Future work

During the standard based assessment of development toolchains not only the used methods are important. The standard specifies that safety arguments are required during the certification process. These safety arguments communicate the relationship between the evidence and objectives.

The arguments can be ordered into a *hierarchical breakdown structure*. There could be some parts of these arguments that are produced by tools in one step or by toolchains in multiple steps. The assessment process will be extended to support the construction of development processes by identifying missing safety arguments.

References

[1] CENELEC, "En 50128: Railway applications - communication, signalling and processing systems - software for railway control and protection systems," URL: http://www.cenelec.eu.

[2] P. M. Anupriya, A. Ankolekar, M. Paolucci, and K. Sycara, "Towards a formal verification of OWL-S," in *In Fourth International Semantic Web Conference (ISWC 2005*.

[3] "VIATRA model transformation framework," URL: http://wiki.eclipse.org/VIATRA2.

[4] J. Shen, Y. Yang, C. Wan, and C. Zhu, "From BPEL4WS to OWL-S: Integrating e-business process descriptions," in *SCC '05: Proceedings of the 2005 IEEE International Conference on Services Computing*, pp. 181–190, Washington, DC, USA, 2005. IEEE Computer Society.

[5] S. Brockmans, P. Haase, P. Hitzler, and R. Studer, "A metamodel and UML profile for rule-extended OWL DL ontologies.," in *ESWC*, pp. 303–316, 2006.