STANDARDS-BASED ASSESSMENT OF DEVELOPMENT TOOLCHAINS

Zoltán SZATMÁRI
Advisor: István MAJZIK

I. Introduction

Software pervasiveness has the consequence that our everyday life depends on software to a considerable extent. To reduce the risks of software design failures, the *software development processes* are more and more subject to regulations fixed in (domain-specific) standards. Accordingly, if software is deployed in a critical environment then an independent assessment is needed to certify that its development process, i.e., the set of applied *methods* and the supporting *tools* or *toolchains* are compliant to the requirements stated in the standard. The need for certification has arisen in several European projects (e.g., DECOS, MOGENTES) that aim at the elaboration of toolchains for the development of embedded and/or safety-critical systems.

The goal of my work is to support the assessment of development processes and toolchains by elaborating *formal verification techniques* that allow the automated checking of the compliance to standards. This vision necessitates the solution of the following tasks:

- Formalization of the requirements in standards that concern the use of methods and tools.
- Definition (or adaptation) of modeling techniques to describe the relation of methods, the capabilities of tools, and the construction of (domain-specific) development processes.
- Elaboration of techniques that check the compliance of concrete development processes (constructed by process designers) to the requirements.

It is worth mentioning that the formalization of the requirements and the model-based description of tools and methods open a way to support also the *synthesis of processes and toolchains* that are compliant to the standard. The process designer can be assisted by (i) identifying missing methods and tools, (ii) proposing alternative solutions, (iii) offering a library of toolchain patterns, (iv) optimizing processes from the point of view of costs, time, safety etc.

In the following we describe ideas and initial results related to the implementation of the above mentioned tasks.

II. Formalization of the requirements

Formalization is a prerequisite of both formal verification and synthesis support. I focused on the development processes for safety critical applications, and analyzed the EN50128 standard for railway applications [1]. 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, highly recommended, recommended and not recommended methods are described in a tabular form. The development methods are refined hierarchically, i.e., several high level methods are decomposed into alternative combinations of lower level ones (See Fig. 1).

A formal representation of the hierarchical structure of methods can be provided by defining an ontology [3] and describing it using description logic, ontology. Here concepts refer to the development methods and their relations include the refinement.

Using the concepts defined in the ontology, the *necessary* and *sufficient conditions* for the selection of methods and the dependency on the safety integrity level can be described for each step in the process using logical statements (interpreted as static constraints). The required temporal ordering of methods can be formalized using *temporal logic* formulae.

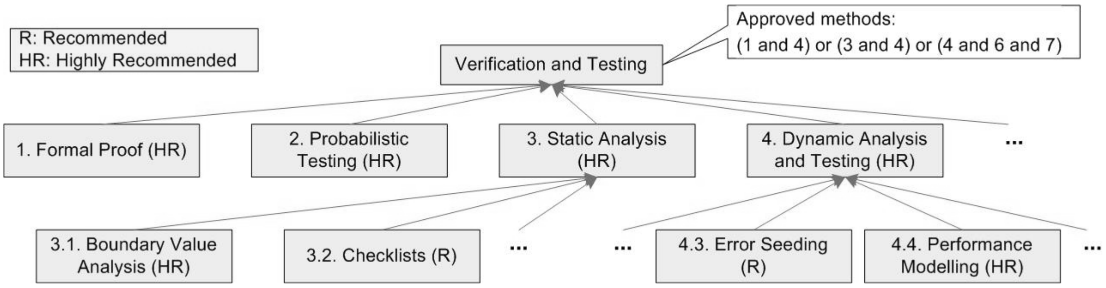

Figure 1: Verification and Testing methods for SIL-4 (EN50128)

III. Assembling an extensible tool repository

The next step of the formalization process is the construction of the tool repository. This repository is a collection of tools that can be used during the construction of the development processes. Each available tool is classified on the basis of the concepts defined in the ontology constructed in the previous step, i.e., for each tool the supported methods are given. Complex methods are supported by *toolchain patterns* that are formed by tools that have to be executed in a predefined sequence.

IV. Modeling the development process

The (domain-specific) development process is formalized using a *process model* which describes the tasks, input and output artifacts, the roles and tools involved in the development process. The MOGENTES project uses the Eclipse Process Framework [2] to model the processes. Using this framework the process designer (i) can construct the specific development process, and can *assign the available tools* to the tasks of the process or (ii) can choose from the available *toolchain patterns*. The tasks of the process implement particular methods that are classified using the ontology.

V. Assessment of domain-specific development toolchains

Using the formalism described above, all of the tasks, the tools and the requirements of development processes are characterized using the concepts represented in the ontology. Accordingly, the *standard conformance* of the selection of methods and their supporting tools in the development process can be checked using an *ontology reasoner*. The requirements about the ordering of the methods can be checked using a temporal logic *model checker*. This way the assessment can be supported by reusing existing formal methods and checker tools.

Note that based on the process model and the available tools in the repository the reasoner can identify missing methods and can give hints about the supporting tools as well.

VI. Conclusion

Formalization of requirements is a wide research area. In this work I proposed an approach to *formalize the requirements of development processes*. This approach forms the basis for the assessment of the standard compliance of specific toolchains and provides support for the process designers to construct certifiable development processes.

References

[1] EN 50128: *Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems*. URL: http://www.cenelec.eu

[2] *Eclipse Process Framework project*, URL: http://www.eclipse.org/epf/

[3] Szeredi P., Lukácsy G., Benkő T.: *A szemantikus világháló elmélete és gyakorlata.,* Typotex, Budapest, 2005.