Experimenting diversity in the formal development of railway signaling systems

Alessandro Fantechi, Stefania Gnesi, Giovanni Lombardi

The FMT group of ISTI-CNR has established since some years a collaboration with ALSTOM FERROVIARIA S.p.A., aiming at the introduction of formal specification and verification tools in the software development process of railway signalling products. Several experimental common projects have been conducted in this direction: the most recent ones have concentrated on the actual feasibility of automatic code generation starting from a specification of the logic of the system given using SCADE, tackling in particular issues of performance over specific proprietary hardware architectures and of integration with existing software modules.

The availability of specifications of real signalling equipment and of the complete environment needed to produce an application at industrial level, from a formal model to the code, has been exploited in ISTI to undertake an additional research effort to review the used development process in terms of safety regulations, in particular by means of the introduction of diversity, to improve safety of the produced equipment.

The introduction of diversity has been considered where an analysis of the safety measures, employed to limit the design faults, has revealed possible weakness of the development process. The considered development process is based on the use of the SCADE tool by Esterel Technologies. A formal model of the (components of the) equipment is first developed using SCADE, with the added possibility of simulating the model and verifying it by model checking, in order to acquire an high confidence on the absence of software faults. Code is then generated by the SIL 4 validated SCADE code generator. A first possible weakness of this process has been identified in the supporting software, that is, the underlying operating system and the compilation environment, which are not validates software components. A first form of diversity has been therefore introduced at the level of the compilation of the generated code, and is aimed at discovering possible faults either due to the compilation environment or to the underlying operating system (figure 1). Two different compilation environment running on two different operating systems have been employed: the proprietary embedded platform with the dedicated compiler and a commercial compiler over the Windows platform. Parallel testing of the two versions with the same set of tests (taken form the official suite of acceptance tests) has been employed with the aim of revealing differences in how the generated code interfaces with the operating system or in how it is compiled.

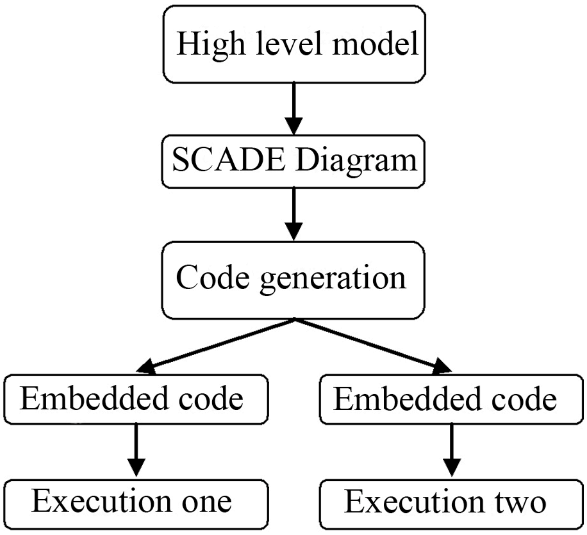

Figure 1 – Compilation and operating system diversity

Although using sophisticated verification tools such as model-checking, it cannot be guaranteed that the process of writing a formal model for the developed system has faithfully captured the (informal) system requirements. Diversity can help also at this stage, by considering two independent formal specifications.

Hence, a second form of diversity is introduced at the level of specification, having therefore an impact over the whole development process.

The specific example from the railway signalling domain has provided a direct way to conceive diverse specifications: the relay schemas which still constitute a common language for railway signalling engineers have been used for one version while a more “modern”, and increasingly popular, notation - UML Sequence Diagrams - have been used for the other one.

From the two given specifications two independent chains of verification / code generation / compilation / deployment have been implemented (figure 2). Again, the final comparison is made by running on both versions the set of official acceptance tests for the developed equipment.

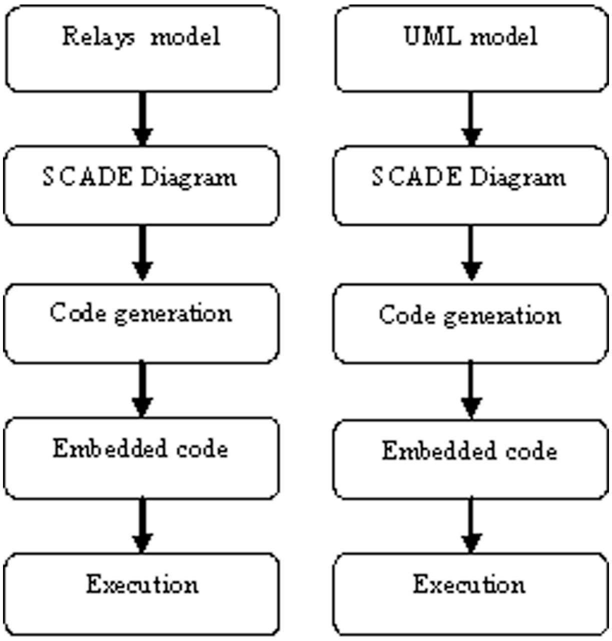

Figure 2 – Specification diversity

The experiments on the introduction of diversity in compilation have been encouraging, due to their relatively low cost that can positively affect the industrial acceptance of the approach.

Indeed, the added cost of the first approach to diversity is limited to repeat compilation and testing on a Windows based machine, with practically no cost for additional hardware and software resources.

Moreover, replication of compilation and testing can be automated to a high extent.

In contrast, the introduction of diversity in specification requires at least the additional effort of writing an independent specification. The overall cost of diverse specifications approaches therefore twice the costs of a single formal specification process.

The higher costs of this form of diversity can actually be justified by the lower testing and debugging costs due to the early discovery of design faults, and by the higher safety objectives achieved.

Links:
http://fmt.isti.cnr.it

Please contact:
Stefania Gnesi
ISTI–CNR
Via G. Moruzzi 1
56124 Pisa, Italy
stefania.gnesi@isti.cnr.it