



ES_PASS – Final Workshop “Industrialization of Abstract Interpretation”

Abstract: “Experiences with ES_PASS Static Analysis Tools and Future Aspects within the Railway Domain”

Document ref. : **ES_PASS_Madrid_2009_Abstract_Busse_Gerlach**
Version : **4.1**

Authors: **Juergen BUSSE, Jens GERLACH**

Date : **2009-10-08**

Abstract:

“Experiences with ES_PASS Static Analysis Tools and Future Aspects within the Railway Domain”

Version 4.1

Page 2 of 4

„Experiences with ES_PASS Static Analysis Tools and Future Aspects within the Railway Domain“**Introduction**

So far static software analysis tools have not penetrated the railway domain. The presentation aims at how to overcome the obstacles, what results have been derived from the ES_PASS experiments on static analysis tools and to what extent tool providers and tool users can gain experience from other industrial domains.

Experimentation results

The experiments of IFB and Fraunhofer FIRST with a number of static analysis tools resulted in the following findings:¹

Polyspace Verifier:

Successfully installed. Polyspace was able to detect faults in our C++ probe. A C source code file that was generated by a graphic-orientated programming environment could not be successfully analysed. (Note that graphic-orientated programming environment is typical for rail vehicle on-board software.)

aiT WCET:

Successfully installed. Unfortunately, aiT WCET can so far not process files in “hex-format”. This format, however is widely used in the railway domain. Moreover, the tool must be configured for the used tool-chain. In particular, the exact combination of compiler and processor must be met.

CAVEAT:

Successfully installed. Has the advantage of being qualified for safety-related verification.

FRAMA-C:

Successfully installed. In principle, this tool can prove that a given C code fulfils a set of formally specified functional requirements. However, as of now, Frama-C has problems processing system-headers and generated C code.

Our main conclusion is that static analysis tools must become more robust and must be better at analysing generated code. At the same time, the vendors of code generators should make sure that they do not rely to much on the dynamic features of C. The reason is clear, because the results of dynamic data structures and weak typing can hardly statically analysed.

¹ For details please refer to ES_PASS deliverable D4.3.1.

Abstract:

“Experiences with ES_PASS Static Analysis Tools and Future Aspects within the Railway Domain”

Version 4.1

Page 3 of 4

Compliance with industrial standards²

EN 50128:2001 forms the software development standard in the railway domain. This standard addresses static analysis. This verification method is highly recommended from SIL 1 upwards. However, the standard recommends using an appropriate choice of the recommended measures. This means that the requirements of the standard can be fulfilled without using static analysis.

In addition, the term “static analysis” in EN 50128:2001 is not limited to tool based verification.

Thus, the EN 50128:2001 recommends the implementation of static analysis. Nevertheless, this standard does not enforce in any way the use of advanced static analysis methods as developed in the ES_PASS project. This holds with respect to the new draft version prEN 50128:2009 as well.

Key factors for improved adaptation of tools and software

Key factors for a better adaptation have been identified as follows:³

Tool usability:

- Pre-defined settings and improved manuals.

Tool qualification for application to safety-critical software:

- Tool qualification is an important pre-condition to allow replacement of other verification activities by static analysis.

Tool integration in the software development process:

Overcoming of technological barriers such as

- specific operating systems for the tools,
- specific processor and compiler types and specific executable formats that differ from commonly used standards in the railway domain and
- less reliance on a *dynamic* programming style with many casts between more or less related types.

Tool marketing:

Integration of tools into the development process. This might be driven by means of

- co-operation between tool provider and key players in the railway domain (as successfully executed in the automotive domain),
- more attractive tool license prizes and
- stronger recommendation of tool usage by safety authorities, assessors and standardization bodies.

² For details please refer to ES_PASS deliverable D2.9.1.

³ For details please refer to ES_PASS deliverable D4.1.2b.

Abstract:

“Experiences with ES_PASS Static Analysis Tools and Future Aspects within the Railway Domain”

Version 4.1

Page 4 of 4

Outlook on succeeding research activities in the static analysis fieldDEVICE-SOFT – Deductive Verification for Industrial Critical Embedded Software

Hoare logic-based deductive verification and similar methods have been successfully studied from an academic point of view. However, their use in real-world applications, developed using contemporary programming languages for embedded software, remains largely undone. In this context, the main goal of the project is to disseminate, improve, integrate, and deploy deductive verification technologies into the industrial domain of safety critical embedded systems.

The project is centered around the Frama-C framework and its deductive verification plug-in. Fraunhofer FIRST works with the main developer of Frama-C, CEA LIST, in order to guide the development of the Frama-C toward relevant real-world issues, and to foster its adoption by an increasingly larger community of users, by making them aware that this tool exists, and is able to tackle their software verification tasks.

STUDY GROUP IFB/FIRST

Based on ES_PASS experiments IFB and Fraunhofer FIRST continue with developing formal specifications of typical safety requirements from the railway domain. This work aims at transforming the informal customer specifications into a standardized formal specification language.

Formal specification of pre-existing software modules and their deductive verification forms another part of the work.

This research work shall support the opening of the deductive verification to pre-existing software development processes.