Title: Applying TLA+ in a Safety-Critical Railway Project

Duration: 20 min.

Speaker:

Stefan Resch
Dependable Systems Architect
Thales
stefan.resch@thalesgroup.com
www.thalesgroup.com

Abstract:

The presentation gives an overview on the experience and insights gained from using TLA+ and PlusCal to model and develop fault-tolerant and safety-critical modules for TAS Control Platform, a platform for railway control applications up to safety integrity level (SIL) 4. It is based on the work also presented in [1] and extended with knowledge we gained in later development phases.

Thales Ground Transportation Systems has bundled the development and maintenance effort of fault-tolerant distributed computing systems in the TAS Control Platform for highly safety-critical applications. In the process of extending the supported set of redundancy architectures of TAS Control Platform, we decided to use formal methods for checking our designs. The chosen method had to fulfill the following three objectives:

1. Show that the algorithms work with respect to the distributed environment,
2. Gain confidence that the model faithfully reflects the environment and algorithms, and
3. Have a direct relation to the implementation to reduce errors introduced during coding.

Achieving the first goal was relatively straightforward with TLA+ and PlusCal. We could check quite elaborate linear temporal properties with the model checker TLC on the specification of algorithms and environment. However, whether these properties are indeed correct or hold due to model oversimplification or simple errors in their specification is not a straightforward task.

In order to gain confidence in our model we came up with a method we call property-driven design. Analogous to test driven design, property-driven design requires the iterative definition and checking of model properties based on requirements followed by extending the functionality of the model. This results in the designers continuously executing the model checker and observing many traces. Additionally, the growth of the state space is observed with each iteration. Finding design flaws early and having a good insight for decisions concerning the modular structure of the overall program are the advantages of this approach.

We finally close the gap to the implementation by writing a simple translator from PlusCal to C code.

This approach enabled us to identify several design bugs at a very early stage of the development. Typically race conditions, which would have been hard to find with testing. Furthermore, after completion of the design phase and during integration of the TLA+ models, we found only two additional bugs in these models. The first was a special case of a generic scenario. This case was not captured in the model due to oversimplification, which in turn was a result of constraining the state space of the model. The second bug was a naming issue in the interfaces of the modules.

Although using TLA+, PlusCal and TLC with the described method was a substantial effort, we believe that it already paid off considering the bugs found during the design phase.

This work has been conducted within the ENABLE-S3 project that has received funding from the ECSEL Joint Undertaking under Grant Agreement no. 692455. This Joint Undertaking receives support from the European Union's HORIZON 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway.

ENABLE-S3 is funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program "ICT of the Future" between May 2016 and April 2019. More information https://iktderzukunft.at/en/

[1] S. Resch and M. Paulitsch, "Using TLA+ in the Development of a Safety-Critical Fault-Tolerant Middleware," 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Toulouse, France, 2017, pp. 146-152. https://doi.org/10.1109/ISSREW.2017.43