Automatic Timing Verification in a Continuous Development Environment

KOSTAL

> “The TA Tool Suite, especially TA Inspector, is a major key to performing an automated integration test in a complex project setup. Several system parameters are analyzed within a fully automated continuous integration system in order to get a fast response regarding system stability and health. TA is a solution-oriented and competent development partner.
> Our project challenges and objectives have been understood, discussed, and always been solved to full satisfaction.”
>
> Björn Kräher , Leopold Kostal GmbH & Co. KG

Leopold Kostal operates among others in the automotive electronic devices areasand integrates the TA Tool Suite into its standard integration process for automatically verifying timing requirements in a fully automated test toolchain.

Tool Solution

TA Inspector

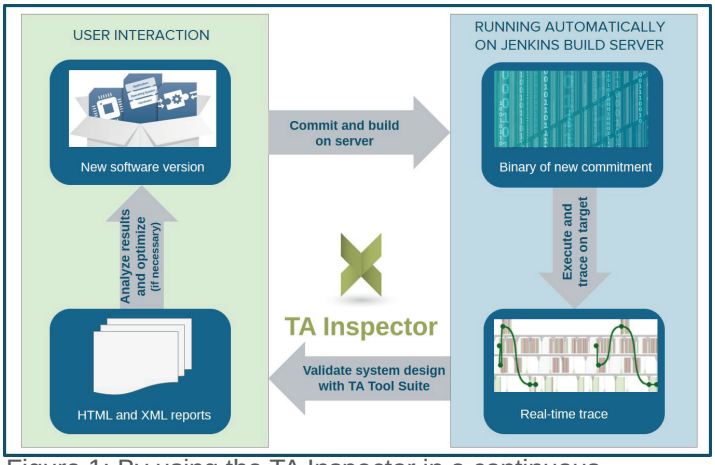

Figure 1: By using the TA Inspector in a continuous integration environment the real-time health state of an application can be monitored automatically.

The Challenge

- Evaluation and verification of real-time requirements via trace based analysis
- Integration into an automated build environment and execution of nightly builds
- Check different application use-cases during load and initialization sequence
- Guarantee correct integration, execution, and interaction between all software modules

The Solution

- Specification of event-chains and definition of event-chain requirements in the standardized AUTOSAR format
- Automated evaluation and requirements verification with the TA Inspector
- Integration of the TA Inspector into the nightly build process via the batch processing interface of the TA Tool Suite
- Overview of the health state of the system via human readable HTML and machine readable XML reports

The Benefits

- Complete automation of the trace verification process into the build process - no manual interaction required
- Early detection of timing problems and missing dependencies by integration into a continuous development process
- Easy continued use and modification of the event-chains and requirements by relying on the standardized AUTOSAR format
- Detect problems at a glance by reporting the evaluation results directly to the Jenkins build server via the JUnit XML format