



© QTronic

Continuous, Platform-independent Testing of Functional Requirements

Thousands of kilometres are driven in simulation and with real car prototypes to ensure that all functional, safety and quality requirements are met with enough coverage of the various operating conditions and driving situations. In this paper, Daimler and QTronic present their approach to build an efficient requirement assessment framework which can be reused for the various test platforms and use cases that contribute to a comprehensive and continuous quality assessment. This approach is currently being reused by several powertrain development projects in the Daimler Group.

AUTHORS



Holger Brückmann
is Development Engineer at Daimler AG in Stuttgart (Germany).



Alexander Waiss
is Development Engineer at Daimler AG in Stuttgart (Germany).



Markus Stix
is Team Leader at QTronic GmbH in Stuttgart (Germany).



Dr. Ingo Matheis
is Team Leader at QTronic GmbH in Stuttgart (Germany).

CHALLENGES

Most often, due to limitations of the representation or of the implementation, the requirement models initially developed for tests in MiL/SiL/HiL simulation, cannot be reused later for all use cases that contribute to a continuous quality assessment. For instance, the test drivers often must rely on pen-and-paper checklists to make sure that they performed all the maneuvers that are planned for a test drive. Moreover, without an immediate feedback from the maneuver assessment, they cannot be sure that the car always reached the operating conditions required by the test. This reduces the completeness, respectively increases the costs of the test drives which are iteratively performed during the development. A further important source of data for the safety and quality assessment under real operating conditions are the measurements recorded by data loggers, either from test benches or from test drives. Requirements tested by classical test automation scripts, being designed to test the system reaction only in the context of a predefined test stimulus, cannot be applied for this case either. Multiple redundant requirement assessment models, hard to maintain in-sync with the development, are often a consequence. That causes additional costs and hinders a continuous and efficient requirement assessment during the development process.

The aim is to cover the following relevant use cases with a unique representation and interpretation of requirements:

- online assessment of manual and automated tests performed in simulation
- offline assessment of measurements from data loggers
- online assessment during test drives. This includes the real-time feedback and guidance for the driver to cover the required tests and maneuvers.

The following usability objectives are important as well:

- Requirement models should be easy to read and write.
- Maintenance, debugging and sharing of models should be easy.
- Multiple product variants should be supported.

FORMAL SPECIFICATION OF REQUIREMENTS

A first step toward platform independent testing is to realize that a requirement model should not depend on a particular test stimulus. The requirement models should detect the necessary operating conditions when the associated system reactions are expected to occur and trigger the assessment in all situations when the pre-conditions are met. This way, the test stimulus is separated from the requirement model and the

requirement can be evaluated for virtually any stimulus.

The “QTronic Requirement Modelling Language (RML)”, which is part of Test-Weaver, provides an easy to use, yet unambiguous specification language that can be translated automatically into executable code, FIGURE 1. The language encodes logical and temporal relations which, depending on the statement, either wait for triggering conditions or check expected reactions. This is done for all requirements simultaneously.

- Online tests performed in simulation with single or multiple virtual ECUs: Silver is used to build virtual ECUs and to simulate networks of virtual ECUs on a PC in closed loop with plant models. The compiled RML modules are used for virtual system tests. These are executed automatically after integrating new software versions. The stimuli for these tests originate from a variety of sources: predefined test scripts, recorded test drives, as well as computer-generated stimuli for automatically increasing the test coverage with TestWeaver [1, 2, 3].
- Offline measurement evaluation: Huge amounts of data recorded by loggers or measurement, calibration and diagnostics (MCD) tools can be checked for requirement compliance offline. This can be performed with Silver with the same compiled RML modules. The measurement evaluation usually runs several times faster than real-time on a PC.
- Real-time assistance and assessment of acceptance tests performed in the

car: Silver running on a laptop can be connected to the real car communication networks (CAN, Lin, Flexray, Ethernet) via a hardware device. The execution of the compiled RML files is performed in real-time for this use case. Feedback about the executed and the remaining tests is provided to the driver on the laptop, FIGURE 3.

All three use cases benefit from the concurrent evaluation of all requirements and the resulting test density. The first two use cases have been in use since a longer time at Daimler AG [1, 2, 3]. The online assessment of acceptance tests performed in the car was the last testing task that was not possible to complete with the same requirement models and assessment tools until now. Therefore, some more details about this setup are given in the following.

An Excel-based frontend documents in real-time the test results and displays hints about the maneuvers expected to be executed in order to complete the untested requirements, FIGURE 4. During a certain maneuver all watchers with

matching preconditions are evaluated and can report a success or a failure. It is possible that a requirement, that initially reported a successful evaluation, reports a failure at a later evaluation, for instance under differing operating conditions. The success state of a requirement evaluation can thus turn into a failed result later (but not the other way around). Measurement snippets in MDF format are automatically recorded for each requirement watcher that reaches

SUPPORT FOR VARIANT DEPENDENT REQUIREMENTS

The Daimler AG builds dozens of different variants for one car model [2]. The differing configurations sometimes expose sets of optional features or differing implementations of the same functions, with differing operational characteristics. Of course, this must be reflected in the set of requirements that are checked with a given system configuration. The requirements associated with a system must be composable and configurable in the same way in which the system itself is composable and configurable.

RML supports this with the declaration of system configuration constraints. This declaration is part of the meta information provided by the “@config section of an RML file.” For instance, the watcher `Change_driving_` program from FIGURE 2 will be considered only when testing a system that has an automatic transmission with type AT or DCT. This configuration mechanism can also be used for (de)activating platform dependent requirements, since