Quality Assurance in Modular Software Development

By Matthias Thullner, Robert Bosch GmbH, and Andre Scholand, ETAS

"CompareComputing" automatically pinpoints errors after migration from rapid prototyping system to ECU platform

It would be a fallacy to assume that software which has been validated on its development system will always run flawlessly on the production ECU. To combat this irksome quirk, ETAS and Bosch's Chassis Systems division jointly developed the process.
In a nutshell, it consists of an automatic real-time comparison of the respective software versions of development system and production ECU. As a result, once function testing on the experimentation system was successful, it need not be repeated on the ECU.

Quite often, the root cause of the flawed performance of software running on the production ECU may be found in optimizations that were aimed specifically at the control unit in question. In the so-called CompareComputing process, both software versions are executed concurrently, with the validated software on the rapid prototyping system serving as the reference for the software onboard the ECU. The benefits of CompareComputing are twofold: On the one hand, the process saves time because it dispenses with the manual software comparison. On the other hand, it may reveal faults that would otherwise manifest themselves much later in the engineering process or, even worse, when a vehicle is being driven by the end user.

Bosch Chassis Systems started the development of "CompareComputing" back in 2001. At the time, the intent was to test the existing optimization tools for ECU code, and to ensure improved software quality.

As a provider of engineering services, ETAS contributed to the CompareComputing project from day one. It was envisioned that ETAS development tools already in use at Bosch were to be suitably configured to form an automated test system.

The CompareComputing approach combines the ES1000 rapid prototyping system and ASCET tool. In the case under scrutiny, it is installed in the laboratory test bench and connected in parallel with the production ECU. The lab test bench consists of LABCAR hardware, LABCAR-OPERATOR, and LABCAR-AUTOMATION.

CompareComputing

The program code for the ES1000 and the respective ECU is generated from the very same ASCET model, along with identical implementation. The use of the single-source principle enables CompareComputing to perform a bit-accurate comparison of the software. The process ensures the early identification of minor faults that would not necessarily be detected in a driving test.

The use of LABCAR facilitates the exchange and modification of all test sequences and parameters of the software under test. The test system generates stimuli used by both production ECU and ES1000 for the simultaneous computation of the control algorithms for the ABS software. The results thus obtained are then subjected to a bit-level comparison on the powerful ES1000 system. LABCAR-AUTOMATION concludes the procedure by generating an HTML or XML report.

The available Ethernet connectivity allows for a distributed hardware configuration. Accordingly, these tests must not necessarily be conducted on the test bench in their entirety but may be run from a networked PC in a spatially distant office.

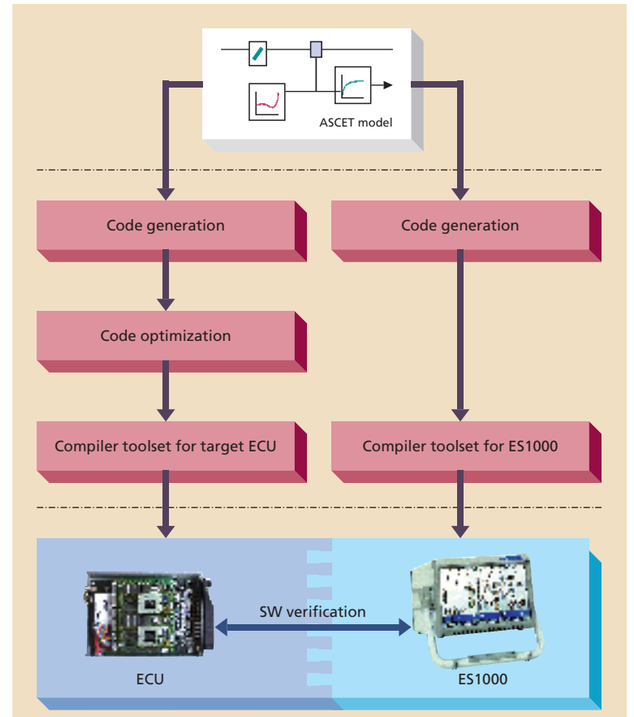
Findings

Effective with the deployment of ASCET for the generation of software for production ECUs, Bosch Chassis Systems have been using CompareComputing as an RTM release test. The results to date have been impressive. For example, Bosch development engineers succeeded in the early detection of several fault variants:

- Compilation errors cause deviations in the ECU.
- Divergent interpolation routines for memory maps and characteristics characterize the implementation on the production ECU, as opposed to that on the experimentation system.
- Overflow of data types as a result of missing or misplaced limiters.
- Functional faults caused by errors in the make and code optimization tools.
- Hardware faults, e.g., defective RAM segments in ECU onboard memory.

Fault analysis

One of the LABCAR functions absolutely essential to proper fault analysis is the ability to download the contents of the ECU fault memory. The auto monitoring program onboard the ECU logs all fault occurrences by writing appropriate information into the fault memory. The resulting entries are consulted and recorded as part of the plausibility performed by the CompareComputing process.

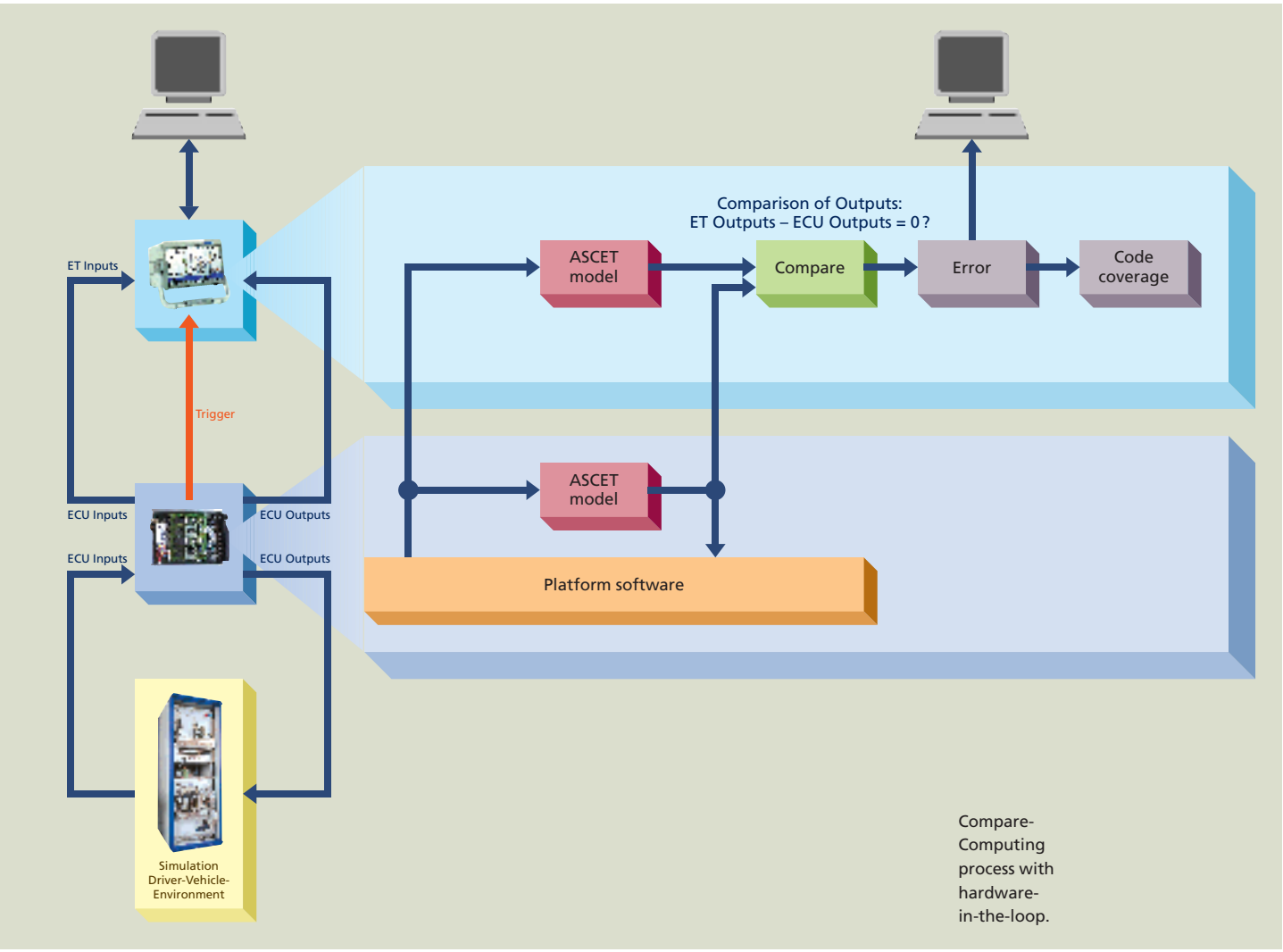


CompareComputing code generation from single source.

At this time, fault identification by CompareComputing still requires project know-how that aids manual troubleshooting. To this end, CompareComputing supports the developer who is able to toggle directly between the test report and the affected C Code with a mere mouse click. The current system comprising a combination of ASCET, LABCAR, and test automation is slated to be improved to a point where the root causes of faults will be pinpointed automatically through iterative testing.

Code coverage analysis

The test intensity of CompareComputing can be checked my means of a code coverage analysis. It ascertains – for both the rapid prototyping and ECU test – how many percent of the program code were processed in response to the LABCAR stimuli. Typically, standard driving maneuvers without hardware-based fault simulation result in a code coverage of two-thirds of all method calls, limiters, and branches. So far, because not all software functions are examined in the laboratory test, it has not been possible to obtain full one-hundred percent coverage. Future plans call for raising the degree of coverage with the aid of additional test cases.



Compare-Computing process with hardware-in-the-loop.

Benefits

The CompareComputing approach is chiefly based on the single-source principle: The ASCET development environment serves as the single source of both rapid prototyping code and production ECU code because this is the only way in which the comparison that lies at the heart of CompareComputing is useful. With the use of this method, the inherent benefit of attaining better-quality software is readily apparent: The system facilitates automated tests by means of LABCAR, and it delivers precise code analysis, high test coverage, and test report generation. Also, the configuration management with respect to tested software versions and test environment ensures test reproducibility.

On the basis of the successes achieved so far, Bosch Chassis Systems plans to add further refinements to the CompareComputing procedure. In addition, Bosch Chassis Systems anticipates using the new LABCAR-AUTOMATION V2.0 software to arrive at full test run automation. (For more information on this subject, see also the article "Automated Testing Cuts Down on Time and Costs" on pages 26-27 in this issue of RealTimes.)
For the first time, LABCAR-AUTOMATION V2.0 contains the widely used TTCN3 description language. As a result, LABCAR-AUTOMATION V2.0 users need not possess comprehensive programming skills in order to create the test specification and perform test procedures.

With the deployment of ASCET V5.0, users of CompareComputing will be able to switch directly from the test report in LABCAR to ASCET's graphical modeling environment. They can then avail themselves of the clearly structured model to backtrack the signal flow all the way to the cause of the problem, and to modify the model specification if the need arises. Several automakers and Tier 1 suppliers have already expressed interest in the CompareComputing approach. In the near future, the AUTOSAR development initiative may be expected to lend even more importance to uncompromising quality assurance for software that is validated on a rapid prototyping system but destined for deployment on a variety of production ECUs.