JÖRN HAASE

On the line

The networking of line-replaceable units is integral to aerospace electronics. A new system is becoming a widely used tool for analysis, simulation, and testing of distributed, embedded systems

BY JÖRN HAASE

Over the past several years, the importance ascribed to aerospace electronics has changed fundamentally. Initially, just a few line-replaceable units (LRUs) were used in aircraft, but innovations were soon increasingly based on electronics, and predominated by a share of functionality residing in software and functional integration.

This rising complexity means that extensive, reproducible and effective tests are more important than ever in all phases of LRU and system development. The widespread use of numerous electronic components causes the number of potential error sources to grow disproportionately. Some weaknesses of the overall system are not revealed until the components are integrated under real conditions during aircraft power-up. This has made testing an interdisciplinary activity that is practiced across departments, manufacturers, and suppliers.

Participants in the aerospace industry have learned lessons from the enormous electronics problems experienced in recent years, and they place great importance on testing. Nonetheless, the complex task of distributed testing can usually be made more reliable by systematic use of available tools.

ANALYSIS, SIMULATION & TESTING OF LRUs

The networking of LRUs forms the backbone of aerospace electronics. In this environment, CANoe from Stuttgart-based Vector Informatik is a widely used tool for analysis, simulation, and testing of distributed, embedded systems. It is often used to implement a remaining bus simulation, and monitor multiple bus topologies simultaneously, and it supports all significant bus systems – in particular AFDX/ARINC 664, ARINC 812/825/826 (CAN), IP, and several others. Any combination of supported bus systems can be used synchronized and in parallel from a single test system. Vector also offers multiple I/O hardware products that supplement general options with test-specific functions such as the ability to apply electrical loads and short circuits directly to the LRU ports or simply to monitor and stimulate discrete lines. In addition, commercially available interface cards can be addressed from CANoe.

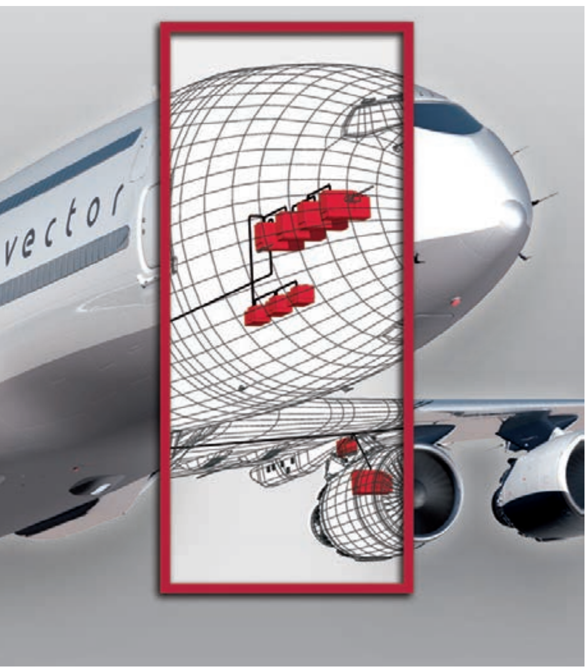
Any standard workplace PC or laptop running under Windows can be used to run CANoe. More powerful test systems with improved real-time capabilities can be set up in a real-time configuration. This approach involves executing the remaining bus simulation and the actual test execution on a dedicated computer under an optimized operating system (Windows XP Embedded), while another dedicated PC is available for the graphic user interface and evaluation. This system configuration can also be used as a test execution environment for a component HIL tester.

INTEGRATION OF TEST & DEVELOPMENT

Today's development models provide tests in various phases of development. In general, the individual tests are considered self-contained and separate activities. In this context, test creation is often also organized as an independent task, detached from other development activities.

This division of work results in a distribution of the many tasks among specialized work groups in the development process. However, if this separation is too strict, the numerous contact points between different development and test tasks cannot be optimally linked. For example, only good coordination between component and system testing can prevent expensive duplicated development of test cases that cover the same content. When compatible tools are used, test cases developed once in the various work areas can serve as the foundation for further developments.

Along with linking the various test phases, development and testing activities should be coordinated with one another. Testing should be understood as an integral part of development. What is important is that the tests should not just be available in the required formal verification phases, but should also support development. Ideally, initial tests could be performed right at the workspace or the LRU developer with the resources available there. For this



ABOVE: Simulation, testing and analysis of avionic bus systems based on CAN and AFDX

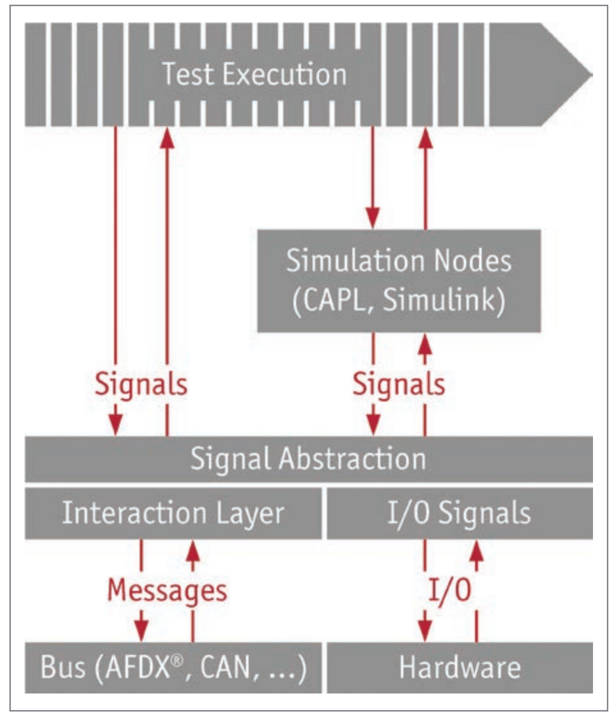
purpose, CANoe offers a runtime environment for test execution that can be used in parallel with the remaining bus simulation and analysis functions.

The remaining bus simulation is an important foundation for the tests. It does not need to be set up manually but should be automatically generated and parameterized from available databases of the system description. The actual work can be performed by so-called modeling DLLs deriving from MATLAB/Simulink – for example, the interaction layer or network management, which are supplied with the tool. Standardized interface description documents can also be used for automated scenario setups. The signals that the remaining bus simulation supplies for the simulated nodes can be acquired directly from the test scripts, stimulated or manually operated.

MATURITY ASSESSMENT & ERROR ANALYSIS

To assess the maturity level of an LRU during development, all executed tests should be comprehensively evaluated. The quality of the individual test results with regard to reliability and relevance is a key feature of a proper verification process. But it is equally important to ensure that suitable tests are used to achieve broad coverage of the required properties. The results of less formally executed tests are helpful for a maturity level analysis. A prerequisite for this is consistent reporting on each test execution.

In each execution of a test with CANoe, whether it is conducted in the test laboratory or at a work bench, a test report is created. The system generates it without intervention by the operator or test case developer, and it is therefore available without additional effort. The XML format of the reports is an open format, so that the report results are available for further processing by other tools. For example, a test management system could evaluate the reports in the context of a maturity level analysis. To ensure a reliable and traceable integration into common development chains, additional DOORS (dynamic object oriented requirements system) modules are available. These modules generate compatible test reports and ensure bidirectional traceability of requirements, test cases, and their results.

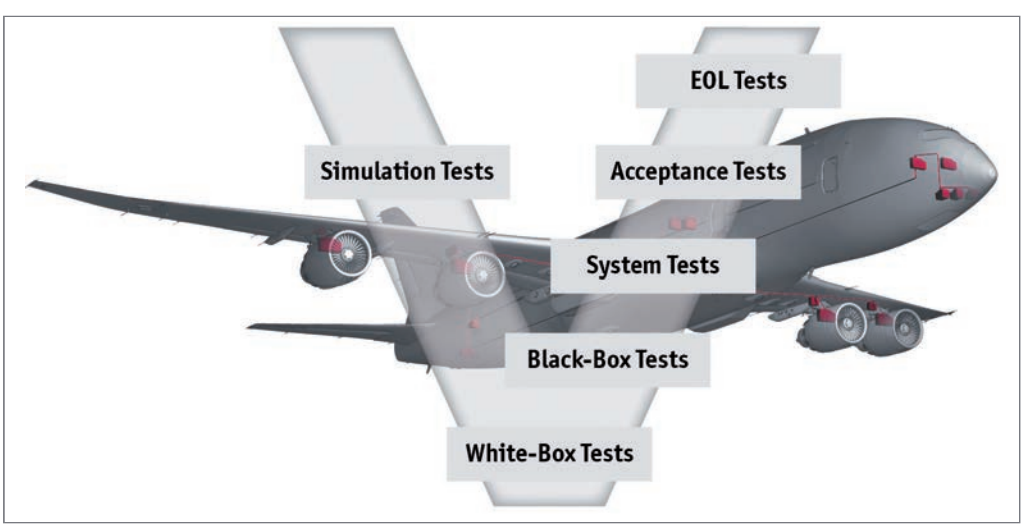


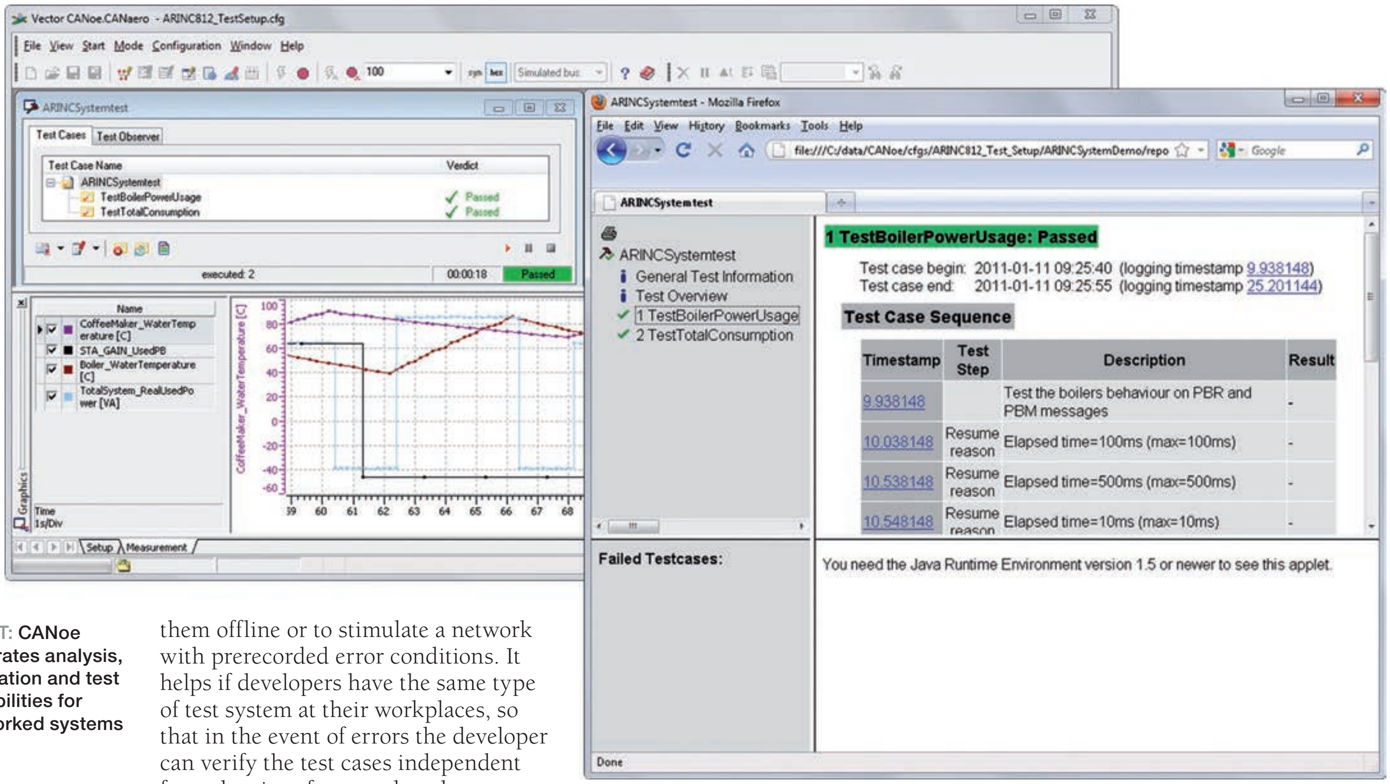
ABOVE: Signals offer an abstraction level between messages and I/O connections, on the one hand, and test definitions and simulation models on the other

At least as important as recording and evaluating the test results is the analysis of the actual error causes. However, most test tools will do this analysis in a rudimentary way at best, partly because error analysis is often considered a separate task, one for developers to do.

Developers are initially confronted with the problem of understanding the errors detected in the test and tracing them to their origins. Especially in the case of errors reported by test laboratories, the developer usually does not even have access to the systems used in the test.

To enable immediate evaluation of the causes of errors it is vital to obtain a precise log of the test flow and every interaction with the device under test, especially the bus communication. In the context of an analysis, the CANoe user can play back any desired recordings (logs) in order to analyze





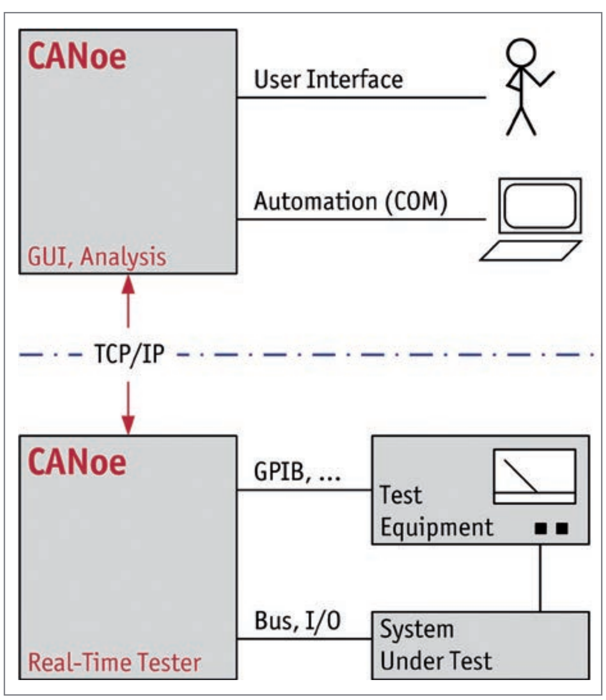
RIGHT: CANoe integrates analysis, simulation and test capabilities for networked systems

BELOW: Two-computer operation of CANoe RT (Real Time) offers improved real-time capabilities

them offline or to stimulate a network with prerecorded error conditions. It helps if developers have the same type of test system at their workplaces, so that in the event of errors the developer can verify the test cases independent from the aircraft or test bench.

SIGNAL ABSTRACTION & DIAGNOSTICS

Increasing functionality in LRUs not only increases the complexity of the systems, but also requires more extensive and complex tests. Selecting the right abstraction level in composing tests reduces the effort (and therefore the costs) required to create the test cases; it also affects the quality of the test cases. Like all other software components, the test cases themselves may even contain errors, and they must be checked before they are used.



In CANoe, the test cases, as well as the analysis options and simulation components, are based on models that are integrated in the form of databases. They might include communication matrices in DBC format for CAN or more general ICD files for various communication means and systems.

Abstraction on the signal level is useful for testing LRU functionality, as is the actual application in the LRU, which is generally placed on a signal abstraction. In the case of ARINC 812/825 (CAN), for example, an interaction layer in the LRU provides the signal abstraction. CANoe also uses an interaction layer; it parameterizes itself from information in network descriptions, which are also used to create the LRU software.

This ensures that the LRU and test environment use the same abstraction layer and are therefore optimally tuned to one another.

At the same time, signal abstraction implements the remaining bus simulation, at least on the protocol level. For example, it ensures that periodic signals are actually transmitted periodically. In testing, this gives the LRU a realistic environment in terms of bus communications. Furthermore, if changes are made to the system's communication matrix, the test cases can continue to be used – usually without modification. With the same application, the abstraction makes it possible to reuse test cases in similar projects.

DIAGNOSTIC & CALIBRATION INTERFACES

Many LRU functions can be tested meaningfully only if deeper access to the LRU is also available. These accesses are provided by the diagnostic and calibration interfaces, which are made via the LRUs' existing bus interfaces. It makes little sense to address these interfaces via simple message sequences, because defined protocols underline the communication. Therefore, suitable abstraction is needed for these interfaces.

Aviation OEMs and suppliers can cope with the growing requirements for LRU tests only by efficient test creation and automatic test execution. With the direct traceability to underlying requirements, even complex duties get manageable. The presented test tool offers a proven solution for implementing test tasks with signal abstraction while incorporating diagnostic, calibration and I/O interfaces. CANoe is a high-performance runtime environment for testing LRUs and networks. The tool permits early creation and execution of tests with a low level of effort beginning right at the developer's workplace. Open interfaces enable users to tightly integrate the tool in a comprehensive test strategy and tool-supported test management. Vector is continually developing CANoe further for use in the noted areas, supporting users with a modern, efficient test platform. ▮

Jörn Haase (Dipl.-Ing.) is the global technical aerospace engineer at Vector Informatik GmbH aviation office in Hamburg, Germany