Functional GUI testing of IVI systems

Software-based embedded systems operated via GUIs are playing an ever more important role in automotive development. Testing both individual software components and the user interface is required.

(Photo: Vector)

In-vehicle infotainment (IVI) systems are part and parcel of today's automotive sector. Their graphical user interfaces (GUIs) serve as the "face to the customer," and externally represent a variety of embedded vehicle applications. Organizations need to adopt and apply measures for quality assurance to the distribution of software-based interfaces, as complex logic (which poses new challenges to the development process) is required to operate embedded systems. Trends such as the expansion of functionality using new software components and flexible adaptation of the user interface can further increase this complexity. As a result, the testing phase of user interface development continues to take on greater importance over time. Verifying IVI systems requires an interdisciplinary approach: one that's capable of handling the complexity resulting from the interplay between software-based user interfaces and embedded systems. To enable efficient testing of the overall system for this purpose, Testplant and Vector have combined their domain tools – Eggplant Functional for GUI tests and CANoe for ECU testing.

Embedded systems are ubiquitous in everyday life today, and are spreading especially quickly in the automotive sector. To operate vehicles, users rely on the functions and data of these systems, which increases the importance of GUIs as the main element of interaction between human and machine.

In vehicles, these interfaces used to be primarily hardware switches but are now being replaced with software implementations because they can be used in a wider variety of ways and can even be easily changed and expanded at runtime. These IVI systems are largely being developed to be the main interface in the vehicle, as they cover an increasing number of functions and access the vehicle on a deep level (Figure 1).

To cope with this increasing importance, functionality must already be ensured during the early stages of development. Frequently, the target hardware is not yet available, or is at least incomplete at that point in time. So, validation must often occur in a purely virtualized environment or on an isolated ECU for each remaining bus simulation.

The interfaces, users want, are characterized by clear and comprehensible visualization of the underlying vehicle applications – despite their complexity – as well as by intuitive and easy operation (i.e., usability). Ultimately, the goal is to make IVI systems as smart as smartphones, which involves highly responsive, touch-based infotainment interfaces with a broad spectrum of functions that ▷

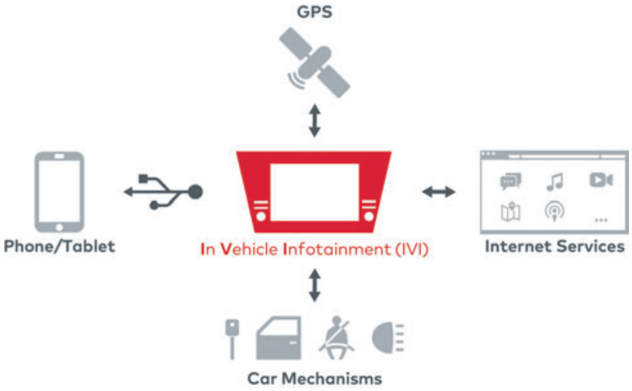


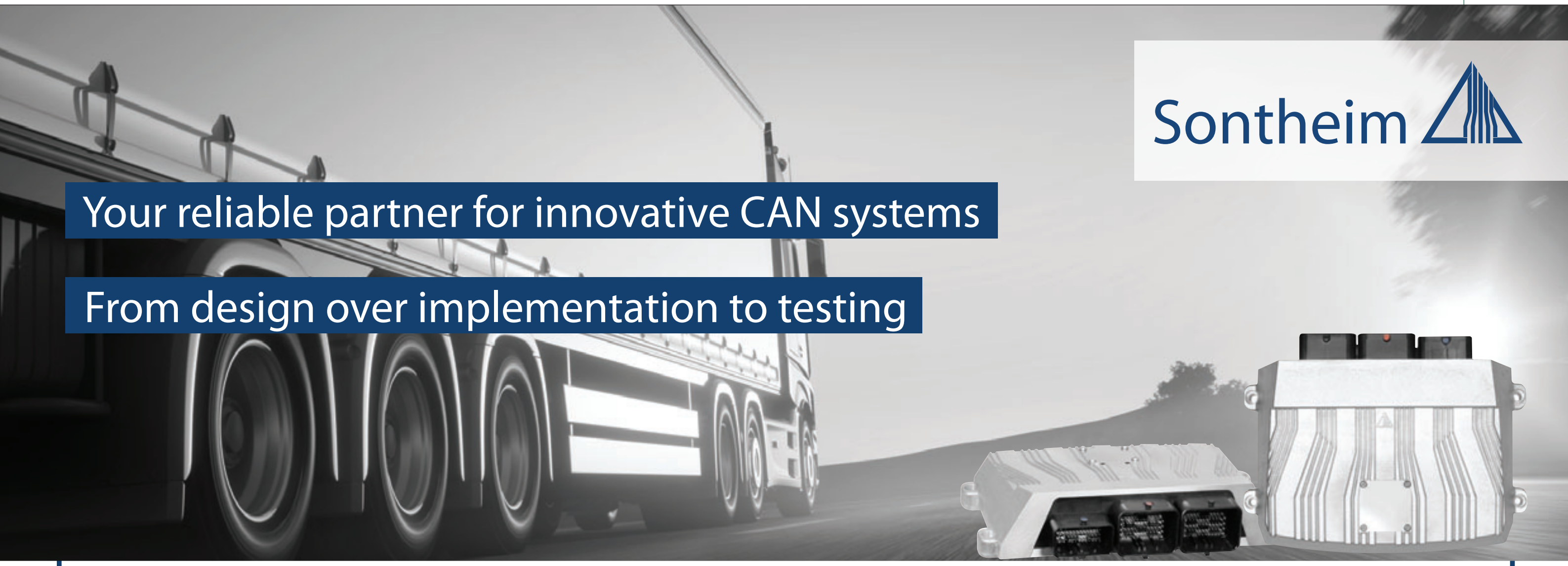
Figure 1: In-vehicle infotainment - the main interface in the vehicle (Photo: Vector Informatik)

also need to be ensured. Customers always expect the technology of the main unit to be state of the art, both in terms of the operating concept and the supported peripherals and audio/video codecs used. The result is a high update frequency which peaks when the car is connected and the software is updated in the field. These framework conditions highlight the requirements for repeatable, automated, and maintainable tests.

An interdisciplinary approach

Satisfying these requirements needs an interdisciplinary approach that combines the tools from both classic GUI testing and ECU testing to ensure comprehensive verification in every stage of development. Eggplant Functional, the software from Testplant for functional GUI testing, enables embedded software applications to be tested based on their user interfaces. With respect to testing on the program code or function level, a greater depth of testing is ensured here. Test automation draws on image and text recognition algorithms to detect buttons and indicators. Should a button be activated, for example, its position is identified through image recognition, and a button press initiated. The transfer of screen content and the initiation of user interactions occur on the system level through remote-control mechanisms such as Virtual Network Computing (VNC), Remote Desktop Protocol (RDP), and Keyboard-Video-Mouse (KVM) over IP. VNC and RDP are protocols common in the PC domain for transferring screen content and keyboard and mouse input. KVM switches typically rely on the hardware level for implementation of the same functionality in order to transfer the input and output information. Changes to the software to be tested, which are made to ensure testability, are thus unnecessary. A non-invasive approach like this ensures that changes, which are otherwise only carried out to test the software, have no effect on the actual test. Since remote-control mechanisms can now be found in every commonly available operating system (Android, iOS, Linux, QNX, VxWorks, Windows, etc.), any ECUs can be tested using this approach. Eggplant Functional runs as a distributed system, and the host application with the test sequence control is executed on a conventional PC for this purpose. Only the remote-control component is required on the ECU to be tested. Using

▷



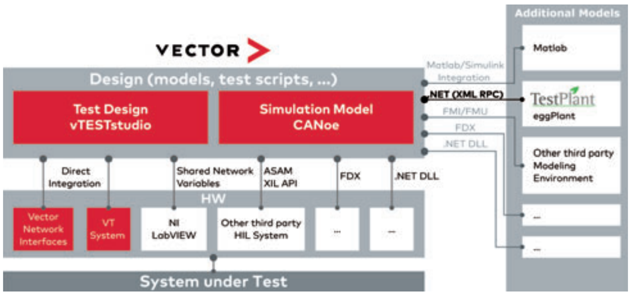


Figure 2: CANoe as an open environment (Photo: Vector Informatik)

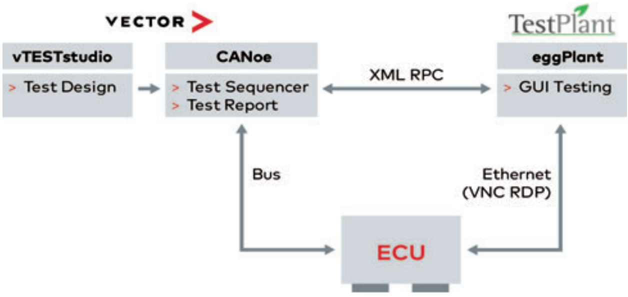


Figure 3: A chain of tools for integrated, functional GUI testing (Photo: Vector Informatik)

various interfaces, Eggplant Functional can interact with test automation systems of other manufacturers and is thus designed as an open system.

The test environment

The CANoe software from Vector enables remaining bus simulation for the ECU to be tested and the analysis of bus sizes. It also provides a testing environment which includes test sequencers and test reporting. The tests themselves are created by Vector in vTeststudio, the authoring tool for editing test sequences for embedded systems. Tabular, graphical, and programming language-based (CAPL/C#) test notations and design methodologies are available for this purpose. CANoe (CAN open environment) was designed explicitly as an "open environment" and offers a host of interfaces for linking external programs (Figure 2). This includes, among others, Vector's own UDP-based Fast Data Exchange (FDX) interface, the Asam XIL API standard, the Functional Mockup Interface (FMI) and Matlab. External DLLs can also be integrated, an option used to connect Eggplant Functional to CANoe.

Eggplant Functional provides an access option for external applications over an XML RPC interface and is thus run as Eggdrive in a mode without a GUI. The interface makes it possible to call functions and test scripts, as well as to read back partial, individual results or the entire test report. Testplant already offers an implementation in the form of a .NET assembly for use of the XML RPC interface. This enables linking to CANoe/vTeststudio (Figure 3). Encapsulating the necessary calls in a C# test library and reproducing the test results in the CANoe test report yields an easy-to-use interface for the combination of CANoe and Eggplant Functional. The test designer creates the ECU tests in the vTeststudio environment they are familiar with and draw on the functions of Eggplant Functional in the same way as the scope of functions belonging to CANoe.

The benefits

The integrated approach from Vector and Testplant provides the user with a testing environment in which to monitor and stimulate the GUI of the ECU, as well as to stimulate and analyze the values transferred on the vehicle bus. From the point of view of the test developer, the level on which they access the information does not play a role.

Because it is also possible to initiate existing Eggplant Functional test scripts, the scripts available from the pure user interface test can be reused in an integrated, functional GUI test. In the combined use case, domain specialists can also use the tools already familiar and available to them. Such an approach increases the testing depth and degree of test automation, and also ensures deep and comprehensive functional testing of the ECU, even in the event of more frequent software releases. Such a test setup makes it possible to already ensure an isolated ECU, including the user interface during the early stages of development. ECUs which communicate with the ECU to be tested are simulated by CANoe and do not actually need to be present here. Functional GUI tests thus enable functionality to be ensured with a transparent hardware expenditure from very early on, which is also ideal for creating a development environment with continuous testing. It also provides the option for daily builds, for example, and enables early responses to problems in new software releases—facilitating frequent software releases and short development cycles, and ensuring a timely response to new requirements or framework conditions.

Conclusion

Software-based embedded systems operated via GUIs are playing an ever more important role in automotive development. To satisfy quality assurance, testing both individual software components and the user interface is required. Thanks to their open interfaces, Eggplant Functional and CANoe test automation tools are ideal to automatically verify the functionality of embedded systems — completely and reliably. Domain specialists can implement tests in their preferred environment and integrate them into an overall test suite — eliminating expenses associated with creating additional tests. This, in turn, enables shorter development cycles with frequent software releases, and a competitive advantage for the OEM or supplier. ◀

Authors

Sebastian Mezger
Vector Informatik
info@vector.com
www.vector.com

Mingzhi Deng
Testplant
sales@eggplant.io
www.testplant.com

EVERYONE CAN

Introducing the new ValueCAN 4 family of tools.

- Great value at a great price - starting at less than €200
- Options for **CAN FD, CAN, Ethernet, and LIN**
- Rugged aluminum case and heavy duty strain relief
- Sealed body and connector
- Small size, easy to keep with your laptop
- Real-time acceleration, bus isolation, high-speed flashing, standalone operation, USB-powered

Read more: www.intrepidcs.com/vcan4

INTREPID CONTROL SYSTEMS GMBH

USA UK Germany Japan Korea China India Australia

+49 (0)721 6633703 –4 icsgermany@intrepidcs.com

IntrepidCS