

EFFICIENT FUNCTIONAL TESTING OF FLIGHT ATTENDANT PANELS

Verification of Flight Attendant Panels (FAPs) requires an interdisciplinary approach. TestPlant and Vector have combined their domain tools for efficient testing of the entire system

// SEBASTIAN MEZGER AND MINGZHI DENG

Ever since Flight Attendant Panels (FAPs) were first introduced in the Airbus A320 series, the number of checking and monitoring functions which they control has grown regularly (Figure 1). Improved touchscreen technologies are continually making the human-machine interface more efficient and convenient. In modern aircraft, the crew uses FAPs to control and monitor many cabin functions, among them lighting, announcements, door status indication, smoke detection and temperatures. The units are also used for functions relevant to maintenance, for instance to add entries to the digital cabin logbook used to log faults. Furthermore, the FAP indicates safety information such as smoke detection or emergency signals. Easy, efficient and reliable operation of FAPs with graphic user interfaces plays an important role in the airlines' satisfaction with the units.

Quality assurance measures must be adapted to meet the increasing prevalence of software-based user interfaces. Complex logic must be applied, especially in the operation of embedded systems, which

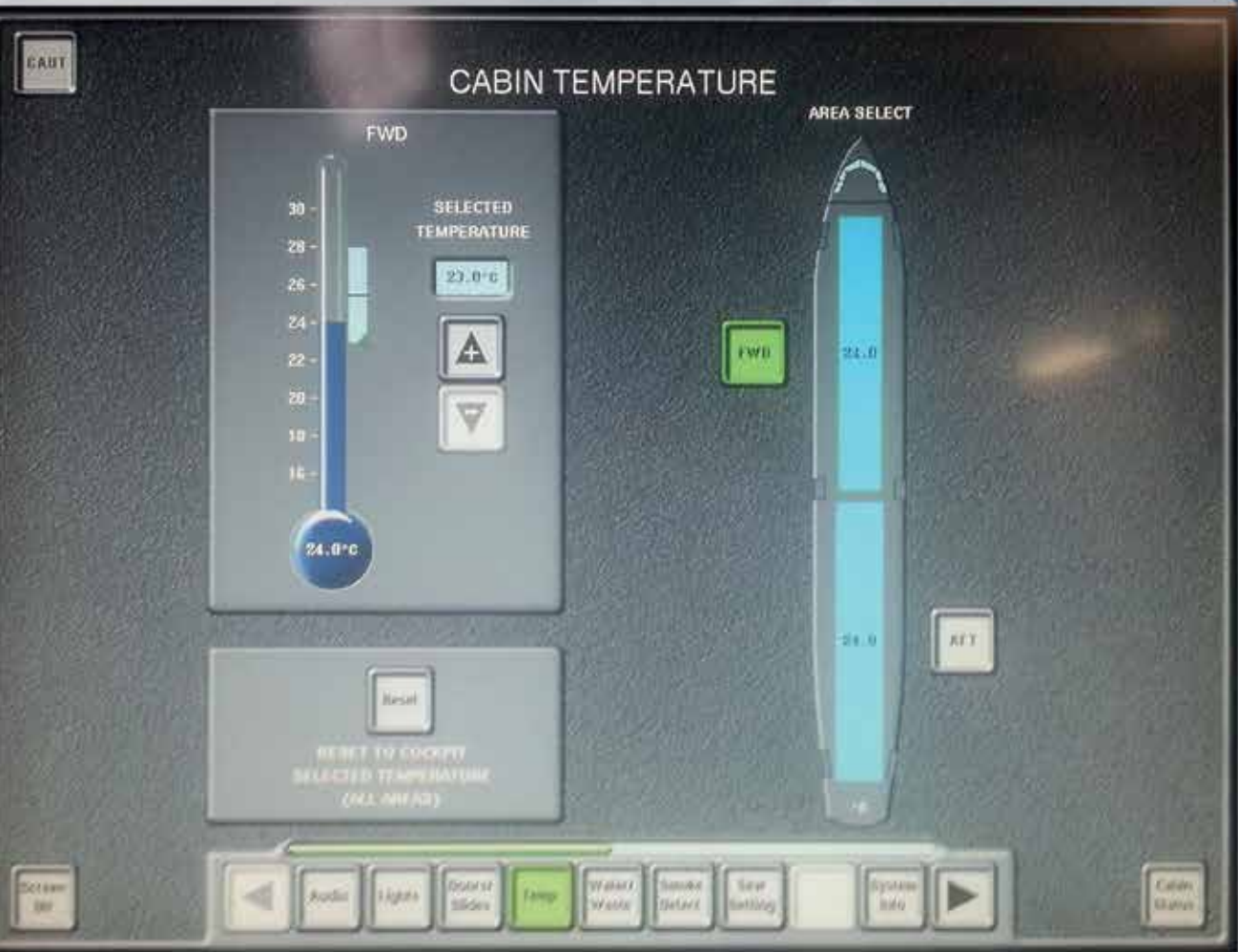
pose new challenges to the development process. Trends such as extending functionality by adding new software components or flexibly adapting the user interface can further increase this complexity. Consequently, the test phase is assuming growing importance in the development of user interfaces.

To properly address this increasing importance, it is necessary to validate functions in the early phases of development. Often the target hardware is still unavailable or at least incomplete by the time testing needs to start. Frequently, the validation must be performed in a purely virtualized environment or on an isolated subsystem using remaining bus simulation.

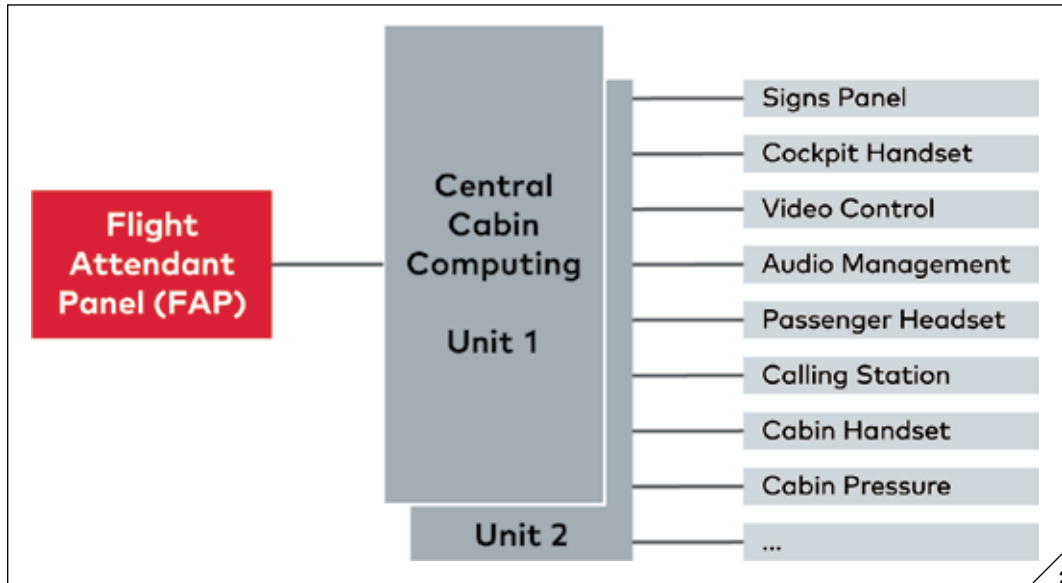
Desirable user interfaces are characterized by visualizing the underlying cabin applications in a way that is both clear and easy to understand – despite their complexity – and by offering intuitive and simple operation (usability). The challenge is to design FAP systems to be as smart as smartphones. This leads to highly responsive touch-based infotainment user



“THE CHALLENGE IS TO DESIGN FAP SYSTEMS TO BE AS SMART AS SMARTPHONES”



1 // Flight Attendant Panels, an essential part of modern aircraft, need reliable quality assurance and testing before aircraft test flights begin



2

interfaces with a broad range of functions, and these must also be validated. The customer always expects the FAPs to show state-of-the-art technology in its operating concept and supported periphery. The units must also lend themselves to reliable software updates at regular intervals. In turn, these constraints place stringent requirements on testing that should be repeatable, automatable and maintainable.

INTERDISCIPLINARY APPROACH

To satisfy these requirements, an interdisciplinary approach is needed that combines the tools from both GUI and ECU testing in order to assure comprehensive verification in all development stages.

eggPlant Functional, the software from TestPlant for functional GUI tests, enables testing of embedded software applications via the user interface. This approach assures greater testing depth compared to testing on the program code or functional level. Here, test automation utilizes image and text recognition algorithms to detect switch surfaces and displays. For instance, if a surface switch should be activated, its position is identified by image recognition, and a button push is initiated. Remote control mechanisms such as Virtual Network Computing (VNC), Remote Desktop Protocol (RDP) or Keyboard-Video-Mouse (KVM) over IP (Internet Protocol) are used on the system level to transfer screen contents and initiate user interactions. VNC and RDP are protocols commonly used in the PC area to transmit screen contents as well as keyboard and

mouse inputs. KVM switches are typically located on the hardware level to implement the same functionality in transmitting input and output information. This makes it unnecessary to modify the testing software to achieve testability.

Such a non-invasive approach ensures that changes which would otherwise only be performed to test the software do not have any effects on the actual test. Any ECUs can be checked using this approach, because remote control mechanisms now exist in all commonly used operating systems (Android, iOS, Linux, QNX, VxWorks, Windows, etc.). eggPlant Functional is operated as a distributed system, in which the host application with test flow control is executed on a conventional PC. Only the remote control component is needed at the ECU under test. By using different interfaces, eggPlant Functional can interact with the systems of other test automation suppliers, and has been designed to be an open system.

CANoe (oe = open environment) software from Vector handles the remaining bus simulation for the ECU under test and the analysis of bus parameters. In addition, it provides a test environment that includes a test sequencer and test reporting. The tests themselves are created in the Vector tool vTESTstudio, an authoring tool for editing test flows for embedded systems. Programming language based methods (CAPL/C#), spreadsheet-based methods or graphic test notation and test design methods are available for this purpose. CANoe was

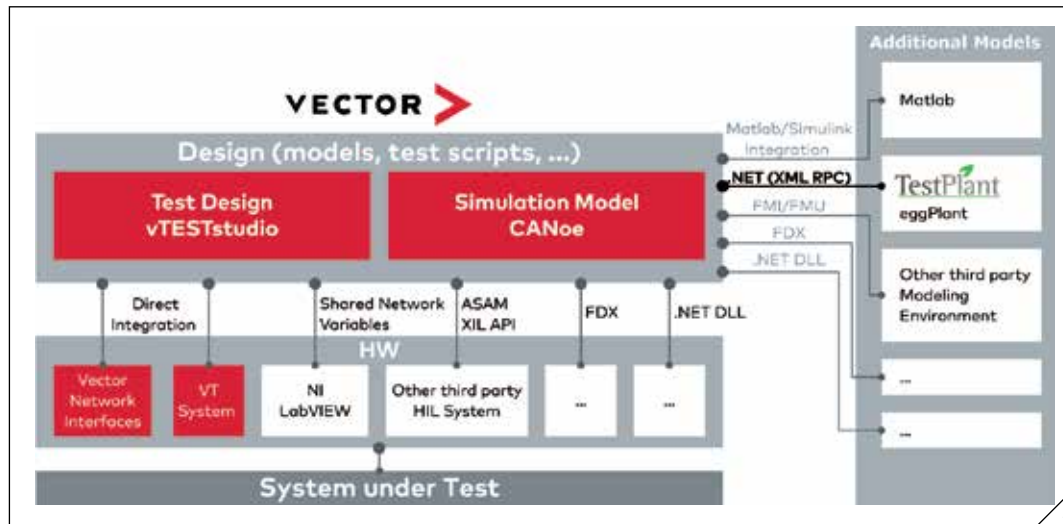
2 // The FAP serves as a central control and display element for a wide variety of cabin functions

explicitly designed to be an open environment, and it offers numerous interfaces for linking to external programs (Figure 2). These include Vector's own UDP-based Fast Data Exchange (FDX) interface, the Functional Mockup Interface (FMI) and MATLAB. It is also possible to integrate external DLLs. The latter is used to link eggPlant Functional to CANoe. eggPlant Functional offers access to external applications over an XML-RPC interface (Extensible Markup Language Remote Procedure Call); this is executed by eggDrive in a mode without a graphic user interface.

The interface lets CANoe call functions and test scripts in eggPlant and read out individual (partial) results of the overall test report. TestPlant offers an implementation in the form of a .NET assembly for use of the XML-RPC interface. This makes it possible to link to the combination of CANoe and vTESTstudio (Figure 3). Encapsulation of the necessary calls in a C# test library and mapping of the test results to the CANoe test report results in an easy-to-use interface for combining CANoe and eggPlant Functional. The test designer creates the ECU tests in a familiar environment such as vTESTstudio and accesses the functions of eggPlant Functional in the same way as the functions of CANoe are accessed.

BENEFITS

The integrated approach of Vector and TestPlant gives the user a test environment in which, in addition to the ability to



3

stimulate and analyze parameters transmitted over the databus in the airplane, the FAP's graphic user interface can also be monitored and stimulated. From the perspective of the test developer, the level on which the information is accessed does not matter. Similarly, one advantage of CANoe is that testing is performed independently of the bus physics. It is only necessary to interchange the hardware interface to match the bus system being used. Consequently, the same models can be used to test the next generation of FAPs, even if a different bus system is used.

Since it is also possible to run existing eggPlant Functional test scripts, scripts from the pure user interface test can be used again in an integrated functional GUI test. The tools that the domain specialists already have and are familiar with can also be used in a combined application case. Such an approach increases the depth of testing or degree of test automation and ensures in-depth and extensive functional testing of an ECU, including in cases where there are frequent software releases. Even in early development stages, such a test setup enables validation of an isolated ECU including the user interface. CANoe is used to simulate ECUs that communicate with the ECU under test and do not need to be present in real form. Functional GUI tests thereby enable validation with modest hardware resources at a very early time.

This test setup is also ideally suited for creating a development environment with continuous testing. It can be used to

3 // CANoe offers an open environment with various interfaces to third-party tools

4 // Toolchain for integrated functional GUI testing of a FAP

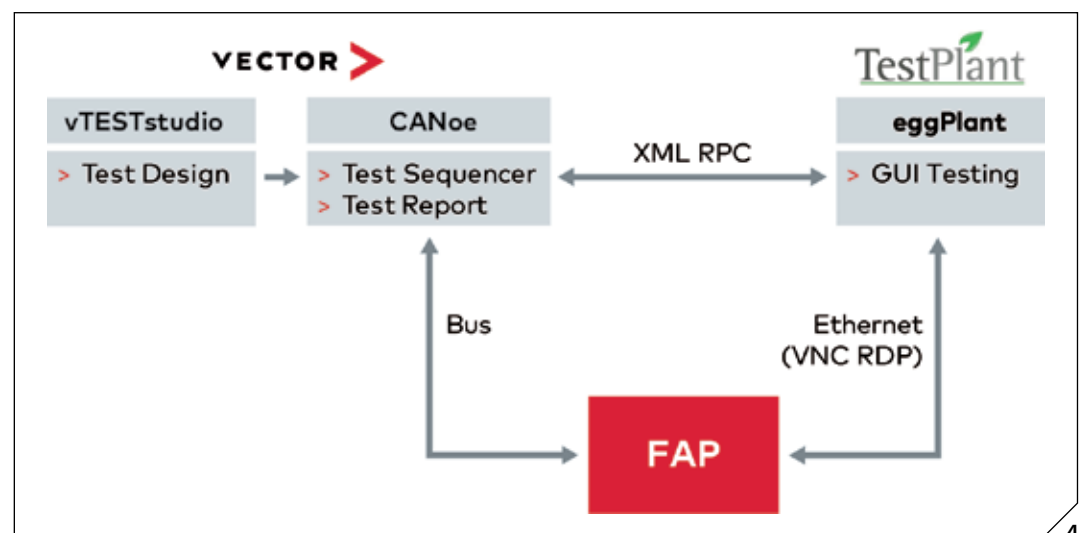
validate daily builds, for example, enabling quick responses to any problems in new software releases. It lends itself to frequent software releases and short development cycles and ensures timely reactions to new requirements and constraints.

SUMMARY

Software-based embedded systems that are operated via graphic user interfaces are becoming increasingly more important in airplane development. To meet quality assurance requirements, it is not only necessary to test the individual software components but also their user interfaces. Therefore, the goal is to verify the functionality of the embedded systems completely and reliably in automated

testing. The test automation tools eggPlant Functional and CANoe come into play here. With their open interfaces they represent a simple tool combination that is ideal for such testing. Domain specialists can implement tests in their familiar environment and then integrate the tools into an overall test suite. This eliminates the effort involved in creating additional tests. It also enables shorter development cycles with frequent software releases, and this produces a clear competitive advantage for the aircraft manufacturer and the supplier. \\\

Sebastian Mezger is team and project leader with Vector Informatik, and Mingzhi Deng is a technical consultant at TestPlant



4