

EFFIZIENTES TESTEN KOMPLEXER MECHATRONISCHER SYSTEME

Prof. Dr.-Ing. habil. K.-D. Thoben, Dr.-Ing. C. Hans, D. Gerke, BIBA - Bremer Institut für Produktion und Logistik GmbH, Hochschulring 20, 28359 Bremen, Deutschland
J. Strahmann, W. Geiwiz, M. Neuhaus, Airbus Operations GmbH, Airbus-Allee 1, 28199 Bremen, Deutschland

Zusammenfassung

Das Bremen Technology Center (BreTeCe) ist ein vom Land Bremen und der EU gefördertes FuE-Vorhaben, das in Kooperation mit der Airbus Operations GmbH die technologischen Grundlagen für innovative Testsysteme erforscht. Die sich in der Entwicklung befindenden Methoden und Werkzeuge sollen Ingenieure unterstützen, Testkampagnen effizienter zu erstellen und wiederzuverwenden sowie Testsysteme miteinander zu vernetzen. Dieser Artikel gibt den derzeitigen Projektstatus wieder, indem er am Beispiel des High Lift Testprozesses die Herausforderungen beim funktionalen Testen komplexer mechatronischer Systeme darstellt, die sich daraus ergebenden Anforderungen benennt und Entwicklungspotenziale aufzeigt.

1. TESTEN MECHATRONISCHER SYSTEME

Mechatronische Systeme setzen sich zusammen aus mechanischen, elektronischen und informationstechnischen Komponenten. Sie werden in die Funktionsgruppen Grundsystem, Sensoren, Informationsverarbeitung und Aktoren unterteilt [1]. Um das spezifikationsgemäße Verhalten eines mechatronischen Systems zu überprüfen, also das korrekte Zusammenwirken aller Funktionsgruppen nachzuweisen, wird ein mechatronisches System einer Vielzahl sogenannter Funktionstests unterworfen [2]. Mittels dieser Tests wird sichergestellt, dass ein mechatronisches System die an es gestellten Anforderungen hinsichtlich Funktionalität, Zuverlässigkeit und Betriebssicherheit erfüllt.

Bei einem Funktionstest wird das Ist-Verhalten einer Komponente oder eines Systems mit seinem spezifizierten Soll-Verhalten verglichen. Ein Testablauf ist dabei zumeist nach einem Reiz-Reaktions-Schema aufgebaut, d. h. der Prüfling bzw. das System unter Test wird, während es sich in einem definierten Zustand befindet, einem Stimulus ausgesetzt und muss darauf den Anforderungen entsprechend reagieren.

Bei Funktionstests wird zwischen zwei Strategien unterschieden: Dem Whitebox-Testen und dem Blackbox-Testen. Während ein Whitebox-Test die interne Funktionsweise einer Komponente oder eines Systems kontrolliert, begutachtet ein Blackbox-Test die Interaktion eines Testobjekts mit seiner Umwelt, d. h. ein Blackbox-Test fokussiert sich auf die Überprüfung von Funktionen, indem er nur deren Eingabe- und Ausgabeparameter respektive deren Werte betrachtet.

BreTeCe adressiert das funktionale Blackbox-Testen komplexer mechatronischer Systeme. In Kapitel 2 wird der High Lift Testprozess bei Airbus Bremen vorgestellt. Im Kapitel 3 werden die Herausforderungen des funktionalen Blackbox-Testens im Testprozess aufgezeigt. Um

diesen Herausforderungen zu begegnen, werden in Kapitel 4 die Anforderungen an einen effizienten Testprozess beschrieben. Kapitel 5 erläutert schließlich die wichtigsten Lösungsansätze von BreTeCe.

2. HIGH LIFT TESTPROZESS

Airbus Bremen entwickelt und produziert Hochauftriebssysteme. Hochauftriebssysteme steuern in einem Flugzeug das Ein- und Ausfahren der Vorflügel und Landeklappen. Das Verfahren der Vorflügel und Landeklappen dient insbesondere dazu, während des Starts und der Landung einen höheren Auftrieb zu erzeugen. Der höhere Auftrieb wird für kürzere Start- und Landestrecken benötigt.

Der Entwicklungs- und Testprozess bei Airbus Bremen richtet sich nach dem V-Modell [3]. Beim V-Modell ist jedem Entwicklungsschritt eine entsprechende Verifikations- bzw. Validierungsaktivität (V&V) zugeordnet. Bei Airbus gibt es die in der Luftfahrt typischen V&V-Ebenen: Equipment, System und Aircraft [4].

Auf Equipment-Level kommen in der Regel entweder bereits getestete „Commercial off the Shelf“-Komponenten (COTS) zum Einsatz oder die Testaufgaben werden von Airbus als Systemhersteller an externe Dienstleister ausgelagert. Die spezifikationsgemäße Umsetzung zugelieferter Module kontrolliert Airbus anhand der Testdokumentation sowie durch die nachfolgenden Tests auf System-Level.

Tests auf System-Level ermöglichen erstmals die Überprüfung komponentenübergreifender Funktionen. Airbus Bremen testet als Systemhersteller insbesondere auf System-Level. Details dazu folgen in den Abschnitten „Testvorbereitung“, „Testdurchführung“ und „Testnachbereitung“.

Das Testen des Hochauftriebssystems auf Aircraft-Level

wird schließlich in Toulouse durchgeführt. Dieses erfolgt zunächst „on ground“ unter Nutzung eines sogenannten „Iron Bird“. Ein „Iron Bird“ ist ein Mock up, der alle primären Flugzeugsysteme im Zusammenspiel testet (Antrieb, Steuerung, Hochauftrieb, usw.). Wurden diese Tests erfolgreich abgeschlossen, kann ein Flugzeug „in flight“ erprobt werden.

Airbus Bremen fokussiert sich bei seinen Testaktivitäten auf den System-Level, d. h. auf das Testen eines Hochauftriebssystems als integriertes Gesamtsystem. Den Schwerpunkt bilden die Verifizierung der Funktionen des Slat Flap Control Computers (SFCC) [5] sowie die Überprüfung der Signalübertragung zwischen dem SFCC und der Sensorik/Aktorik. Prinzipiell gestaltet sich der Testprozess auf System-Level genauso wie auf den anderen V&V-Ebenen: Es gibt eine Testvorbereitung, eine Testdurchführung und eine Testnachbereitung.

2.1. Testvorbereitung

In der Testvorbereitung wird zunächst eine geeignete Teststrategie gewählt. Airbus Bremen testet nach der Blackbox-Strategie. Dieser Ansatz gewährleistet eine möglichst große Unabhängigkeit zwischen dem Entwicklungsbereich und dem Testbereich. Gemeinsame Basis beider Bereiche sind die Anforderungen. Sie sind der Ausgangspunkt sowohl für den Entwicklungs- als auch für den Testprozess. Das Ziel der Testvorbereitung ist es, eine möglichst adäquate Menge an Testfällen aus den Anforderungen abzuleiten.

Ein Team von Ingenieuren, welches fundierte Kenntnisse über das Hochauftriebssystem und den SFCC besitzt, ist bei Airbus Bremen für die Definition der Testfälle verantwortlich. Eine Testdefinition beschreibt in einem kurzen, natürlichsprachlichen Text ein Testszenario sowie die Zustände, die es zu überprüfen gilt, sprich die das zu testende System in dem Szenario annehmen soll. Eine Testdefinition wird darüber hinaus durch das Team mit einer Reihe von Anforderungen verlinkt, die durch den Test abgedeckt werden.

Ein zweites Team codiert anschließend auf Basis der Testdefinitionen sogenannte Testprozeduren. Testprozeduren enthalten konkrete Anweisungen für den Aufbau und den Ablauf eines Tests nach dem Reiz-Reaktion-Schema. Testsystemspezifische Anweisungen und Befehle enthalten die Testprozeduren bei Airbus nicht mehr. Von einer hardwarenahen Beschreibung wird mittels einer Hochsprache abstrahiert. Die Hochsprache ermöglicht die Notation funktionaler Testprozeduren in einer für den Menschen lesbaren Form unter Angabe von Systemzuständen, parametrisierten Einflussgrößen und des erwarteten Systemverhaltens einschließlich Reaktionszeiten. Die Testprozeduren können mittels eines Compilers in maschinenlesbare Skripte überführt werden. Der Compiler kann an die Eingabeformate verschiedener Testanlagen angepasst werden, so dass sich die Testprozeduren wiederverwenden lassen.

2.2. Testdurchführung

Airbus Bremen nutzt für das funktionale Testen von Hochauftriebssystemen zwei Arten von Testsystemen: Zum einen Real Time System Simulatoren und zum anderen hydromechanische Prüfstände, sogenannte Rigs. Die Testdurchführung ist durch den Einsatz dieser Anlagen sowie durch die Anwendung von „Software und Hardware in the Loop“-Verfahren (SiL und HiL) bereits weitgehend automatisiert. In Kürze stellt sich der Testablauf wie folgt dar: Die für einen Testlauf benötigten Informationen (Konfigurationsdaten, Testskripte, Simulationsmodelle, usw.) werden zunächst auf das Testsystem übertragen. Dann wird das Testsystem initialisiert. Anschließend arbeiten die Anlagen ein Testskript Schritt für Schritt in Echtzeit ab. Nach Beendigung des Testlaufs werden die Log-Dateien sowie die erzielten Testergebnisse in einer entsprechenden Form (ASCII-Dateien, Datenbank) bereitgestellt.

2.3. Testnachbereitung

Die Testnachbereitung beinhaltet bei Airbus Bremen vor allem die Analyse, Visualisierung und Auswertung einzelner Testläufe. Signale und Messwerte einzelner Parameter eines Testlaufs können bei der Auswertung einzeln betrachtet oder mit Signalen und Messwerten anderer Parameter beliebig kombiniert und aggregiert werden.

Der Testerfolg orientiert sich an der erreichten Testabdeckung, d. h. wie viele Funktionen verifiziert werden konnten. Hierüber werden ausführliche Berichte, sogenannte Test-Reports angefertigt. Die Nachverfolgbarkeit der Anforderungen, also die Abbildung einer oder mehrerer Anforderungen auf einen einzelnen oder einer Menge von Tests, wird durch das eingesetzte Testmanagementsystem gewährleistet.

3. HERAUSFORDERUNGEN IM TESTPROZESS

Der bei Airbus Bremen etablierte Testprozess gewährleistet bereits eine sehr hohe Betriebssicherheit und Zuverlässigkeit für Hochauftriebssysteme. Die folgenden Punkte adressieren deshalb vor allem Effizienzschwächen im Prozess.

3.1. Vielzahl von Testfällen

Der Nachweis über die Funktionalität eines mechatronischen Systems lässt sich in aller Regel nicht auf einem formalen, analytischen Wege erbringen. Bspw. bedingt die Einbindung von analoger Signalen und Schnittstellen in die Flugsteuerung (Fly-by-wire) einen unendlichen Zustandsraum der Systemfunktionen. Folglich kann Airbus oder ein anderer Flugzeughersteller keinen mathematischen Beweis für die Korrektheit der Funktion „Flugsteuerung“ bzw. „Hochauftrieb“ liefern. Der Nachweis über die spezifikationsgemäße Herstellung eines Flugzeugs oder eines komplexen Subsystems, wie dem

Hochauftriebssystem, kann deshalb nur über Tests erfolgen. Die Menge der möglichen Testfälle ist dabei neben der Anzahl der zu testenden Systemfunktionen auch von der Variabilität der Systemumgebung und den spezifizierten Anforderungen abhängig. Unter der Annahme, dass aufgrund begrenzter zeitlicher und monetärer Ressourcen niemals alle Funktionen unter allen Bedingungen getestet werden können, wird eine Untermenge von Testfällen ausgewählt. Allerdings ist gerade für sicherheitskritische Systeme diese Auswahl problematisch, da sich sicherheitskritische Systeme jederzeit in einem definierten Systemzustand befinden müssen. Aufgrund geringer Eintrittswahrscheinlichkeiten bestimmte Systemzustände einfach zu vernachlässigen, ist hinsichtlich einer Systemverifizierung keine zufriedenstellende Lösung. Um den strengen Sicherheitsanforderungen dennoch gerecht zu werden, wird in der Luftfahrt das umfangreiche und wiederholte Testen der Systeme mit anderen Strategien kombiniert, wie z. B. der mehrfach redundanten Systemauslegung. Während Redundanzen in den Systemen also bekannt und gewollt sind, werden bei den Testfällen ungewollte Redundanzen im niedrigen zweistelligen Prozentbereich vermutet. Es bleibt festzuhalten:

- Die Korrektheit von komplexen mechatronischen Systemen lässt sich nicht beweisen.
- Bei komplexen mechatronischen Systemen können niemals alle Funktionen unter allen Bedingungen getestet werden.
- Redundanzen in den Testkampagnen sind unerwünscht. Sie kosten Zeit und Geld und binden Ressourcen.

3.2. Multifunktionalität

Die Zunahme der Komplexität mechatronischer Systeme lässt sich insbesondere mit der fortschreitenden Integration und der daraus resultierenden Multifunktionalität heutiger Systeme begründen. Multifunktionalität bedeutet, dass Funktionen eines Gesamtsystems nicht mehr nur durch ein System sondern durch das Zusammenwirken mehrerer Systeme realisiert werden. Im System „Flugzeug“ zeichnet sich die Multifunktionalität bspw. in der zunehmenden Vernetzung der primären Flugsteuerung mit dem Hochauftriebssystem ab. Die durch die Kopplung der Systeme gewonnene Funktionalität und Flexibilität hat allerdings auch einen entscheidenden Nachteil: Es gibt deutlich mehr Schnittstellen, Wechselwirkungen und Abhängigkeiten, die es beim Testen zu berücksichtigen gilt. Allerdings mangelt es an Lösungen, um Multisystemtests frühzeitig durchzuführen. Bestehende Testanlagen sind in der Regel nur für einzelne, lokale Systeme ausgelegt. Häufig handelt es hierbei um Sonderanlagen, die nur wenige, sehr spezielle Aufgaben leisten. Generell mangelt es in diesem Bereich an Standards, was u. a. dazu führt, dass Testsysteme nicht oder nur schwer angepasst, wiederverwendet und miteinander vernetzt werden können. Das Testen einer systemübergreifenden Funktion kann aber erst nach der Integration aller an ihr beteiligten Systeme erfolgen. Ein Ge-

samtsystemhersteller wie Airbus muss dadurch in seiner Rolle als Integrator ggf. mehr Testaktivitäten übernehmen, als es wünschenswert wäre. Es lässt sich resümieren:

- Multifunktionalität führt zu komplexen Testscenarien.
- Heutige Testsysteme sind nicht für Multisystemtests ausgelegt.

3.3. Proprietäre Testsysteme

Testsystemhersteller liefern in der Regel geschlossene Lösungen. Selten werden offene Schnittstellen oder Protokolle verwendet. Die eingesetzten Hardwarekomponenten sowie die Betriebs- und Anwendungssoftware sind in der Regel proprietär. Freie oder den Anwendern vertraute Software kann oftmals nicht eingesetzt werden. Unkomfortable oder fehlende Migrationsmöglichkeiten für Testkampagnen erschweren den Umstieg auf ein System eines anderen Anbieters. Für einen Systemhersteller ergeben sich durch proprietäre Testsysteme folgende Nachteile:

- Unzureichende Transparenz über die Funktionsweise des Testsystems,
- schlechte Übertragbarkeit von Testkampagnen und Anwendungssoftware sowie
- eine erhöhte Gefahr der Abhängigkeit von einem Testsystemlieferanten.

3.4. Aufwendige Testvorbereitung

Bei Betrachtung des High Lift Testprozesses sticht der unterschiedliche Automatisierungs- bzw. Unterstützungsgrad der einzelnen Prozessphasen hervor. Während die Testdurchführung durch SiL- und HiL-Verfahren bereits weitgehend automatisiert und die Testauswertung durch kommerzielle Tools zur Datenverwaltung und -verarbeitung unterstützt wird, erfolgt die Erstellung der Testdefinitionen und Testprozeduren manuell. Dabei ist besonders die Erstellung der Testprozeduren sehr zeitintensiv. Beide Tätigkeiten sind darüber hinaus sehr anspruchsvoll und erfordern hochqualifiziertes Personal. Für die Prozedurerstellung müssen die Mitarbeiter detailliertes Wissen über das zu prüfende System als auch das Testsystem und seine Programmierschnittstelle aufweisen. Bei hochkomplexen Systemen ist der Anwender damit schnell überfordert, was in der Folge zu Umsetzungsfehlern und notwendiger Nacharbeit führt. Die Vorteile einer automatisierten Testdurchführung wie Wiederholbarkeit und Determinismus wird folglich mit einem erhöhten Aufwand in der Testvorbereitung erkaufte. Zusammengefasst heißt das:

- Eine automatisierte Testdurchführung schafft zusätzliche Aufwände in der Testvorbereitung.
- Die Testvorbereitung, insbesondere die Erstellung von Testprozeduren, wird bisher unzureichend unterstützt.

4. ANFORDERUNGEN AN EINEN EFFIZIENTEN TESTPROZESS

Der Testprozess ist durch entsprechende Mechanismen unter Aufrechterhaltung höchster Standards bzgl. Qualität und Robustheit zeitlich zu verkürzen. Eine höhere Zuverlässigkeit der Systeme wird bereits in den frühen Testphasen gefordert. Daher sollten Systeme am besten gleichmäßig über alle Funktionen getestet werden. Die Umsetzung von Normen, wie z. B. DO-178B ist zu fördern.

Um die spezifikationsgemäße Funktionalität eines komplexen mechatronischen Systems sicherzustellen, müssen alle Anforderungen an das System durch eine adäquate Menge von Tests verifiziert werden. Die Definition einer adäquaten Menge von Testfällen ist nicht trivial. Es stellt sich die Frage nach der Relevanz eines Testfalls und wie sich relevante Testfälle ermitteln lassen. Folglich bedarf es der Formulierung und Umsetzung von Entscheidungsverfahren, die Ingenieure zukünftig bei der Identifizierung relevanter Testfälle unterstützen. Darüber hinaus gilt es, ungewollte Redundanzen in den Testfällen aufzudecken bzw. von vornherein zu vermeiden. Auch hierfür werden leicht anwendbare Methoden und Werkzeuge benötigt. Auch sollen aus Zeit- und Kostengründen nicht mehr Tests als notwendig durchgeführt werden, d. h. ein Mechanismus zur fortwährenden Überprüfung der Testabdeckung während einer Testkampagne wäre wünschenswert. Für sicherheitskritische Systeme reicht es ferner nicht aus, ein System nur im Normalbetrieb unter Normalbedingungen zu testen (Positivtests). Unvorhergesehene, nicht definierte oder von der Norm abweichende Einwirkungen blieben dann unberücksichtigt. Um zu gewährleisten, dass ein sicherheitskritisches System niemals einen Menschenleben gefährdenden Zustand einnimmt, müssen auch alle erdenklichen Sonderfälle geprüft werden (Negativtests). Inwiefern es bei dieser komplizierten Aufgabe möglich ist, den Testingenieuren eine informationstechnische Unterstützung anzubieten, ist ebenfalls zu evaluieren.

Die zunehmende Kooperation von Unternehmen im Bereich der Produktentwicklung und Fertigung erfordert neue Strategien im Bereich funktionaler Systemtests, wie z. B. eine Parallelisierung von Prozessschritten. Testaktivitäten müssen zeitlich verkürzt werden und Testsysteme derart aufgebaut sein, dass sie Multisystemtests ermöglichen. Multisystemtests müssen frühzeitiger erfolgen, was im Bereich der Luftfahrt bedeutet, nicht erst auf Aircraft-Level. Voraussetzung für frühzeitige Multisystemtests ist eine vorhergehende, zumindest temporär existierende, Integration von Testanlagen. Hierzu bedarf es einer Technologie, die die räumliche Entfernung zwischen verschiedenen Prüfständen überbrückt und als eine Art Universalübersetzer fungiert. Die Konfigurations- und Managementwerkzeuge für eine integrierte Testlösung müssen einfach gestaltet sein und dürfen sich in ihrer Handhabung nicht von etablierter Anwendungssoftware unterscheiden.

Die Standardisierung von Testsystemen ist voranzutreiben. Offene Systeme und modulare Plattformen mit definierten Schnittstellen sind gefordert. Sie begünstigen die Möglichkeit einer Vernetzung und Clusterbildung von Testanlagen. Die eingesetzte Software muss vom Kernsystem bis zur Anwendungsschicht beliebig erweitert und ausgetauscht werden können. Die Anwender erwarten Plug-in Technologie, wie sie sie bereits von Softwarewerkzeugen aus dem Entwicklungsbereich kennen. Import- und Exportfunktionalitäten für Standarddatenformate werden bei allen Softwarelösungen vorausgesetzt.

Um die Testvorbereitung effizienter zu gestalten, bedarf es eines Ansatzes zur teilautomatisierten Erstellung von Testprozeduren. Der Ansatz muss es ermöglichen, verschiedene Teile einer Prozedur zu generieren, wie bspw. Kontrollstrukturen oder Methodenrumpfe. Ferner wäre es wünschenswert, wenn aus einer formalisierten Testdefinition weitere Fragmente einer Prozedur generiert bzw. automatisiert übernommen werden könnten. Es sollen Notationen eingesetzt werden, die das Verständnis der Benutzer erhöhen und ihnen helfen Fehler zu vermeiden.

5. BREMEN TECHNOLOGY CENTER

BreTeCe hat das Ziel, den Aufwand für das Testen komplexer mechatronischer Produkte zu verringern und dadurch den Testprozess zeitlich zu verkürzen. Der verfolgte Lösungsansatz berücksichtigt technische, methodische und organisatorische Aspekte und führt diese zu einem integrierten Ansatz zusammen. Die Schwerpunkte dabei bilden:

- die methodisch gestützte Optimierung einzelner Prozessschritte.
- Technologien und Werkzeuge zur Errichtung einer Infrastruktur als Basis für verteilte Testumgebungen.
- eine offene, modulare Testplattform, die die standardisierte Einbindung von Testequipment und Softwaremodulen unterstützt.

5.1. Methodengestützter Testfallgenerator

Die Definition von Testfällen obliegt dem Testingenieur. Anhand seines Test-Know-Hows, den Anforderungen und seiner Kenntnisse über das zu testende System überlegt er sich in einem intellektuellen Prozess mögliche Testfälle. Dabei verfolgt er Best-Practice-Ansätze, wie z. B. das die Menge der Testfälle zu je einem Drittel alte, neue und sicherheitskritische Funktionen adressiert.

BreTeCe möchte Ingenieure bei ihrer Tätigkeit unterstützen und im Rahmen einer Prozessoptimierung die Spezifikation relevanter Testfälle effizienter gestalten. Dafür wird ein prototypisches Werkzeug basierend auf der Klassifikationsbaummethode entwickelt [BILD 1]. Die Klassifikationsbaummethode ermöglicht die Ermittlung konkreter Testdaten für eine zu testende Systemfunktion unter Ausschluss von Redundanzen [6].

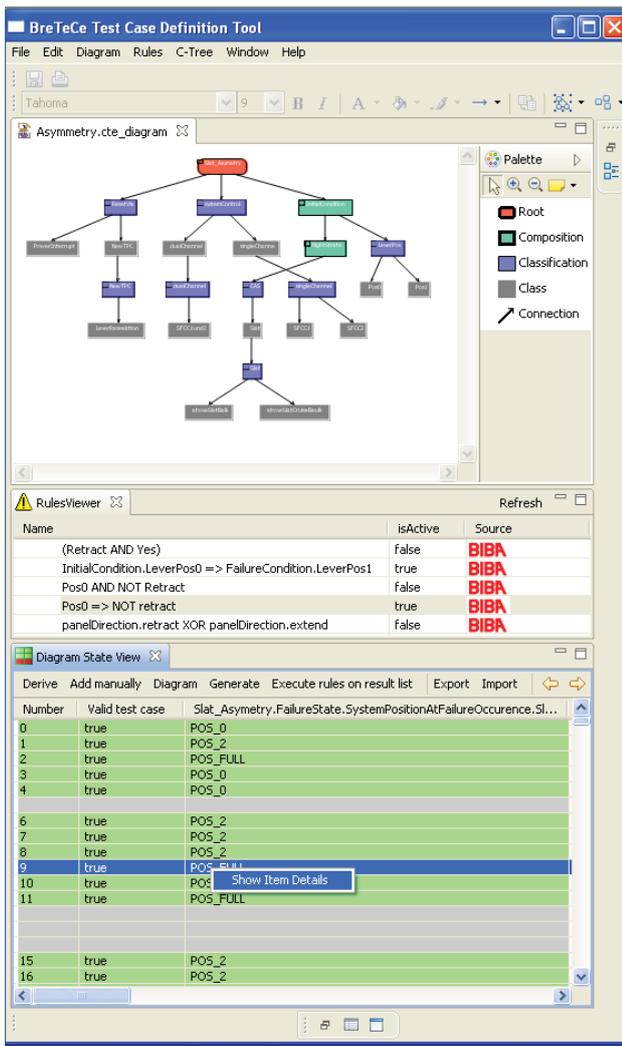


BILD 1. Werkzeug zur Erstellung von Testfalldefinitionen

Ein Ingenieur kann unter Einsatz des Werkzeugs Testdaten wiederholt und deterministisch erzeugen. Dafür modelliert er in einem ersten Schritt einen Klassifikationsbaum mittels eines graphischen Editors (oberes Fenster). Das Werkzeug erlaubt dem Anwender nur das Zeichnen von methodenkonformen Bäumen. Im nächsten Schritt kann der Anwender zur Priorisierung einzelner Testdaten Gewichtungen innerhalb bestimmter Bauelemente, den Klassen, setzen (nicht dargestellt). Hierdurch legt er fest, wie oft einzelne Funktionsparameter bei der Generierung der Testdaten berücksichtigt werden sollen. Auch in diesem Arbeitsschritt unterstützt das Werkzeug den Benutzer, indem es Fehleingaben meldet oder lückenhafte Eingaben automatisiert vervollständigt. Weiterhin ermöglicht das Werkzeug dem Anwender, semantische Abhängigkeiten im Testobjekt sprich im Klassifikationsbaum über sogenannte Regeln zu annotieren (mittleres Fenster). Auch diese Angaben werden dann bei der Generierung der Ergebnismenge berücksichtigt. Der Anwender erhält schließlich die Testdaten, mit denen das System unter Test bzw. die zu testende Systemfunktion parametrisiert werden muss (unteres Fenster).

Testdaten geben den strukturellen Aufbau eines Testobjekts wieder, d. h. die Zustände denen das System im Test auszusetzen ist. Das dynamische Verhalten des Systems sowie der Testablauf werden durch den Testfallgenerator nicht adressiert. Um auch diesbezüglich eine Optimierung zu erreichen, bedarf es weiterer methodengestützter Ansätze. Das Ziel muss es insbesondere sein, Ingenieure bei der Erstellung von Testprozeduren zu unterstützen. In diesem Kontext untersucht BreTeCe derzeit sowohl modellbasierte Ansätze als auch etablierte Verfahren der Textersetzung hinsichtlich ihrer Anwendbarkeit.

5.2. Infrastruktur für verteilte Testumgebungen

BreTeCe möchte eine „virtuelle“ Testumgebung für dezentrale Testsysteme schaffen. Das Hauptziel ist es, Multisystemtests in einer früheren Phase des Entwicklungsprozesses durchzuführen unter Nutzung existierenden Anlagen. Voraussetzung hierfür ist ein Werkzeug zur Vernetzung räumlich getrennter Testsysteme. Dabei soll es abhängig vom Testfokus möglich sein, Originalteile durch Simulationsmodelle zu substituieren.

BreTeCe entwickelt dazu den Node Controller, eine universelle Vermittlungsschnittstelle für Kommunikationsprotokolle wie z. B. ARINC 429 und AFDX, Bussysteme wie z. B. CAN sowie für analoge und diskrete Signale. Die Vision des Node Controller beliebige Testsysteme mittels eines einheitlichen Steckers miteinander zu vernetzen, ist in BILD 2 symbolhaft dargestellt. BILD 2 zeigt einen Real Time System Simulator (oben), der mittels zweier Node Controller über das Internet mit einem Test-Rig für Hochauftriebssysteme (unten) verbunden werden soll. In einem solchen BreTeCe-Netzwerk wären die Node Controller u. a. für die Gewährleistung der Signalkonformität und der Signalqualität verantwortlich. Die Leistungsanforderungen bzgl. der Signalübertragungen konnten bereits mittels Simulationen validiert werden. Eine prototypische Umsetzung der Node Controller ist in Arbeit.

Das Node Controller Konzept zur temporären Errichtung von verteilten Testumgebungen unter akzeptablem Aufwand würde einen enormen Flexibilitätszuwachs für das Testen mechatronischer Systeme bedeuten. Durch die Kombination existierender Testanlagen wären weniger spezialisierte Systeme notwendig und Funktionstests auf Multisystemlevel könnten frühzeitiger beginnen. Der Testprozess würde die verteilte Ausprägung heutiger Entwicklungs- und Fertigungssysteme reflektieren und damit den Anforderungen von Produktionsnetzwerke besser gerecht werden.

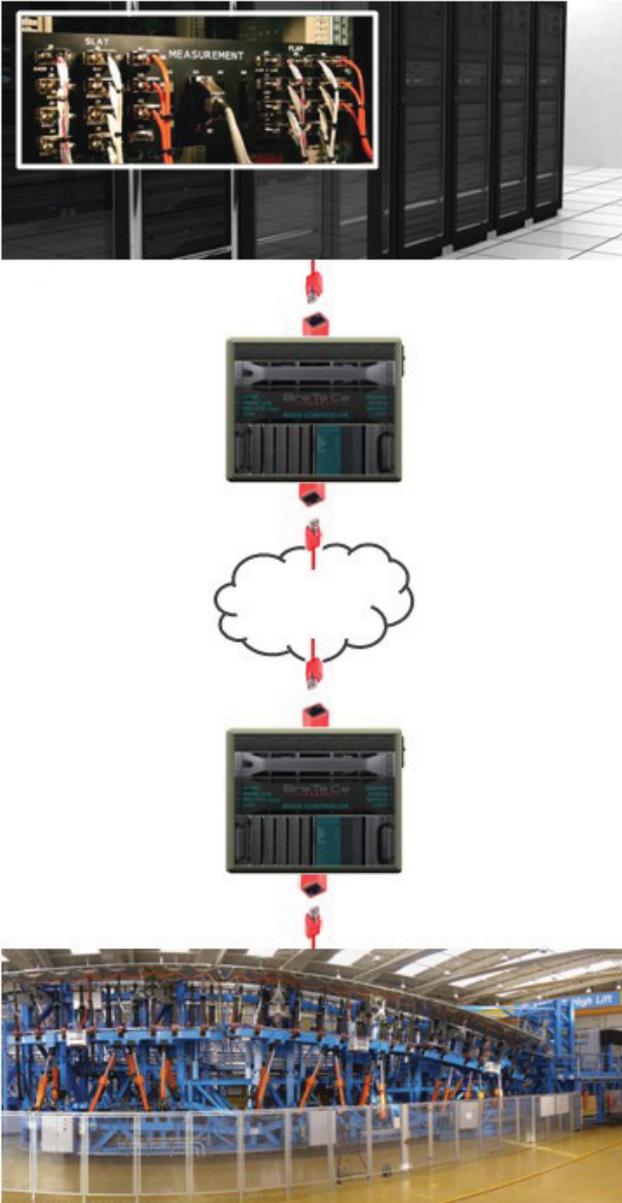


BILD 2. Node Controller zur Vernetzung von Testanlagen

5.3. Standardisierte Kontrollschnittstelle

Funktionstests erfordern neben dem Prüfling (System unter Test) eine Reihe spezifischer Komponenten, die ihrerseits die Testumgebung darstellen. Die Testumgebung beinhaltet neben einem Testsystem also eine Reihe weiterer insbesondere softwaretechnischer Komponenten. Diese dienen u. a. der automatisierten Testdurchführung, der manuellen Einflussnahme durch Testingenieure oder der Speicherung, Visualisierung und Auswertung von Testergebnissen. Häufig werden Testumgebungen für einen bestimmten Zweck bzw. Prüfling realisiert. Die dazu erforderlichen Systemkomponenten sind in der Regel über proprietäre Schnittstellen gekoppelt. Für einen Hersteller bedeutet dies, dass er eine vorhandene Testumgebung nicht ohne weiteres für geänderte bzw. neue Systeme verwenden kann.

Eine standardisierte Kontrollschnittstelle würde dazu beitragen, eine verbesserte Interoperabilität von Testequipment und damit einhergehend eine weitgehende Wiederverwendung von bestehenden Testanlagen zu ermöglichen. Durch eine standardisierte Kontrollschnittstelle ließe sich die aufwändige Inbetriebnahme von Testumgebungen, welche häufig auf Kompatibilitätsproblemen zwischen Anwendungssoftware und Testsystem beruht, deutlich verkürzen. Deshalb konzipiert und implementiert BreTeCe eine Kontrollschnittstelle, die die besonderen Anforderungen von Testumgebungen im Kontext mechatronischer Systeme adressiert. Die sich in der Umsetzung befindende Kontrollschnittstelle für Testumgebungen beinhaltet eine Abstraktionsschicht, die die hardwarenahen Signale und Befehle des Echtzeitsystems einer Testanlage gegenüber der Anwendungssoftware kapselt sowie eine Konfigurationsschicht, die im Bedarfsfall die komfortable Anpassung eines Testsystems an einen Prüfling ermöglicht [BILD 3].

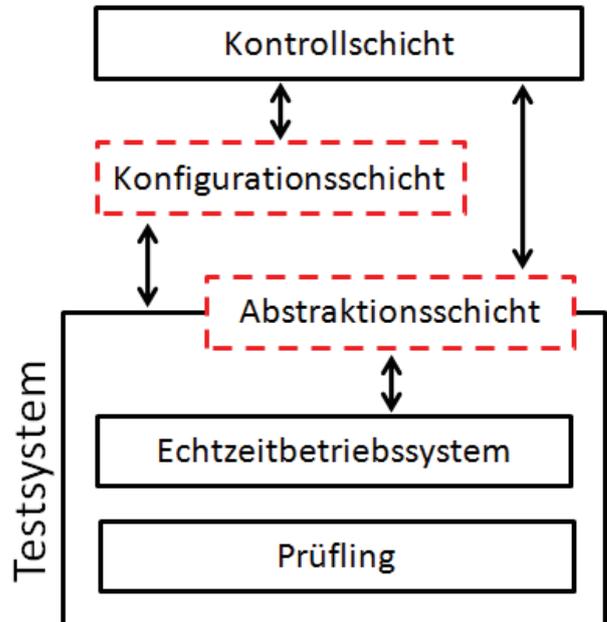


BILD 3. Kontrollschnittstelle zum Zugriff auf Testsysteme

Die in BILD 3 dargestellte Architektur einer Kontrollschnittstelle bewirkt eine Entkopplung der Anwendungssoftware von den spezifischen Ausprägungen einer Testanlage. Testsystemunabhängige Anwendungssoftware bewirkt einen effizienteren Testprozess, da sie sich kampagnenübergreifend einsetzen lässt und ihr Funktionsumfang kontinuierlich erweiterbar ist. Ferner werden durch den Einsatz von testsystemunabhängiger Anwendungssoftware Testingenieure bei einem Systemwechsel nicht mehr dazu gezwungen, sich fortwährend in neue Bedienungsabläufe und Programmoberflächen einzuarbeiten.

6. KONKLUSION

Die in BreTeCe verfolgten Ansätze und Entwicklungen

zeigen bereits grundsätzlich das Optimierungspotential für das funktionale Testen von komplexen mechatronischen Systemen. Der prototypisch realisierte Testfallgenerator belegt, dass eine Unterstützung der Identifikation relevanter Testfälle informationstechnisch sichergestellt werden kann. Eine auf dem Node Controller Konzept basierende Infrastruktur erlaubt die Vernetzung von Testsystemen und infolgedessen eine frühzeitige Durchführung von Multisystemtests. Die bisherige prototypische Umsetzung zeigt jedoch, dass speziell hinsichtlich der Skalierbarkeit und Handhabbarkeit von verteilten Testumgebungen weitere Forschungsaktivitäten erforderlich sind. Die Entkopplung von Anwendungssoftware und Testsystem durch eine generische Schnittstelle ermöglicht die Wiederverwendung existierender Software.

Literatur

- [1] Verein Deutscher Ingenieure: VDI 2206 – Entwicklungsmethodik für mechatronische Systeme. Richtlinie. Beuth Verlag GmbH, Düsseldorf (2004)
- [2] Henry, P.: The Testing Network - An Integral Approach to Test Activities in Large Software Projects, pp. 60-62. Springer-Verlag Berlin Heidelberg (2008)
- [3] V-Modell XT Autoren und Andere: V-Modell XT Gesamt 1.3. URL: <http://vmodell.iabg.de/dmdocuments/V-Modell-XT-Gesamt-Englisch-V1.3.pdf>, last visit: 18.08.11
- [4] Ott, A.: System Testing in the Avionics Domain, S. 3-6. Dissertation (2007). urn:nbn:de:gbv:46-diss000108814
- [5] AVACS H3: The Flap Controller. Fallstudie, S. 1-4. URL: <http://www.avacs.org/fileadmin/Benchmarks/Open/flapcontroller.pdf>, last visit: 18.08.11
- [6] Grochtmann, M.: Test Case Design using Classification Trees. Proceedings of the International Conference on Software Testing Analysis & Review (STAR 1994), Washington D.C., USA (1994). doi: 10.1.1.83.9731