

SiLEST

Entwicklung und Erprobung von Methoden und Werkzeugen für den Test und die Sicherheitsanalyse eingebetteter Echtzeitsoftware

Dr. Olaf Maibaum

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Simulations- und Softwaretechnik
Lilienthalplatz 7
38108 Braunschweig

Kurzfassung

Es wird ein Überblick über den in SiLEST verfolgten Ansatz des Tests von eingebetteten Systemen in einer Software in the Loop Testumgebung gegeben. Ein besonderes Augenmerk liegt dabei auf der Überprüfung der softwarebeeinflussten Systemsicherheit. Die hohe Flexibilität des rein softwarebasierten Ansatzes läßt erwarten, dass die falsche Behandlung von Fehlern und Fehlverhalten von Sensoren und Aktuatoren schon frühzeitig in einer Regelungssoftware aufgedeckt und durch die gebotenen Debugging-Möglichkeiten schneller behoben werden können. Hierdurch werden eine Erhöhung der Softwarequalität und eine Senkung der Kosten für den Test eingebetteter sicherheitskritischer Systeme erwartet.

1. Einleitung und Vorstellung des Themenkomplexes

Unsere technische Umwelt umgibt uns mit einer immer weiter steigenden Anzahl von Computersystemen, welche als eingebettete Systeme bezeichnet werden. Eingebettete Systeme übernehmen in Geräten die Steuer- und Regelungsaufgaben des Systems und sind als Computer durch den Benutzer nicht zu erkennen. So besitzen zum Beispiel bereits heute Automobile mehr als 40 eingebettete Systeme für die Bereitstellung von Service- und Sicherheitsfunktionen bis hin zum aktiven Eingriff in die Fahrdynamik. Systeme mit ähnlichen Anforderungen lassen sich auch in der Luft- als auch in der Raumfahrt finden.

Der Stellenwert von Software aktueller und zukünftiger eingebetter Systeme besitzt dabei eine große Bedeutung. Die immer weiter steigenden Anforderungen hinsichtlich der Funktionalität und der nachträglichen Erweiterung der Funktionalität machen den Einsatz von Software unverzichtbar. Die jüngste Vergangenheit hat jedoch gezeigt, dass durch den gesteigerten Einsatz von Software die Zuverlässigkeit der Produkte gesunken ist. Die Ursachen für diesen Makel sind in unzureichenden Methoden für die Entwicklung und die Qualifizierung von komplexen Softwaresystemen zu finden, welche eine besondere Sorgfalt hinsichtlich der Schnittstellen zwischen den einzelnen Systembestandteilen benötigen. Erschwerend kommt hinzu, dass die Methoden für den

Test von Standardsoftware nicht direkt auf den Test von Software eingebetteter Systeme übertragbar sind.

Durch den Einsatz dieser Systeme in sicherheitskritischen Umgebungen wie beispielsweise in einem Automobil und dem größer werdenden Anteil der Software an einem eingebetteten System, kommt dieser Software auch eine immer größer werdende Rolle an der Kritikalität eines eingebetteten Systems zu. Da die Software im Lauf der Zeit keinen Änderungen unterworfen ist, sofern sie nicht im Rahmen der Softwarewartung oder durch einen Defekt des Massenspeichers geändert wird, liegen auftretende Softwarefehler bereits bei der Auslieferung vor. Im Gegensatz hierzu können an der Hardware eines eingebetteten Systems auch Defekte auftreten, welche durch die Alterung des Systems entstehen. Die Alterung der Hardware kann zum Ausfall von Sensoren oder Aktuatoren, der Verschlechterung von Messdaten der Sensoren oder ein schlechteres Wirkverhalten von Aktuatoren führen.

Software eingebetteter Systeme in sicherheitskritischen Umgebungen muss mit diesen Alterungserscheinungen der Hardware umgehen können. Im Falle eines kompletten Ausfalls oder der Verfälschung von Sensordaten muss der Regelalgorithmus der Software eingebetteter Systeme dafür sorgen, dass von dem eingebetteten System trotzdem keine Gefahr ausgeht. Dies bedeutet, dass die Software den Ausfall von Hardware oder die Verfälschung von Sensordaten erkennen und das Regelverhalten an die veränderte Umgebungssituation anpassen muss. Wenn kein sicherer Betrieb des technischen Systems mehr möglich sein sollte, so ist das System in einen sicheren Zustand zu bringen. Dieses Verhalten wird auch als softwarebeeinflusste Systemsicherheit bezeichnet.

Die softwarebeeinflusste Systemsicherheit muss während der Qualifikation eines sicherheitsrelevanten Systems überprüft werden. Das heißt, die Software eines eingebetteten Systems ist neben dem nominalen Verhalten auch hinsichtlich des Verhaltens im Falle des Ausfalls oder der Alterung der Hardware zu testen. Derartige Tests können nicht losgelöst von der Umwelt des technischen Systems durchgeführt werden, da das eingebettete System über die Aktuatoren und Sensoren in einen Regelkreis eingebunden ist. Für den Softwaretest nach der Integration sind daher Testumgebungen notwendig, welche die zu testende Software in den Regelkreis einbettet. Neben dem Test des Ausfalls und der Alterung der Hardware ist auch das Softwareverhalten in Extremsituationen zu untersuchen. Die Software sollte hierbei immer noch ein Regelverhalten aufweisen, welches in solchen Situationen zu keinem Schaden führen.

Für den Test der Software eines eingebetteten Systems im Regelkreis stehen drei Möglichkeiten zur Verfügung,

- die Verwendung von Prototypen des technischen Systems in der realen Umgebung,
- die Verwendung von Hardware in the Loop (HiL) Simulationen oder
- die Verwendung von Software in the Loop (SiL) Simulationen.

Die Möglichkeit der Verwendung von Prototypen für die Qualifizierung von Software eingebetteter Systeme bietet jedoch nur die Möglichkeit der Überprüfung des Nominalverhaltens. Die Überprüfung des Verhaltens im Falle des Ausfalls von Hardware während des Betriebs ist mit einem Prototyp nur schwer möglich. Auch der Test der Software in Extremsituationen ist nur für solche Situationen sinnvoll, welche nicht zum Verlust des Prototyps führen würden. Desweiteren lässt sich insbesondere im Bereich der Raumfahrt die reale Umgebungssituation nicht perfekt nachbilden. Insofern kommt für die Qualifizierung der softwarebeeinflussten Systemsicherheit die Verwendung eines Prototyps nicht in Frage.

Die zweite Möglichkeit des Tests mit einer HiL Simulation ist die heute gängige Praxis für den Test von eingebetteten Softwaresystemen. Für die HiL Simulation wird eine Simulation der Umwelt, der Sensoren und der Aktuatoren benötigt. Die Kopplung zwischen der Simulation und dem

eingebetteten System geschieht dabei über VME-Buskarten zur Umsetzung der Sensordaten in die elektrischen Signale der Sensoren und zur Umsetzung der elektrischen Steuersignale an die Aktuatoren. Die Ein- und Ausgänge der VME-Buskarten werden über einen Kabelbaum mit den elektrischen Ein- und Ausgängen des Rechnerboards des eingebetteten Systems verbunden.

Der große Vorteil des Tests mit einer HiL Simulation ist, dass zum Aufbau des Testbeds kein Eingriff in die zu testende Software notwendig ist. Die HiL Simulation muss jedoch Echtzeitanforderungen erfüllen. Das heißt das zeitliche Verhalten der simulierten Sensoren und Aktuatoren als auch der Rückkopplungseffekte müssen dem realen zeitlichen Verhalten entsprechen. Der Einsatz von HiL Simulationen ist jedoch durch die Auslegung des Testbeds in Hardware kostenintensiv und zeitlich erst sehr spät im Entwicklungsprozess möglich. Auch stellt das Testbed einen Engpass im Testprozess dar, da nicht mehrere Entwicklergruppen gleichzeitig auf das Testbed zugreifen können.

Aus Sicht des Softwareentwicklers bietet der Test mit einer HiL Simulation durch die lose Kopplung von Simulation und eingebetteten System noch weitere Probleme in sich. So erfolgen die Tests aus einem Initialzustand heraus und die Tests können nicht unterbrochen werden. Eine Unterbrechung eines Testlaufs ist jedoch nötig, um im Falle der Fehlersuche einen genauen Blick auf den Zustand des Systems vor dem Auftreten eines Fehlers zu ermöglichen.

Die letzte Möglichkeit des Tests mit einer SiL Simulation kehrt die Nachteile und Vorteile einer HiL Simulation in das Gegenteil um. Die Erweiterung von SiL Simulationen zum Test bieten das Potenzial von erheblichen Kosteneinsparungen gegenüber der Verwendung einer HiL Simulation. Das notwendige Testbed läßt sich beliebig oft reproduzieren, sofern ausreichend Entwicklerboards für das eingebettete System zur Verfügung stehen. Hierdurch stellt das Testbed keinen Engpass für den Softwaretest mehr da.

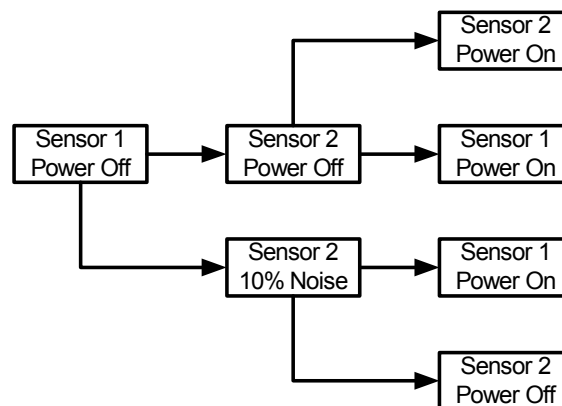


Abbildung 1: Baumartige Fehlerabfolge

Auch die Debuggingmöglichkeiten für den Softwareentwickler werden durch den Einsatz einer SiL Simulation verbessert, weil durch die mögliche enge Kopplung zwischen Simulation und dem zu testendem System ein Start/Stop-Betrieb ermöglicht wird. Hierdurch wird es möglich, Momentaufnahmen der Simulation und des eingebetteten Systems zu erzeugen, welche auch als Startwert für weitere Tests verwendet werden können. Dies ermöglicht die Ausführung von baumartigen aufgebauten Fehlerabfolgen in der Hardware zur Simulation von Mehrfachfehlern ohne die erneute Ausführung aller bisherigen Fehler aus einem Initialzustand heraus.

Im Rahmen des Projektes SiLEST¹ soll die Überprüfung der softwarebeeinflussten Systemsicherheit eingebetteter Systeme mit Hilfe von SiL-Simulationen untersucht werden. Die dabei zu entwickelnden Methoden und Werkzeuge werden innerhalb des Projektes für die Anwendungsdomänen Automobil und Raumfahrt auf vorhandene Software angewandt und die Ergebnisse mit der Anwendung von HiL Testprozessen und real gemachten Erfahrungen verglichen. Dabei ist im Einzelnen

- ein verallgemeinerter Testprozess für den Softwaretest mit einer SiL Simulation zu definieren.
- ein Satz von Tailoringregeln für den verallgemeinerten Testprozess für die Anwendungsdomänen Automobil und Raumfahrt aufzustellen.
- eine Testumgebung für den Softwaretest mit einer SiL Simulation zu schaffen.
- eine allgemeine Schnittstellenspezifikation für die Simulationsmoduln zu beschreiben.
- mit einer Werkzeugkette die Automatisierung des Softwaretests mit einer SiL Simulation zu ermöglichen.

Sollten sich die Methoden des Softwaretests eingebetteter Systeme mit einer SiL Simulation zur Qualifizierung der softwarebeeinflussten Systemsicherheit anwenden lassen, so ist durch deren Einsatz eine Kostenreduzierung, und durch die Flexibilität und Automatisierbarkeit eine Steigerung der Produktqualität zu erwarten.

2. Projektstatus

Da das erste Hauptarbeitspaket von SiLEST noch nicht abgeschlossen wurde und daher noch keine Ergebnisse im Projekt vorliegen, können sich noch Änderungen in den in diesem Abschnitt vorgestellten Konzepten und Problemlösungen ergeben, sofern die im Projekt gewonnenen Erkenntnisse dies notwendig machen.

Der Test der softwarebeeinflussten Systemsicherheit unterscheidet sich von den klassischen Vorgehensweise für den Nachweis der funktionalen Korrektheit dadurch, dass die Testfälle nicht auf Basis der Softwareanforderungen generiert werden, sondern auf Basis der bekannten und vorstellbaren Fehler und Fehlfunktionen der Softwareumgebung. So können beispielsweise

- der Totalausfall,
- ein elektrischer Offset,
- ein Rauschen
- oder eine Wellenfunktion auf einer Signalleitung oder
- ein geändertes Signal/Rausch-Verhältnis

eines Sensors die Grundlage für einen Testfall bilden. Die Anforderungen liefern für diese Art des Softwaretests lediglich die Information, wie das korrekte Verhalten des Systems aussehen sollte.

Die Basis für den Aufbau der Simulation und der Erstellung der Testfälle bildet das Systemmodell des eingebetteten Systems. Das Systemmodell liegt dabei idealerweise als UML-Beschreibung in Form eines Komponentendiagramm vor wie in Abbildung 2 dargestellt. Aus dem Komponentendiagramm des Systemmodells lassen sich die Sensoren und Aktuatoren und die Schnittstellen zum eingebetteten System als Basis für die Umgebungssimulation ableiten.

¹ Software in the Loop for Embedded Software Test

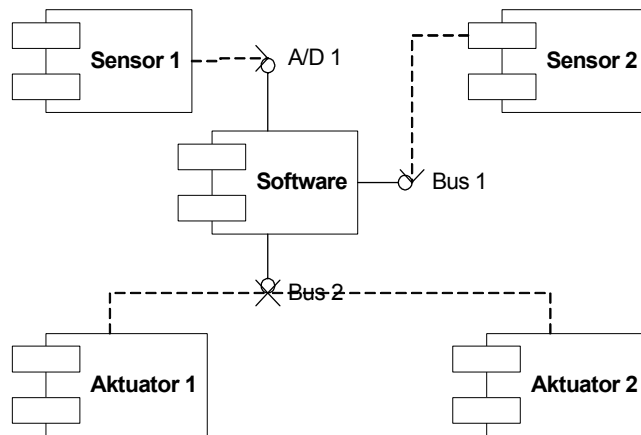


Abbildung 2: Komponentendiagramm des Systemmodells

Dieser Vorgang der Ableitung von Simulationsmoduln aus dem Komponentendiagramm des Systemmodells wird durch eine Simulationsmodellbibliothek unterstützt, welche vorgefertigte Simulationsmoduln der verwendeten Sensoren und Aktuatoren enthält. Jeden dieser Simulationsmoduln haften die Schnittstellen zum eingebetteten System und zu der physikalischen Umgebungssimulation an. Im resultierenden Testbed wird die Schnittstelle zum eingebetteten System durch spezielle Simulationsmoduln in der Umweltsimulation und angepassten Betriebssystemschnittstellen der Software unter Test repräsentiert. Die Schnittstellen zur physikalischen Umgebungssimulation werden durch SI-Einheiten repräsentiert, welche zwischen den Sensor- und Aktuator-simulationsmoduln und den physikalischen Simulationsmoduln ausgetauscht werden. Die Abbildung 3 zeigt das diesem Ansatz zu Grunde liegende Schichtenmodell für einen Sensor.

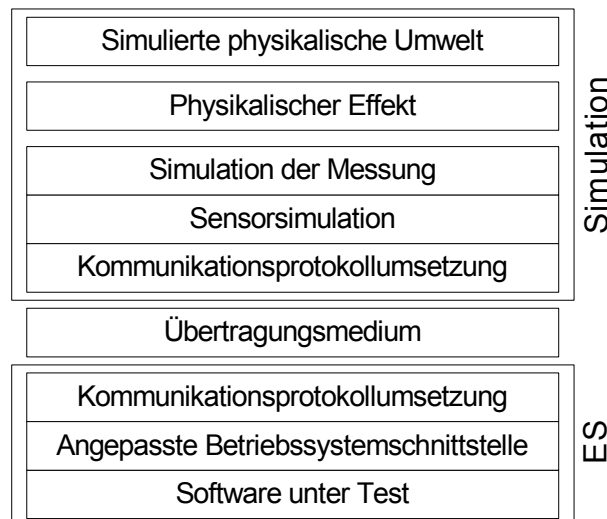


Abbildung 3: Schichtenmodell für einen Sensor

Um den Aufbau der Umweltsimulation zu vervollkommen, sind aus der Simulationsbibliothek vorhandene Simulationsmoduln für die physikalische Umwelt auszuwählen und die Sensor und Aktuatormoduln mit Ihnen in Beziehung zu setzen und zu parametrisieren. Falls keine vorgefer-

tigten Simulationsmoduln zur Verfügung stehen, so sind entsprechende Simulationsmoduln zu entwickeln. Für spätere Tests können sie in die Simulationsbibliothek aufgenommen werden.

Die Simulationsmoduln für Sensoren und Aktuatoren sollten neben der Simulation des Nominalverhaltens auch Simulationen für bekannte oder theoretisch mögliche Fehler und Fehlverhalten enthalten. Dabei sollte ein objektorientiertes Konzept verfolgt werden, so dass es beispielsweise möglich wird, durch eine Spezialisierung eines Analogensors einen Temperatursensor abzuleiten. Fehlverhalten, wie ein aufmoduliertes Rauschen auf der Signalleitung, oder die Kommunikation mit dem eingebetteten System werden in diesem Fall von der Klasse eines Analogensors ererbt. Sensorspezifische Fehler und Fehlverhalten sind Teil der Spezialisierung. Auf diese Art ist es möglich, schnell neue Simulationsmoduln aufzubauen.

Für die Testdurchführung stehen zwei unterschiedliche Vorgehensweisen zur Verfügung. Beiden Vorgehensweisen gemeinsam ist, dass der Testingenieur zunächst anhand der Anforderungen an das zu testende System quantitative Grenzen für ein normales Verhalten während eines Handlungsstrangs einer Nutzung festlegt. Dies könnte beispielsweise die Drehrate eines Satelliten während eines Datatakes in der Simulation sein, falls das Lageregelungssystem eines Satelliten das Testobjekt ist, oder die Umdrehungsgeschwindigkeit eines Motors für ein Motormanagementsystem als Testobjekt. Während des Testablaufs werden dem Handlungsstrangs Fehler und Fehlverhalten der Sensorik und Aktuatorik aufmoduliert. Bei Verletzung der vorgegebenen Grenzen für ein Nominalverhalten wird ein Testablauf automatisch durch die Testumgebung im Testprotokoll als fehlerhaft eingestuft und bedarf einer weiteren Untersuchung durch die Entwicklungsingenieure.

Bei der ersten Möglichkeit der Testdurchführung werden die auftretenden Fehler und Fehlverhalten und ihre Abfolge aus der Menge der durch die Simulation bereitgestellten Fehler und Fehlverhalten für Sensoren und Aktuatoren bereitgestellt. Diese Möglichkeit ermöglicht es dem Testingenieur bestimmte interessante Fehlerabfolgen zu überprüfen, welche durch das System unter Test abgefangen werden müssen.

Bei der zweiten Möglichkeit der Testdurchführung gibt der Entwicklungsingenieur für die Fehler und Fehlverhalten Eintrittswahrscheinlichkeiten und die mittlere Zeit zwischen zwei Fehlern an. Die Testumgebung variiert dann selbstständig Fehler und Fehlverhalten über den Handlungssträngen bis eine untere Schranke für die Eintrittswahrscheinlichkeit der Mehrfachfehler unterschritten wird. Diese Art der Testdurchführung kann sehr zeitaufwendig sein. Die Auslegung der Testumgebung in reiner Software erlaubt es jedoch, diese Tests ohne einen Eingriff durch einen Testingenieur durchzuführen.

Ein grundlegendes Problem, welches im Rahmen von SiLEST gelöst werden muss, ist die Behandlung des zeitlichen Verhaltens im Test mit einer SiL Simulation. Eingebettete Systeme unterliegen im Regelfall Echtzeitanforderungen. Für die Kopplung der Software unter Test mit der Simulation ist jedoch der Austausch von Betriebssystemschnittstellen notwendig. Dies hat ein anderes zeitliches Verhalten der zu testenden Software zur Folge.

In SiLEST ist beabsichtigt, dieser Problematik durch die Einführung einer virtuellen Simulationszeit zu begegnen. Die Grundlage hierfür liefert der Ansatz von Müller-Olm [Mü97], dass die Zeit nur zu Zeitpunkten der Ein- und Ausgabe sichtbar ist. Zu Zeiten zwischen Ein- und Ausgaben können keine zeitlichen Aussagen getroffen werden, da die Zeit kommutativ zu den inneren Prozessen definiert ist. Für den Aufbau des SiL Testbed bedeutet dies, dass ein Snap Shot von einem Zustand nur zu Zeiten einer Ein- und Ausgabe des eingebetteten Systems durchgeführt werden kann, da dies die Synchronisationszeitpunkte des eingebetteten Systems mit der Simulation sind. Für die Realisation bedeutet dies, dass sämtliche Kommunikation zwischen dem eingebetteten System mit der Software unter Test und dem Simulationssystem nur zu Zeiten stattfindet, an de-

nen die Software unter Test einen Betriebssystemschnittstellenaufruf zur Kommunikation mit der Sensorik oder Aktuatorik getätigt hat. Die Realzeituhr des eingebetteten Systems ist dann vor Verlassen der Betriebssystemschnittstelle entsprechend der Laufzeit der originalen Betriebssystemschnittstelle zu korrigieren. Der Effekt ist, dass die Software unter Test kein anderes Zeitverhalten zu sehen bekommt, obwohl sie ein anderes Zeitverhalten aufweist. Die einzigen Abweichungen die bei dieser Vorgehensweise für die Software unter Test sichtbar werden, sind zeitliche Seiteneffekte durch Caches und Pipelines.

Zur Bestimmung der Laufzeit von Betriebssystemschnittstellen lassen sich Messverfahren oder analytische Verfahren zur Laufzeitmessung verwenden [Mai03].

3. Erfahrungen, Bewertungen

Bei den beiden Verbundpartnern Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR) und Ingenieurgesellschaft für Automobil und Verkehr (IAV) liegen umfangreichen Erfahrungen in der Entwicklung und dem Test von eingebetteten Systemen in den Anwendungsdomänen Raumfahrt und Automobil vor.

Das DLR setzt für die Entwicklung von onboard Software, der Kommandierungssoftware von Raumfahrtmissionen und Pilotenassistenzsystemen der Luftfahrt u.a. schon langjährig HiL Simulationssysteme ein. Mit der Etablierung der Verkehrsforschung im DLR werden HiL Simulationen auch für Steuerungssoftware der Verkehrs- und Fahrzeugtechnik eingesetzt werden (z.B. für aktive Fahrerassistenzsysteme).

Bei den meisten dieser Systeme besteht das Problem, dass diese Simulationssysteme nicht die wirklichen Umgebungsbedingungen während des Betriebes widerspiegeln, da z.B. im Weltraum kein Schwerkrafteinfluss vorhanden ist oder problematische Betriebssituationen sind durch den entstehenden Schaden nicht realistisch testbar. Mit HiL Simulationen ist in solchen Fällen die notwendige Vertrauenswürdigkeit in die Systemkorrektheit, -zuverlässigkeit und in die Software beeinflusste Systemsicherheit nicht vollständig herstellbar. Die SiL Simulation soll hierfür künftig als effektiveres und ergänzendes Testverfahren zur HiL Simulation eingesetzt werden.

Die Entwicklung von sicherheitskritischer eingebetteter Software im Rahmen neuartiger innovativer Produktkonzepte der Raumfahrt, Luftfahrt und der Verkehrs- und Fahrzeugtechnik wird im DLR künftig weiter zunehmen, wobei der Zeit-, Kosten- und Qualitätsdruck auch bei diesen wissenschaftlichen Entwicklungen zum Einsatz effektiverer System und Software Engineering Methoden zwingt.

Für die Verbesserung der Verkehrssicherheit im Straßen- und Schienenverkehr werden im DLR neue, innovative Konzepte und prototypische Lösungen für aktive Fahrerassistenzfunktionen entwickelt. Die hochkritische eingebettete Echtzeit-Steuerungssoftware dieser Systeme erfordert bereits in den frühen Phasen ihrer Entwicklung ein umfassendes Safety- und Dependability Management. Die mit SiLEST möglichen Tests der zu entwickelnden Software werden hierzu einen effektiven Beitrag leisten können.

Aus Sicht der IAV wird der Softwaretest sicherheitskritischer Systeme zukünftig aufgrund des zunehmenden Einsatzes so genannter x-by-Wire Steuerungen (z.B. elektronische Bremse oder Lenkung) an Bedeutung gewinnen.

Bisher wurden die Softwaretests für sicherheitskritische Systeme im Fahrzeug und an HiL Prüfständen durchgeführt. Durch den Einsatz der SiL Simulation können viele Tests unabhängig von kostenintensiven HiL Testsystemen angeboten und durchgeführt werden.

Dies bedeutet für die IAV GmbH eine höhere Flexibilität. Durch die niedrigeren Kosten ist darüber hinaus zu erwarten, dass die Akzeptanz der Simulation zum Softwaretest bei den Kunden gesteigert wird.

Desweiteren wird erwartet, dass sich der Einsatz von SiL Methoden auf den gesamten Bereich der Funktionsentwicklung von Steuergerätesoftware übertragen lässt und somit gerade für Modultests verstärkt eingesetzt werden kann. Daraus würde sich ein beträchtliches Potenzial an einzusparender Zeit ergeben, was folglich Kosten reduzierend wirken würde.

Im Rahmen von SiLEST wird der Verbundpartner Fraunhofer FIRST das verlässliche Echtzeit-Betriebssystem BOSS erweitern und eine Bewertung der erarbeiteten Methoden und Werkzeuge hinsichtlich ihrer Safety-Eigenschaften vornehmen. Wie bei jedem anderen Echtzeit-Betriebssystem ist das Zeitmanagement von BOSS statisch. Im SiLEST Projekt wird ein dynamisch anpassbares und ausdehnbares Zeitmodell erarbeitet, so dass BOSS für eine simulierte Echtzeit eingesetzt werden kann.

Dadurch werden unter anderen eine bessere Echtzeitanalyse und gemischte Hard- und Software-in-the-Loop-Simulationen ermöglicht. Die Simulatoren werden in SiLEST-Demonstratoren eingesetzt, aber generisch gestaltet, dass sie auch für andere Systeme eingesetzt werden können. FhG FIRST erreicht dadurch eine höhere Flexibilität, Effektivität und Verlässlichkeit der Systementwicklungen. Dadurch kann die Qualität und Effektivität der eigenen Entwicklungen erhöht werden.

Der Verbundpartner Webdynamix GmbH ist mit der Umsetzung der erarbeiteten Methoden in einen Werkzeugdemonstrator betraut. Das 1999 gegründete Unternehmen besitzt umfangreiche Entwicklungserfahrungen in den Bereichen Netz- und Datenbankanwendungen, Contentmanagement- und Risikomanagementsysteme unter Einsatz aktueller Softwareentwicklungsmethoden.

4. Ausblick

Sollte die Anwendungserprobung der in SiLEST entwickelten Methoden und des Werkzeugdemonstrators positiv evaluiert werden, so sollen sie, nach der Weiterentwicklung des Werkzeugdemonstrators zu einem marktfähigen Werkzeug durch die Webdynamix GmbH, in zukünftigen Projekten der Verbundpartner zum Einsatz kommen. Ein besonderes Potential liegt dabei in der zu entwickelnden Simulationsmodellschnittstelle, insbesondere für die Sensoren und Aktuatoren.

Diese Schnittstelle soll durch das DLR zur Standardisierung gebracht werden. Geeignet erscheint hier die Standardisierung als Zusatz des Step-Standards ISO 10303 zum Produktdatenaustausch. Dieser Standard regelt zurzeit den Austausch von Produktdaten insbesondere zwischen CAD Werkzeugen. Damit ließe sich der nicht unerhebliche Aufwand zur Entwicklung der Sensor und Aktuator Simulationen erheblich reduzieren, indem die Hersteller von Sensorik und Aktuatorik zu ihren Produkten auch vereinheitlichte Simulationsmoduln für ihre Produkte anbieten können.

Auch die momentane Problematik der fehlenden Interaktion zwischen den Simulationspaketen unterschiedlicher Fachdisziplinen kann mit einer standardisierten Simulationsmodulschnittstelle begegnet werden. Dies würde es beispielsweise ermöglichen, Thermal-, Energie- und Orbit Simulationen ohne größeren Aufwand miteinander zu koppeln. Es steht allerdings zu befürchten, dass die vorhandenen Simulationspakete einzelner Fachdisziplin nicht an eine neue Schnittstelle angepasst werden.

Das Potential der Schnittstellenstandardisierung liegt dabei nicht alleine bei dem in SiLEST verfolgtem Ansatz des Tests mit einer SiL Simulation, sondern auch für den Test mit einer HiL Simulation. Der Unterschied liegt dabei alleine in der Art der Kopplung zwischen eingebetteten System und der Simulation.

Sollte die für den Test mit einer SiL Simulation notwendige enge Kopplung zwischen dem eingebetteten System und der Umgebungssimulation ohne große zeitliche Nebenwirkungen möglich sein, so ließe sich dieser Ansatz einer virtuellen Zeit auch auf den Test von verteilten eingebetteten Systemen übertragen. Durch die hohe Flexibilität einer reinen Softwareumgebung sind dabei auch Untersuchungen hinsichtlich des Jitter innerhalb der Kommunikation zwischen den einzelnen Systemen möglich.

Ein völliger Verzicht auf Integrationstest mit einem Labormuster oder einer HiL Testumgebung wird in Zukunft jedoch nicht möglich sein, da die bislang gemachten Erfahrungen zeigen, dass die Realität sich von einer simulierten Umwelt unterscheidet, unabhängig von der Qualität der Simulation. Der Einsatz von SiL Testmethoden verspricht aber, die sicherheitskritischen Fehler in einer Software schon früher und kostengünstiger zu entdecken.

Akronyme

CAD	Computer Aided Design
ES	Eingebettetes System
HiL	Hardware in the Loop
ISO	International Organization for Standardization
SiL	Software in the Loop
SiLEST	Software in the Loop for Embedded Software Test
VME	Versa Module Europa

Literatur

- [Mai03] Olaf Maibaum: *Bestimmung symbolischer Laufzeiten in eingebetteten Echtzeitsystemen*. Dissertation. Nummer 2/03 in Berichte aus dem Department für Informatik, Carl von Ossietzky Universität Oldenburg. ISSN 0946-2910. Juni 2003.
- [Mü97] Markus Müller-Olm: *Modular Compiler Verification*. Nr. 1283 in Lecture Notes in Computer Science. Springer Verlag, 1997.
-