

Modellintegration: Architektursimulation von verteilten Steuergerätesystemen

Joachim Schlosser
BMW AG, München

Definition Die *Architektursimulation von verteilten Steuergerätesystemen* ist eine Methode, das Zusammenwirken von Funktionen, die verschiedenen Ausführungseinheiten eines Steuergeräteverbundes zugewiesen sind, zu simulieren. Die Architektursimulation berücksichtigt dabei die Einflüsse der zugrundeliegenden *Hardware-* und *Systemplattform*.

Mit Funktionen sind in diesem Zusammenhang Anwendungen wie z. B. die Grundbremsfunktion eines Bremssystems bezeichnet, die durch die sogenannte *Funktionsnetzarchitektur* in verschiedene *Komponenten* zusammengefaßt werden. Die Funktionen liegen dabei entweder als textuelle Beschreibung oder als ausführbare Modelle vor.

Die Hardware- und Systemplattform, kurz Plattform, bezeichnet die Hardware, bestehend aus Busleitungen, Prozessoren, Speicher, Sensoren und Aktoren, sowie systembezogene Komponenten wie Busprotokolle, Betriebssysteme, und *Middleware* wie z. B. Netzwerkmanagement.

Rein *funktionale Simulation* betrachtet das Verhalten von Funktionskomponenten unter Vernachlässigung von Laufzeiten und Verzögerungszeiten.

Abbildung 1 zeigt, wie zwei Funktionen *Funktion_A* und *Funktion_B* auf logischer Ebene mittels Signal *Signal_1* direkt miteinander kommunizieren. Die Architektursimulation bildet nun den Weg nach, den die Nachricht im realen System nimmt. Hier werden das Betriebssystem *OSEK* und dessen Ressourcen, sowie der Versand der aus dem Signal generierten Nachricht auf den redundanten Bussen berücksichtigt.

Pilotprojekt In einem aktuellen Vorentwicklungsprojekt bei BMW, dem Drive-by-Wire (DbW) Projekt, wird unter anderem diese Architektursimulation pilotiert. Dabei soll auch geklärt werden, unter welchen Umständen die Architektursimulation eine Ergänzung zu bekannten Evaluierungsmethoden wie rein funktionaler Simulation, Rapid Prototyping oder Hardware-in-the-Loop (HiL) bieten kann. Zusatznutzen kann zum einen erbracht werden, in dem Tests in der Architektursimulation schneller als in oben genannten Techniken durchgeführt werden können. Dies eröffnet sowohl die Möglichkeit kürzerer Iterationzyklen, als auch einer größeren Testabdeckung. Zum anderen ist Zusatznutzen darin zu erkennen, daß insgesamt eine größere Testtiefe erreicht wird, also durch Modellevaluierung andere Eigenschaften des Systems abgeprüft werden können als mit Hardware. Diese Eigenschaften können entweder in der Hardware nur schwer oder gar nicht prüfbar, oder deren Zustände nicht reproduzierbar sein.

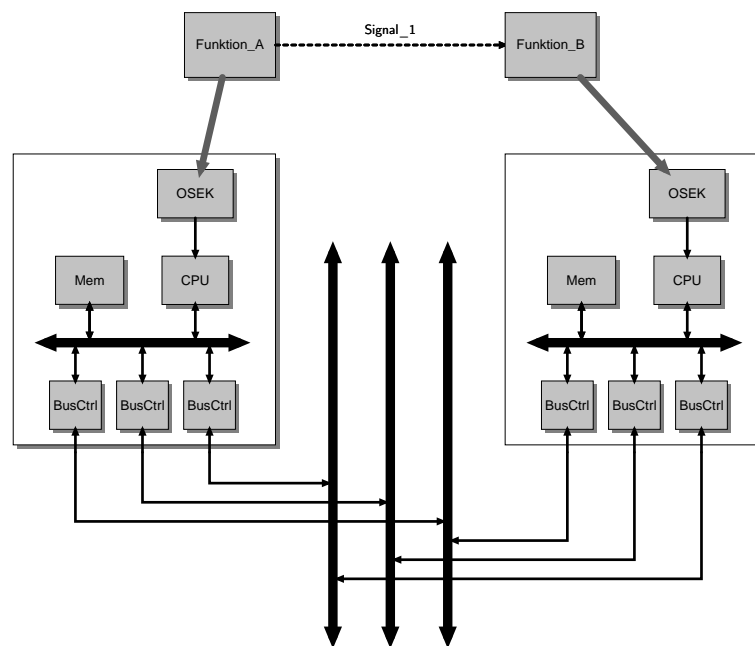


Abbildung 1: Prinzip der Architektursimulation

Mit Hilfe von Architektursimulation können Plattformscheidungen fundierter getroffen werden und Probleme, die bei der Integration von Plattformbausteinen und Funktionskomponenten auftreten, früher erkannt und behoben werden.

Die dabei untersuchten *Modellarten* umfassen Modelle von *Funktionsstruktur*, *Funktionsverhalten*, *Hardware* inklusive Bus und Busprotokoll, *Plattform* und *Verteilung*. Das Verteilungsmodell beinhaltet die Information, welche Funktionskomponente auf welche Plattformkomponente abgebildet wird. Darin eingeschlossen ist im Falle von Betriebssystemen die Konfiguration von Tasks und Prozessen, die entweder separat vorliegt oder aus detaillierten Funktionsverhaltensmodellen generiert wird.

Die Untersuchung der Umstände, unter denen Architektursimulation angewendet werden kann, umfasst

- eine Analyse der Anforderungen an die verschiedenen Modellarten hinsichtlich Abstraktionsebene und Strukturierung,
- die Anforderungen an den Entwicklungsprozeß hinsichtlich der Verfügbarkeit von Entwurfsentscheidungen und Modellen in den verschiedenen Entwicklungsabschnitten und
- die Anforderungen an die Werkzeugkette für die Architektursimulation und deren Ankopplung an die übrige Werkzeugkette der Entwicklung.

Hauptsächlich werden für das Vorentwicklungsprojekt Untersuchungen durchgeführt, die dazu beitragen sollen, die Plattformarchitektur für das DbW-System zu definieren und zu optimieren. Für die Architektursimulation wird hier das Werkzeug Cadence Sysdesign verwendet, ein Aufsatz auf Cierto VCC, das hauptsächlich für System-on-Chip-Anwendungen gedacht war. In Zusammenarbeit der Firmen Cadence und BMW entstanden, erweitert Sysdesign das ursprüngliche Cierto VCC um zusätzliche Modelle für das OSEK-Betriebssystem und Feldbusse wie CAN oder FlexRay und Integrationsmöglichkeiten für den Einsatz im Kontext der Entwicklung verteilter, eingebetteter Systeme. Letzteres umfaßt hauptsächlich den Import von ASCET SD Modellen und eine skriptgesteuerte Schnittstelle zum halbautomatischen Verteilen der Funktionen.

Kritische Aspekte Da die Architektursimulation eine rein virtuelle Absicherungsmethode ist, sind bestimmte Abstraktionen bezüglich der Systemplattform zu treffen, um die Simulation beherrschbar hinsichtlich Laufzeit, Erstellung und Parametrierung zu halten. Eine Nebenuntersuchung stellt die Abstraktionen und deren Auswirkung auf die Belastbarkeit der Aussagen, die durch die Simulation gewonnen wurden, gegenüber.

Dabei wird zwischen verschiedenen Arten von gewünschten Aussagen unterschieden. So wird die Simulation u. a. für die Validierung des Sicherheitskonzepts, sowie für die Validierung – teilweise auch Optimierung – der Konfigurierung der Betriebssysteme und der Kommunikation eingesetzt. Ein besonderes Augenmerk wird dabei gerichtet auf folgendes Problem gerichtet: In der frühen Phase der Entwicklung stehen kaum detaillierte Modelle zur Verfügung, was die Genauigkeit der Simulation und damit deren Aussagekraft einschränkt.

In der Realisierung stellt sich speziell die Beherrschung der Werkzeugkette und der Modellgrößen bzw. der Komplexität der Modelle als Herausforderung dar. Funktionsmodelle, die nicht von vornherein auch für Architektursimulation vorgesehen sind, müssen manuell verändert werden, um ihre virtuelle Integration zu ermöglichen. Deswegen sollte die Entscheidung, daß während einer Entwicklung eines Systems Architektursimulation verwendet werden soll, bereits in der Initialphase erfolgen.

Entwurfsprozeß Der bereits realisierte Teil der Architektursimulation ist in den Entwurfsprozeß des DbW-Vorentwicklungsprojektes wie folgt eingebunden:

Nachdem das Funktionsnetz und Teile der Plattform in UML spezifiziert sind, werden die einzelnen Module im graphischen Programmierwerkzeug ETAS ASCET SD implementiert. Der Kommunikationsschedule wird in einem mehrstufigen Prozeß aus der Kommunikationsmatrix und den Laufzeitanforderungen gewonnen. Dieser Schedule fließt mit in die Middleware ein. Diese wird zusammen mit dem aus ASCET SD generierten Code auf die Steuergeräte gebracht. Am HiL-Prüfstand und in einem Versuchsfahrzeug werden die endgültigen Tests durchgeführt.

Die Architektursimulation setzt als zusätzliche Testmethode hauptsächlich während der Implementierung in ASCET SD an, soll zukünftig aber auch während der Funktionsnetzmodellierung zum Einsatz kommen.

Aus dem in ASCET SD vorliegenden *Verhaltensmodell* wird ein Funktionsmodell generiert, das Information über die Module, Prozesse und den erzeugten C-Code enthält, sowie ein Modell der Betriebssystemkonfiguration, das Tasks, Prozesse, deren Periode und Offset definiert. Zusätzlich fließt ein *Architekturmodell* ein, das entsprechend den Vorgaben in der GUI des Simulationswerkzeugs Cierto VCC modelliert wird. Aus der Kommunikationsdatenbank wird mittels ASAM FIBEX, einem Austauschformat für Schedulinginformationen, halbautomatisch das Task- und Kommunikationsschedule generiert. Dieses beinhaltet Signale und deren Zuordnung sowohl zu Frames und Slots, als auch zu Tasks und Steuergeräten. Zusammen mit dem sogenannten *Mapping*, also der Information, welche Funktionskomponente auf welcher Plattformkomponente ablaufen soll, und den Simulationseinstellungen, also der Festlegung über Prüfpunkte und Stimuli, wird ein *Simulationsmodell* generiert und ausgeführt. Die daraus gewonnenen Aufzeichnungen von Signalen, Busbelegungen und Betriebssystemabläufen werden analysiert liefern Erkenntnisse sowohl über die zu prüfende Aussage als auch über nicht vorhergesehene, systemimmanente Wechselwirkungen. Abbildung 2 beinhaltet eine Übersicht über die für die Architektursimulation verwendeten Modellarten und deren Generatoren.

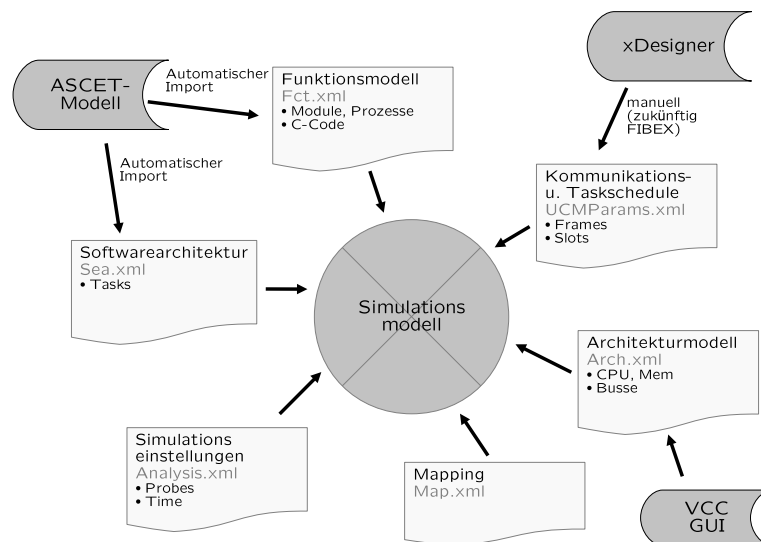


Abbildung 2: Artefakte für die Architektursimulation

Um Tests automatisch ablaufen lassen zu können, wurde eine Anbindung an ein Testautomatisierungswerkzeug implementiert, die die Möglichkeiten von Stimulierung und punktueller Untersuchung des Simulationsmodells bietet. Desweiteren werden alle Abläufe bezüglich Import von Funktionsmodellen und Aufbau des Simulationsmodells soweit als möglich automatisiert, um bei Verfügbarkeit der Modelle rasch Ergebnisse liefern zu können.