Herausforderungen und Erfahrungen

Modellbasierte Softwareentwicklung in der Bahntechnik

Die moderne Bahntechnik weist heutzutage einen hohen Grad an Automatisierung auf. So wird mit nur sieben Betriebsleitzentralen der gesamte Bahnfernverkehr auf Deutschlands Hauptstrecken gesteuert und überwacht. Elektronische Stellwerke müssen daher höchsten Anforderungen an Sicherheit, Verfügbarkeit und Wartbarkeit genügen.

Um eine hohe Qualität sicherzustellen, sind hohe Aufwände bei Tests, Begutachtungen und Zertifizierungen von Systemen vorgeschrieben. In der Bahntechnik wird das durch die europäische Norm CENELEC EN 50126 und ihren assoziierten Normen geregelt. Hinzu kommt, dass der Lebenszyklus bahntechnischer Produkte sehr lang ist – 20 oder 30 Jahre sind hier der Standard. Außerdem bestehen heutige Systeme zu einem wesentlichen Anteil aus Software. Innerhalb des Lebenszyklus eines Produktes ist es daher unvermeidlich, dass sich die zugrunde liegende Hardware, die Betriebssysteme und die eingesetzten Entwicklungstools während der Laufzeit ändern. Eine Möglichkeit, diese Herausforderungen zu meistern ist die modellbasierte Vorgehensweise, die in den letzten Jahren in der Softwareentwicklung klar an Bedeutung gewonnen hat.

Modellbasierte Softwareentwicklung

Zentrales Merkmal der modellbasierten Entwicklung ist der Einsatz von Modellen, die sich an den Problemen orientieren – anstatt an der Lösungsdomäne. Softwaresysteme werden durchgängig mit Hilfe von Modellen beschrieben, die eine aufeinander aufbauende Abstraktionshierarchie bilden und den gesamten Softwareentwicklungsprozess durchziehen. Die Übergänge zwischen den Modellen auf unterschiedlichen Abstraktionsebenen werden über Transformationen durchgeführt, die im Idealfall vollständig automatisiert sind. Durch diesen Ansatz kann in jedem SW-Entwicklungsabschnitt (SW, Software) von unnötigen Festlegungen abstrahiert werden, während besonders wichtige und kritische Aspekte explizit und frühzeitig modelliert werden. Zusätzlich können moderne Analyseverfahren (z. B. Model Checking zur formalen Verifikation) eingesetzt werden, das zu einer effizienten Entwicklung hochqualitativer Software führt. Die modellbasierte SW-Entwicklung stellt somit eine mittel- bis langfristig wirkende Maßnahme zur Steigerung der Produktivität in der SW-Entwicklung dar. Die Steigerung der Produktivität wird dadurch erreicht, dass die in der SW-Entwicklungsvergangenheit getrennte Entwurfs- und Implementierungsphase zu einem einzigen Prozessschritt zusammengeführt wird.

AUTOREN



Heike Burghardt ist Communication Consultant, Dr. Ralf Pinger Manager of Software-Initiative und Dr. Stefan Milius Expert for model-based Software Development. Alle drei arbeiten bei der Siemens AG, Industry Sector, Mobility Division, Rail Automation

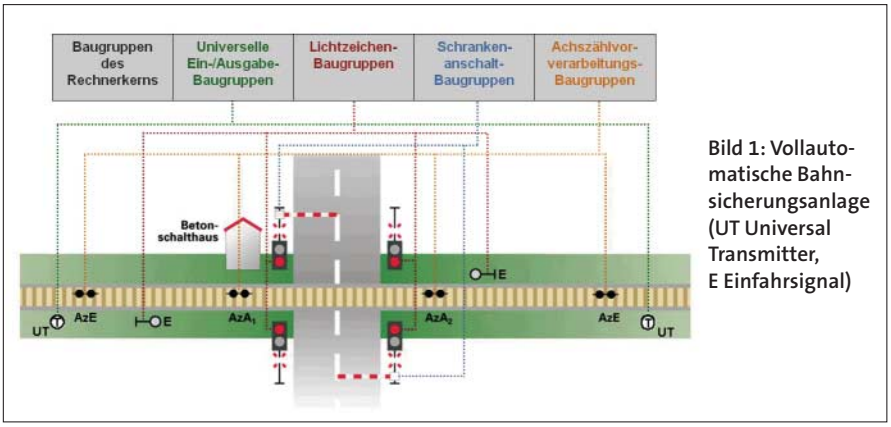

Bild 1: Vollautomatische Bahnsicherungsanlage (UT Universal Transmitter, E Einfahrsignal)

Treten Änderungen am System auf, werden diese ausschließlich im Modell durchgeführt. Informationen bezüglich der Anwendung oder der Systemumgebung werden also prinzipiell im Modell statt im Code gepflegt. Durch den höheren Abstraktionsgrad von Modellen, wird der Aufwand für Systemänderungen und für die Systemwartung erheblich reduziert. Dadurch wird die Softwareerstellung effizienter und Fehler, die sich durch eine Abweichung der Implementierung zum Modell ergeben, werden vermieden.

Rail Automation von Siemens

Für den erfolgreichen Einsatz der modellbasierten SW-Entwicklung ist ein durchgängiges Vorgehen unter Verwendung ausgereifter Entwicklungswerkzeuge von entscheidender Bedeutung. Im FutureLab der Software-Initiative von Rail Automation in Braunschweig wird der Einsatz der modellbasierten Entwicklung momentan anhand eines produktiven Entwicklungsprojektes erfolgreich erprobt (**Bild 1**). Das Beispiel eines vollautomatischen Bahnübergangs ist bereits komplex genug, um die Vorteile eines modellbasierten Ansatzes zur Geltung kommen zu lassen.
Die gezeigten Achszählsensoren (AzE Einschaltpunkt, AzA_1 und AzA_2 Ausschaltpunkte) beruhen auf dem induktiven Prinzip und werden durch Überfahrung eines Rades beeinflusst, wobei auch die Fahrtrichtung eines Rades ermittelt werden kann. Die Auswertung der Sensordaten findet im Systemrechner statt, der sich im Betonschalthaus befindet. Bei dem System, handelt es sich um einen sicheren Rechner nach dem Simis-Prinzip (Simis: Sicheres Mikrocomputersystem von Siemens), der den höchsten definierten SIL 4 (SIL, Sicherheitsintegritätslevel) erfüllt. Für die auf dem Rechner laufende Software gilt entsprechend der Software-SIL 4, das laut der Norm EN 50128 eine Reihe von Maßnahmen an den Softwareentwicklungsprozess impliziert. Diese Maßnahmen liefern eine weitere Motivation für den Einsatz modellbasierter Methoden.

Modellierung mit SCADE

Die Beschreibung der logischen Funktionalität der Softwarekomponenten erfolgt mit SCADE (Safety Critical Development Environment) von Esterel Technologies. SCADE ist eine umfangreiche Tool-Suite zur Entwicklung von Software für sicherheitskritische Systeme. Die zugrundeliegende Modellierungssprache erlaubt die Beschreibung von Software als deterministische zyklisch synchron getaktete Datenfluss- und Zustandsmaschinen. Neben einer rigorosen mathematischen Semantik haben die Sprachelemente grafische Repräsentationen, mit denen man im SCADE-Editor arbeitet. Dies führt zu einer sehr klaren und übersichtlichen Beschreibung der logischen Funktionalität der Software. **Bild 2** zeigt einen Teil des SCADE-Modells aus dem Pilotprojekt.
Die Modelle werden mittels eines Code-Generators in C-Code übersetzt. Die in Form von Modellen beschriebene Lösung ist hierbei völlig getrennt von der konkreten Programmiersprache. Ändert sich diese, muss lediglich der Codegenerator ausgetauscht werden. Die Modelle bleiben unverändert erhalten. Die Qualifizierung des Codegenerators für die Entwicklung sicherheitsbezogener Software nach den gängigen Normen aus der Luftfahrt ▶

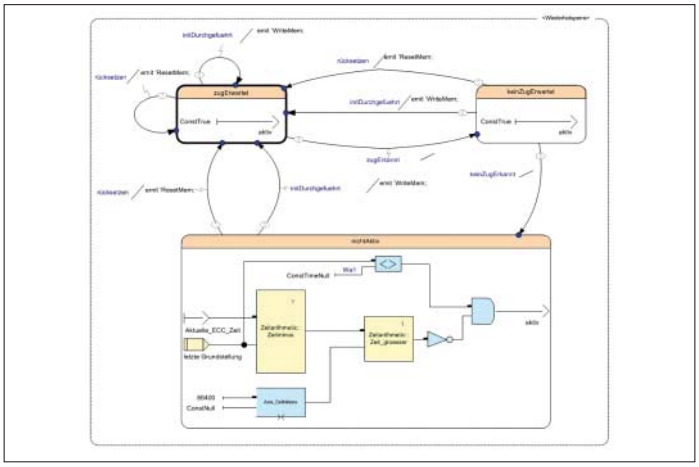

Bild 2: Beispiel eines SCADE-Modells

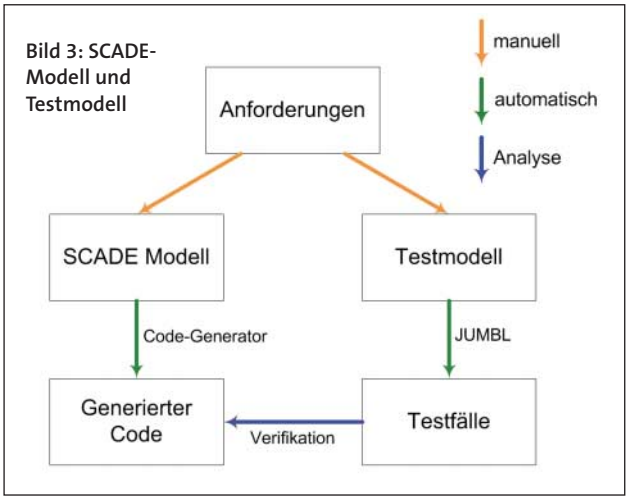

Bild 3: SCADE-Modell und Testmodell

und dem Eisenbahnwesen (DO-178B, Level A und nach CENELEC EN 50128 für SIL 4) wird ab Mitte Juli 2008 vorhanden sein. Zusammen mit dem bei Siemens Mobility in der Rail Automation eingesetzten zertifizierten Cadul-Compiler ergibt sich eine vollständige automatisierte und zertifizierte Generierung vom Modell bis ins Zielsystem. Es gibt aber auch ein paar Problempunkte – zum Beispiel sind im generierten Code die Schnittstellendaten nicht vollständig von den internen Daten getrennt.

Neben dem Code-Generator bietet SCADE einen Simulator, der es erlaubt, den generierten Code auf Modellebene zu testen und zu debuggen. Mit Hilfe des Requirements Gateway der SCADE-Suite werden die Anforderungen, die in DOORS (Dynamic Object Oriented Requirements System) erfasst und verwaltet werden, mit den entsprechenden Modellelementen verlinkt. Die durch die Normen geforderte bidirektionale Nachverfolgung zwischen Anforderungen und Modell ist somit gegeben.

Verifikation und statistisches Testen

Bei der modellbasierten Vorgehensweise werden Modelle insbesondere im Bereich der Verifikation eingesetzt. Im Pilotprojekt wird aus den Anforderungen ein statistisches Testmodell konstruiert. Aus diesem Testmodell werden mit Hilfe des vom Software Quality Research Laboratory der University of Tennessee entwickelten Tools JUMBL (Java Usage Model Builder Library) Testfälle generiert. **Bild 3** zeigt das grundsätzliche Vorgehen.

Wichtig ist, dass das Testmodell unabhängig von dem Implementierungsmodell in SCADE aus den Anforderungen abgeleitet wird. Dadurch wird durch die generierten Testfälle die Funktionalität im Implementierungsmodell getestet. Durch gezielte Veränderung der Wahrscheinlichkeiten, die in das Testmodell eingehen, können bestimmte Anwendungsszenarien häufiger oder weniger häufig getestet werden. Es hat sich gezeigt, dass durch dieses Vorgehen Fehler bereits in der Modellierungsphase erkannt werden, die sonst erst in späteren Phasen entdeckt würden. Im Zusammenhang mit dem Testmodell kommt auch das Model Test Coverage Feature der SCADE-Suite zum Einsatz, mit dem die strukturelle Abdeckung des SCADE-Modells durch die generierten Testfälle gemessen wird. Der Abdeckungsnachweis ist ebenfalls eine Forderung aus der Norm für den Software-SIL 4.

Problematisch bei der Verifikation ist die Einbettung der Modelle in klassisch codierte Software. Hierfür muss ein so genannter Wrapper implementiert werden, der die Schnittstellen zwischen den klassischen Softwarekomponenten und den Modellen ineinander überführt. Hier wird ein Integrationstest notwendig. Dieser Test geschieht innerhalb der vorhandenen Testumgebungen für die im Achszählprojekt der RT-Tester der Firma Verified International eingesetzt wird.

Zusammenfassung

Durch die Arbeit mit Modellen ergibt sich eine gute Trennung zwischen logischer Funktionalität der Software und der Hardwareplattform. Die Toolunterstützung vereinfacht das Nachverfolgen von Anforderungen sowohl in das Modell als auch zu den Testfällen. Da Modelle die Funktionalität vollständig spezifizieren und danach ein Code-Generator verwendet wird, gibt es keinen Bruch zwischen Modell und Code. Für die Bahntechnik ist wichtig, dass der Code-Generator einem sehr hohen Qualitätsanspruch gerecht wird, das durch die Qualifizierung begründbar wird. Zusätzlich erscheint der Einsatz moderner Analyseverfahren sehr vielversprechend. Leider ist zum jetzigen Zeitpunkt die Komponente für formale Verifikation in der aktuellen Version 6 der SCADE-Suite noch nicht vollständig einsetzbar – die Fertigstellung dieser Komponente ist für Ende 2008 geplant. Es bleibt offen, ob formale Verifikation lediglich der Testunterstützung dient oder, ob sich die Zertifizierung modellierter Software vereinfacht. Hierzu sind weitere Gespräche mit den Zertifizierungsbehörden erforderlich.

Insgesamt lässt sich feststellen, dass mit der modellbasierten Entwicklung ein deutlicher Gewinn an Effizienz in der Softwareentwicklung erreicht wird. Die erzeugte Software hat eine höhere Qualität. Fehler werden im Entwicklungsprozess viel früher aufgedeckt. Letztendlich wird erreicht, dass Produkte schnell und in hoher Qualität beim Kunden vorliegen. (jj)

infoDIRECT **509ei0908**

www.elektronik-industrie.de
- ▶ Link zu **Siemens AG**
- ▶ Link zu **Esterel Technologies**