

# Modellgetriebene Entwicklung eingebetteter Systeme mit Eclipse

Axel Terfloth, Lothar Wendehals

itemis AG  
Am Brambusch 15-24  
44536 Lünen  
axel.terfloth@itemis.de, lothar.wendehals@itemis.de

## 1 Integrierte Werkzeugketten für eingebettete Systeme

Entwicklungsmethoden und -werkzeuge für eingebettete Systeme und Enterprise Applikationen nähern sich zusehends an. Insbesondere die modellgetriebene Entwicklung hilft beim Umgang mit zunehmend komplexeren, eingebetteten Softwaresystemen. Zur Automatisierung der Entwicklung ist die Werkzeugunterstützung entscheidend. Die Open-Source-Entwicklungsplattform Eclipse stellt mit dem Eclipse Modeling Project eine leistungsfähige Infrastruktur zur Entwicklung der Werkzeuge. Außerdem ist Eclipse eine Integrationsplattform für Werkzeugketten. Anhand praktischer Beispiele werden wir Aspekte Eclipse-basierter MDSD-Werkzeugketten für eingebettete Systeme diskutieren.

### 1.1 Werkzeug- & Modellintegration

Die Werkzeuglandschaft für eingebettete Systeme ist von Speziallösungen bis hin zu universell einsetzbaren Entwicklungsumgebungen weit gefächert. Zur Integration der Modelle verschiedener Werkzeuge gibt es grundsätzlich zwei Methoden: ein gemeinsames Modell oder die Synchronisation der verschiedenen Modelle. Bei einem gemeinsamen Modell wird das spezialisierte Modell eines Werkzeugs durch Modell-zu-Modell-Transformation aus dem gemeinsamen Modell gewonnen. Nach der Bearbeitung werden Änderungen des spezialisierten Modells zurück in das gemeinsame Modell transformiert. Bei der Synchronisation einzelner Modelle werden dagegen Änderungen paarweise von einem in das andere Modell propagiert und eventuelle Transformationen durchgeführt.

### 1.2 Metamodellierung, DSLs und Editoren

Zur Entwicklung eingebetteter Systeme findet zunehmend die UML mit speziell zugeschnittenen Profilen wie MARTE Anwendung. Daneben sind jedoch auch etablierte Metamodelle wie Blockschaltssysteme, die Architecture Analysis and Design Language (AADL) oder Standards wie AUTOSAR im Automotive-Bereich relevant. Eine weitere Klasse sind die Domain Specific Languages (DSL), die sowohl in grafischer als auch textueller Form vorliegen können. Das Eclipse Modeling Project bietet eine Basis für DSLs. Das darin enthaltene Eclipse Modeling Framework (EMF) hat sich als Standard für Metamodellierung und -modelle etabliert. Neben grafischen Modellierungswerkzeugen werden auch Grammatik-basierte Ansätze zur textuellen Modellierung unterstützt.

### **1.3 Validierung**

Die sehr detaillierten Modelle eingebetteter Systeme erfordern eine frühe, automatisierte Überprüfung der Korrektheit, damit Auswirkungen von Änderungen nachvollziehbar bleiben und unnötige Build/Deploy/Test-Zyklen vermieden werden. Metamodelle können jedoch nur teilweise die syntaktische Korrektheit des Modells sicherstellen. Auch Einschränkungen bei der Freiheit der Modelleditoren sind nur bis zu einem gewissen Grad praktikabel. Einige syntaktische, aber vor allem semantische Vorgaben an das Modell können weder vom Metamodell noch von speziellen Editoren gewährleistet werden. Deshalb ist eine explizite Validierung des Modells durch teils sehr umfangreiche Regelsätze notwendig.

### **1.4 Simulation und Visualisierung**

Die Simulation von Teilsystemen ist wesentlich in der Entwicklung eingebetteter Systeme, da während der Softwareentwicklung das eigentliche Zielsystem für Tests oftmals nicht zur Verfügung steht. Die Simulation ist prinzipiell auf alle Verhaltensmodelle anwendbar. Existieren Simulations-Engines, können diese mit dem Editor gekoppelt werden. Dafür sollte die Notation der Modelle um zusätzliche dynamische Elemente zur Darstellung der Simulation erweitert werden. Bei der Simulation eines Zustandsautomaten können so zur Visualisierung zum Beispiel aktive Zustände oder durchgeführte Transitionen hervorgehoben werden.

### **1.5 Codegenerierung**

Die Codegenerierung für eingebettete Systeme unterscheidet sich bezüglich der zugrunde liegenden Techniken nicht wesentlich von der für Enterprise-Anwendungen. Da eingebettete Systeme oftmals jedoch sicherheitskritisch sind, sind formale Testverfahren, die unter anderem Code-Coverage-Messungen des Generator-Codes enthalten, notwendig. Bei der Produktlinien-Entwicklung, bedingt durch die extrem große Vielfalt der Zielplattformen, ist des Weiteren ein modularer Aufbau der Codegenerierungs-Templates hilfreich. Die für die Zielplattform benötigten Templates können dadurch gezielt zusammengesetzt werden. Durch Aspekt-orientierte Programmierung (AOP) können außerdem je nach Zielplattform zusätzliche Informationen in die Templates eingebracht werden. Xpand aus dem openArchitectureWare-Projekt unterstützt diese Techniken.

## **2 Das Beispiel**

Die angesprochenen Aspekte werden an einer konkreten, Eclipse-basierten Beispiel-Werkzeugkette vorgestellt, die als Open-Source verfügbar ist. Die Werkzeugkette umfasst Editoren zur grafischen Modellierung von Zustandsautomaten sowie zur grafischen und textuellen Modellierung von Blockschaltssystemen. Die Validierung und Simulation der Modelle sowie die Codegenerierung gehören ebenfalls zur Werkzeugkette. Die verschiedenen Modelle werden mit Hilfe von Modelltransformationen in die Werkzeugkette integriert.