

Funktionale Modellierung varianter mechatronischer Systeme

I. Dürrbaum¹, J. Petersen², K. Stübbe¹,
M. Bourhaleb², H. D. Kochs², A. Lapp¹, J. Schirmer¹, D. Ziegenbein¹

¹Robert Bosch GmbH, FV/SLN
70442 Stuttgart, Germany
[mail@ingo-duerrbaum.de]

²Universität Duisburg-Essen
Institut für Informationstechnik
47057 Duisburg, Germany

1 Einleitung

Der wachsenden Vernetzung und zunehmenden Variantenvielfalt mechatronischer Systeme im Kraftfahrzeug bei gleichzeitig kürzeren Produktzyklen und einem erhöhten Kostendruck lässt sich mit einer systemfamilienspezifischen Plattformentwicklung begegnen. Durch eine möglichst hohe Wiederverwendung im Sinne eines Baukastenprinzips können die Entwicklungszeiten und –kosten reduziert werden. Eine zentrale Rolle spielen dabei funktionale Varianten, die einen Ausgangspunkt für die Entwicklung von Produktfamilien darstellen. Wir skizzieren hier ein Modellierungskonzept für Funktionalitäten, welches Basisfunktionen und funktionale Varianten darstellbar macht. Zwischen der funktionalen Modellierung und den Varianten in der Software-Architektur (SWA) muss ein Zusammenhang hergestellt werden, der unter Berücksichtigung nicht-funktionaler Anforderungen zu einem durchgängigen Entwicklungsprozess führt (Abb. 1).

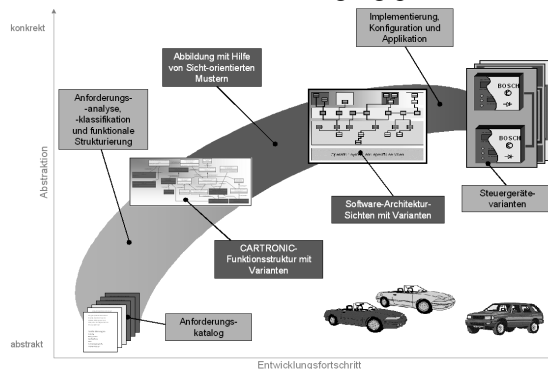


Abb. 1: CARTRONIC-basierter Entwicklungsprozess

2 Varianten in der Funktionsstruktur

Um die Komplexität interagierender Funktionalitäten zu beherrschen, wurde das Strukturierungskonzept CARTRONIC [La01][Pe02] entwickelt. Im Gegensatz zur bisherigen Praxis, Funktionalitäten weitgehend wechselwirkungsfrei zu spezifizieren und anschließend im Fahrzeug zu integrieren, unterstützt CARTRONIC die zunehmende Vernetzung von Funktionalitäten. CARTRONIC bietet die Möglichkeit, Funktionalitäten und funktionale Abhängigkeiten formal zu beschreiben und die Funktionalitäten nach logischen Kriterien zu Komponenten zusammenzufassen. Die funktionalen Komponenten werden durch eine hierarchische Dekomposition in ihrer Komplexität reduziert und mit abgestimmten funktionalen Schnittstellen ausgestattet.

Da CARTRONIC keinen eigenen Entwicklungsprozess definiert, ist es notwendig CARTRONIC in einen Entwicklungsprozess einzubinden. Im Rahmen einer Anforderungsanalyse werden zunächst Anforderungen als funktionale und nicht-funktionale und weiter als universelle und nicht-universelle Anforderungen klassifiziert (Abb. 3).

Die Klassifikation in universelle und nicht-universelle Anforderungen berücksichtigt die Tatsache, dass nach [Vo00] ein vollständiger Anforderungskatalog eine nicht formale Beschreibung **aller** Variabilitäten enthält. Diese motivieren in der weiteren Entwicklung funktional und nicht-funktional getriebene Varianten. Nach der Anforderungsklassifikation lassen sich die Funktionalitäten mit ihren funktionalen Varianten gesamtheitlich auf Basis von CARTRONIC [Pe02] strukturell modellieren.

Um die Komplexität variantenreicher Systeme zu reduzieren und Austauschbarkeit, Wiederverwendung und Skalierbarkeit im Sinne eines Baukastenprinzips zu ermöglichen, werden in einem CARTRONIC-basierten Entwicklungsprozess zunächst nur Systemanteile und Varianten betrachtet, die durch funktionale Anforderungen motiviert sind. Die Fokussierung auf funktionale Aspekte trennt dabei Funktionalität von der späteren Umsetzung – z.B. in Hard- oder Software – da derartige Entscheidungen i.A. nicht-funktional motiviert sind. Die Modellierung erfolgt in einem Systemfamilienmodell, welches die Spezifikationen der enthaltenen Systemmodelle vereinigt (Abb. 2).

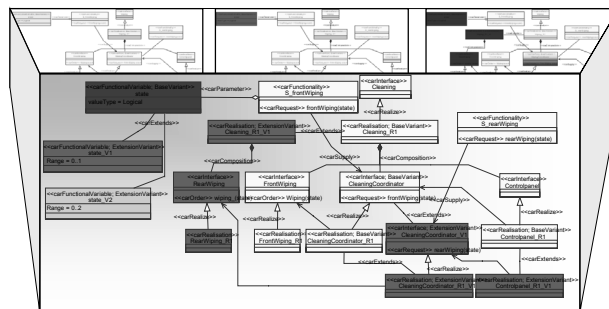


Abb. 2: Systemmodelle und Systemfamilienmodell

Um Variabilität zu berücksichtigen wird nach dem hier vorgestellten Verfahren, entsprechend [Dü03] dazu zunächst der funktionale Gleichanteil (universelle funktionale Anforderungen) als sogenannte „Basisvariante“ modelliert und so vom varianten Anteil

separiert. Die nicht-universellen funktionalen Anforderungen werden erst im Anschluss als sogenannte Erweiterungsvarianten modelliert und mit Hilfe eines speziellen Konnektors mit den Basisvarianten verbunden. Ist bei unterschiedlichen Erweiterungsvarianten ein Gleichanteil identifizierbar, so kann dieser mithilfe einer Variantenhierarchie modelliert werden. Mit Hilfe von Variationspunkten und/oder einer solchen Variantenhierarchie lassen sich Abhängigkeiten zwischen funktionalen Varianten eines funktionalen Aspekts unter dem Gesichtspunkt der möglichen Konfigurierbarkeit modellieren (vgl. Abb. 3). Durch diesen Ansatz lässt sich eine Separation von Gleichanteilen und varianten Anteilen erreichen, die eine explizite Modellierung von Variabilität zulässt. Eine Anwendung dieses Konzeptes führt einerseits zu einem Verständnis der Abhängigkeiten innerhalb einer varianten Funktionalität und andererseits zwischen den verschiedenen funktionalen Varianten eines Systemfamilienmodells. Erreicht wird dies durch die explizite Modellierung der Variabilität.

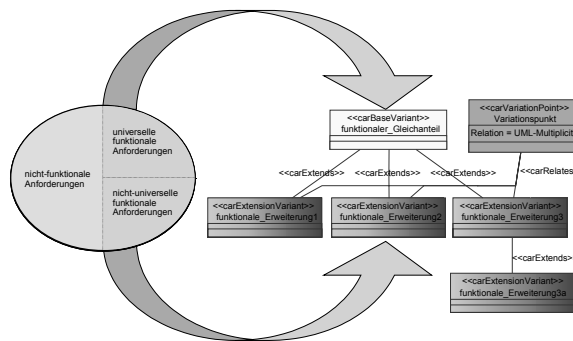


Abb. 3: Modellierungskonzept funktionaler Varianten

Durch die Betrachtung der verschiedenen Aspekte funktionaler Strukturierung nach [La01][Pe02] und der Anwendung des skizzierten Modellierungskonzeptes lässt sich eine funktionale Typisierung von Variabilität in funktionalen Modellen durchführen, die die in Tabelle 1 aufgeführten funktionalen Variantentypen identifiziert.

Tabelle 1: funktionale Variantentypen

Schnittstellenvariante
Dekompositionsvariante
Kommunikationsvariante
Funktionsvariante
Parametervariante

3 Zusammenhang zwischen Funktions- und Softwarevarianten

Softwarearchitektur wird i.A. durch einen Satz möglichst orthogonaler domänen-relevanter Sichten beschrieben, die in ihrem Aufbau mehr oder weniger stark voneinander abhängen und im wesentlichen durch nicht-funktionale Anforderungen

beeinflusst werden (vgl. [BCK98]). Jede verwendete Sicht kann sichtspezifische Variantentypen beinhalten, die sich aufgrund der Wechselwirkungen zwischen den Sichten gegenseitig beeinflussen und im Zusammenhang eine für den Architekturf Entwurf bedeutende Variantenart (Tabelle 2) hervorrufen. So kann beispielsweise eine funktionale Schnittstellenvariante im Architekturf Entwurf zu einer Schnittstellenvariante eines Komponentenmodells (statische Sicht) **und** einer Verhaltensvariante (dynamische Sicht) führen.

Tabelle 2: Variantenarten im Architekturf Entwurf

Applikationsvariante	Dekompositionsvariante
Kennlinienvariante	Scatteringvariante
Algorithmusvariante	Hardwarevariante
Typvariante	Bestückungsvariante
Testvariante	Konfigurationsvariante
Schnittstellenvariante	Verhaltensvariante

4 Ausblick

Um eine Abbildungsmöglichkeit varianter Funktionsstrukturen auf Softwarearchitektur zu ermöglichen, ist es notwendig, die möglichen Variantentypen der für eine Domäne relevanten Sichten zu identifizieren und den bekannten Variantenarten des Architekturf Entwurfs zuzuordnen. Außerdem müssen die zwischen den Sichten bestehenden Wechselwirkungen beschrieben werden, was nach [St02] eine vereinfachte Nachvollziehbarkeit von Entwurfsentscheidungen ermöglicht und die Auswirkungen von Änderungen eines Systemaspekts auf andere Aspekte des Systems identifizierbar macht. Ziel ist die Entwicklung von sicht-orientierten Mustern, die nicht-funktionale Anforderungen berücksichtigen und eine Unterstützung bei der Übertragung von varianter Funktionsstrukturen in die Sichten einer SWA-Beschreibung bieten können.

Literaturverzeichnis

- [La01] Lapp, A.; Torre Flores, P.; Schirmer, J.; Kraft, D.; Hermsen, W.; Bertram, T.; Petersen, J.: Softwareentwicklung für Steuergeräte im Systemverbund - Von der CARTRONIC-Domänenstruktur zum Steuergerätecode, 2001, VDI-Gesellschaft "Fahrzeug- und Verkehrstechnik", VDI-Berichte 1646 "Elektronik im Kraftfahrzeug", S. 249-276.
- [Pe02] Petersen, J.; Bertram, T.; Lapp, A.; Knorr, K.; Torre Flores, P.; Schirmer, J.; Kraft, D.; Hermsen, W.: UML Metamodel Extensions for Specifying Functional Requirements of Mechatronic Components in Vehicles. In: P. P. Hofmann, A. Schürr. OMER - Object-oriented Modeling of Embedded Real-Time Systems. Lecture Notes in Informatics. S. 84ff. Gesellschaft für Informatik, Bonn, 2002.
- [Vo00] Voget, S. et. al.: Behandlung von Variabilitäten in Produktlinien mit Schwerpunkt Architektur, Beitrag zum 1. Deutschen Produktlinien Workshop, IESE-Report No. 076.00/E, Kaiserslautern, 2000. http://www.iese.fhg.de/pdf_files/iese-076_00.pdf
- [Dü03] Dürrbaum, I. et. al.: Modellierung funktionaler Varianten von mechatronischen Systemen im Kraftfahrzeug. HdT Essen: TM. Nr. 2 (2003), S. 71-77, ISSN: 0040-1439
- [St02] G. Starke: Effektive Software-Architekturen. Hanser Verlag, München, 2002
- [BCK98] L. Bass, P. Clements, R. Kazman: Software Architecture in Practice. Addison-Wesley, 1998