

Ein Klassifikationsansatz zur Variabilitätsmodellierung in E/E-Entwicklungsprozessen

Cem Mengi und Ibrahim Armaç
Lehrstuhl für Informatik 3 - Softwaretechnik
RWTH Aachen, Ahornstr. 55, 52074 Aachen
{mengi | armac}@i3.informatik.rwth-aachen.de

Abstract: Das Erfassen und Verwalten von Varianten in E/E-Entwicklungsprozessen ist für Automobilhersteller mit einem großen Aufwand verbunden. Hinzu kommt, dass sie derzeit ihre Prozesse auf den AUTOSAR Standard anpassen. Die Auswirkungen in Bezug auf Variabilität sind dabei sehr vielfältig und können sehr komplex werden, wenn der Prozess nicht geeignet organisiert wird. In diesem Artikel wird daher ein Klassifikationsansatz vorgestellt, der die Auswirkungen von AUTOSAR im Hinblick auf Variabilität berücksichtigt. Dabei betrachten wir drei Ebenen: die Feature-Ebene, die Funktions-Ebene und die Architektur-Ebene. Durch die Klassifikation kann der Prozess besser strukturiert und analysiert werden, so dass Varianten in den unterschiedlichen Ebenen genauer beleuchtet werden.

1 Einleitung

Die Automobilindustrie bietet derzeit ihren Kunden beim Kauf eines Fahrzeugs individuelle Ausstattungsmöglichkeiten, wie beispielsweise Parkassistenten, Regensensoren und Komfort-Schließsysteme. Die Vielfalt an softwarebasierten Sonderausstattungen und ihre Kombinationen ermöglichen eine große Anzahl an Fahrzeugvarianten zu konfigurieren.

Für Fahrzeughersteller bedeutet dies, dass im Verlauf des E/E-Entwicklungsprozesses Varianten bzw. Variationspunkte, die unterschiedliche Varianten hervorrufen, in den Entwicklungsartefakten geeignet und effizient erfasst werden müssen.

Die Existenz von Varianten erstreckt sich im gesamten E/E-Entwicklungsprozess. Varianten können in der Anforderungsspezifikation, in der Systemspezifikation, im Architektur-entwurf, im Code, aber auch in der Test- und Integrationsphase vorhanden sein. Darüber hinaus entstehen Varianten auch im Laufe der Produktion und in der Betriebs- und Wartungsphase, so dass im gesamten Produktlebenszyklus, der bei einem Fahrzeug ca. 20-25 Jahre andauert, sehr viele und unterschiedliche Ausprägungen von Varianten entstehen können.

Offensichtlich ist das Erfassen und Verwalten von Variationspunkten mit einem erheblichen Aufwand verbunden. Eine einheitliche und nahtlose Methodik inklusive ihrer Werkzeugunterstützung, die den gesamten Produktlebenszyklus unterstützt, ist derzeit nicht im Einsatz.

Hinzu kommt, dass die Automobilindustrie derzeit ihre Prozesse auf den AUTOSAR

Standard abstimmt [aut], um der enorm steigenden Komplexität entgegen zu wirken [PBKS07, Bro06]. Eines der wichtigsten Entwicklungen im Standard ist, dass durch Einführung einer Abstraktionsschicht die Software von der Hardware entkoppelt werden kann. Die Auswirkungen auf den Entwicklungsprozess in Bezug auf Variabilität sind sehr vielfältig und können sehr komplex werden, wenn der Prozess nicht geeignet organisiert wird.

In diesem Artikel wird daher unter Berücksichtigung des AUTOSAR Standards und dessen Einfluss auf den E/E-Entwicklungsprozess im Hinblick auf Variabilität ein Klassifikationsansatz vorgestellt, so dass Varianten in den unterschiedlichen Ebenen genauer beleuchtet und analysiert werden können. Dabei betrachten wir drei Ebenen: die Feature-Ebene, die Funktions-Ebene und die Architektur-Ebene. Insbesondere werden dabei die Abhängigkeiten der Variationspunkte und der Konfigurationen zu spezifischen Varianten in den drei Ebenen genauer beschrieben. Weitere Phasen im Produktlebenszyklus, wie beispielsweise Test und Integration, werden nicht betrachtet.

Der Artikel ist wie folgt strukturiert. In Abschnitt 2 wird zunächst die Auswirkung des AUTOSAR Standards auf den E/E-Entwicklungsprozess im Hinblick auf die Handhabung der Variabilität untersucht. Auf Basis dieser Untersuchung wird in Abschnitt 3 ein Klassifikationsansatz vorgeschlagen, der die wesentlichen Prozessschritte bis zur Architektur-Ebene berücksichtigt. Schließlich wird in Abschnitt 4 der Artikel zusammengefasst.

2 Modellierung der Variabilität in E/E-Entwicklungsprozessen

Es gibt viele Techniken zur Modellierung der Variabilität. *Sinnema* und *Deelstra* haben diese in ihrem Artikel klassifiziert [SD07]. Im Wesentlichen unterscheiden sie drei Techniken: die Modellierung der Variabilität mit *Features*, mit *Use Cases* und mit anderen Techniken.

Features beschreiben dabei Merkmale des Systems. Ein Beispiel wäre die Zentralverriegelung im Auto. Durch eine Baumstruktur, die zusätzliche Sprachelemente zur Beschreibung der Variabilität besitzt, können Features organisiert werden. Mit Use Cases können funktionale Anforderungen des Systems und deren Beziehungen zu Aktoren in Anwendungsfällen beschrieben werden [uml]. Durch geeignete Erweiterungen der Modellierungselemente kann hier auch Variabilität ausgedrückt werden [vdML]. Auf zusätzliche Techniken wird hier nicht weiter eingegangen, sondern auf [SD07] verwiesen. In diesem Artikel beschränken wir uns auf die Modellierung der Variabilität mit Features, da sie insbesondere im Automobilbereich bereits Anwendung findet [TH02].

Bei der Betrachtung der Variabilitätsmodellierung im E/E-Entwicklungsprozess unterscheiden wir zwei Ansätze: den klassischen hardware-getriebenen Ansatz und den funktions-getriebenen Ansatz. In Abbildung 1 sind beide Ansätze beispielhaft dargestellt.

Untersuchen wir den hardware-getriebenen Ansatz (siehe Abbildung 1(a)), so werden die Merkmale des Systems und dessen Varianten mit Feature-Modellen erfasst [KCH⁺90]. Sie bieten die Möglichkeit vorgeschriebene, optionale, partielle und exklusive Features zu modellieren.

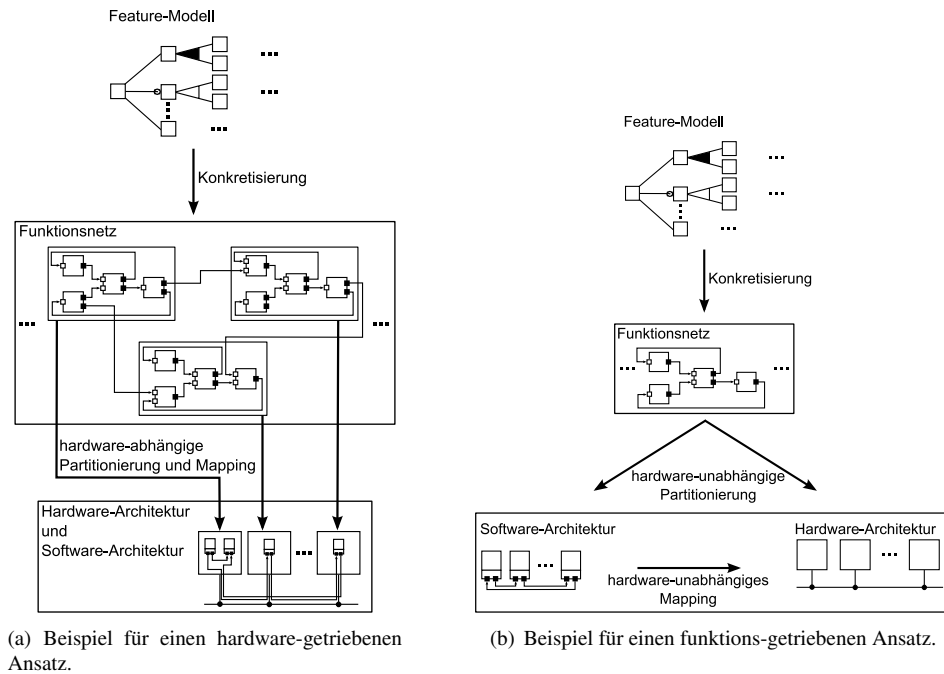


Abbildung 1: Zwei Beispiele für die Variantenhandhabung im E/E-Entwicklungsprozess.

In einer weiteren Phase werden in einem Funktionsnetz die spezifizierten Features modelliert [vdB04, GHK⁺08]. Ein Funktionsnetz besteht aus Funktionen, die miteinander kommunizieren. Sie stellt in dieser Phase die virtuelle Realisierung des Systems dar. Auf diese Weise lassen sich funktionale Anforderungen konkretisieren und Spezifikationsfehler frühzeitig erkennen. Außerdem wird der Einstieg in die Problematik erleichtert.

Varianten in Funktionsnetzen werden nicht explizit modelliert. Stattdessen wird ein maximales Funktionsnetz erstellt, das alle Features beinhaltet. Die Varianten werden dann durch Entfernen bestimmter Teile des Funktionsnetzes gebildet (in Abbildung 1(a) durch die inneren Rechtecke dargestellt). In diesem Ansatz ist diese Art der Variantenhandhabung möglich, da das Funktionsnetz mit dem Wissen über technische Details, wie beispielsweise die verwendete Kommunikationsinfrastruktur und Mapping-Informationen, entworfen wird. Daher kann das Entfernen bestimmter Teile des Funktionsnetzes mit dem Entfernen von Steuergeräten aus der Fahrzeugtopologie gleichgesetzt werden. Auf diese Weise werden Varianten hardware-abhängig gebildet.

Durch die Einführung des AUTOSAR Standards wird gleichzeitig ein Paradigmenwechsel von einem hardware-getriebenen Ansatz zu einem funktions-getriebenen Ansatz vollzogen. In Abbildung 1(b) ist ein auf den AUTOSAR Standard angepasster funktions-getriebener Ansatz beispielhaft dargestellt. Variable Merkmale des Systems werden weiterhin mit Feature-Modellen erfasst.

Funktionsnetze werden jetzt unabhängig von der Hardware modelliert, so dass hier ei-

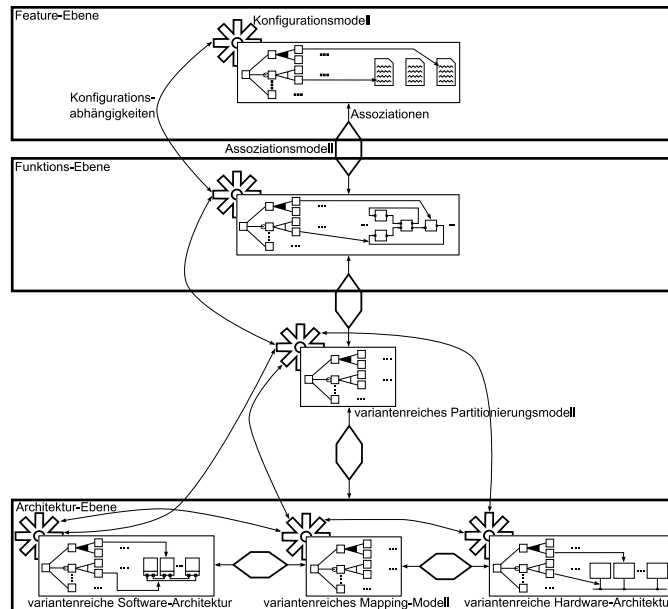


Abbildung 2: Ein Klassifikationsansatz zur Variabilitätsmodellierung in E/E-Entwicklungsprozessen auf Basis von AUTOSAR.

ne weitere Abstraktionsstufe eingeführt werden kann. Dies bewirkt, dass das Funktionsnetz deutlich kleiner wird, da technische Details vernachlässigt werden können. Varianten müssen allerdings mit einer alternativen Technik modelliert werden.

Weitere wesentliche Schritte auf die AUTOSAR im Hinblick auf Variabilität Einfluss hat, sind, die Partitionierung des Funktionsnetzes, die Modellierung der Software- und Hardware-Architektur, sowie deren Mapping.

3 Klassifikationsansatz zur Variabilitätsmodellierung in E/E-Entwicklungsprozessen

Berücksichtigen wir die Einflüsse des AUTOSAR Standards auf den E/E-Entwicklungsprozess hinsichtlich der Variabilität, so unterscheiden wir zwischen drei Ebenen: die Feature-Ebene, die Funktions-Ebene und die Architektur-Ebene (siehe Abbildung 2).

Auf der Feature-Ebene werden Varianten durch Feature-Bäume erfasst. Hierbei ist es wichtig Features mit Anforderungen in Beziehung zu setzen, um die Verfolgbarkeit und Konsistenz bei Änderungen gewährleisten zu können. Ein Konfigurationsmodell auf dieser Ebene leitet den Konfigurationsprozess ein. Dabei werden Informationen mit weiteren Ebenen geteilt, um Konfigurationsabhängigkeiten aufzulösen.

Auf der Funktions-Ebene werden die Funktionsnetze ohne technische Details entworfen. Dadurch kann abstrakter modelliert werden, was zu einer erheblichen Reduzierung hinsichtlich der Größe und der Anzahl an Varianten, die aufgrund technischer Details existierten, führt. Die für diese Ebene typischen Varianten können beispielsweise separat durch Feature-Bäume modelliert werden (angedeutet durch den Feature-Baum in Abbildung 2). Abhängigkeiten zwischen dem Variantenmodell und den Funktionsnetzen, sowie ein Konfigurationsmodell sind wesentliche Bestandteile.

Die Entscheidung ob eine Funktion vollständig als Software oder Hardware, bzw. als Software und Hardware realisiert wird erfolgt bei der Partitionierung. Insbesondere kann diese Variabilität einen enormen Kosteneinfluss haben. Eine gut gewählte Partitionierungsvariante kann die Entwicklungskosten erheblich senken.

Die Applikations-Software kann nun nach dem AUTOSAR Ansatz unabhängig von der Hardware entworfen werden. Variabilität in der Software-Architektur als auch in der Hardware-Architektur werden weitestgehend unabhängig behandelt. Beispielsweise können die Varianten *Standard-Zentralverriegelung* und *Komfort-Zentralverriegelung* in der Software-Architektur durch Feature-Bäume erfasst werden. In der Konfigurationsphase werden die entsprechenden Varianten so konfiguriert, dass die gewünschte Architektur modelliert wird. Da das Mapping nicht festgelegt ist, existieren vielfältige Möglichkeiten die Software auf die Hardware abzubilden. Während im hardware-getriebenen Ansatz die Standard- und Komfort-Zentralverriegelung durch zwei Steuergeräte realisiert werden musste, um die Komplexität so gering wie möglich zu halten, ist es jetzt möglich die Software auf einem Steuergerät laufen zu lassen. Genau wie für die Partitionierung, kann auch die Variabilität beim Mapping einen enormen Kosteneinfluss haben.

Die separate Variantenerfassung wird zwar anfangs mit mehr Aufwand verbunden sein, doch auf langer Sicht werden die Vorteile dieses Ansatzes insbesondere in Bezug auf AUTOSAR deutlich werden. Eine Klassifikation auf drei Ebenen scheint hier geeignet zu sein, da sie die wesentlichen Prozessschritte berücksichtigt.

4 Zusammenfassung

In diesem Artikel wurde ein Klassifikationsansatz vorgestellt, der unter Berücksichtigung des AUTOSAR Standards Variabilität im E/E-Entwicklungsprozess bis zur Architektur-Ebene beschreibt. Eine Einteilung in die Feature-, Funktions- und Architektur-Ebene scheint dabei geeignet zu sein, da sie die wesentlichen Prozessschritte beachtet. Welche Modellierungstechnik zur Erfassung der Variabilität in den jeweiligen Ebenen passend ist, wurde dabei nicht untersucht. Es scheint allerdings, dass ein Ansatz basierend auf Feature-Modellen anwendbar ist. Die separate Modellierung stellt eine wichtige Voraussetzung dar, um die Komplexität im eigentlichen Modell so gering wie möglich zu halten. Der Ansatz ermöglicht eine genauere Analyse in den Ebenen, so dass Lösungen entwickelt werden können, die auf die Problemdomäne zugeschnitten sind. Beispielsweise sind neue Techniken zur Variabilitätsmodellierung und ihre Werkzeugunterstützung in der Funktions-Ebene wichtige Bestandteile, um die Anwendung der Funktionsnetze zu fördern.

Literatur

- [aut] AUTOSAR Website. <http://www.autosar.org>.
- [Bro06] Manfred Broy. Challenges in automotive software engineering. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, Seiten 33–42, New York, NY, USA, 2006. ACM.
- [GHK⁺08] Hans Grönniger, Jochen Hartmann, Holger Krahn, Stefan Kriebel, Lutz Rothhardt und Bernhard Rumpe. Modelling Automotive Function Nets with Views for Features, Variants, and Modes. In *4th European Congress ERTS - Embedded Real Time Software*, 2008.
- [KCH⁺90] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak und A. S. Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Bericht, Carnegie-Mellon University Software Engineering Institute, November 1990.
- [PBKS07] Alexander Pretschner, Manfred Broy, Ingolf H. Kruger und Thomas Stauner. Software Engineering for Automotive Systems: A Roadmap. In *FOSE '07: 2007 Future of Software Engineering*, Seiten 55–71, Washington, DC, USA, 2007. IEEE Computer Society.
- [SD07] Marco Sinnema und Sybren Deelstra. Classifying variability modeling techniques. *Information and Software Technology*, 49(7):717–739, July 2007.
- [TH02] Steffen Thiel und Andreas Hein. Modeling and Using Product Line Variability in Automotive Systems. *IEEE Softw.*, 19(4):66–72, 2002.
- [uml] Unified Modeling Language Website. www.uml.org.
- [vdB04] Michael von der Beeck. Function Net Modeling with UML-RT: Experiences from an Automotive Project at BMW Group. In Nuno Jardim Nunes, Bran Selic, Alberto Rodrigues da Silva und José Ambrosio Toval Álvarez, Hrsg., *UML Satellite Activities*, Jgg. 3297 of *Lecture Notes in Computer Science*, Seiten 94–104. Springer, 2004.
- [vdML] T. von der Maßen und H. Lichter. Modeling Variability by UML Use Case Diagrams. In *IEEE Joint International Requirements Engineering Conference (RE02), Essen, Germany*, Proc. of International Workshop on Requirements Engineering for Product Lines.