

Merkmalsmodellierung für Anforderungsartefakte mit Doors®

Ekaterina Boutkova¹, Dr. Shahram Hami-Nobari¹, Benjamin Kolb², Laura Löffler³

¹Group Research and Advanced Engineering, Daimler AG, Germany

²sprylab technologies GmbH

³Daimler Protics GmbH

^{1,3}{ekaterina.boutkova, shahram.hami-nobari, laura.loeffler}@daimler.com

²benjamin.kolb@sprylab.com

Einführung

Alle Industriezweige sind heutzutage durch eine große Anzahl von Produktvarianten geprägt. Die Unternehmen wollen mit Hilfe der Variantenvielfalt Wettbewerbsvorteile sichern bzw. gewinnen. Eine wesentliche Voraussetzung für Wettbewerbserfolg ist, dass die Entwicklung mehrerer Produktvarianten die Entwicklungszeit nicht verlängert. Hierfür stellt ein effektives und effizientes Variantenmanagement, welches die Wiederverwendung aller Entwicklungsartefakte ermöglicht, eine wichtige Grundlage dar.

Spezifikationen hoch-komplexer Produkte sind sehr umfangreich [3,4] und deren Erstellung und Pflege ist kostenintensiv. Gleichzeitig bieten sie ein großes Wiederverwendungspotential.

Problem

Für die Wiederverwendung von Anforderungen der hoch-komplexen Produkte ist sowohl die methodische als auch die Werkzeugunterstützung nötig.

Bei einer Analyse der existierenden Ansätze bei der Daimler AG wurden mehrere Verbesserungspotentiale entdeckt [1]. Die zwei wichtigsten Probleme entstehen durch den hohen manuellen Aufwand bei der Zuordnung von Anforderungen zu Varianten und die fehlende Werkzeugunterstützung für die Zuordnungsüberprüfung.

Die Analyse der Forschungsmethoden hat gezeigt, dass das größte Potenzial im Bereich der Variabilitätsmodellierung natürlichsprachlicher Spezifikationen die Merkmalsmodellierung [5] hat. Um die Merkmalsmodellierung anwenden zu können, wurde allerdings ein Paradigmenwechsel notwendig - die Anforderungen werden nicht den Produkt-Varianten sondern den Produkt-Merkmalen und deren Ausprägungen zugeordnet [2].

Die Analyse der Werkzeuge, die mit dem bei der Daimler AG eingesetzten Tool zur Anforderungsdokumentation, IBM Rational Doors® [6], kompatibel sind, hat gezeigt, dass auf dem Markt verfügbare Werkzeuge nicht in allen Punkten den Bedürfnissen der Daimler AG entsprechen. Ein wichtiger Aspekt ist, dem Benutzer zu ermöglichen mit dem gleichen

Werkzeug, in dem er seine Anforderungen dokumentiert und verwaltet, die Produktvarianten zu konfigurieren und zu verwalten. Da keines der betrachteten Werkzeuge diesen Aspekt erfüllt, wurde eine Erweiterung des bereits eingesetzten Werkzeugs Doors angestrebt, die alle notwendigen Anforderungen der Daimler AG erfüllt.

Varianten Modul

Zur Lösung des vorgestellten Problems wurde das sogenannte Varianten-Modul (VM) entwickelt. VM ist ein formales Doors-Modul, das die Entwickler bei der Erstellung der Spezifikationen von Produktvarianten, die in Doors dokumentiert wurden, unterstützt.

Für die Darstellung von Merkmalen und der dazugehörigen Beziehungen wurde das Attribut *Object Type* und Doors-proprietäre Links verwendet. Dem Benutzer stehen folgende Werte des Attributes *Object Type* zur Verfügung:

- *Variant Document*: Dieser Wert markiert das erste Objekt im VM. Es ist die Voraussetzung für die Herstellung der Verknüpfung zum Spezifikationsmodul.
- *Criteria*: Dieser Wert wird den Merkmalen zugeordnet. Ein Merkmal ist eine Eigenschaft eines Produktes.
- *Criteria Value*: Jedes Merkmal hat mindestens zwei (disjunkte) Ausprägungen.
- *Combination*: Eine Kombination ist eine Auswahl der Merkmalsausprägungen unter Berücksichtigung der Regeln. Die einzelnen Merkmalsausprägungen werden über Links miteinander verknüpft (AND).
- *Configuration*: Eine Konfiguration ist die Selektion von Merkmalsausprägungen und/oder Kombinationen, die in einer bestimmten Produktvariante enthalten sind. Die einzelnen Elemente werden über Links miteinander verknüpft (AND).
- *Negation*: Negation ist ein Hilfskonstrukt, das den logischen Operator NOT darstellt. Der ausgehende Doors-Link vom Negationsobjekt aus zeigt auf die negierte Merkmalsausprägung.
- *Group*: Gruppierung ist ebenfalls ein Hilfskonstrukt und stellt den logischen Operator XOR dar.

- **Rule:** Eine Regel beschreibt die Beziehungen zwischen zwei Merkmalsausprägungen. Der eingehende Link auf *Rule* wird als Bedingung (IF) und der ausgehende Link von *Rule* auf einer Ausprägung als Wirkung (THEN) interpretiert. Zusätzlich stellt VM folgende fünf Attribute zur Verfügung:
 - **Object Heading and Object Text** ermöglicht es, die Merkmale, Regeln, Kombinationen und Konfigurationen textuell zu beschreiben;
 - **Short Text** ermöglicht es, die Texte aus dem vorherigen Attribut durch Abkürzungen zu ersetzen, um Formeln lesbar zu gestalten.
 - **Visible**, mit den Werten *true* und *false*, erlaubt die Steuerung des Merkmalsimports in das Spezifikationsmodul.
 - **Simple Formula** stellt die formale Zusammensetzung der *Negationen*, *Groups*, *Rules*, *Combinations* und *Configuration* dar.
 - **Input Criteria Generator** ermöglicht die Anotation von nicht disjunkten Merkmalen. Die Ausprägungen sind *Target Container*, *Source Container* und *Source Criteria*.

Der Prozess der Variantendokumentation besteht aus fünf folgenden Schritten:

1. **Merkmale identifizieren:** Während des Spezifikationsreviews werden von Experten variable (nicht für alle Produktvarianten gültige) Anforderungen identifiziert. Der Grund für die Anforderungsvariabilität ist ein Produktmerkmal bzw. Merkmalsausprägung. Die variablen Anforderungen können mit einem Variationspunkt markiert werden, um später die Zuordnung zu Merkmalen zu erleichtern.

2. **Merkmale dokumentieren:** Die identifizierten Merkmale und deren Ausprägungen werden in dem VM dokumentiert. Die Hierarchie zwischen Objekten mit dem Object Type Criteria und Criteria Value beschreibt die Vater/Kind-Beziehungen zwischen gefundenen Merkmalen und deren Ausprägungen. Dabei werden nur die untersten Vater/Kind-Beziehungen eines Merkmalsmodells durch die Doors-Objekt-Hierarchie abgebildet. Alle anderen (sowohl hierarchischen als auch nicht-hierarchischen) Beziehungen des Merkmalsmodells müssen mit Hilfe von *Rules* abgebildet werden.

3. **Anforderungen zu Merkmalen zuordnen:** Das Lastenheftmodul muss mit dem entsprechenden VM verknüpft werden. Es ist möglich ein VM mit mehreren Lastenheftmodulen zu verbinden. Ein Lastenheftmodul kann sich jedoch nur auf ein VM beziehen. Die Verknüpfung erfolgt über ein Doors-proprietäres Linkmodul. Die Merkmale werden über die Importfunktion als Doors-Attribute des Typs Multi-Enumeration angelegt. Die Merkmalsausprägungen stehen dann als Werte des Attributes zur Verfügung. Initial gelten die Anforderungen für alle Merkmale. Die variablen Anforderungen werden den relevanten Merkmalen bzw. Merkmalsausprägungen zugeordnet.

Dabei stehen mehrere Funktionen (*Apply to all Children*, *Apply to Parent*) zur Verfügung, die diesen Vorgang unterstützen. Außerdem stehen mehrere Funktionen, die die semi-automatische Validierung der Zuordnung ermöglichen, zur Verfügung.

4. **Konfigurationen erstellen:** Benötigt der Entwickler eine neue Produktvariante so kann er eine Konfiguration im VM anlegen. Zur Unterstützung des Entwicklers steht die syntaktische Validierung der erstellten Konfiguration sowie des VMs bereit.

5. **Spezifikation für neue Produktvariante ableiten:** Wird eine Spezifikation für eine neue Produktvariante benötigt, so kann diese innerhalb weniger Minuten erstellt werden. Im VM wird die neue Produktvariante mit den eigenen Merkmalsausprägungen als Konfiguration angelegt und in das Spezifikationsmodul als Doors-Filter importiert. Für die Erstellung der Spezifikation der neuen Produktvariante muss nur der entsprechende Doors-Filter aktiviert werden.

Ausblick

Die Erweiterung der Doors-Funktionalitäten ermöglicht die Unterstützung der Entwickler bei der Merkmalsmodellierung in den Anforderungsdokumenten. Die vorgestellte Erweiterung ist hauptsächlich für die Dokumentation von Merkmalsmodellen mit wenigen Merkmalen (ca. 10 Merkmalen und jeweils weniger als 15 Ausprägungen) und nur wenigen Beziehungen zwischen Merkmalen geeignet. Allerdings entsprechen die meisten Merkmalsmodelle auf System- bzw. Komponenten-Ebene diesen Restriktionen, so dass die vorgestellte Vorgehensweise bereits in mehreren aktuellen Entwicklungsprojekten erfolgreich evaluiert wurde. Die Weiterentwicklung von VM beschäftigt sich mit der Unterstützung des Änderungs- und Versionsmanagements sowie der Entwicklung einer GUI.

Referenzen

- [1] Boutkova, E.: Variantenmanagement in Anforderungsdokumenten: State of the Practice. In: Softwaretechnik-Trends 29(1), 2009, pp.4-5.
- [2] Boutkova, E.: Herausforderungen für Variantenmanagement in Anforderungsdokumenten. In: PIK 2010, Paderborn, 2010.
- [3] Houdek, F.: Challenges in Automotive Requirements Engineering. Industrial Presentations by REFSQ 2010, Essen, 2010.
- [4] Leuser, J., Ott, D.: Tracking Semi-automatic traceability in large specifications. In: R. Wieringa and A. Pesson (Hrsg.): REFSQ 2010, LNCS 6182, 2010, pp. 203-217.
- [5] Kang, K.C., Cohen, G., Hess, S.J.A., Novak, W.E., Spencer, A.: Feature-Oriented Domain Analysis (FODA) Feasibility Study. Peterson (1990)
- [6] IBM Rational Doors <http://www-142.ibm.com/software/products/de/de/ratidoor>