

Software-Varianten im Griff mit textuellen DSLs

Erfahrungsbericht

Johannes Reitzner

CoC Model Driven Development
msg systems ag
Robert-Bürkle-Straße 1
85737 Ismaning
johannes.reitzner@msg-systems.com

Abstract:

Anhand eines Projektes aus der Automobilindustrie wird gezeigt, wie textuelle DSLs zur Beherrschung von länderspezifischen Varianten eines Anwendungssystems eingesetzt werden können. Bei konventionellen Modellierungs- und Programmieransätzen besteht in solchen Fällen die Gefahr vieler Softwareversionen. Die Weiterentwicklung und Wartung wird dabei mit jedem zusätzlichen Land erschwert. Mit aktivem Management der Länderspezifika in Featuremodellen, kombiniert mit geeigneten DSLs und Tools, kann die Entwicklung weiterer Ländervarianten deutlich beschleunigt werden.

1 Ausgangssituation

Für einen Automobilhersteller soll zur Unterstützung des Planungsprozesses von Bestellvorschlägen ein Anwendungssystem erstellt werden, das in mehrere Länder ausgerollt werden soll. Die Planungskernfunktionalität ist für alle Länder gleich, aber es gibt länderspezifische Anforderungen, die sich in geänderten fachlichen Abläufen, Datenstrukturen und Benutzeroberflächen widerspiegeln.

2 Ziele der technischen Umsetzung

Wesentliches Ziel bei der Konzeption und technischen Umsetzung ist es aus einer einzigen Codebasis für jede Landesvariante ein Softwaresystem erstellen zu können, das keine Codeteile enthält, die nur für andere Ländervarianten relevant sind.

Damit wird ein weiteres Ziel unterstützt, nämlich die speziellen Anforderungen und die damit verbundenen technischen Auswirkungen für später neu hinzukommende Länder aktiv managen und kontrollieren zu können.

3 Lösungsansatz und Vorgehen

Zur technischen Unterstützung des aktiven Variantenmanagements wurde ein modell- und generatorgetriebener Lösungsansatz gewählt.

Sämtliche Anforderungen aller Länder werden dahingehend untersucht, ob sich sogenannte Variationspunkte identifizieren lassen, an denen sich ein Landessystem unterschiedlich zu einem anderen verhält. Diese Variationspunkte werden in einem Featuremodell mit dem Eclipse-Werkzeug `pure::variants` modelliert und gepflegt. Aus dem Featuremodell wiederum entstehen durch Selektion bestimmter Features landesspezifische Variantenmodelle.

Die Brücke zur Technik wird durch eine TMF/Xtext basierte Architektur- und Design spezifische Sprache (ADSL) geschlagen, die komponentenorientierten Entwurf und direkte Bezüge von Architekturartefakten zum Featuremodell ermöglicht.

Codegeneratoren erzeugen aus der Kombination von ADSL- und Featuremodell wesentliche Teile der allgemeinen Codebasis, aus der durch Auswertung eines Variantenmodells das zugehörige landesspezifische System durch automatisierte Tailoring-Workflows „zurechtgeschnitten“ und anschließend assembliert wird.

4 Erfahrungen

Der skizzierte Entwicklungsweg hat sich als praktikabel und effizient erwiesen.

Wesentlich dazu beigetragen hat der Aufbau der benötigten Werkzeugkette als durchgängige, vollständig in Eclipse integrierte Workbench, die Toolbrüche vermeidet. Diese umfasst neben `pure::variants`, Xtext und `openArchitectureWare` weitere Eclipse-Plugins für das featureabhängige Zurechtschneiden von manuell erstelltem bzw. generiertem Code.

Die Möglichkeit Modelle schon frühzeitig während ihrer Erstellung IDE gestützt validieren zu können, steigert die Effizienz im Projekt durch weniger fehlschlagende Modellauswertungsläufe.

Verbesserungspotenzial gibt es bei der Unterstützung der Entwickler während der manuellen Erstellung von variantenabhängigen Codeteilen. Hier wären Code-Sichten innerhalb der IDE sehr hilfreich, die dem Zustand nach dem Code-Tailoring auf Basis eines Variantenmodells entsprechen.

Für das Entwicklungsteam hat sich die grafische Visualisierung der textuellen DSL-Modelle als sehr wichtig erwiesen. Diese Funktionalität wird von einem neu entwickelten Eclipse-Plugin bereit gestellt, das durch leicht zu schreibende Modell zu Modell Transformationen für beliebige EMF-Modelle allgemein nutzbar ist.