

Modellbasierter Datenaustausch

Das Beispiel GDV-Datensatz

Für den Austausch von Vertrags- und Antragsinformationen zwischen Versicherungsunternehmen und Vermittlern ist eine Standardisierung notwendig. Das Standardformat dafür bietet der GDV-Datensatz – doch bevor er den Austausch vereinfacht, sind spezielle Verfahren zum Einlesen und Verarbeiten notwendig, um eine richtige Zuordnung zu den Datenmodellen der jeweiligen Bestandssysteme sicherzustellen. In diesem Beitrag wird eine modellbasierte Methode vorgestellt, die außer der Implementierung auch eine vollständige und aktuelle Dokumentation erzeugt.

Versicherungsprodukte bestehen, vereinfacht gesagt, aus Dokumenten und Geldflüssen. Nach dem Abschluss eines Versicherungsvertrags werden die benötigten Daten in die Bestandssysteme der Unternehmen übertragen, ab hier erfolgt die Abwicklung der Geschäftsprozesse elektronisch. Solange die Daten in einem System verbleiben, ist dies problemlos möglich, da die einzelnen verwendeten Datenformate bekannt und intern beherrschbar sind. Verlassen die Daten die Grenzen des Bestandssystems, wird die elektronische Kommunikation schwieriger. Einer der entscheidenden Erfolgsfaktoren für effiziente Geschäftsprozesse ist jedoch ein reibungsloser elektronischer Datenaustausch. Der Gesamtverband der deutschen Versicherungswirtschaft e.V. (GDV) ist ein Treiber für die Standardisierung von Datenformaten. Besonders die elektronische Datenverteilung von Versicherungsunternehmen an die

Vermittler wurde durch den GDV-Datensatz „VU-Vermittler“ standardisiert. Dieser Datensatz kann Vermittlern Versicherungsdaten in Form von standardisierten Bestands-, Inkasso- oder Schadeninformationsdatensätzen zur Verfügung stellen.

Modellbasierter Ansatz für die Abbildung des Datenaustauschs

Auszuarbeiten bleibt jedoch die Abbildung zwischen dem Datenmodell des jeweiligen Bestandssystems und dem GDV-Datensatz – wie transformiert man die Daten von einem Format zum anderen? Außerdem stellt sich im Rahmen von Änderungen und Erweiterungen auch die Anforderung, die Dokumentation der Abbildung zwischen Bestandsdatenmodell und GDV-Datensatz stets aktuell zu halten. Um diese Aufwände zu minimieren, wurde ein modellbasierter Ansatz zur Realisierung von Implementierung und Dokumentation entwickelt. Modellbasierte Ansätze haben in einigen Bereichen der Softwareentwicklung zu einer deutlichen Verbesserung geführt – sowohl in Bezug auf die Produktivität als auch im Hinblick auf Kosten und Zeit. Der Vorteil liegt in der durchgängigen Verwendung eines Modells durch alle Beteiligten. Dieses Modell dient oft als Grundlage der Implementierung und als Teil der Dokumentation, da durch die Verbindung zum Code Inkonsistenzen vermieden werden.

Automatisierte Quellcode-Erzeugung durch ein Meta-Modell

Zur Modellierung werden zwei Modell-Ebenen verwendet: Auf der einen Ebene liegt die Instanz eines Modells, auf der anderen liegt das Meta-Modell der Instanz. Auf Basis dieses Meta-Modells kann automatisiert kompilierbarer Software-Quellcode erzeugt werden. Dadurch lassen sich Modelländerungen schnell in neue Programmversionen übertragen. Durch den hohen Automatisierungsgrad der Quellcodeerzeugung ist außerdem die Fehleranfälligkeit minimiert. Die Trennung des Modells von generiertem Software-Quellcode ergibt darüber hinaus Plattformunabhängigkeit. Die Möglichkeiten der modellgetriebenen Entwicklung stehen und fallen damit, wie mächtig die verwendeten Werkzeuge sind. Je mehr Programmieraufwand durch Werkzeuge abgenommen wird, desto schneller kann die Entwicklung neuer Versionen erfolgen. Ein Beispiel für ein potentes Werkzeug ist das Eclipse Mode-

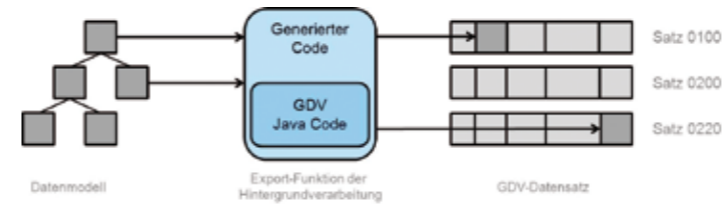


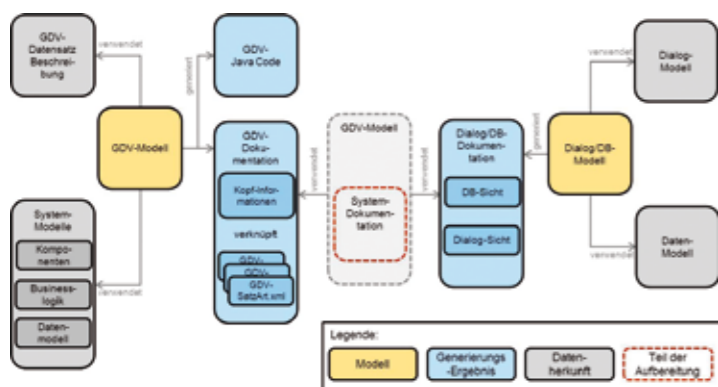
Abbildung 2

ling Framework: Durch Integration der Modellierungs- und Implementierungsumgebung und Unterstützung durch Generierungswerkzeuge schafft es eine einheitliche Basis für die Softwareentwicklung. Ein Wechsel des Werkzeugs zwischen Entwurfs- und Implementierungsphase ist daher nicht mehr notwendig. Das in Abbildung 1 dargestellte GDV-Modell basiert auf der Zuordnung zwischen der jeweiligen GDV-Satzartebene und der Datenherkunft auf Seiten des Bestandssystems.

Aus dieser Zuordnung werden zu jeder Satzart sowohl Implementierung als auch Dokumentation generiert. Die generierte Implementierung wird innerhalb der GDV-Export-Funktion zur Erzeugung der GDV-Sätze eingesetzt – siehe Abbildung 2. Dieses GDV-Modell lässt sich mit weiteren Modellen der Anwendung kombinieren. Dadurch können auch unterschiedliche Sichten generiert werden, z.B. zwischen Dialog, GDV und Datenmodell – dies kann als eine Form der Systemdokumentation verwendet werden. In Abbildung 1 (rechte Seite) ist hierzu ein Dialog/DB-Modell zu sehen, das eine Zuordnung zwischen Dialog- und DB-Informationen ermöglicht. Zur Extrahierung dieser Modelle kann auf bereits bestehende modellbasierte Ansätze zurückgegriffen oder eine automatisierte Codeanalyse eingesetzt werden. Eine Integration der Generierungsschritte in den Buildprozess der Anwendung stellt auch während der Entwicklung eine Zuordenbarkeit zwischen Implementierung und Dokumentation sicher. Der Einsatz des modellbasierten Datenaustauschs bei intersoft bietet viele Vorteile: Wie die Erfahrungen zeigen, verkürzt sich die Entwicklungszeit. Anforderungen können anhand des Modells geklärt werden und finden so schneller ihren Weg in die Realisierung. Frühzeitige Aussagen über Arbeitsaufwände sind möglich. Basierend auf der Modellierung können Implementierung und Dokumentation generiert werden. Die Fehlerhäufigkeit hat hierdurch spürbar abgenommen.

Der modellbasierte Datenaustausch bietet zahlreiche Vorteile:

- Durchgängige Verwendung eines Modells durch alle Beteiligten. Änderungen lassen sich schnell in neue Programmversionen übertragen. Ständiger Aktualität der Dokumentation ist garantiert.
- Es ergibt sich eine hohe Nachverfolgbarkeit, da Dokumentationsinformationen und Anforderungen auf einer sehr feingranularen Ebene zuordenbar sind.
- Folgeaufwände ergeben sich nur durch Änderungs- oder neuartige Anforderungen an der Schnittstelle. Beispiele dafür sind die Realisierung neuer Sparten oder Erweiterungen der Funktionalität bestehender Softwarekomponenten.
- Das Modell dient als zentrale Ablage: Hier sind Informationen zur Code- und Dokumentengenerierung abgelegt. Aktualisierungen des Datenmodells erfolgen automatisch durch den Softwareentwicklungsprozess.
- Durch den Einsatz von vor- und nachgelagerten Automatisierungskomponenten ist eine Grundqualität gewährleistet. Fehler lassen sich schnell und gezielt zurückverfolgen.



Modellbasierter Ansatz für die Abbildung des Datenaustauschs



Autoren:
Clemens große Kreymborg
Teamleiter im Bereich Softwarearchitektur bei der intersoft AG



Dipl.-Ing. (FH)
Ronny Heinsberg
Softwarearchitekt bei der intersoft AG