

Modellgetriebene Integration von Altsystemen

Dieter Hildebrandt¹

Michael Gründler²

Heiner Feislachen²

¹Betriebliches Informationsmanagement, OFFIS Institut für Informatik
Escherweg 2
D-26121 Oldenburg
dieter.hildebrandt@offis.de

²Softwarelösungen, BTC AG
Escherweg 5
D-26121 Oldenburg

michael.gruendler@btc-ag.com, heiner.feislachen@btc-ag.com

Abstract: Modellgetriebene Entwicklungsverfahren zur Integration bestehender heterogener betrieblicher Informationssysteme können insbesondere für KMU eine wesentliche Unterstützung darstellen, da sie den flexiblen Umgang mit sich rasch ändernden Anforderungen vor dem Hintergrund gewachsener Anwendungslandschaften erlauben.

In dem vorliegenden Beitrag wird ein Vorgehen zur modellgetriebenen Integration von Altsystemen beschrieben. Bei diesem Vorgehen werden Altsysteme bereits auf der Ebene von Geschäftsprozessen integriert, wobei die Geschäftsprozesse mit einer für diesen Zweck geschaffenen domänenspezifischen Sprache modelliert werden. Neben der Darstellung des Vorgehens liegt der Fokus dieses Beitrages auf der Beschreibung von Aspekten der Modellierung eines Altsystems und der Art und Weise der Verwendung von Altsystemmodellen bei der modellgetriebenen Integration von Altsystemen.

1 Einleitung

Das vom BMBF geförderte Projekt *Modellgetriebene Integration von Informationssystemen* (MINT) [SSMG06] befasst sich unter anderem mit der Entwicklung eines modellgetriebenen Verfahrens für die Integration bestehender Systeme, im Folgenden auch als Altsysteme bezeichnet. Der Fokus des Projekts liegt dabei auf betrieblichen Informationssystemen für kleine und mittlere Unternehmen (KMU). Das entwickelte Verfahren soll KMU die Möglichkeit geben, ihre Softwaresysteme kosteneffizient an sich ändernde Geschäftsprozesse und neue Anforderungen anpassen zu können. Ein Schwerpunkt des Projektes ist die Vereinfachung der Beschreibung von Softwaresystemen unter Einbindung von Fachexperten durch den Einsatz domänenspezifischer Sprachen (Domain Specific Languages, DSL [MHS05]).

Der hier vorgestellte Ansatz grenzt sich zu anderen Ansätzen der modellgetriebenen Entwicklung durch die Integration bestehender Systeme in den Entwicklungsprozess ab. Dieser Aspekt wird weder in der von OMG vorgeschlagenen *Model Driven Architecture*

(MDA) [MM03], noch in den bisherigen praxisorientierten Ansätzen zur modellgetriebenen Entwicklung, wie z. B. [SV05], berücksichtigt. Zudem integriert der vorgestellte Ansatz bereits auf der Ebene von Geschäftsprozessen, und nicht wie häufig anzutreffen auf der Ebene von Datenmodellen oder Anwendungen.

In dem Projekt MINT wird u. a. der Frage nachgegangen, ob sich die üblicherweise mit der modellgetriebenen Softwareentwicklung verbundenen Vorteile (vgl. z. B. [SV05]) auch auf Integrationsvorhaben übertragen lassen, und wie eine modellgetriebene Integration von Altsystemen konkret umgesetzt werden kann. Der Fokus dieses Beitrages liegt auf der Beschreibung von Aspekten der Modellierung von Altsystemen für den Zweck der modellgetriebenen Integration (Abschnitt 2) und der Darstellung eines Vorgehens zur modellgetriebenen Integration (Abschnitt 3). In Abschnitt 4 wird kurz auf die verwendete technische Plattform eingegangen. Der Beitrag schließt mit einer Zusammenfassung in Abschnitt 5.

2 Modellierung von Altsystemen

Viele der älteren, in Unternehmen häufig noch im Einsatz befindlichen Anwendungssysteme sind monolithisch aufgebaut. Die Herausforderung der Integration neuer Software mit diesen Anwendungssystemen besteht darin, dass die dort bereits implementierten Fachverfahren sowie der mitunter über Jahrzehnte aufgebaute Datenbestand in den neu zu realisierenden Geschäftsprozessen aufgehen müssen. Da in der Regel für die bestehenden Anwendungen keine Modelle existieren, müssen ihre Daten, Verfahren und mögliche Schnittstellen zunächst identifiziert und modelliert werden.

Informationen über bestehende Dienste der zu integrierenden Anwendungssysteme sind aus unterschiedlichen Quellen verfügbar:

- Technische und fachliche Dokumentation
- Anwender und Entwickler
- Quelltexte und Datenbanken
- Repositories, wie z. B. UDDI (Universal Description, Discovery and Integration)
- Explizite Beschreibungen von Funktionalitäten, z. B. in der Form von WSDL (Web Services Description Language)

Im Kontext des Projektes MINT ist es notwendig, dass die gesammelten Informationen über die bestehenden Anwendungssysteme bzw. Altsysteme in einer Sprache vorgehalten werden, die sowohl von Softwarearchitekten, als auch von den im weiteren Integrationsprozess verwendeten Generatoren interpretierbar ist. Zur Modellierung des Altsystems wurde als Modellierungssprache die UML2-Notation [Obj07] gewählt. Eine Evaluation hat ergeben, dass die UML die an eine Modellierungssprache von Altsystemen im Kontext des Projektes gestellten Anforderungen am besten erfüllt:

- **Angemessenheit:** Die Modellierungssprache muss sinnvoll zur Modellierung von Altsystemen anwendbar sein und alle relevanten Eigenschaften dieser Systeme modellieren können.
- **Verbreitung:** Die Modellierungssprache muss sowohl in aktuell vorhandenen Werkzeugen unterstützt als auch bei Softwarearchitekten verbreitet sein.
- **Gute Werkzeugunterstützung:** Die Modellierungssprache muss durch bestehende Werkzeuge gut unterstützt werden. In der UML repräsentierte Modelle können beispielsweise im Generatorframework *openArchitectureWare* (oAW) [ope] bei den Modell-zu-Modell-Transformationen direkt verarbeitet werden.

Modelle von Altsystemen bestehen in erster Linie aus Komponentendiagrammen. Diese Komponentendiagramme modellieren eine Zerlegung der zu integrierenden, aus Altsystemen bestehenden Anwendungslandschaft in Komponenten und Schnittstellen dieser Komponenten. Die Modellierung beschränkt sich dabei allerdings auf die für das jeweilige Ziel der Integration notwendige, öffentliche Funktionalität der Altsysteme. Weitere relevante Informationen, die nicht über die vorgegebenen Sprachmittel der UML abgebildet werden können, werden über ein für diesen Zweck erstelltes UML-Profil mit Stereotypen und *tagged values* aufgenommen.

In der derzeitigen, ersten Realisierung des Vorgehens sind Altsystemmodelle manuell durch einen Softwarearchitekten zu erstellen. Diese Vorgehensweise ist aufwendig aber für weniger komplexe Systeme vertretbar, da nur die für die jeweilige Integration notwendigen, öffentlichen Funktionalitäten modelliert werden müssen. Das erstellte Altsystemmodell kann dabei in aufeinanderfolgenden Integrationsprojekten wiederverwendet und inkrementell angepasst und erweitert werden. Es ist geplant zu einem späteren Zeitpunkt die Altsystemmodelle automatisiert durch geeignete Reengineering Methoden zu generieren. Wenn die Altsysteme beispielsweise im günstigsten Fall bereits Funktionalität in der Form von Web Services anbieten, ist eine automatisierte Erzeugung eines UML-Altsystemmodells aus WSDL-Beschreibungen mittels einer WSDL-zu-UML-Transformation möglich.

3 Das MINT-Vorgehensmodell zur Integration von Altsystemen

Das MINT-Vorgehensmodell zur modellgetriebenen Integration von Altsystemen basiert auf der MDA von OMG [MM03] und unterstützt alle vier dort vorgeschlagenen Ebenen: das *Computation Independent Model* (CIM), das *Platform Independent Model* (PIM), das *Platform Specific Model* (PSM) und die Code-Ebene. Abbildung 3.1 zeigt das MINT-Vorgehensmodell des Integrationsprozesses im Überblick.

Das im vorherigen Abschnitt dargestellte Altsystemmodell ist sowohl der CIM-, PIM- als auch PSM-Ebene zuzuordnen. Diese Zuordnung reflektiert, dass (a) das Altsystemmodell primär geschäftsbezogene Funktionalität modelliert, die als Teil von Geschäftsprozessen verwendet wird (CIM), (b) bestimmte Aspekte der Architektur der Altsysteme modelliert (PIM) und (c) zusätzlich plattformspezifische Angaben enthält (PSM), wie beispielsweise

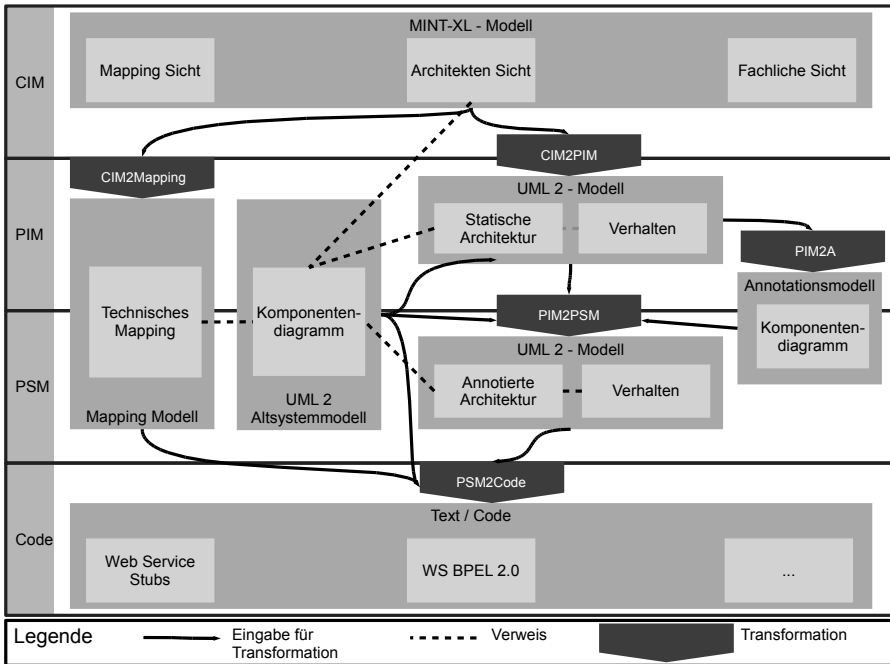


Abbildung 3.1: Das MINT-Vorgehensmodell zur Integration von Altsystemen

Hostinformationen von Web Services. Das Vorgehensmodell setzt voraus, dass die Funktionalitäten der Altsysteme durch Web Services gekapselt sind. Für Altsysteme, deren Funktionalität noch nicht auf diese Weise gekapselt ist, werden durch das Vorgehen im weiteren Verlauf automatisch Web Service Stubs bzw. Adapter generiert (s. u.).

Auf der CIM-Ebene sind unter Verwendung der im Projektverlauf entwickelten domänen-spezifischen Sprache *MINT-XL* drei Modelle zu erstellen. Den Ausgangspunkt stellt die von einem Fachexperten zu erstellende *fachliche Sicht* dar. Sie definiert einen Geschäftsprozess, der das Zusammenspiel der zu integrierenden Altsysteme beschreibt. Die *Architekten Sicht* wird von einem Softwarearchitekten erstellt und besteht aus einer Menge von Annotationen, die sich auf die fachliche Sicht beziehen. Durch diese Annotationen können u. a. Beziehungen zwischen dem Altsystemmodell und der fachlichen Sicht hergestellt werden, indem angegeben wird, durch welche Elemente des Altsystems einzelne Prozessschritte realisiert werden. Neben den Prozessmodellen ist von einem Softwarearchitekten eine *Mapping Sicht* zu erstellen, die definiert, wie Daten beim Austausch zwischen den beteiligten Altsystemen zu transformieren sind. Mit der Mapping Sicht wird dem Umstand Rechnung getragen, dass bestehende Altsysteme häufig unterschiedliche Datenmodelle verwenden und auch die Durchsetzung eines unternehmensweit einheitlichen Datenmodells [Sch88] als nicht realistisch gilt [KBS04]. Die Erstellung der drei genannten Modelle wird durch drei spezialisierte, domänenspezifische *MINT-XL Editoren* unterstützt.

Bei der Transformation des CIM in ein PIM wird die fachliche Sicht, die Architekten

Sicht, die Mapping Sicht und das Altsystemmodell verarbeitet und ein PIM in der UML2 Notation erzeugt. Das PIM beschreibt sowohl *Statische Struktur* in der Form eines Komponentendiagramms, als auch *Verhalten* in der Form eines Aktivitätsdiagramms des durch die Integration erzeugten Systems. Dieses System besteht aus den Altsystemen, die gemäß des auf der CIM-Ebene spezifizierten Geschäftsprozesses verknüpft worden sind.

Für die Transformation des PIM in ein PSM sind neben dem PIM zusätzliche Informationen bzgl. der gewünschten Plattform notwendig, auf der der integrierte Geschäftsprozess später ausgeführt werden soll. Diese Informationen werden, wie bereits analog auf der CIM-Ebene, in Form von Annotationen hinterlegt, die in diesem Fall das PIM mit Informationen anreichern. Um zu verhindern, dass Annotationen an einem generierten PIM bei einer erneuten Generierung überschrieben werden, werden diese Informationen in dem separaten *Annotationsmodell* abgelegt. In die Transformation des PIM in das PSM geht neben dem PIM und dem Annotationsmodell auch das Altsystemmodell ein.

Bei der letzten Transformation geht neben dem PSM auch ein *technisches Mapping* genanntes Modell ein. Dieses ergänzt die bereits eingeführte Mapping Sicht, in der im Wesentlichen lediglich deklarativ Zuordnungen von Elementen unterschiedlicher Datenmodelle zueinander vorgenommen werden, um detaillierte Spezifikationen, wie die Abbildungen algorithmisch umgesetzt werden sollen. Die Transformation von PSM und technisches Mapping erzeugt ausführbaren Code in unterschiedlichen Zielsprachen. Der auf der CIM-Ebene modellierte Geschäftsprozess und Teile des technischen Mappings werden in die Business Process Execution Language (BPEL, [OAS]) transformiert. Die erzeugten BPEL-Skripte können direkt in entsprechenden BPEL-Engines ausgeführt werden.

4 Technische Plattform

Technische Plattform des MINT-Integrationsprozesses ist die Open Source Entwicklungsumgebung *Eclipse* [Ecla]. Eclipse erlaubt das Laden unterschiedlicher Plugins, die u. a. das Erstellen von domänenspezifischen Sprachen, Editoren und Generatoren ermöglichen. Bei einer zum Projektstart durchgeführten Evaluation von unterschiedlichen MDA-Werkzeugen (*openArchitectureWare*, *AndroMDA*, *Software Factories*) stellte sich heraus, dass das Framework *openArchitectureWare* (oAW) [ope] die im Projekt MINT bestehenden Anforderungen am besten erfüllt [FM07]. Ein wesentliches Kriterium war neben der Unterstützung von Modell-zu-Code-Transformationen die Unterstützung von Modell-zu-Modell-Transformationen, die einzig von oAW in ausreichendem Maße geboten wird. Aus diesem Grund bietet oAW als einziges der untersuchten Werkzeuge ausreichend Unterstützung für alle vier Ebenen der MDA. In oAW kann zur Erstellung des Meta-Metamodells das *Eclipse Modeling Framework* (EMF) [Eclb] verwendet werden. EMF nutzt zur Modellierung *Ecore*, das weitestgehend kompatibel zu *EMOF* [Obj06] ist.

5 Zusammenfassung

In diesem Beitrag wurden Aspekte der Modellierung von Altsystemen zur Verwendung in der modellgetriebenen Integration dargestellt. Zudem wurde ein Vorgehen zur modellgetriebenen Integration vorgeschlagen, das Altsysteme bereits auf der Ebene von Geschäftsprozessen integriert. Der beschriebene Ansatz basiert dabei auf der von der OMG vorgeschlagenen MDA. Bei dem Vorgehen sind neben den Geschäftsprozessmodellen das Altsystemmodell und die Mappings zwischen den Datenmodellen der Altsysteme wichtige Eingaben und Ausgangspunkte für eine Kette von Transformationen, an deren Ende ein ausführbares, integriertes Teilsystem steht.

Das Projekt befindet sich derzeit noch in der Durchführung. Zukünftige Arbeiten richten sich insbesondere auf die Generierung von Adaptern für Altsysteme und die automatisierte Erzeugung des Altsystemmodells durch Methoden des Reengineering. Evaluationsergebnisse des Ansatzes liegen derzeit noch nicht vor.

Literatur

- [Ecla] Eclipse Foundation. Eclipse. <http://www.eclipse.org>.
- [Eclb] Eclipse Foundation. Eclipse Modeling Framework. <http://www.eclipse.org/emf/>.
- [FM07] Heiner Feislachen und Jürgen Meister. Evaluation von MDA-Werkzeugen. Bericht, BTC-AG, 2007. http://bi.offis.de/publicProjectDocument/Bericht_Evaluation_MDA-Werkzeugauswahl.pdf?docId=nbqy8mgsvk42&groupId=awb01zr338sz.
- [KBS04] Dirk Krafczig, Karl Banke und Dirk Slama. *Enterprise SOA: Service-Oriented Architecture Best Practices*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.
- [MHS05] Marjan Mernik, Jan Heering und Anthony M. Sloane. When and how to develop domain-specific languages. *ACM Computing Surveys*, 37(4):316–344, 2005.
- [MM03] Joaquin Miller und Jishnu Mukerji. *MDA Guide Version 1.0.1*. Object Management Group (OMG), June 2003.
- [OAS] OASIS. *BPEL*. OASIS. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>.
- [Obj06] Object Management Group (OMG). *Meta Object Facility (MOF) Core Specification Version 2.0*. Object Management Group (OMG), January 2006.
- [Obj07] Object Management Group (OMG). *Unified Modeling Language: Superstructure Version 2.1.1 (formal/2007-02-05)*, 2007.
- [ope] openArchitectureWare.org. openArchitectureWare. <http://www.openarchitectureware.org>.
- [Sch88] August-Wilhelm Scheer. *Wirtschaftsinformatik*. Springer Verlag, Berlin, Heidelberg, New York, 1988.

- [SSMG06] Niels Streekmann, Ulrike Steffens, Claus Möbus und Hilke Garbe. Model-Driven Integration of Business Information Systems. *Softwaretechnik-Trends*, 26(4):9–13, November 2006.
- [SV05] Thomas Stahl und Markus Völter. *Modellgetriebene Softwareentwicklung*. Dpunkt Verlag, 2005.