

Modellgetriebene Transformation von Legacy Business-Software

Matthias Kühnemann

Fakultät für Informatik
Technische Universität Chemnitz
kumat@informatik.tu-chemnitz.de

Gudula Rünger

Fakultät für Informatik
Technische Universität Chemnitz
ruenger@informatik.tu-chemnitz.de

Zusammenfassung

In zahlreichen Unternehmen befinden sich Business-Softwaresysteme im Einsatz, deren Programmarchitektur und Softwareinfrastruktur den heutigen Anforderungen, die an die Sicherheit, Leistungsfähigkeit, Adaptierbarkeit sowie an die Kosten und Verfügbarkeit moderner Zielplattformen gestellt werden, oftmals nicht mehr gewachsen sind. Das Forschungsprojekt TransBS¹ beschäftigt sich mit der Realisierung eines modellgetriebenen Transformationswerkzeuges, das Softwareentwickler bei der Transformation erprobter, monolithischer Business-Software in eine moderne, verteilte und workflowbasierte Client-Server Architektur unterstützen soll. Das Werkzeug soll für eine inkrementelle, parametrisierbare und adaptierbare Überführung verschiedene Zwischendarstellungen der Ausgangs- und Zielsoftware auf der Basis der modellgetriebenen Softwareentwicklung nutzen.

1 Einführung

Eine monolithische Business-Software repräsentiert oftmals typische Aspekte einer Legacy-Problematik, d.h. ein über Jahre gewachsenes Softwaresystem, welches beständig durch Expertenwissen erweitert und adaptiert wurde [2]. Die Transformation der monolithischen Ausgangssoftware in eine prototypische verteilte Referenzimplementierung kann dabei in inkrementellen, transparenten und konfigurierbaren Transformationsschritten erfolgen. Die für diesen Ansatz hilfreiche Infrastruktur kann auf der *Modellgetriebenen Softwareentwicklung* (MDA - Model Driven Architecture) [4] basieren. Die MDA ermöglicht das Speichern von Zwischendarstellungen der Ausgangs- und Zielsoftware in Form von plattformunabhängigen und plattformspezifischen Softwaremodellen. MDA-Werkzeuge erlauben die automatisierte und interaktive Generierung dieser Softwaremodelle und der damit assoziierten und kompilierbaren Quelltextstrukturen.

¹Das Projekt TransBS - Transformation monolithischer Business-Softwaresysteme in verteilte, workbasierte Client-Server-Architekturen wird seit 2006 durch das Bundesministerium für Bildung und Forschung (BMBF) unter dem Förderkennzeichen 01 IS F10 gefördert. Weitere Informationen sind unter www.transbs.de zu finden. Das Projektkonsortium setzt sich aus der Berndt & Brungs Software GmbH, der Universität Bayreuth und der Technischen Universität Chemnitz zusammen.

Als exemplarische Ausgangssoftware betrachten wir das monolithische Softwaresystem GBware², das sich in einer Vielzahl von Unternehmen erfolgreich im Einsatz befindet. Ein Ziel des Forschungsprojektes TransBS ist eine prototypische Überführung des GBware Legacy-Programmcodes in ein verteiltes Softwaresystem GBwareD (GBware distributed) im Client-Server-Stil. GBwareD soll zu diesem Zweck ein offenes und erweiterbares Client-Server-Framework CFrame für Business-Software nutzen können [5]. CFrame soll modular aufgebaut sein und über ein Kommunikationsinterface verfügen, in dem Teile existierender Business-Softwaresysteme als Komponenten integriert werden können. Das Framework wird dabei unterschiedliche Aspekte der Verteiltheit, Sicherheit und Performance abdecken. Für die Überführung der GBware Module in GBwareD Komponenten, die in einem Client-Server-System verteilt ausgeführt werden können, soll ein Transformationswerkzeug (im folgenden TransBS genannt) entwickelt werden. Das funktions-, korrektheits- und merkmals-erhaltene Transformationswerkzeug soll eine inkrementelle Überführung unterstützen sowie geeignete Beschreibungs- und Spezifikationsprachen für die Modulinteraktion bereitstellen.

2 Modellgetriebene Transformation

Die OMG (Object Management Group) will mit MDA einen Standard für eine modellgetriebene Softwareentwicklung schaffen. MDA-Werkzeuge unterstützen einen Softwareentwickler bei der Transformation eines plattformunabhängigen Softwaremodells (PIM) in ein plattformabhängiges Modell (PSM) und in einen kompilierbaren Quelltext [1]. Ein inkrementeller und konfigurierbarer Transformationsansatz kann durch die Überführung eines leicht adaptierbaren und erweiterbaren PIMs in potentiell unterschiedliche PSMs ermöglicht werden, wobei das erzeugte PSM die Zielarchitektur festlegt. Die OMG favorisiert UML (Unified Modeling Language) weitestgehend für die Erstellung eines PIMs. Oftmals wird zur Modelldarstellung eine höhere, domänenspezifische Modellierungssprache (DSL - Domain Specific Language) verwendet, deren Definition man oftmals durch ein UML-

²GBware© (General Businessware), Berndt & Brungs Software GmbH

Profil erhält, das an eine Business-Domäne angepasst wurde. Es können neue UML-Profile definiert oder vorgefertigte Profile (z.B. J2EE/EJB Profile Example) verwendet werden. MDA-Werkzeuge unterstützen die Transformation in verschiedene Zielarchitekturen (z.B. CORBA, J2EE, .NET) und unterschiedliche Zielsprachen (z.B. Java, C++). Nur durch die Erweiterung eines UML-Klassendiagramms um zusätzliche Informationen ist die Erzeugung eines kombinierbaren Quellcodes überhaupt möglich. Zu diesen Informationen zählen Stereotypen, Eigenschaftswerte (tagged values) und Bedingungen (constraints). Soll beispielsweise eine beliebige Klasse als Datenbankklasse abgespeichert werden, so beschreibt ein Stereotyp die Datenbankklasse, der Primärschlüssel ist ein Eigenschaftswert, und die Bedingung legt einen Wertebereich fest. In einem nächsten Schritt kann das PSM in kompilierbaren Quelltext übersetzt werden, wobei die Geschäftslogik und Prozessalgorithmen der Business-Software oftmals in Form von Templates (die Platzhalter enthalten) implementiert werden. Neben templateorientierten Transformationsansätzen werden auch patternorientierte und metamodellorientierte Transformationsansätze für die Generierung von Quellcode eingesetzt.

Die MDA bietet bisher keine Unterstützung zur Behandlung von Legacy-Software. Dieser Ansatz soll im Forschungsprojekt TransBS untersucht werden. Ein allgemeiner Ansatz für eine solche Überführung besteht in der Analyse des Legacy-Programmcodes, um eine (gedankliche) Umstrukturierung einer Referenzimplementierung in 3 getrennte Codebereiche vorzunehmen: *Individueller Code*, *Generischer Code* und *Schematischer, sich wiederholender Code*. Individueller Code bezeichnet Programmcode, der für jede spezielle Applikation neu implementiert werden muss [6]. Generischer Code kann unabhängig von der speziellen Applikation wieder verwendet werden. Der Transformationsansatz aus einem Softwaremodell in einen verteilten, workflowbasierten Programmcode eignet sich insbesondere für den schematisch, sich wiederholenden Codebereich.

Die Frage, wie die Analyse des Legacy-Programmcodes durchgeführt werden soll, um ein passendes PIM zu generieren, ist noch offen. In Betracht kommen sowohl manuelle Analyseverfahren (Programmierer, Dokumentation, bestehende Programmstruktur), als auch automatisierte, compilerbasierter Techniken (Parser, Code-to-Code, Source-Code-Graph für Darstellung der Modulabhängigkeiten).

3 Anforderungsspezifikation TransBS

Ein erster Entwurf für die Realisierung eines Transformationswerkzeuges TransBS auf der Basis der MDA könnte wie folgt aussehen. Das Werkzeug TransBS verwendet ein PIM und ein (oder mehrere) PSM(s) zur internen Darstellung der verteilten Codestruk-

tur der Zielsoftware GBwareD. Das PIM erhält man aus dem Legacy-Programmcode GBware durch verschiedene (Analyse-)Ansätze in einem ersten Schritt (*code-to-model Transformation*). Dazu zählen Reengineering, das Studium der GBware-Dokumentation, sowie compilerbasierte und manuelle Analyseverfahren. Konfigurationen und Adaptionen der Zielsoftware können in einem zweiten Schritt bei der Transformation des PIMs in ein PSM gegebenenfalls interaktiv durch den TransBS Anwender vorgenommen werden (*model-to-model Transformation*) [3]. Dies schließt inkrementelle, transparente und interaktive Transformationschritte nicht aus, da zum einen Transformationen in immer konkretere Modelle (sowohl PIMs als auch PSMs) durch die MDA-Idee angeregt werden und zum anderen auch mehrere verschiedene PSMs durch den TransBS Anwender interaktiv erzeugt werden können. Der dritte und letzte Schritt erzeugt den kompilierbaren Quellcode der verteilten Zielsoftware GBwareD durch die Überführung des PSM in Programmcode (*model-to-code Transformation*). Ermöglicht wird diese Überführung durch Abbildungen und Transformationsregeln von PSM-Modellelemente auf Klassenelemente des Frameworks CBFrame. Die Referenzimplementierung von CBFrame ist hierfür in einer Zielarchitektur realisiert und kann auf einer ausgewählten Zielplattform prototypisch ausgeführt werden. Als Zielarchitekturen für eine verteilte Implementierung des Client-Server-Frameworks kommen EJB (Enterprise Java Beans) und CORBA (Component Object Request Broker Architecture) in Betracht.

Literatur

- [1] A. Andresen. *Komponentenbasierte Softwareentwicklung*. Hanser Verlag, 2004.
- [2] M.C. Feathers. *Working Effectively with Legacy Code*. Prentice Hall, 2004.
- [3] V. Gruhn, D. Pieper, and C. Röttgers. *MDA. Effektives Softwareengineering mit UML2 und Eclipse*. Springer, 2006.
- [4] MDA Model Driven Architecture, <http://www.omg.org/mda>.
- [5] G. Muller, J.L. Lawall, J.M. Menaud, and M. Südholt. Constructing component-based extension interfaces in legacy system code. In *Proc. of the 11th workshop on ACM SIGOPS European workshop*. AMC Press, 2004.
- [6] T. Stahl and M. Völter. *Modellgetriebene Softwareentwicklung: Techniken, Engineering, Management*. dpunkt Verlag, 2005.