

Vorwort

Warum ein Buch über QVT Operational Mappings?

Beim Paradigmenwechsel von der strukturierten zur objektorientierten Software-Entwicklung Mitte der Neunziger gab es im Grunde genommen zwei Wege, sich den neuen Methoden und Techniken zu nähern: zum einen den datenorientierten, ausgehend von der Lehre der Datenbanken und Datenmodellierung, zum anderen den sprachlichen, ausgehend von den Programmiersprachen, die zur der Zeit den Weg der objektorientierten Entwicklung bereiteten. So ähnlich verhält es sich auch mit dem Thema des modellgetriebenen Vorgehens. Auch hier gibt es grundsätzlich zwei alternative Blickrichtungen, die zum einen auf die modellgetriebene Code-Entwicklung mit generativen Techniken und zum anderen auf den Weg der Entwicklung über Modellierung und Modelltransformation ausgerichtet sind. Letzteres ist der Vorschlag, den die *Object Management Group* (OMG) mit dem *Model Driven Architecture*-Konzept (MDA) propagiert und den ich in diesem Buch konsequent aufnehmen und verfolgen möchte.

Die OMG ist ein herstellerunabhängiges Gremium, welches den Auftrag hat, objektorientierte Techniken und Technologien zu standardisieren. So gibt es zum Beispiel die *Unified Modeling Language* (UML) und das *Meta Object Facility*-Konzept (MOF), die sich als OMG-Standards etabliert haben. Das eine ist eine formale, einheitliche, universelle Modellierungssprache, das andere beschreibt einen Ansatz, auf der Basis von formalen Modellen formale Modellierungssprachen zu entwickeln. Beides dient als Fundament für das MDA-Konzept, welches ebenfalls von der OMG als Standard herausgegeben worden ist.

Bisher handelte es sich um Bausteine zur Modellierung, sicher eine der Säulen eines modellgetriebenen Vorgehens. Die MDA geht allerdings weiter und schlägt neben der Modellierung die Transformation von Modellen vor, um aus einer Modellierungsebene in eine folgende zu gelangen, bis hin zu einer modellgestützten Code-Generierung. Sprachen zur Modellierung sind definiert und eingeführt, mindestens die oben erwähnte UML. Was in diesem MDA-Konzept noch fehlte, sind Sprachen zur Modelltransformation.

Von anderer Seite, zum Beispiel aus der Szene der modellgetriebenen Software-Entwicklung, sind Generierungssprachen entwickelt und eingeführt worden, die durchaus eine gewisse Verbreitung und Anerkennung gefunden haben. Doch auch die OMG hat die Lücke mit der im April 2008 als Standard veröffentlichten Spezifikation MOF/QVT (*Query Views Transformation*) mittlerweile geschlossen.

Die QVT ist ein Konzept, in dem drei alternative Sprachen angeboten werden, um auf verschiedene Weise eine Transformation von Modellen beschreiben und durchführen zu können. Die *Operational Mappings* ist eine von ihnen. Nun ist es meiner Meinung nach sehr schwierig, auf der Basis der OMG-Spezifikationen neue Techniken und Technologien wie die der Transformation zu erlernen – ein Problem, dem ich mit diesem Fachbuch Abhilfe schaffen möchte.

Meine ersten intensiven Kontakte mit Modelltransformationen mit einer operationalen Sprache hatte ich 2006. Borland hatte mit Together 2006 als einer der ersten Hersteller auf der Basis des bewährten Modellierungswerkzeugs ein MDA-Werkzeug auf den Markt gebracht, welches Operational Mappings unterstützte, allerdings in einer frühen Ausprägung, die nur sehr schwer zu erlernen war, da es weder von Borland geeignete Literatur gab, noch die QVT-Spezifikation zum Erlernen des Together-Dialektes taugte. Dies war einmal mehr für mich eine Motivation, selbst ein Fachbuch zu dem Thema anzubieten, ein Buch, in dem ich mich recht konsequent auf die Seite der OMG gestellt habe.

Auch von Borlands Together – mittlerweile in der Version 2008 – ist zu sagen, dass sich der herstellereigene QVT/OM-Dialekt etwas mehr der standardisierten Fassung angenähert hat, so dass die hier entwickelten Beispiele auch mit Together anwendbar sein sollten. Aber nicht nur aus dem Hause Borland gibt es QVT-Werkzeuge. Einige weitere werde ich vorstellen.

Für wen ist das Buch gedacht?

Modellierung von Sachverhalten der realen Welt ist ein wesentlicher Bestandteil des ingenieurmäßigen Software-Entwicklungsprozesses. Schwerpunkt dieses Buches ist jedoch nicht die Modellierung mit einer Modellierungssprache, und auch die Generierung von Code auf der Basis von Modellen steht nicht im Fokus der Betrachtung. Modelle, die mit formalen Modellierungssprachen erarbeitet worden sind, sind Mittel zum Zweck, um daraus neue und konsistente Modelle abzuleiten – mittels Transformation. Und vor dem Hintergrund, dass die Entwicklung von komplexen Anwendungen in der Regel über mehrere aufeinanderfolgende Phasen erfolgt, wird die systematische und modellgetriebene Entwicklung auch und gerade bei einem Phasenübergang mit Einsatz von Modelltransformationen von zunehmender Bedeutung sein. Erreichen möchte ich also auf jeden Fall IT-Architekten, Analytiker und Designer, zu deren wesentlichen Aufgaben es gehört, reale Sachverhalte zu beschreiben, zu strukturieren und in abstrahierter Form einer weiteren Entwicklungsarbeit zuzuführen.

Wenn auch das Thema MDA nicht unbedingt im Mittelpunkt steht, so möchte ich doch verantwortlichen Projektleitern für IT-Projekte wie auch IT-Führungskräften einen Einblick in den Dialog zwischen Modellierung und Transformation geben. Zum Thema MDA gibt es weiterführende Literatur, die ich hiermit ergänzen und erweitern möchte, und das Thema UML ist sicher recht erschöpfend behandelt. Ebenso würde ich mir wünschen, dass auch Entwickler, die sonst eher einen Zugang zu Programmiersprachen und sprachlich repräsentierten Darstellungsweisen haben, durch die Lektüre dieses Buches einige Anregungen finden für neue Wege und neue Horizonte.

Dieses Buch diskutiert das Thema Modelltransformation mit *Operational Mappings*, wie ich hoffe, ergiebig und ausführlich an vielen Beispielen, die sämtlich mit den zurzeit verfügbaren QVT-Werkzeugen erarbeitet worden sind, so dass ich denke, ein umfassendes Lehrwerk geschaffen zu haben, das nicht nur zum theoretischen Erlernen, sondern auch zum begleitenden Experimentieren und Üben geeignet ist. Grundkenntnisse der Modellierung mit UML, insbesondere im Zusammenhang mit Modellen und Metamodellen, setze ich dabei voraus, Erfahrungen mit höheren Programmiersprachen sind sicher hilfreich.

Wie sollte man es lesen?

Das erste Kapitel, die Einleitung, soll einen Überblick geben und eine Einordnung ermöglichen. Das nächste Kapitel beschäftigt sich umfassend mit der Entwicklung und Repräsentation von Metamodellen. Ein umfassendes Verständnis von Metamodellen ist wesentlich für die weitere Arbeit mit der Transformationssprache, da QVT als MOF-Konzept grundsätzlich metamodellbasiert ist. Demjenigen, der sich zunächst nur mit den Zusammenhängen und den Einordnungen beschäftigen möchte, empfehle ich ein intensiveres Studium der einleitenden Kapitel; die Beschreibung und das Arbeiten mit der Sprache, kann dann ein wenig oberflächlicher gelesen werden.

Das Kapitel 3 geht nun in die Tiefe und beschreibt die Transformationssprache *Operational Mappings* auf eine recht formale, an der Syntax ausgerichtete Weise. Das ist wahrscheinlich, auch wenn ich mich um spannende Formulierungen bemüht habe, etwas mühsam zu lesen. Allerdings habe ich damit bezweckt, einerseits auf die OCL-Grundlagen der QVT einzugehen und andererseits die Sprachkonzepte vollständig zu präsentieren und zu erläutern. An Beispielen soll es nicht fehlen, diese behandeln im Wesentlichen das berühmte „HelloWorld“.

Kapitel 4 schließlich beschäftigt sich mit zwei komplexen und zusammenhängenden Beispielen, einmal dem Standardbeispiel der OMG-Spezifikation – Uml-ToRdbm –, auf der Basis eigener simpler Metamodelle, und zum anderen einem Beispiel – UML2EJB – mit pragmatischeren Modellen auf der Grundlage des UML2-Metamodells. Beide Beispiele werden Schritt für Schritt entwickelt, so dass man das Kapitel sequentiell abarbeiten sollte.

Meiner Meinung nach ist dieses Vorgehen recht gut geeignet, um einem eher praktisch veranlagten Leser einen schnelleren Zugang zu dem Thema zu eröffnen, den man sich dann nach und nach mit der Methode „Versuch und Irrtum“ erschließen kann. Ein so veranlagter Leser wird vielleicht mit diesem Kapitel einsteigen und die erforderlichen Grundlagen bedarfsgerecht nachlesen wollen. Zuletzt werde ich einige fortgeschrittene Konzepte erläutern, zum Beispiel die Arbeit mit UML-Profilen, BlackBoxes etc., die nicht einfach, aber sicher spannend sind.

Wem bin ich zu Dank verpflichtet?

Einen großen Anteil an der Entstehung dieses Buches hat der Springer-Verlag, dem ich für die Zusammenarbeit herzlich danken möchte. Einen weiteren erheblichen Anteil daran, dass dieses Buch entstehen konnte, wie es ist, haben die beiden wichtigsten Frauen in meinem Leben, Konstanze und Christine, die sich während der Arbeiten an diesem Buch das ein oder andere Mal durch frühe Fassungen hindurch gearbeitet haben. Als „Nicht-IT-Experten“ gehören sie nicht gerade zur klassischen Zielgruppe; und dafür, dass sie mich und die Entstehung des Buches geduldig ertragen und mit Kritik und Anregungen unterstützt haben, bin ich ihnen sehr dankbar.

Letztendlich ist auch zu erwähnen, dass verschiedene Initiativen und Produkthersteller dazu beigetragen haben, dass es nicht nur bei einer theoretischen Ausarbeitung geblieben ist, sondern dass ich im Laufe der Arbeiten an dem Buch viele Beispiele entwickeln und diskutieren konnte. Im Einzelnen zu nennen sind die Sourceforge-Initiative mit der Herausgabe des Produktes SmartQVT, die Firma Borland für die Freigabe des Produktes Operational QVT (QVTO) und die *Eclipse Modeling Toolkit*-Initiative für die Integration von QVTO in die Eclipse-Plattform. Für die Beispiele im Kapitel „Metamodelle“ ist das freie UML-Werkzeug Topcased eingesetzt worden. Bei fast allen Abbildungen handelt es sich um UML-Diagramme, die mit dem Werkzeug MagicDraw der Firma NoMagic Inc. erstellt worden sind. Allen Herstellern und Initiativen gilt mein Dank.

Und schließlich möchte ich natürlich auch der Gemeinde der Leser danken, insbesondere dann, wenn ich mit anregender Kritik versehen werde.



<http://www.springer.com/978-3-540-92292-6>

QVT - Operational Mappings

Modellierung mit der Query Views Transformation

Nolte, S.

2010, XIII, 267 S. 52 Abb., Hardcover

ISBN: 978-3-540-92292-6