Kurz erklärt: Model Management

Erhard Rahm
Universität Leipzig, *http://dbs.uni-leipzig.de*

Model Management bezeichnet ein ambitioniertes Framework zur generischen Metadaten-Verwaltung, das vom bekannten Microsoft-Forscher Phil Bernstein und Kollegen als Vision zur drastisch vereinfachten Erstellung und Anpassung Metadaten-getriebener Anwendungen vorgeschlagen wurde [2, 3, 5]. Zielsetzung dabei ist die Bereitstellung einer Infrastruktur, mit der unterschiedliche Modelle wie Schemas und Ontologien sowie Abbildungen (*Mappings*) zwischen Modellen in einheitlicher Weise repräsentiert und mittels mächtiger, deklarativer Operatoren automatisiert verarbeitet werden können. Wesentlich ist einerseits die Generizität, d.h. die Anwendbarkeit des Ansatzes für unterschiedliche Anwendungsbereiche und für unterschiedliche Modellrepräsentationen (Metamodelle). Andererseits soll durch die Operatoren der manuelle Aufwand zur Metadatenverarbeitung stark reduziert werden.

Die Notwendigkeit einer mächtigen Metadatenverwaltung ergibt sich vor allem bei der Entwicklung und Anpassung interoperabler Informationssysteme, bei denen mehrere Schemas zur Beschreibung von Daten oder Dienstschnittstellen Verwendung finden. Dies ist in zahlreichen Anwendungsgebieten erforderlich, zum Beispiel beim Austausch von Daten/Nachrichten zwischen E-Business-Anwendungen, zur Integration mehrerer Datenquellen in ein Data Warehouse oder zur Erzeugung von Wrappern für den Zugriff auf Web-Datenquellen. Die hierbei benötigten Datentransformationen können durch Mappings zwischen den beteiligten Schemas beschrieben werden. Die Erstellung solcher Mappings sowie deren Anpassung, z.B. nach der Änderung eines Schemas, ist derzeit jedoch ein sehr aufwändiger und hochgradig manueller Prozess. Dies liegt u.a. an den unterschiedlichen Datenmodellen und Schemasprachen (relationale Datenbanken, XML-Schemas, OWL-Ontologien, etc.) sowie vor allem an der semantischen Heterogenität, da die zu verarbeitenden Schemas, Ontologien und Datenbestände oft unabhängig voneinander von verschiedenen Personen für unterschiedliche Verwendungszwecke entwickelt wurden.

Eine Vielzahl von Forschungs- und Entwicklungsprojekten befasste sich mit den damit zusammenhängenden Problemstellungen, jedoch meist bezogen auf eine bestimmte Anwendungsklasse und bestimmte Repräsentationsformate. Derzeitige Repository-Systeme ermöglichen zwar eine einheitlichen Speicherung unterschiedlicher Schemas, bieten jedoch nur eine geringe Funktionalität zur automatisierten Verarbeitung der Metadaten. Typischerweise werden nur fein-granulare, navigierende Programmierschnittstellen für den Zugriff auf Schemakomponenten angeboten (object-at-a-time), womit die Erstellung von Anwendungen oder Metadaten-Werkzeugen sehr aufwändig wird.

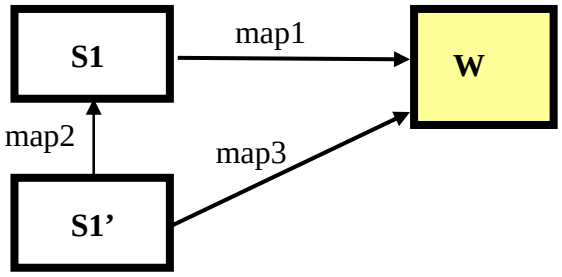
Model Management (MM) strebt anstelle der einfachen Object-at-a-Time-Operationen die Bereitstellung mächtiger Operatoren an, welche auf vollständigen Modellen und Mappings arbeiten. Damit wird für die Metadatenverarbeitung ein ähnlicher Quantensprung und Produktivitätsgewinn angestrebt, wie für die Daten(bank)verarbeitung beim Übergang von satzorientierten Operationen auf die mengenorientierten Operatoren der Relationenalgebra. Zu den wesentlichen MM-Operatoren zählen:

- *Import/Export:* Überführung eines realen Modells (relationales Datenbankschema, XML-Nachrichtenformat, OWL-Ontologie, etc.) in die generische Repräsentation des

MM-Systems bzw. Erzeugung eines realen Modells aus der internen Repräsentation

- *Match*: Generierung eines Mappings zwischen zwei Modellen (Schemas, Ontologien). Das Mapping beinhaltet dabei sämtliche semantischen Korrespondenzen zwischen den Eingabemodellen. Die Erstellung eines auf Instanzdaten anwendbaren Mappings, z.B. zur Datentransformation, kann bereits Teil des Match-Operators sein oder durch einen Operator *TransGen* erfolgen [5], der ein einfacheres Match-Mapping als Eingabe erhält.
- *Compose:* Kombination zweier aufeinander folgender Mappings in ein einziges Mapping
- *Merge:* Mischen zweier Modelle auf Basis eines gegebenen Mappings zwischen den Modellen [15]
- *Diff:* Für ein gegebenes Modell und Mapping wird ein Teilmodell bestimmt, das nicht am Mapping teilnimmt
- *ModelGen:* Überführung eines Modells in einer Sprache in ein äquivalentes Modell einer anderen Sprache (z.B. objektorientiert-relational oder relational- XML) [1].

Die Abbildung illustriert den Einsatz von MM-Operatoren für ein Data-Warehouse-Szenario. Zur Integration einer Datenquelle mit Schema S1 in ein Data Warehouse mit Schema W soll zunächst durch eine Match-Operation ein Mapping *map1* ermittelt werden, das alle für das Warehouse relevanten S1-Komponenten ermittelt sowie eine Abbildung zu den korrespondierenden W-Komponenten. Die Match-Operation ist aufgrund semantischer Heterogenitätsprobleme und zur Bestimmung komplexerer Abbildungsfälle i.a. nur teilautomatisch durchführbar, d.h. automatisch ermittelte Korrespondenzen sind zu bestätigen bzw. zu korrigieren. Dieser nach wie vor erforderliche manuelle Aufwand sollte zur Bestimmung anderer Mappings nicht wiederholt notwendig werden, z.B. nach einer Änderung von S1 nach S1' (Schemaevolution). Durch eine – vergleichsweise einfach durchführbare - Match-Operation lässt sich ein Mapping *map2* zwischen S1' und S1 ermitteln, in dem vor allem alle unveränderten Schemateile berücksichtigt sind. Um das geänderte Schema S1' auf das Warehouse abzubilden, ermöglicht die Komposition der beiden Mappings u.a eine Wiederverwendung (Re-use) von *map1*.



1. map1 = match (S1, W)
2. map2 = match (S1', S1)
3. map3 = compose (map2, map1)

In den vergangenen Jahren beschäftigten sich viele Forschungsarbeiten mit Model Management bzw. wesentlichen Teilaufgaben wie generischen Metamodellen zur einheitlichen Repräsentation heterogener Schemas [1, 16], generischen Mapping-Repräsentationen sowie der automatisierten Realisierung einzelner Operatoren. Einen detaillierten Überblick zu dem erreichten Stand der Forschung gibt [5], so dass hier nur auf einige ausgewählte Ergebnisse eingegangen wird.

Besonders intensiv bearbeitet wurden in den letzten Jahren Verfahren zum Schema- und Ontologie-Matching und damit zur Realisierung des Match-Operators [17, 8]. Die erzielten

Ergebnisse zeigen, dass dieses Problem generisch behandelt werden kann, wobei für eine hohe Vollständigkeit und Genauigkeit bei der Ermittlung von Korrespondenzen möglichst mehrere Einzelverfahren (z.B. Nutzung von Attributnamen, Datentypen, Wörterbüchern oder Beispielinstanzen) kombiniert werden sollten. Einige Prototypen unterstützen auch die Wiederverwendung früherer Match-Ergebnisse, um den manuellen Aufwand zu reduzieren [7, 12]. Im Rahmen des Clio-Projektes wurde die Generierung ausführbarer Mappings, und damit die Realisierung eines TransGen-Operators, intensiv untersucht [14, 10, 19]. Verfügbare Werkzeuge zur Generierung ausführbarer Mappings sind jedoch meist noch auf einfache Abbildungsfälle beschränkt [11]. Einige Forschungsarbeiten untersuchten die Realisierung des Compose-Operators [9, 4] sowie seine Nutzung zur Anpassung von Mappings aufgrund von Schemaänderungen [20, 18].

Mit Rondo [13] wurde ein erster Prototyp eines MM-Systems entwickelt, der jedoch nur sehr einfache (syntaktische) Mappings unterstützt, die nicht unmittelbar auf Dateninstanzen anwendbar sind. Neuere Arbeiten zeigen, dass eine inhärente Herausforderung des Model Management in der Unterstützung einer generischen, aber semantisch ausdrucksstarken Mapping-Sprache liegt [5]. Ein generischer Ansatz ist notwendig, um Abbildungen zwischen Schemas unterschiedlicher Metamodelle zu ermöglichen; eine hohe semantische Ausdrucksstärke ist Voraussetzung für eine automatisierte Umsetzung der Mappings in auf Dateninstanzen anwendbare Transformationen (z.B. in SQL, XQuery oder XSLT). Die größten Erfolgsaussichten werden derzeit Mapping-Sprachen auf Basis logischer Regeln bzw. algebraischer Ausdrücke eingeräumt, die jedoch nicht die volle Mächtigkeit von Sprachen wie SQL oder XQuery abdecken. Wie auch die jüngsten Arbeiten zu Compose zeigen, ist die generische Realisierung der MM-Operatoren umso schwieriger, je mächtiger die Mapping-Sprache ist.

Ausblick

Der Model-Management-Ansatz zur generischen Verwaltung und Manipulation von Modellen und Mappings ist sehr ambitioniert und bisher nur partiell umgesetzt. Dennoch zeigt sich, dass bereits Teillösungen wie die teilautomatisierte Generierung und Anpassung von Mappings für viele praktische Einsatzfälle sehr hilfreich sind. Die noch offenen Probleme ermöglichen eine Vielzahl interessanter Forschungsarbeiten mit hohem Praxispotenzial, insbesondere die Unterstützung semantisch aussdrucksstarker Mapping-Sprachen, die effiziente Realisierung noch wenig untersuchter Operatoren (z.B. Merge, Diff) und deren Anwendung, z.B. für Datenintegration und Schemaevolution.

Literatur

1. P. Atzeni, P. Cappellari, P.A. Bernstein: *ModelGen: Model Independent Schema Translation*. Proc. ICDE 2005
2. P.A. Bernstein, A.Y. Levy, R.A. Pottinger: *A Vision for Management of Complex Models*. ACM Sigmod Record, 2000
3. P.A. Bernstein. *Applying Model Management to Classical Meta Data Problems.* Proc. Conf. on Innovative Data Systems Research (CIDR), 2003.
4. P.A. Bernstein, T.J. Green, S. Melnik, A. Nash: *Implementing Mapping Composition*. Proc. 32nd VLDB, 2006
5. P.A. Bernstein, S. Melnik: *Model Management 2.0 – Manipulating Richer Mappings*. Proc. ACM Sigmod Conf., 2007
6. P.A. Bernstein, R.A. Pottinger: *Merging Models Based on Given Correspondences*. Proc. 29th VLDB, 2003
7. H. Do, E. Rahm: *COMA - A System for Flexible Combination of Schema Matching Approaches*, Proc. VLDB 2002
8. J. Euzenat, P. Shvaiko: *Ontology Matching*. Springer 2007
9. R. Fagin, P.G. Kolaitis, L. Popa, W.C. Tan: *Composing Schema Mappings: Second-Order Dependencies to the Rescue*. TODS 2005

10. L.M. Haas, M.A. Hernández, H. Ho, L. Popa, M. Roth: *Clio Grows Up: From Research Prototype to Industrial Tool.* Proc. ACM Sigmod Conf. 2005
11. F. Legler, F. Naumann: *A Classification of Schema Mappings and Analysis of Mapping Tools.* Proc. BTW 2007
12. J. Madhavan, P. A. Bernstein, A. Doan, A. Y. Halevy: *Corpus-based Schema Matching.* Proc. ICDE 2005
13. S. Melnik, E. Rahm, P.A. Bernstein. *RONDO – a programming platform for generic model management.* Proc. ACM Sigmod Conf., 2003.
14. R. Miller, L. Haas, M. Hernandez: *Schema Mapping as Query Discovery*. Proc. 26th VLDB, 2000
15. R. Pottinger, P.A. Bernstein: *Merging Models Based on Given Correspondences*. Proc. VLDB 2003
16. C. Quix, D. Kensche, M.A. Chatti: *Rollenbasierte Metamodellierung zur Datenintegration.* Datenbank-Spektrum 15, 2005
17. E. Rahm, P.A. Bernstein. *A Survey of Approaches to Automatic Schema Matching.* VLDB Journal, 2001
18. E. Rahm, P.A. Bernstein: *An Online Bibliography on Schema Evolution.* ACM Sigmod Record, 2006
19. M. Roth, et al.: *XML mapping technology: Making connections in an XML-centric world.* IBM Systems Journal 45(2), 2006
20. *C.* Yu, L. Popa: *Semantic Adaptation of Schema Mappings when Schemas Evolve. Proc.* VLDB 2005