ERHARD RAHM

# Model Management

Model Management bezeichnet ein ambitioniertes Framework zur generischen Metadatenverwaltung, das vom bekannten Microsoft-Forscher Phil Bernstein und Kollegen als Vision zur drastisch vereinfachten Erstellung und Anpassung metadatengetriebener Anwendungen vorgeschlagen wurde [Bernstein et al. 2000; Bernstein 2003; Bernstein & Melnik 2007]. 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 aufwendiger 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.

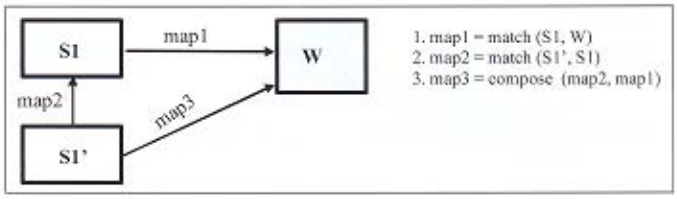
Eine Vielzahl von Forschungs- und Entwicklungsprojekten befasste sich mit den damit zusammenhängenden Problemstellungen, jedoch meist bezogen auf eine bestimmte Anwendungsklasse und bestimmte Repräsentationsformate. Derzeitige Repository-Systeme ermöglichen zwar eine einheitliche Speicherung unterschiedlicher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

Model Management (MM) strebt anstelle der einfachen Object-at-a-Time-Operationen die Bereitstellung mächtiger Operatoren an, die 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 [Bernstein & Melnik 2007], der ein einfacheres Match-Mapping als Eingabe erhält.
- *Compose:* Kombination zweier aufeinanderfolgender Mappings in ein einziges Mapping.
- *Merge:* Mischen zweier Modelle auf Basis eines gegebenen Mappings zwischen den Modellen [Pottinger & Bernstein 2003].
- *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) [Atzeni et al. 2005].

Abbildung 1 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* bestimmt 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 im Allgemeinen 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

**Abb. 1: Einsatz von MM-Operatoren**



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*.

In den vergangenen Jahren beschäftigten sich viele Forschungsarbeiten mit Model Management bzw. wesentlichen Teilaufgaben wie generischen Metamodellen zur einheitlichen Repräsentation heterogener Schemas [Atzeni et al. 2005; Quix et al. 2005], generischen Mapping-Repräsentationen sowie der automatisierten Realisierung einzelner Operatoren. Einen detaillierten Überblick zu dem erreichten Stand der Forschung gibt [Bernstein & Melnik 2007], sodass 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 [Rahm & Bernstein 2001; Euzenta & Shvaiko 2007]. 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 [Do & Rahm 2002; Madhavan et al. 2005]. Im Rahmen des Clio-Projektes wurde die Generierung ausführbarer Mappings, und damit die Realisierung eines TransGen-Operators, intensiv untersucht [Miller et al. 2000; Haas et al. 2005; Roth et al. 2006]. Verfügbare Werkzeuge zur Generierung ausführbarer Mappings sind jedoch meist noch auf einfache Abbildungsfälle beschränkt [Legler & Naumann 2007]. Einige Forschungsarbeiten untersuchten die Realisierung des Compose-Operators [Fagin et al. 2005; Bernstein et al. 2006] sowie seine Nutzung zur Anpassung von Mappings aufgrund von Schemaänderungen [Yu & Popa 2005; Rahm & Bernstein 2006].

Mit Rondo [Melnik et al. 2003] 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 [Bernstein & Melnik 2007]. 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


[Atzeni et al. 2005] *Atzeni, P.; Cappellari, P.; Bernstein, P. A.:* ModelGen: Model Independent Schema Translation. Proc. ICDE 2005.

[Bernstein 2003] *Bernstein, P. A.:* Applying Model Management to Classical Meta Data Problems. Proc. Conf. on Innovative Data Systems Research (CIDR), 2003.

[Bernstein & Melnik 2007] *Bernstein, P. A.; Melnik, S.:* Model Management 2.0 – Manipulating Richer Mappings. Proc. ACM Sigmod Conf., 2007.

[Bernstein & Pottinger 2003] *Bernstein, P. A.; Pottinger, R. A.:* Merging Models Based on Given Correspondences. Proc. 29th VLDB, 2003.

[Bernstein et al. 2000] *Bernstein, P. A.; Levy, A. Y.; Pottinger, R. A.:* A Vision for Management of Complex Models. ACM Sigmod Record, 2000.

[Bernstein et al. 2006] *Bernstein, P. A.; Green, T. J.; Melnik, S.; Nash, A.:* Implementing Mapping Composition. Proc. 32nd VLDB, 2006.

[Do & Rahm 2002] *Do, H.; Rahm, E.:* COMA – A System for Flexible Combination of Schema Matching Approaches. Proc. VLDB, 2002.

[Euzenat & Shvaiko 2007] *Euzenat, J.; Shvaiko, P.:* Ontology Matching. Springer-Verlag, 2007.

[Fagin et al. 2005] *Fagin, R.; Kolaitis, P. G.; Popa, L.; Tan, W. C.:* Composing Schema Mappings: Second-Order Dependencies to the Rescue. TODS, 2005.

[Haas et al. 2005] *Haas, L. M.; Hernández, M. A.; Ho, H.; Popa, L.; Roth, M.:* Clio Grows Up: From Research Prototype to Industrial Tool. Proc. ACM Sigmod Conf. 2005.

[Legler & Naumann 2007] *Legler, F.; Naumann, F:* A Classification of Schema Mappings and Analysis of Mapping Tools. Proc. BTW 2007.

[Madhavan et al. 2005] *Madhavan, J.; Bernstein, P. A.; Doan, A.; Halevy, A. Y.:* Corpus-based Schema Matching. Proc. ICDE 2005.

[Melnik et al. 2003] *Melnik, S.; Rahm, E.; Bernstein, P. A.:* RONDO – a programming platform for generic model management. Proc. ACM Sigmod Conf., 2003.

[Miller et al. 2000] *Miller, R.; Haas, L.; Hernandez, M.:* Schema Mapping as Query Discovery. Proc. 26th VLDB, 2000.

[Pottinger & Bernstein 2003] *Pottinger, R.; Bernstein, P. A.:* Merging Models Based on Given Correspondences. Proc. VLDB 2003.

[Quix et al. 2005] *Quix, C.; Kensche, D.; Chatti, M. A.:* Rollenbasierte Metamodellierung zur Datenintegration. Datenbank-Spektrum, Heft 15, 2005, S. 5-11.

[Rahm & Bernstein 2001] *Rahm, E.; Bernstein, P. A.:* A Survey of Approaches to Automatic Schema Matching. VLDB Journal, 2001.

[Rahm & Bernstein 2006] *Rahm, E.; Bernstein, P. A.:* An Online Bibliography on Schema Evolution. ACM Sigmod Record, 2006.

[Roth et al. 2006] *Roth, M. et al.:* XML mapping technology: Making connections in an XML-centric world. IBM Systems Journal 45(2), 2006.

[Yu & Popa 2005] *Yu, C.; Popa, L.:* Semantic Adaptation of Schema Mappings when Schemas Evolve. Proc. VLDB, 2005.



**Erhard Rahm**
ist Lehrstuhlinhaber für Datenbanken am Institut für Informatik der Universität Leipzig. Er promovierte und habilitierte an der Universität Kaiserslautern und verbrachte Forschungsaufenthalte in den USA bei IBM sowie Microsoft Research. Er ist Autor mehrerer Bücher und zahlreicher Konferenz- und Zeitschriftenbeiträge. Derzeit ist er Sprecher des GI-Arbeitskreises »Web und Datenbanken«.

Prof. Dr. Erhard Rahm
Universität Leipzig
Abteilung Datenbanken
Postfach 100920
04009 Leipzig
rahm@informatik.uni-leipzig.de
http://dbs.uni-leipzig.de