Informatik, Forschung und Entwicklung manuscript No.
(will be inserted by the editor)

Informationsfusion auf heterogenen Datenbeständen*

Oliver Dunemann, Ingolf Geist, Roland Jesse, Gunter Saake, Kai-Uwe Sattler

Fakultät für Informatik, Otto-von-Guericke Universität Magdeburg, Postfach 4120, D-39016 Magdeburg

Received: 27. November 2001

Key words Information Fusion, Meta Data Management, Data Analysis, Temporality, Heteroeneous Data Sources

Zusammenfassung Die Informationsfusion als Prozess der Integration und Interpretation heterogener Daten mit dem Ziel der Gewinnung von Informationen einer neuen, höheren Qualität eröffnet eine Vielzahl von Anwendungsgebieten. Dabei erfordert dieser Prozess eine enge Verzahnung der bislang häufig noch isoliert vorliegenden Werkzeuge und Techniken zum Zugriff auf heterogene Datenquellen, deren Integration, Aufbereitung, Analyse der syntaktischen, semantischen sowie temporalen Strukturen und Visualisierung derselben. In diesem Beitrag werden Rahmenbedingungen zur Informationsfusion ebenso dargestellt wie die sich aus ihnen ergebenden Aufgaben. Es werden Lösungsansätze zur Erstellung einer Workbench vorgestellt, die eine durchgängige Unterstützung von Fusionsschritten ermöglicht. Dabei wird das Ziel einer konsequenten Nutzung von Datenbanktechniken verfolgt.

Abstract Information Fusion is the process of integration and interpretation of heterogeneous data gaining information of a higher quality. This concept is open to a variety of applications. The process of information fusion requires a tight connection between isolated tools and techniques: accessing heterogeneous data sources, their integration, preparation and transformation, analysis of syntactic, semantic and temporal structures as well as their visualization. In this paper we present the framework for a workbench, which supports the individual steps of information fusion in a continous and uniform manner by using database technology.

* Diese Arbeit wird gefördert von der DFG (FOR 345/1).

1 Einleitung und Motivation

„Stellen Sie sicher, dass Sie durch Ihren Wissensdurst nicht in der Flut von Informationen ertrinken."

(Anthony J. D'Angelo)

Durch die Steigerung der Leistungsfähigkeit der Informationstechnologie ist heute die Verwaltung sehr großer Datenbestände technisch möglich. Da durch die Zunahme des Datenvolumens in gleichem Maße dessen Verständnis abgenommen hat, wird die Möglichkeit zur automatisierten Analyse der Daten immer wichtiger. Hierzu müssen relevante Teile der oftmals verteilt und heterogen vorliegenden Datenbestände identifiziert werden, um sie anschließend in einer integrierten Form darstellen und analysieren zu können. Die wissenschaftliche Arbeit konzentriert sich dabei zunächst auf das Erarbeiten von Lösungen für einzelne Teilschritte. So werden die technischen Voraussetzungen zum Zugriff auf verteilte, heterogene Datenbestände und Methoden für die Integration auch über Paradigmengrenzen hinaus geschaffen. Werkzeuge für spezielle Nachbearbeitungs- und Analyseschritte wie beispielsweise Data Mining oder Visualisierung werden entwickelt. Eine Integration dieser Komponenten in Rahmen einer Workbench dient zum Einen dazu, den manuellen Bearbeitungsaufwand zum Transformieren der Daten zwischen den Bearbeitungsschritten zu minimieren. Zum Anderen kann eine Instanz, welcher der gesamte Fusionsprozess bekannt ist, Optimierungen in der Art durchführen, dass beispielsweise gemeinsame oder wiederholt auszuführende Schritte erkannt und zusammengefasst werden. Zusätzliche Dienste, die von einer solchen Workbench angeboten werden, sind unter Anderem ein Authentifizierungsmechanismus, eine einheitliche Fehlerbehandlung oder vielfältige Möglichkeiten der Visualisierung.

An der Universität Magdeburg wird zur Zeit unter dem Arbeitstitel InFuse eine Workbench entwickelt, die die Zusammenführung der Komponenten der Informationsfusion zum Ziel hat [31]. Durch ein offenes und mo-

dulares Konzept wird ein Rahmen aus den oben angesprochenen Basisdiensten und einer Benutzerschnittstelle zur Definition und Ausführung von Fusionsprozessen geschaffen, in den weitere Komponenten eingefügt werden können. Dabei werden bereits in der Analysephase Aspekte der Teilaufgaben der Informationsfusion berücksichtigt, indem von Praxisbeispielen ausgehend beispielhaft Fusionsprozesse modelliert werden.

Im Weiteren ist der Beitrag wie folgt gegliedert. Zunächst definiert der Abschnitt 2 den Begriff „Informationsfusion“ näher und beschreibt mögliche Anwendungen. Anschließend wird im Abschnitt 3 ein Beispiel entwickelt, welches durchgängig in der Arbeit benutzt werden soll. Der Abschnitt 4 stellt den Hauptteil der Arbeit dar und zeigt die Integration verschiedener Methoden in einer Workbench zur Unterstützung der Informationsfusion. Dabei wird auf die Architektur, mögliche Integrations- und Analysemethoden, die Verwaltung der Metadaten sowie die Berücksichtigung temporalen Verhaltens während des Fusionsprozesses eingegangen. Den Abschluss dieses Abschnittes bildet die Beschreibung des entstandenen Prototyps. Abschnitt 5 zeigt den aktuellen Stand der Technik auf und stellt verwandte Arbeiten vor. Eine Zusammenfassung und ein Ausblick auf weitere Forschungsschwerpunkte zur Informationsfusion beschließen die Arbeit.

2 Informationsfusion – Begriff und Anwendungen

Mit dem Begriff der Informationsfusion wird im hier dargestellten Umfeld[1] der Prozess der Integration und Interpretation von Daten aus heterogenen, verteilten Quellen sowie die darauf aufbauende Konstruktion von Modellen für einen bestimmten Problembereich mit dem Ziel der Gewinnung von Informationen einer neuen, höheren Qualität bezeichnet. Auf Basis dieser neu generierten Informationen können Anwender aus verschiedenen Gebieten in die Lage versetzt werden, fundiertere Entscheidungen zu treffen. Somit ist der Prozess durch die Entdeckung von für den Anwender nützlichen, interessanten Informationen getrieben. Diese Definition erklärt die Informationsfusion als ein interdisziplinäres Gebiet, welches auf Methoden und Techniken verschiedener Bereiche, wie z. B. Datenbanken, Statistik, Maschinellem Lernen und Visualisierung zurückgreift.

Der Fusionsprozess beinhaltet dabei die verschiedenen Aspekte Datenzugriff, Datenintegration, Analyse sowie Verdichtung, Präsentation und Weiterverarbeitung sowie die Verwaltung der Metadateninformationen. Eine Anforderungsanalyse und Beschreibung der einzelnen Bereiche sind in [15] gegeben.

[1] In der Literatur wird der Begiff der Informationsfusion auch für das Zusammenführen von Sensordaten verwendet [5, 34], hier jedoch liegt der Fokus auf der Fusion von Daten aus Datenbanken

Die Schritte der Integration und Analyse der Daten sowie die Abhängigkeiten untereinander können formal durch die Verwendung eines Graphen modelliert werden. In diesem Graphen repräsentieren die Knoten Datenquellen beziehungsweise Operationen auf diesen Quellen, während die Kanten die Aufeinanderfolge der Operationen und Quellen beschreiben. Somit beschreibt ein solcher Graph einen Fusionsprozess und dient im Weiteren als Modell für ein *Worksheet*, welches die verschiedenen Sichten auf den Prozess modelliert. Hierbei kann der Fusionsgraph einmal direkt grafisch modelliert beziehungsweise implizit durch die Anwendung der verschiedenen Operationen auf die Daten in einer „Spreadsheet“-Ansicht erzeugt werden.

In vielen betriebswirtschaftlichen und wissenschaftlichen Anwendungsgebieten werden Aufbereitung und Präsentation von Daten zur Unterstützung von Entscheidungsprozessen benötigt, die durch die Informationsfusion unterstützt werden können. Exemplarisch sei die Analyse von Gensequenzen aus verschiedenen Gen- und Stoffwechseldatenbanken in der Bioinformatik (Comparative Genomics) genannt. Weiterhin stellen der Produktionsvorbereitungsprozess in der Giesserei-Industrie sowie das Stoffstrommanagement Anwendungsgebiete für die Informationsfusion dar. Ein weiteres Anwendungsszenario, aus dem das im folgenden Abschnitt eingeführte Beispiel entnommen worden ist, wird durch die Analyse von Konto- und Kundendaten in Kreditinstituten zum Zweck des *Database Marketing* gebildet [16, 19].

3 Beispiel

In diesem Abschnitt wird beispielhaft ein Anwendungsszenario beschrieben, in dem mit Hilfe der Informationsfusion abwanderungsgefährdete Kunden eines Kreditinstituts identifiziert werden sollen. Mit dieser Information und den Gründen für eine potenzielle Abwanderung kann derselben in bestimmten Fällen durch geeignete Maßnahmen entgegengewirkt werden. Das Institut verfüge über ein Data Warehouse, in dem historische Kunden- und Kontodaten vorliegen. Ausserdem existiere eine Datenbasis mit ehemaligen Kunden, die ihre Kundenbeziehung beendet haben. In dieser liegen auch die Gründe für diese Beendigung vor. In einem ersten Schritt kann so ein Datensatz erstellt werden, der sowohl abgewanderte wie auch nicht abgewanderte Kunden mit einigen kundenspezifischen Kennzahlen wie den Volumina der verschiedenen Produktarten enthält. Dieser hat etwa das in Tab. 1 dargestellte Aussehen.

Um die Gründe für die Abwanderung von Kunden erkennen zu können, muß dieser Datensatz um Abwanderungen bereinigt werden, die nicht im Einflußbereich des Instituts liegen. So wurde beispielsweise die Beziehung mit dem Kunden 25894 durch dessen Tod beendet, weshalb sie keine Relevanz für die Klassifizierung und Abhängigkeitsanalyse abwanderungsgefährdeter Kunden

1999				2000				2001				Abw.-Grund
Kunde	Giro	Spar	...	Kunde	Giro	Spar	...	Kunde	Giro	Spar	...	
12345	207,20	6000,00		12345	-800,24	7000,00		12345	2198,05	8000,00		
13521	3820,48	4832,99		13521	12,78	0,00						Direktbank A
17846	1457,58	8569,47		17846	5,46	0,00						Filialbank B
25894	526,92	7894,15		25894	632,14	8314,02						Tod
29587	-1258,49	0,00		29587	-4897,15	0,00		29587	-3874,95	0,00		
⋮												

Tabelle 1 Historische Kundenentwicklung (über drei Jahre)

besitzt. Das Ergebnis dieser Bereinigung wird mit soziodemographischen Informationen angereichert, um mehrere Kriterien für eine spätere Klassifizierung zu erhalten. So werden Kundendaten wie das Alter und Informationen aus Selbstbeurteilungen der den Kunden betreuenden Geschäftsstellen mit einbezogen (Tab. 2). Diese geschäftsstellenspezifischen Informationen sind von 1 (sehr schlecht) bis 9 (sehr gut) skaliert und umfassen die folgenden Kriterien:

- Verkehrsmäßige Lage,
- Räumliche Verhältnisse,
- Konkurrenzaktivitäten,
- Siedlungsstruktur und
- Einwohnerentwicklung.

Dieser Datensatz wird nun als Trainingsmenge verwendet, um im Data Mining-Schritt mit Entscheidungsbäumen und sequentiellen Mustern die abgewanderten Kunden von den anderen zu unterscheiden. Ergebnisse könnten beispielsweise Entscheidungsbäume (Abb. 1) beziehungsweise folgende Merkmale überdurchschnittlich abwanderungsgefährdeter Kunden sein (vgl. [19]):

- Kunden älter als 65 Jahre,
- Nur eine aktive Kontoverbindung,
- Geringes Volumen (unter 500 €),
- Reine Anlage bzw. Finanzierungskunden und
- Kurze Dauer der Geschäftsbeziehung (weniger als fünf Jahre).

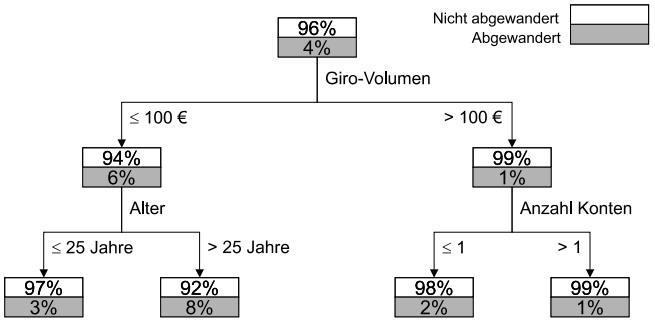

Abbildung 1 Entscheidungsbaum von Kundenabwanderungen

Mit diesen Ergebnissen kann ein Kundenbestand in Hinblick auf abwanderungsgefährdete Kunden untersucht werden. Allerdings stellt nicht jeder Abbruch einer Kundenbeziehung einen monetären Verlust für die Bank dar. Daher sind in einem weiteren Schritt die profitablen von den nicht profitablen Kundenverbindungen zu trennen. Hierzu kann beispielsweise eine Potezialanalyse nach dem *Customer Lifetime Value*-Prinzip durchgeführt werden [17]. Durch einen weiteren Fusionsschritt können somit abwanderungsgefährdete Kunden, die eine lukrative Entwicklung der Kundenbeziehung erwarten lassen, extrahiert werden. Da auch die Gründe bzw. die Richtung der Abwanderung bekannt sind (beispielsweise schlechte Lage der Filiale und Abwanderung hin zu einer Direktbank), kann dieser gezielt entgegen gewirkt werden. Die Abb. 2 zeigt den gesamten Prozess im Überblick, wobei die genutzten und entstandenen Tabellen, die Operationen bzw. Unterprozesse sowie die erzeugten Modelle (Kundenklassifikation, Kundenpotential) dargestellt werden.

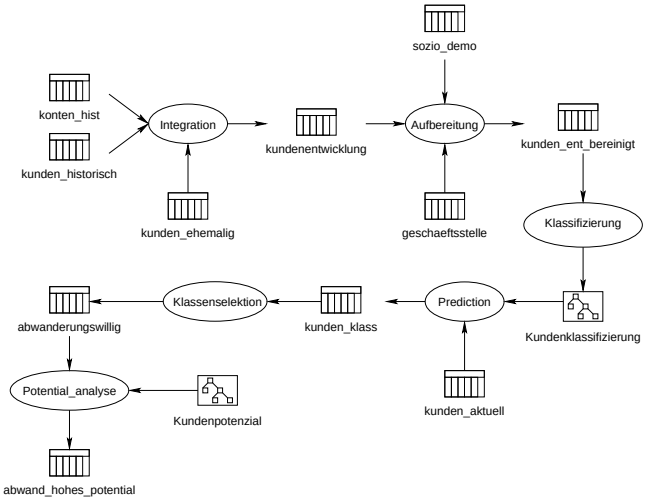

Abbildung 2 Beispielprozess

4 Werkzeugunterstützung für die Informationsfusion

Eine enge Verzahnung der Integration heterogener Daten mit ihrer Aufbereitung sowie Analyse bildet die Basis

2000						2001						Abw.-Grund
Kunde	Giro	Spar	Alter	Lage	···	Kunde	Giro	Spar	Alter	Lage	···	
12345	-800,24	7000,00	47	7		12345	2198,05	8000,00	48	7		
13521	12,78	0,00	21	3								Direktbank A
17846	5,46	0,00	29	2								Filialbank B
29587	-4897,15	0,00	53	6		29587	-3874,95	0,00	54	6		
⋮												

Tabelle 2 Bereinigte und angereicherte Kundenentwicklung (über zwei Jahre)

der Informationsfusion. Das Ineinandergreifen der einzelnen Bestandteile ermöglicht die interaktive Arbeitsweise, welche dem iterativen Charakter des Fusionsprozesses gerecht wird. Ein solcher Prozess benutzt Methoden aus einem Vorrat von Werkzeugen, die innerhalb der Workbench verwaltet werden. Im folgenden Abschnitt werden ein Vorschlag für eine Architektur einer Workbench für die Informationsfusion sowie die zu verwaltenden Metadaten diskutiert. Die nachfolgenden Abschnitte beschreiben verschiedene Klassen von Werkzeugen und ihre Anwendung in der Integration, Analyse und Visualisierung. Der Abschnitt schließt mit der Beschreibung des entstandenen Prototypen.

4.1 Architektur

Eine Workbench zur Unterstützung der Informationsfusion muss eine effiziente und interaktive Analyse großer, zum Teil heterogener Datenbestände ermöglichen. Diese Aufgaben umfassen die Definition und Ausführung von Anfragen, die Transformation von Daten sowie die Anwendung von Analyseoperationen und die Visualisierung der Zwischenstände und Endergebnisse. Vergleichbare Anforderungen sind auch in OLAP-Anwendungen zu finden, so dass für die Fusionsworkbench ein ähnlicher Architekturansatz gewählt wurde wie derjenige, der in OLAP-Systemen zum Einsatz kommt (Abb. 3).

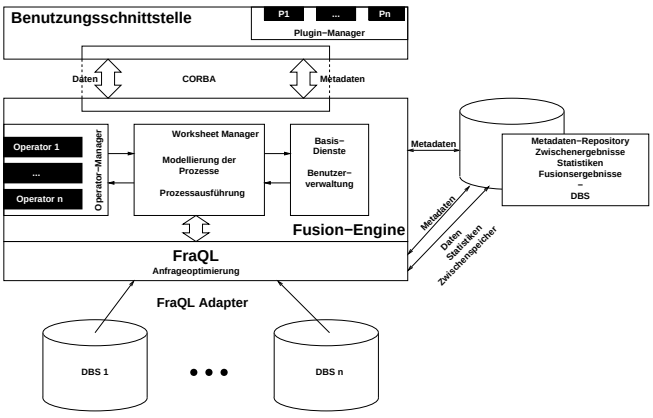

Abbildung 3 Architektur der Workbench

Die Basis des Systems bildet die *Fusion-Engine*, die im Kern aus einem Anfragesystem für Multidatenbanken besteht. Dieses Anfragesystem ermöglicht einen transparenten Zugriff auf verschiedene Datenquellen und stellt Mechanismen zu deren Integration bereit [42]. Das Anfragesystem (in Abb. 3 die FraQL-Schicht) umfaßt weiterhin eine lokale Datenbank für temporäre Daten (z.B. Materialisierungen und Metadaten) und Ergebnisse sowie die eigentlichen Fusionsoperatoren, die ähnlich gespeicherten Prozeduren direkt auf den integrierten Datenbeständen ausgeführt werden können. Die Fusionsoperatoren werden als Plugin in die Workbench eingebunden und können somit zur Laufzeit in das System integriert oder aus diesem entfernt werden. Der Operator-Manager verrichtet diese Aufgabe mit Hilfe der gespeicherten Metadaten. Der Worksheet-Manager ist für die Verwaltung komplette Fusionsprozesse zuständig, indem die Abhängigkeiten und Zustände einzelner Aufbereitungs- und Analyseschritte berücksichtigt werden und so bei Daten- oder Parameteränderungen nur die betroffenen Schritte neu ausgeführt werden müssen. Es können voneinander unabhängige, ausführbare Operationen (es existieren keine Vorgänger oder alle Vorgängeroperationen sind bereits erfolgreich ausgeführt worden) parallel abgearbeitet werden. Tritt in einem Zweig des Graphen ein Fehler auf, der einen Abbruch an der entsprechenden Stelle zur Folge hat, können dennoch davon nicht abhängige Zweige weiter ausgeführt werden.

Die Benutzungsschnittstelle wird durch das Workbench-Frontend (Benutzungsschnittstelle) bereit gestellt. Dieses ist zunächst ein grafisches Analyse- und Definitionswerkzeug, mit dem der Anwender über die Fusion-Engine Zugriff auf die Daten der einzelnen Quellen hat. So können interaktiv Integrations- und Fusionsoperationen ausgeführt, Anfragen formuliert und die Ergebnisse visualisiert werden. Die angebotene Funktionalität hängt dabei von den Rechten des angemeldeten Anwenders ab. So wird zum Beispiel zwischen reinen Prozessnutzer (zum Beispiel ein Manager in einem Kreditinstitut) und Prozessersteller (Knowledge-Ingenieur) unterschieden. Zusätzlich können durch die Unterscheidung von Benutzungsschnittstelle und Fusion-Engine spezialisierte Werkzeuge zur Anwendung kommen.

Die Architektur der Workbench ist mit der Trennung in Fusion-Engine und Frontend mit Ansätzen aus dem OLAP-Bereich vergleichbar. Ein wesentlicher Un-

terschied besteht jedoch darin, dass die zu analysierenden Daten nicht vorab extrahiert, transformiert, bereinigt und redundant in einem Warehouse abgelegt werden. Statt dessen ermöglicht die Verwendung eines Multidatenbank-Anfragesystems innerhalb der Fusions-Engine den transparenten Zugriff auf die Quellen und die Anwendung von Transformations- und Integrationsoperationen. Auf diese Weise können einerseits erste Analysen durchgeführt werden, ohne dass zuvor Daten aufwendig migriert und transformiert werden müssen. Andererseits können die aktuellen Daten betrachtet werden. So lassen sich relevante Datenausschnitte selektieren und Operationen parametrisieren. Für die tiefergehende Analyse können die Ergebnisse einzelner Schritte anschließend materialisiert werden, um so eine effiziente Ausführung zu erreichen.

4.2 Metadatenverwaltung

Für eine flexible Integration und Analyse von heterogenen Quellen mit Hilfe einer Reihe von verschiedenen Werkzeugen ist eine zentrale Metadatenverwaltung von entscheidender Bedeutung. Dazu müssen Daten über die Objekte *Datenquellen*, *Operationen* und *Worksheet* modelliert werden.

Datenquelle: Als Datenquellen werden einerseits Relationen (Tabellen) als auch Ergebnisse der Datenanalyse bezeichnet. Ergebnisse eines Analyseschrittes können ihrerseits wieder Datenquellen eines nachfolgenden Schrittes bilden. Alle Relationen sind Bestandteil eines Schemas innerhalb des Kataloges. Ein Schema kann dabei entweder eine lokale Datenbank oder die globale Sicht repräsentieren.
Die während der Datenanalyse entdeckten Muster oder Hypothesen werden als Modelle bezeichnet. Diese werden im globalen Schema abgelegt. Die Modelle beschreiben neben den gefundenen Mustern auch deren Qualität über eine Interessantheitsfunktion. Weiterhin wird für jedes Modell dessen mögliche Weiterverwendung, wie z.B. die Art der Daten, die klassifiziert werden können, vermerkt.

Operator: Operatoren stehen jeweils für eine Aktion innerhalb des Fusionsprozesses. Sie werden neben ihrem Namen durch ihre Ein- und Ausgabeparameter beschrieben. Zur Auswahl eines Operators wird dieser in eine Klassenhierarchie eingeordnet, die die Verwaltung der Operatoren strukturiert. Hierdurch kann beispielsweise in Analyseoperationen und Integrationsoperatoren unterschieden werden. Eine weitere Klasse von Operatoren sind die „Routing"-Operationen, die mit Hilfe von Abhängigkeiten und Bedingungen den Workflow innerhalb eines Worksheets steuern. Von jedem Operator-Typ können in einem Worksheet beliebig viele, unterschiedlich parametrisierte Instanzen zum Einsatz kommen. Diese nutzen und erzeugen Relationen und Modelle.

Worksheet: Ein Worksheet beinhaltet einen Fusionsprozess. Wie Operationen haben auch Worksheets Ein- und Ausgabeparameter, wodurch die Schachtelung von verschiedenen Prozessen ermöglicht wird. In einem solchen Prozess werden Operationen und Datenquellen verknüpft und somit die Erstellungsschritte einer bestimmten Datenquelle beschrieben. Eine Worksheet-Instanz stellt wiederum die vollständig parametrisierte Form eines Prozesses dar, welche in anderen Prozessen als Teilprozess benutzt werden kann.

Abb. 4 zeigt die Struktur im Metadatenkatalog der vorgestellten Workbench. Zusätzlich zu den schon beschriebenen Objekten ist eine Verwaltung der Benutzer und ihrer verschiedenen Rollen aufgeführt. Auf die Objekte einer Datenbank wird mit Hilfe einer Datenzugriffsmethode (Adapter) zugegriffen.

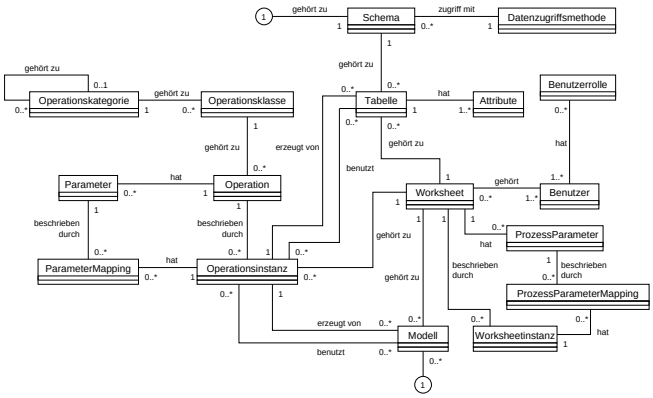

Abbildung 4 UML-Beschreibung der Metadatenverwaltung

4.3 Aufbereitung und Integration der Daten

Bevor die Analyseoperationen ausgeführt werden können, müssen alle dafür notwendigen Daten und Modelle in einem einheitlichen (objektrelationalen) Format vorliegen, welches die Operatoren verarbeiten können. In den Aufbereitungs- und Integrationsschritten werden die Datenquellen in dieses Format transformiert, wobei die Datenqualität zumindest erhalten und in vielen Fällen noch verbessert werden muss. Ohne qualitativ hochwertigen Daten lassen sich keine relevanten und zur Prognose geeigneten Analyseergebnisse ableiten. Für diesen Schritt sind oft manuelle Eingriffe des Analysten notwendig.

Im Data Warehouse-Bereich werden Data-Cleaning-Werkzeuge bzw. Programme zur Extraktion, Transformation und zum Laden von Daten (ETL) zur Aufbereitung und Integration der Daten benutzt. Um diese Aktivitäten zu ermöglichen, muss ein Werkzeug folgende Eigenschaften aufweisen:

Integration der Werkzeuge: Mit Hilfe verschiedener Aktionen und Algorithmen zur Konfliktentdeckung und -lösung wird die Aufbereitung der Daten durchgeführt. Da sich bei diesem Prozess Konflikte gegenseitig bedingen können und somit nicht sofort zu erkennen sind, ist eine Integration der Werkzeuge in einem System notwendig. Hierdurch wird die Erkennung und Lösung dieser verschachtelten Konflikte erst ermöglicht.

Schnelle Reaktionszeiten: Schon die Erkennung und insbesondere die Auflösung von Konflikten erfordert einen Dialog mit dem Anwender. Um diese interaktive Arbeitsweise zu ermöglichen, müssen die Werkzeuge erste Ergebnisse der Aufbereitungsschritte schnell an den Anwender ausgeben, so dass dieser sie in einem frühen Stadium beurteilen kann. Somit ist die Anwendung von langlaufenden Batch-Prozessen, die auf dem gesamten Datenbestand arbeiten, nicht gewünscht. Eine Möglichkeit, diesem Umstand zu begegnen, ist, zunächst Stichproben zu untersuchen, um anschließend die angepassten Transformationsschritte auf den gesamten Datenbestand anzuwenden. Ein alternativer Ansatz ist die iterative Ausführung von Integrationsschritten, so dass diese bei Auftreten eines Konfliktes unterbrochen und gegebenenfalls im Anschluss an die Konfliktbehebung weiter ausgeführt werden können.

Grafische Benutzerführung: Die Interaktion mit den Werkzeugen soll möglichst durch eine durchgängige grafische Benutzerführung erleichtert werden. Somit können einerseits die Zeiten zur Einarbeitung in komplexe Programmierumgebungen verringert werden, andererseits wird auch die Entdeckung und Lösung von Konflikten während der Aufarbeitung und Integration der Daten erleichtert. Zur Unterstützung der Iteration im Integrationsprozess ist eine Undo-Funktion von zentraler Bedeutung. Somit sollten alle Aktionen zunächst virtuell ablaufen und nicht sofort materialisiert werden, damit die Auswirkungen eingeschätzt werden können.

Aus diesen Überlegungen ergibt sich ein interaktiver Ansatz der Datenaufbereitung und -integration. Dabei wird davon ausgegangen, dass das globale Schema bereits vorliegt und die lokalen Schemata auf dieses abgebildet werden müssen. Für diese Abbildung werden die Anfrage- und Integrationfähigkeiten der Multidatenbanksprache FraQL eingesetzt. Diese ist mit ihren Eigenschaften in [42] ausführlich beschrieben.

Zunächst erfolgt die Anwendung der Integrationsschritte auf einer Stichprobe des gesamten Datenbestandes. Diese kann durch bekannte Sampling-Verfahren generiert werden [48]. Dazu müssen die Verfahren geeignet in die dem System zu Grunde liegende Datenzugriffsschicht integriert werden [11,37]. Dadurch kann bereits auf dieser tiefen Ebene eine Optimierung dieser Form von Anfragen durchgeführt werden. Entscheidend dabei ist neben der Laufzeit der Sampling-Algorithmen, dass die Verteilungen der Attributwerte erhalten bleiben [45].

Mit dieser Form der interaktiven Integration auf Basis von Stichproben können frühzeitig Konflikte in den Daten erkannt werden. Die nachfolgende Beschreibung zeigt, wie die einzelnen Konfliktklassen behandelt werden. Voraussetzung ist dabei, dass die Daten in relationaler bzw. objektrelationaler Form vorliegen. Dieses wird durch die Multidatenbanksprache im Kern der Fusions-Engine mit Hilfe von Adaptern zu den einzelnen Datenquellen gewährleistet. Durch die anfängliche Benutzung kleiner Stichproben, deren Umfang nach und nach vergrößert werden kann, können auf Grund der geringeren Datenmenge schnellere Antworten des Systems erreicht werden. Allerdings ist es so unmöglich, alle Ausreißer bzw. Fehler in den Daten zu finden, hierzu muss weiterhin die gesamte Datenmenge betrachtet werden.

Zunächst müssen die lokalen Schemata auf das globale Schema abgebildet werden. Dieses erfolgt mit Umbenennungen, Hinzufügen und Löschen von Attributen. Hierbei werden die Möglichkeiten der Multidatenbanksprache FraQL genutzt. Weiterhin können Metadatenkonflikte auftreten. Diese zeichnen sich dadurch aus, dass ein Teil der Daten in den Metadaten, wie z.B. den Attributnamen, und ein Teil in den Datenwerten modelliert ist. Eine Lösung dieser Konflikte ist durch die Benutzung von Metadaten in den Anfragen gegeben.

Nach der Anpassung der Schemata werden Konflikte in den Datenwerten betrachtet. Diese sind allerdings nicht unabhängig von der Schemaanpassung zu sehen, da eine Abhängigkeit in beide Richtungen besteht. Zur Aufbereitung der Daten können arithmetische Ausdrücke oder auch Zeichenkettenfunktionen angewendet werden. Hiermit können Beschreibungskonflikte gelöst werden. Sind diese Mittel nicht ausreichend, kann der Anwender eigene benutzerdefinierte Funktionen auf die Daten anwenden. Hier wird der Vorteil der einfachen Anwendbarkeit der Funktionen zugunsten einer erhöhten Flexibilität aufgegeben. Alle diese Operationen können zunächst auf einer Stichprobe ausgeführt werden, da hierdurch bereits bestimmte Zusammenhänge schnell erkannt werden können. Darüber hinaus sind oftmals weitere Bereinigungsschritte wie Transformation, Normalisierung, Duplikatentfernung oder das Aufdecken von Ausreißern notwendig. Diese Operation werden ebenfalls in Form von Anfrageprimitiven in der Multidatenbanksprache bereitgestellt [44].

Somit fließen alle Operationen zunächst in eine Sichtdefinition ein, sind also effizient zu modifizieren. Da die Fusions-Engine die Entscheidung über die Materialisierung von Zwischenergebnissen treffen kann, ist eine transparente Implementierung der Undo-Funktion möglich.

Eng damit verbunden ist die Berücksichtigung von temporalen Eigenschaften des Fusionsprozesses. Zur grafischen Unterstützung ist diese in einer Visualisierung des Prozesses selbst sowie seiner zu unterschiedlichen Zeitpunkten anfallenden Ergebnisse zu berücksichtigen.

Ein entsprechendes Modell, welches auch die zeitlichen Eigenschaften von Nutzerinteraktionen berücksichtigt, wird in Abschnitt 4.5 beschrieben.

Hat der Anwender unter Verwendung der vorgestellten Integrationsmechanismen die Daten entsprechend seinen Wünschen aufbereitet, existieren zwei Möglichkeiten der weiteren Verwendung des Ergebnisses. Zum Einen kann die erzeugte globale Sicht auf die Daten direkt weiterverwendet werden, zum Anderen kann eine Materialisierung in eine lokale Datenbank vorgenommen werden, um weitere Bearbeitungs- und Analyseschritte der Daten zu beschleunigen.

4.4 Datenanalyse

In den meisten Anwendungsfällen wird allein durch die Integration verschiedener Datenquellen noch kein Gewinn erzielt. Gerade bei einer größeren Anzahl von Quellen bleiben aufgrund des resultierenden Datenvolumens interessante Aspekte oft verborgen. Daher sind die integrierten Daten weiter zu analysieren, um etwa Muster, Tendenzen, Regelmäßigkeiten oder Ausreißer aufzudecken. Das Suchen von Mustern und Zusammenhängen in Daten (*Data Mining*) als Teil des *Knowledge Discovery in Databases* (KDD) ist ein Forschungsbereich, der zunehmend an Bedeutung gewinnt [28]. Dabei befindet er sich an der Schnittstelle zwischen verschiedenen Bereichen wie beispielsweise Statistik, Datenbanken, Entdeckung von Mustern, Optimierung und Visualisierung [2]. In der Vergangenheit wurde eine Vielzahl von Verfahren entwickelt [20]. In Verbindung mit Zugriffs- und Integrationsmechanismen für heterogene Datenquellen versprechen diese Techniken neue, vielfältige Einsatzmöglichkeiten.

Ein Defizit aktueller Ansätze zur automatischen Datenanalyse in großen Datenbeständen – im Vergleich zu OLAP-Anwendungen, die eher eine anwendergesteuerte, navigierende Form unterstützen – ist die unzureichende Kopplung zum Datenbanksystem. So arbeiten viele Data Mining-Tools hauptspeicherbasiert und sind damit zwar sehr schnell, aber hinsichtlich der zu untersuchenden Datenmenge beschränkt. Obwohl vielfach die zu analysierenden Daten bereits in DBMS vorliegen, werden diese selten für Data Mining-Operationen eingesetzt. Ein Grund hierfür ist, dass moderne DBMS kaum Unterstützung für Data Mining-Verfahren in Form spezieller Operatoren oder Optimierungsstrategien anbieten. Ein neues Problem, welches durch den erhöhten Abstraktionsgrad gegenüber OLAP entsteht, ist, wie die Anfragen zu formulieren sind: Wie kann ein Anwender dem System mitteilen, nach was es suchen soll, wenn das Ziel der Suche noch nicht genau bekannt ist? Trotzdem wurden auch für diesen Bereich Anfragesprachen wie DMQL [26] und MSQL [30] entwickelt. Der Standardentwurf SQL/MM Part 6 enthält ebenfalls eine Schnittstelle für Data-Mining-Abfragen mittels SQL [47]. Eine enge Kopplung von Data Mining-Verfahren und DBMS bietet eine Reihe von Vorteilen, wie die Nutzung der durch das DBMS bereitgestellten Zugriffsstrukturen und Optimierungsstrategien, der Speicherverwaltung für die Bearbeitung großer Datenmengen sowie der ausgereiften Parallelisierungsmechanismen moderner Systeme [14]. Erst mit diesen Möglichkeiten wird ein interaktives *Ad-hoc Mining* möglich [12].

Vor diesem Hintergrund wird im Rahmen der hier vorgestellten Workbench eine enge Verbindung zwischen den Analysetechniken und der Datenbankfunktionalität angestrebt. So werden die einzelnen Analyseoperationen als SQL-Programme ähnlich gespeicherten Prozeduren implementiert und in der Fusions-Engine ausgeführt. Auf diese Weise lassen sich einzelne SQL-Anweisungen als Teil einer Analyseoperation direkt auf den Quelldaten bzw. auf den (materialisierten) Ergebnisrelationen anwenden.

Die Umsetzung von Data Mining-Verfahren auf der Basis von SQL ist eine aktuelle Herausforderung. Erste Arbeiten beschäftigen sich im Wesentlichen mit Klassifikationsverfahren [13] und der Ableitung von Assoziationsregeln [41]. Dabei wurde deutlich, dass noch Performance-Probleme bestehen, die durch neue Datenbankprimitive und Optimierungstechniken zu lösen sind [12]. Der erste Schritt ist also die Identifikation von Primitiven, die eine möglichst große Anzahl von Data Mining-Verfahren unterstützen. Dazu sind die Verfahren zu analysieren und gemeinsame, laufzeitintensive Anfragetypen zu bestimmen.

In Abschnitt 3 wurde der Entscheidungsbaum als ein für das Beispiel relevantes Data Mining-Verfahren genannt. Anhand dieses Beispiels soll im Folgenden die Entwicklung von oben angesprochenen Datenbankprimitiven aufgezeigt werden. Für die Erstellung von Entscheidungsbäumen sind viele Algorithmen wie ID3, C4.5, PUBLIC, SLIQ oder SPRINT bekannt, die häufig auf dem Greedy-Prinzip basieren. In der ersten Phase wird der Baum aufgebaut. Dazu wird an der Wurzel mit dem gesamten Datensatz begonnen und ein Kriterium gesucht, anhand dessen der Datensatz am besten partitioniert wird. Für das Beispiel wird das Giro-Volumen identifiziert. Die Definition des Kriteriums ist in den einzelnen Entscheidungsbaum-Verfahren unterschiedlich. Anschließend wird der weitere Baum rekursiv mit den entsprechenden Partitionen aufgebaut. In der zweiten Phase werden Teilbäume des Baumes abgeschnitten (Pruning), um eine zu genaue Anpassung an die Trainingsdaten (Overfitting) zu verhindern.

Die Bestimmung des Punktes, an dem eine Aufspaltung stattfindet, ist während der Aufbauphase der aufwändigste Teil. Hierbei müssen für jeden Knoten alle Tupel bestimmt werden, die zu der entsprechenden Partition gehören. Anschließend wird für diese Partition das Aufspaltungskriterium für alle noch nicht berücksichtigten Attribute berechnet. Hierzu ist eigentlich kein Zugriff auf die Basisdaten notwendig, da lediglich die Häu-

attrib_name	attrib_value	class	count
Giro	100,00	abgewandert	687
Spar	2000,00	nicht abgew.	154.830
Alter	64	nicht abgew.	87.533
Lage	5	abgewandert	34
⋮	⋮	⋮	⋮

Tabelle 3 CC-Tabelle

figkeiten von Attributkombinationen relevant sind. Diese Informationen können auch aus einer einfachen Tabelle entnommen werden, die aus Attributnamen, Attributwert, Klassenzugehörigkeit und Anzahl besteht. Bei nummerischen Attributen kann eine Diskretisierung sinnvoll sein, um die Anzahl der Daten gering zu halten. Eine solche Struktur wird als *CC-Tabelle* [13] oder in ähnlicher Form als *Attribute-Value-Class-Group* [22] bezeichnet. Für das Beispiel könnte sie etwa das in Tab. 3 dargestellte Aussehen haben. Eine solche Tabelle kann mit einer Anfrage der folgenden Form erzeugt werden [13]:

```
select 'A1' as attrib_name, A1 as attrib_value,
    C as class, count(*)
from BaseRelation
where <condition>
group by A1, C
    union all
select 'A2' as attrib_name, A2 as attrib_value,
    C as class, count(*)
from BaseRelation
where <condition>
group by A2, C
    union all
...
```

Die meisten Optimierer von aktuellen DBMS sind nicht in der Lage, einen Ausführungsplan zu generieren, der diese Anfrage mit einem einzigen Scan über die Basisrelation verarbeitet. Somit stellt diese Form von Anfragen einen geeigneten Kanditaten für eine Erweiterung der Fähigkeiten von Optimierern dar. Es besteht auch die Möglichkeit, direkt die Berechnung des Aufspaltungskriteriums zu optimieren. Damit würde aber lediglich ein einzelnes Verfahren beschleunigt, während die Optimierung der Verarbeitung von CC-Tabellen eine breite Palette von Verfahren unterstützt.

Neben der Berechnung der Häufigkeiten von Attributkombinationen wurde oben auch die Bestimmung der Tupel, die zu den einzelnen Partitionen gehören als gemeinsamer Teil aller Entscheidungsbaumverfahren genannt. Dieses wird typischerweise mittels sogenannter *Partial-Match*-Anfragen durchgeführt. Dieses sind Anfragen, die eine Bedingung der Form $P_1 \wedge P_2 \wedge \cdots \wedge P_m$ enthalten, wobei P_i ein Prädikat $A_j\ \theta\ v, \theta \in \{<, \leq, >, \geq, =, \neq, \ldots\}$ ist. Um beispielsweise die Partition des linken Knotens für den Entscheidungsbaum in Abb. 1 zu bestimmen, ist folgende Anfrage zu formulieren:

```
select *
from BaseRelation
where giro_volumen ≤ 100 and alter ≤ 25
```

Hier kann mit einer speziellen Zugriffsstruktur, die Partial-Match-Anfragen auf mehrdimensionalen Daten besonders gut unterstützt, ein Vorteil gegenüber den herkömmlichen Indexstrukturen, wie B-Baum oder Bitmap-Index erzielt werden. In [43] wurde das multidimensionale Hashing (MDH) als eine solche Zugriffsstrukur identifiziert. Dort werden auch Primitive für weitere Schritte untersucht, wie z.B. die Berechnung der AVC-Gruppen und die Anwendung des Baumes zur Vorhersage der Klassenzugehörigkeit neuer Daten.

Neben der Unterstützung für Entscheidungsbaumverfahren ist die Untersuchung weiterer Data Mining-Algorithmen auf Basis von SQL notwendig. Hierzu zählen Algorithmen zur Aufdeckung von Assoziationsregeln und Clustering-Verfahren.

4.5 Repräsentation der Temporalität

Die Definition des Fusionsprozesses spiegelt den zeitlichen Ablauf unterschiedlicher Methoden und ihrer Ergebnisse wider. Verschiedene Operatoren zur Datenaufbereitung und -analyse, wie sie in den vorangegangenen Abschnitten diskutiert wurden, bedingen einander oder werden unabhängig voneinander in zeitlicher Folge ausgeführt. Eine visuelle Repräsentation dieses Prozesses selbst sowie insbesondere seiner Ausführungsergebnisse sollte somit um temporale Aspekte angereichert werden, um zeitliche Abhängigkeiten dem Benutzer zugänglich machen zu können. Anderweitig verdeckte Strukturen des Fusionsprozesses können somit offensichtlich gemacht werden.

Als interaktives System ist die Workbench auch selbst durch temporale Parameter charakterisiert. Nutzerinterkation drückt sich in Ereignissen aus, die nacheinander vom System aufgenommen und bearbeitet werden. Jedes Interaktionsereignis ist ein potenzieller Aktionsauslöser und somit in der Lage, den Systemzustand zu modifizieren. Diese Änderung bleibt gültig bis zu einem weiteren Nutzerereignis oder bis sie durch Systemvoranschritt veraltet. Daraus resultiert, dass der Effekt jedes nutzerbasierten Ereignisses für einen spezifischen Zeitintervall gültig ist.

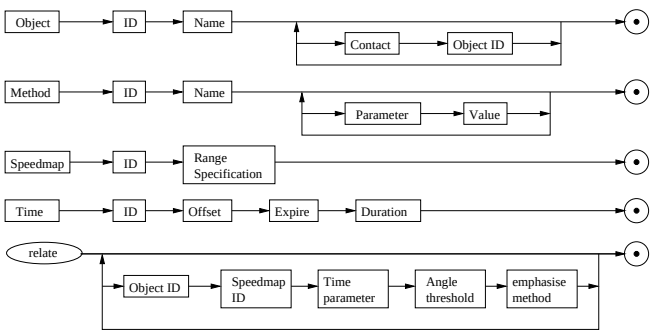

Abbildung 5 Modellierung von temporal geprägten Objektbeziehungen

In die Modellierung von Beziehungen zwischen Objekten fließt somit der Faktor *Zeit* mit ein. Abb. 5 veranschaulicht ein entsprechendes Modell. Darin werden Objekte in Beziehung zu einander gestellt, womit sie zu Segmenten (Clustern) zusammen gefasst werden können. Die visuelle Darstellung einer Objektbeziehung (hier als *emphasise method* bezeichnet) wird zeitlich parameterisiert. Weitere Modellbestandteile beschreiben zusätzliche Visualisierungs- sowie Interaktionselemente. So dient die *Speedmap* der Modellierung unterschiedlicher Interaktionsbereiche, die durch Geschwindigkeit und somit zeitlich abgegrenzt sind. Der *Angle threshold* dient ebenfalls der Interaktionsmodellierung und beschreibt Toleranzbereiche, die bei gleichzeitiger Interaktion mit mehreren Objekten berücksichtigt werden können.

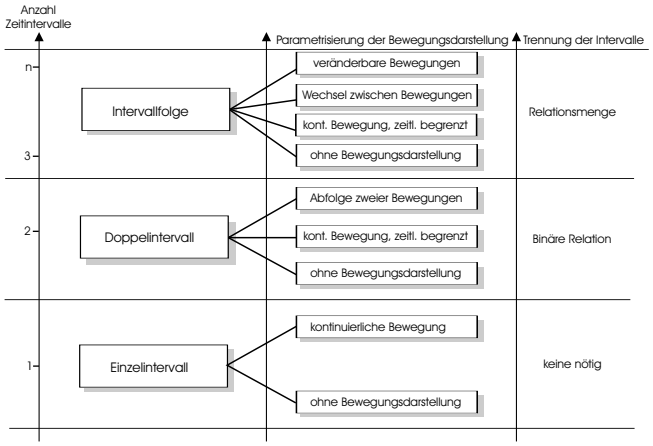

Abbildung 6 Ebenenmodell für zeitabhängige Bewegungsdarstellung

Die Verwaltung der temporal geprägten Objektbeziehungen erfolgt auf der Basis der Relationenalgebra von Allen [3]. Anlehnend an Freksas Betrachtungen in [21] können die zeitlichen Abhängigkeiten allerdings auch mit einer eingeschränkten Relationenmenge vollständig modelliert werden. Die Beziehung zwischen zwei gegebenen Intervallen $\mathcal{I}_i$ und $\mathcal{I}_j$ lässt sich somit beschreiben durch $\mathcal{I}_i \circ \mathcal{I}_j, \circ \in \{<, \leq, =, \geq, >\}$.

Zur Illustration sei das Ebenenmodell aus Abb. 6 betrachtet. In Abhängigkeit der modellierten Zeitintervalle schildert es die Möglichkeit, verschiedene Darstellungen zur Zeitrepräsentation zu verwenden. Für diesen Zweck sei die Verwendung von Objektbewegungen als eigenständige Präsentationsdimension motiviert [32]. Aufgrund ihrer zeitlichen Variabilität sind Bewegungen gut geeignet, um temporale Charakteristika von Objekten zu repräsentieren. Adäquat angewendet, bieten Bewegungen ferner die Möglichkeit, mit geringem kognitiven Aufwand ausdrucksstarke Darstellungsmuster zu erstellen [38]. Der Einsatz von Bewegungen ist abhängig von der Anzahl der für spezifische Objektbeziehungen bestehenden Zeitintervalle. Beispielhaft sei das Doppelintervall betrachtet. Zwei verschiedene Möglichkeiten zur Bewegungsdarstellung bieten sich hier: Zum einen die zeitlich begrenzte Darstellung einer kontinuierlichen Bewegung; zum anderen die Abfolge zweier unterschiedlicher Bewegungsdarstellungen. Ihre Dauer ergibt sich aus den Intervallgrenzen. Diese werden durch die drei Zeitpunkte t_1 (Beginn des ersten Intervalls), t_2 (Ende des ersten und Beginn des zweiten Intervalls) sowie t_3 (Ende des dritten Intervalls) festgelegt. Die Dauer der zeitlichen Begrenzung einer kontinuierlichen Darstellung beträgt somit $t_2 - t_1$. Da Intervallrelationen transitiv sind, gilt $t_2 \leq t_3 \Rightarrow t_2 - t_1 \leq t_3 - t_1$. Das verbleibende Intervall $t_3 - t_2$ kann entweder zur Darstellung des zweiten Bewegungsmusters oder aber zur Darstellung mit Hilfe klassischer Präsentationsvariablen [36] eingesetzt werden.

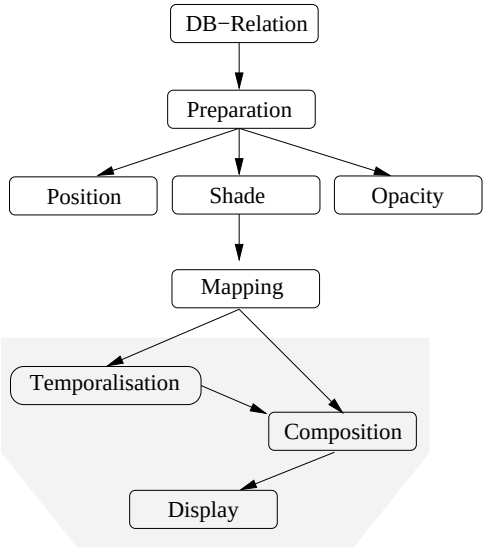

Abbildung 7 Visualisierungspipeline

Abb. 7 spiegelt den prinzipiellen Aufbau einer Visualisierungspipeline in Anlehnung an [46, S. 88f] wider. Die zentrale Funktion dieser Pipeline liegt in der Überführung von Daten, die nicht notwendigerweise inhärent geometrischer Natur sind, in eine geometrische Repräsentation, gewöhnlich *Glyphs* genannt [46, S. 185ff]. Dieser Schritt wird als *Mapping* bezeichnet. Die Bestimmung klassischer Präsentationsvariablen – Position, Form, Transparenz – zu diesen Glyphs erfolgt ebenfalls während der Transformation.

In einem separaten Schritt werden gegebenenfalls temporale Charakteristika der Darstellungsinformationen ausgewertet. Die Parameterisierung von Bewegungsdarstellungen wird somit auf geometrische Objekte appliziert. Eine generelle Einsetzbarkeit einmal modellierter Bewegungsmuster bleibt auf diesem Wege gewährleistet. Für die Übertragung auf andere Anwendungsbereiche ist lediglich die Visualisierungspipeline entsprechend modifiziert zu parameterisieren. Sollte direkt die Darstellung geometrischer Objekte Betrachtungsgegenstand sein, ist

selbige gar ohne eine Mappingphase vollständig realisierbar.

Um eine effiziente Ausführung zu gewährleisten, wird eine Visualisierungspipeline nur bedarfsgerecht ausgewertet. Änderungen in der temporalen Repräsentation (wie Bewegungsdarstellungen) führen lediglich zur Neuausführung des markierten Teilabschnittes der Visualisierungspipeline. Nicht notwendige Operationen werden somit vermieden.

4.6 Prototyp

Die Umsetzung der geschilderten Konzepte erfolgt auf Basis der in Abschnitt 4.1 geschilderten Architektur. Das System wird auf verschiedenen Unixplattformen implementiert. Die Verwendung einer standardisierten Datenbank-API, des Visualization Toolkits [46] sowie Open Inventors™ als Basis der Visualisierungskomponente sowie von Qt als Grundlage für die Benutzungsschnittstelle gewährleistet eine potenziell weiterführende Plattformunabhängigkeit. Der Prototyp benutzt als Anfragesystem FraQL und hat damit Zugriff auf heterogene Quellen und Integrationsoperationen.

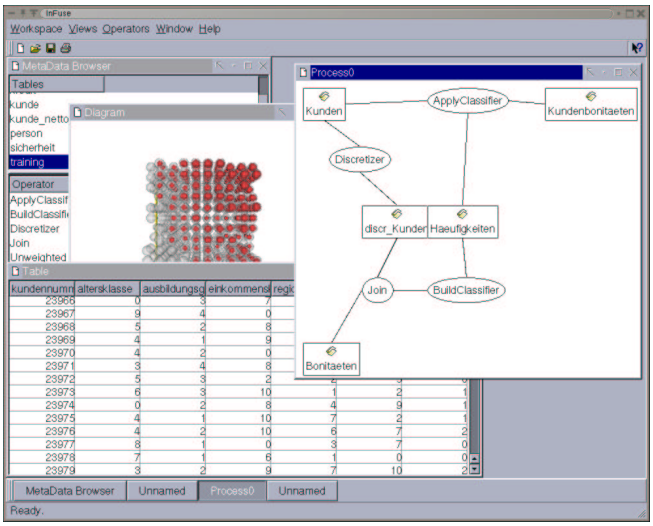

Abbildung 8 Prototyp mit Darstellung eines einfachen Prozessgraphen sowie verschiedenen Sichten auf einen exemplarischen Datensatz

Abb. 8 zeigt eine Beispielsitzung mit der Workbench. Der dargestellte Prozessgraph zur Bestimmung der Abfolge einzelner Fusionsschritte wird interaktiv aufgebaut. Eine Übersicht über die verfügbaren Datenquellen (Relationen, Views, etc.) sowie Operatoren liefert der Metadaten-Browser, im Bild links oben dargestellt. Exemplarisch sind als zwei Sichten auf die Relation mit den Trainingsdaten eine Tabelle und ein einfaches Diagramm abgebildet.

Weitere Entwicklungen sind in Arbeit. Zur Vervollständigung des Prototyps wird in erster Linie die Integration weiterer Fusionsoperatoren angestrebt. Neben datenbankorientierten Optimierungs- und Analysefunktionen sind hier insbesondere interaktive Data-Mining-Verfahren sowie Methoden zum automatischen Lernen interessant. Erstere ermöglichen beispielsweise mittels Methoden des interaktiven Clusterings eine weiterführende Benutzerunterstützung, während letztere die Analyse dahingehend unterstützen, dass sie wiederkehrende Verhaltens- und Abhängigkeitsmuster in größeren Datensätzen zu entdecken helfen.

Zusätzlich zur Einbettung weiterer Fusionsoperatoren in die Workbench ist eine Erweiterung der möglichen grafischen Sichten auf bestehende, generierte sowie abzuleitende Informationsbestände derzeit in der Entwicklung. Bezüglich der Diagrammdarstellungen, die der Unterstützung von Beziehungsdarstellungen einzelner Dimensionen der zu Grunde liegenden Daten dienen, um Kausalitäten zwischen Datensätzen erkennbar zu gestalten, ist die Erstellung automatisierter Filter von besonderem Interesse. Diese dienen der Aufbereitung des Eingabestroms innerhalb der Visualisierungspipeline, so dass die Notwendigkeit zur manuellen Darstellungsbearbeitung marginalisiert wird.

5 Verwandte Arbeiten

In den letzten Jahren wurden in der Literatur verschiedene Vorschläge zur Integration von heterogenen Datenquellen gegeben. Dabei lag zunächst der Schwerpunkt auf der Schemaintegration [8]. Aktuell, hervorgerufen durch die starke Verbreitung des Data-Warehouse-Konzeptes, wird stärker auf die Integration und Aufbereitung der Inhalte Wert gelegt [9,25].

Einen Überblick über die Data Warehouse-Architektur und den Ablauf des Prozesses von der Extraktion aus den lokalen Quellen bis zur Auswertung der Daten wird in [10] gegeben. [1] zeigt eine Übersicht über verschiedene kommerzielle Werkzeuge, die zur Extraktion, Transformation und zum Laden (ETL) der Daten benutzt werden. Beispiele für grafische ETL-Systeme sind die Microsoft Data Transformation Services und die Oracle Data-Mart Suite.

Weitere Forschungsprojekte zur interaktiven Datenaufbereitung und -integration sind unter anderem Clio [24] und Potter's Wheel [39]. Diese haben das Ziel einer interaktiven, datenorientierten und iterativen Aufbereitung der Daten für die weitere Analyse. In [24] verwenden die Autoren hierfür als Datenbank-Middleware das Multidatenbanksystem Garlic. Potter's Wheel ist ein ähnliches Projekt für ein Framework zur Unterstützung der interaktiven Datenaufbereitung. Dieses verwendet eine grafische Benutzungsschnittstelle in Form eines skalierbaren Spreadsheets. Mit diesem kann der Anwender seine Aktionen zur Datenaufbereitung sofort auf einer Stichprobe der Daten ausführen und validieren.

Wie am Beispiel des oben vorgestellten Projekts Clio erwähnt, stellen Multidatenbanksysteme mit ihren Mög-

lichkeiten des Zugriffs auf heterogene Datenquellen und der Integration von Daten die Grundlage für eine virtuelle und interaktive Aufbereitung dar. Beispiele für solche Systeme sind u.a. MSQL [23], SchemaSQL [33] oder auch FraQL [42].

Nach der Bereinigung und Integration der Daten erfolgen üblicherweise verschiedene numerische, statistische oder grafische Analysen zur Gewinnung von neuen Erkenntnissen aus dem aufbereiteten Datenbestand. Im Data Warehouse-Bereich erfolgt diese zumeist durch On-Line Analytical Processing-Werkzeuge (OLAP). Hierbei ist zu sagen, dass diese Software oft gänzlich abgekoppelt von den oben genannten ETL-Werkzeugen vorliegt.

Zur Datenanalyse werden verschiedene Algorithmen und Methoden benutzt, deren Spektrum von Ad-hoc-Anfragen bis zu lang laufenden Data-Mining-Methoden reicht. Um eine effiziente Verarbeitung der Daten in einem Datenbanksystem zu ermöglichen, müssen diese Algorithmen in das Datenbankmanagementsystem (DBMS) integriert werden. Eine Untersuchung der verschiedenen Möglichkeiten der Integration dieser Algorithmen wurde am Beispiel der Ableitung von Assoziationsregeln in [41] durchgeführt. Das System DBMiner [27] zum Beispiel integriert verschiedene Data-Mining-Algorithmen für On-Line Analytical Mining in großen Datenbanken bzw. Data Warehouses.

Bei der Exploration von Datenbankinhalten ist die Standardbenutzungsschnittstelle noch immer eine Tabellensicht [39]. Verschiedene Techniken wurden entwickelt, um multidimensionale Daten dem Anwender leichter zugänglich aufzubereiten [18], [27], [29]. Diese sind geprägt durch eine selektive Beschränkung der zu Grunde liegenden Dimensionalität zum Vorteil der besonders hervorgehobenen Darstellung von einzelnen Merkmale der Ausgangsdaten. Zu ihrem besseren Verständnis werden große Pivottabellen somit auf mehrere kleine Tabellen aufgeteilt. Die Darstellung derselben erfolgt partiell auf visuell reichere, aber kognitiv weniger belastende Variationen. Beispiele hierfür sind Kombinationen aus Bubble Plots, parallelen Koordinaten sowie Boxengraphiken [18,27]. Alternativ werden zur Darstellung sehr großer Datensätze Abbildungen auf Volumendarstellungen eingesetzt. Dabei existiert Information im 3D-Raum und wird nicht nur in Form von 2D-Daten in den 3D-Raum abgebildet. Volumenrendering ist im Gegensatz zu herkömmlichen Renderingmethoden nicht an die Vorgabe von geometrischen Informationen gebunden. Es eignet sich somit in besonderem Maße zur Darstellung relationaler Daten. Eine Methode der Abbildung dieser Daten in eine Volumendarstellung ist das Splatting [35]. Vorhergehend in die einzelnen räumlichen Dimensionen abgebildete Attribute werden dabei als Voxel auf die Bildebene projiziert. Sämtliche dieser Voxel werden anschliessend durchlaufen und ihr jeweiliger Einfluss auf die Pixel des Ergebnisbildes festgestellt. Die Voxel können somit jeweils als eine Art Energiequelle verstanden werden, die ihre Energie über ein spezifisches Bildgebiet ausbreitet. Ein weiterer Ansatz besteht in der Abbildung dreidimensionaler Würfelausschnitte im Cyberspace [4]

6 Zusammenfassung und Ausblick

Aufbauend auf den Grundkonzepten der Informationsfusion, die sich einzeln betrachtet inzwischen in einem weitgehend ausgereiften Stadium befinden, wird ein Rahmen entwickelt, der den gesamten Prozess der Informationsfusion von der Integration heterogener Datenquellen bis zur Ableitung neuer Informationen abdeckt. Dieser bietet die benötigten Basisdienste in einer einheitlichen Schnittstelle an, die es ermöglicht, dass zusätzliche Operatoren und Visualisierungsmethoden entwickelt und in das System eingebunden werden können.

Anhand von Beispielen und verschiedenen Anwendungsszenarien soll die praktische Relevanz der gebotenen Unterstützung evaluiert und erweitert werden. Somit wird der Satz von Präsentationsvariablen (Form, Farbe, Position, Transparenz) um Objektbewegungen [7] angereichert. Außerdem werden nicht nur weitere Operatoren wie beispielsweise zusätzliche KDD-Verfahren implementiert, sondern auch die Basisdienste ergänzt und verbessert. Zur Zeit werden Datenbankprimitive erarbeitet, die Effizienzsteigerungen insbesondere von Data-Mining-Algorithmen erlauben. Weitere Teilprojekte befassen sich mit der Generierung von Samples und Zwischenergebnissen mit iterativ zunehmender Genauigkeit, um den interaktiven Charakter der Fusionsaufgabe besser abbilden zu können.

Neben der Erweiterung der Basisdienste und der Entwicklung weiterer Operatoren ist geplant, Erkenntnisse verwandter Forschungsgebiete in die Workbench einfließen zu lassen. Denkbar sind hier Lernverfahren zur Optimierung einzelner Prozessschritte oder sogar des Gesamtprozesses, sowie Methoden der Wissensakquisition zur Einbindung natürlichsprachlicher Texte in den Fusionsprozess.

Literatur

1. Data Extraction, Transformation, and Loading Tools (ETL), http://www.dwinfocenter.org/clean.html, August 2000.
2. Pieter Adriaans and Dolf Zantinge. *Data Mining.* Harlow, 1996.
3. James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.
4. Ayman Ammoura, Osmar R. Zaïane, and Yuan Ji. Immersed Visual Data Mining: Walking the Walk. In Read [40], pages 202–218.
5. H. Arabnia and D. Zhu, editors. *Proc. of the Int. Conf. on Multisource-Multisensor Information Fusion - FUSION '98*, Las Vegas, NV, 1998. CSREA Press.
6. Ramon C. Barquin and Herbert A. Edelstein, editors. *Building, Using, and Managing the Data Warehouse.*

The Data Warehouse Institute Series. Prentice Hall PTR, New Jersey, USA, 1997.
7. Lyn Bartram. Enhancing Visualizations With Motion. In *Hot Topics: Information Visualization 1998*, North Carolina, USA, 1998.
8. C. Batini, M. Lenzerini, and S. B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys*, 18(4):323–364, December 1986.
9. George Burch. Building High Data Quality Into Your Data Warehouse. In Barquin and Edelstein [6], pages 69–83.
10. S. Chaudhuri and U. Dayal. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, 26(1), 1997.
11. S. Chaudhuri, R. Motwani, and V.R. Narasayya. On Random Sampling over Joins. In A. Delis, C. Faloutsos, and S. Ghandeharizadeh, editors, *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadephia, Pennsylvania, USA*, pages 263–274. ACM Press, 1999.
12. Surajit Chaudhuri. Data Mining and Database Systems: Where is the Intersection? *Data Engineering Bulletin*, 21(1):4–8, 1998.
13. Surajit Chaudhuri, Usama M. Fayyad, and Jeff Bernhardt. Scalable Classification over SQL Databases. In *Proceedings of the 15th International Conference on Data Engineering, 1999, Sydney, Austrialia*, pages 470–479. IEEE Computer Society, 1999.
14. John Clear, Debbie Dunn, Brad Harvey, Michael L. Heytens, Peter Lohman, Abhay Mehta, Mark Melton, Lars Rohrberg, Ashok Savasere, Robert M. Wehrmeister, and Melody Xu. NonStop SQL/MX Primitives for Knowledge Discovery. In *Proc. 5th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining 1999, San Diego, CA USA*, pages 425–429, 1999.
15. S. Conrad, G. Saake, and K. Sattler. Informationsfusion - Herausforderungen an die Datenbanktechnologie. In A. P. Buchmann, editor, *Datenbanksysteme in Büro, Technik und Wissenschaft, BTW'99, GI-Fachtagung, Freiburg, März 1999*, Informatik aktuell, pages 307–316, Berlin, 1999. Springer-Verlag.
16. Oliver Dunemann. Unterstützung der Quantifizierung von Kreditrisiken mit Methoden der Informationsfusion. Präsentiert bei: 5. Internationale Tagung Wirtschaftsinformatik / 3. Tagung Informationssysteme in der Finanzwirtschaft. Augsburg, September 2001.
17. Jochen Dzienziol, Nina Schroeder, and Christoph Wolf. Kundenwertorientierte Unternehmenssteuerung. In Hans Ulrich Buhl, Nina Kreyer, and Werner Steck, editors, *e-Finance - Innovative Problemlösungen für Informationssysteme in der Finanzwirtschaft*, pages 63–86, Augsburg, 2001. Springer, Berlin, Heidelberg, New York.
18. Stephen G. Eick. Visualizing Multi-Dimensional Data. *Computer Graphics*, pages 61–67, February 2000.
19. Jens Eickbusch. *Kundenabwanderungen in Kreditinstituten – Eine empirische Analyse mittels Data Mining-Methoden für das Privatkundengeschäft einer Großsparkasse.* Inauguraldissertation, Gerhard Mercator-Universität Duisburg, Duisburg, 2000.
20. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI Press / The MIT Press, Menlo Park, California, 1996.
21. Christian Freksa. Temporal Reasoning Based on Semi-Intervals. *Artificial Intelligence*, 52(1-2):199–227, 1992.
22. Johannes Gehrke, Raghu Ramakrishnan, and Venkatesh Ganti. Rainforest - a framework for fast decision tree construction of large datasets. In Ashish Gupta, Oded Shmueli, and Jennifer Widom, editors, *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, pages 416–427. Morgan Kaufmann, 1998.
23. J. Grant, W. Litwin, N. Roussopoulus, and T. Sellis. Query Languages for Relational Multidatabases. *The VLDB Journal*, 2(2):153–171, April 1993.
24. Laura M. Haas, Renée J. Miller, B. Niswonger, Mary Tork Roth, Peter M. Schwarz, and Edward L. Wimmers. Transforming heterogeneous data with database middleware: Beyond integration. *IEEE Data Engineering Bulletin*, 22(1):31–36, 1999.
25. Katherine Hammer. Migrating Data from Legacy Systems: Challenges and Solutions. In Barquin and Edelstein [6], pages 27–40.
26. J. Han, Y. Fu, K. Koperski, W. Wang, and O. Zaiane. DMQL: A Data Mining Query Language for Relational Databases. In H. V. Jagadish and Inderpal Singh Mumick, editors, *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, June 4-6, 1996*. ACM Press, 1996.
27. Jiawei Han. Towards On-Line Analytical Mining in Large Databases. *ACM SIGMOD Record*, (27):97–107, 1998.
28. Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2001.
29. Marti A. Hearst and Chandu Karadi. Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy. In *Proceedings of the* 20^{th} *Annual International ACM/SIGIR Conference*, Philadelphia, PA, July 1997.
30. Tomasz Imielinski and Aashu Virmani. MSQL: A Query Language for Database Mining. *Data Mining and Knowledge Discovery*, 3(4):373–408, December 1999.
31. R. Jesse, I. Geist, and O. Dunemann. Konzeption einer datenbankbasierten Plattform für die Informationsfusion. Preprint 9, Fakultät für Informatik, Universität Magdeburg, Magdeburg, 2001.
32. Roland Jesse and Thomas Strothotte. Motion Enhanced Visualization in Support of Information Fusion. In Hamid R. Arabnia, editor, *Proceedings of International Conference on Imaging Science, Systems, and Technology (CISST'2001)*, pages 492–497. CSREA Press, June 2001.
33. L.V.S. Lakshmanan, F. Sadri, and I.N. Subramanian. SchemaSQL – A Language for Interoperability in Relational Multidatabase Systems. In T.M. Vijayaraman, A.P. Buchmann, C. Mohan, and N.L. Sarda, editors, *VLDB'96, Proceedings of 22nd International Conference on Very Large Data Bases, September 3-6, 1996, Bombay, India*, pages 239–250. Morgan Kaufmann, 1996.
34. R.C. Luo and M.G. Kay, editors. *Multisensor Integration and Fusion for Intelligent Machines and Systems*. Ablex Publishing Corporation, Norwood, NJ, 1995.

35. Klaus Mueller, Torsten Möller, and Roger Crawfis. Splatting Without The Blur. In *Proceedings of IEEE Conference on Visualization 1999*, pages 363–371, October 1999.
36. Emanuel G. Noik. A Space of Presentation Emphasis Techniques for Visualizing Graphs. In *Proceedings of Graphics Interface '94*, pages 225–233, 1994.
37. F. Olken and D. Rotem. Simple Random Sampling from Relational Databases. In W.W. Chu, G. Gardarin, S. Ohsuga, and Y. Kambayashi, editors, *VLDB'86 Twelfth International Conference on Very Large Data Bases,August 25-28, 1986, Kyoto, Japan, Proceedings*, pages 160–169. Morgan Kaufmann, 1986.
38. Zenon W. Pylyshyn, J. Burkell, B. Fisher, C. Sears, W. Schmidt, and L. Trick. Multiple parallel access in visual attention. *Canadian Journal of Experimental Psychology*, 1993.
39. Vijayshankar Raman and Joseph M. Hellerstein. An Interactive Framework for Data Cleaning. http://control.cs.berkeley.edu/abc/, 2000. Working draft.
40. Brian Read, editor. *Advances in Databases, Proceedings of the 18th British National Conference on Databases (BNCOD 18)*, volume 2097 of *Lecture Notes in Computer Science*, Chilton, UK, July 2001. Springer-Verlag, Berlin, Heidelberg.
41. Sunita Sarawagi, Shiby Thomas, and Rakesh Agrawal. Integrating Mining with Relational Database Systems: Alternatives and Implications. In Laura M. Haas and Ashutosh Tiwary, editors, *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*, pages 343–354. ACM Press, 1998.
42. K. Sattler, S. Conrad, and G. Saake. Adding Conflict Resolution Features to a Query Language for Database Federations. *Australian Journal of Information Systems*, 8(1):116–125, 2000.
43. K. Sattler and O. Dunemann. SQL Database Primitives for Decision Tree Classifiers. In *Proc. of the 10th ACM CIKM Int. Conf. on Information and Knowledge Management, November 5–10, 2001, Atlanta, Georgia, USA*, 2001.
44. K. Sattler and E. Schallehn. A Data Preparation Framework based on a Multidatabase Language. In M. Adiba, C. Collet, and B.P. Desai, editors, *Proc. of Int. Database Engineering and Applications Symposium (IDEAS 2001)*, pages 219–228, Grenoble, France, 2001. IEEE Computer Society.
45. Kai-Uwe Sattler, Oliver Dunemann, Ingolf Geist, Gunter Saake, and Stefan Conrad. Limiting Result Cardinalities for Multidatabase Queries using Histograms. In Read [40], pages 152–167.
46. Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit – An Object-Oriented Approach to 3D Graphics*. Prentice Hall PTR, 2. edition, 1998.
47. Friedemann Schwenkreis. New Working Draft of SQL/MM Part 6: Data Mining based on BHX008 and BHX033-BHX039. Technical Report, International Organization for Standardization (ISO), Mai 2000.
48. J.S. Vitter. An Efficient Algorithm for Sequential Random Sampling. *ACM Transactions on Mathematical Software*, 13(1):58–67, March 1987.