INFUSE – Eine datenbankbasierte Plattform für die Informationsfusion*

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

Fakultät für Informatik, Universität Magdeburg
Postfach 4120, D-39016 Magdeburg
fusion@cs.uni-magdeburg.de
http://fusion.cs.uni-magdeburg.de

Zusammenfassung Informationsfusion als Prozess der Integration und Interpretation heterogener Daten mit dem Ziel der Gewinnung neuer Informationen einer höheren Qualität eröffnet eine Vielzahl von Anwendungsgebieten. Gleichzeitig erfordert dieser Prozess aber auch eine enge Verzahnung der bislang häufig noch isoliert vorliegenden Werkzeuge und Techniken zum Zugriff auf heterogene Datenquellen, deren Integration, Aufbereitung, Analyse und Visualisierung. In diesem Beitrag werden erste Ergebnisse der Entwicklung einer Workbench vorgestellt, die durch konsequente Nutzung von Datenbanktechniken eine durchgängige Unterstützung dieser Schritte ermöglicht.

1 Einleitung und Motivation

Die Bedeutung der Aufgabe, verteilt und heterogen vorliegende Datenbestände in einer integrierten Form darzustellen und zu analysieren, wurde in letzter Zeit zunehmend erkannt. Die wissenschaftliche Arbeit konzentrierte sich dabei zunächst auf das Erarbeiten von Lösungen für Teilschritte. So wurden 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 wurden entwickelt. Da jedoch die Integration dieser Komponenten bisher nicht oder nur teilweise durchgeführt wurde, konnte erhebliches Optimierungspotential nicht ausgeschöpft werden. Hier ist beispielsweise eine dynamische und anpassungsfähige Entscheidungsfindung bezüglich der Materialisierung von Zwischenergebnissen vorgesehen.

An der Universität Magdeburg wird zur Zeit unter dem Arbeitstitel INFUSE eine Workbench entwickelt, die die Zusammenführung der Komponenten der Informationsfusion zum Ziel hat. Durch ein offenes und modulares Konzept wird ein Rahmen aus Basisdiensten geschaffen, auf dem aufbauend weitere Komponenten entwickelt werden können. Dabei werden bereits in der Analysephase Aspekte der Teilaufgaben der Informationsfusion berücksichtigt, indem von Praxisbeispielen ausgehend beispielhaft Fusionsprozesse modelliert werden. Solche Basisdienste stellen neben den Integrations-

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

und Darstellungskomponenten einen zentralen Authentifizierungsmechanismus und eine einheitliche Fehlerbehandlung zur Verfügung.

Im Weiteren ist die Arbeit wie folgt gegliedert. Zunächst zeigt der Abschnitt 2 den Stand der Technik auf und stellt verwandte Arbeiten vor. Anschließend wird im Abschnitt 3 der Begriff, der Prozess sowie mögliche Anwendungsfelder der Informationsfusion dargestellt. Ein Beispiel, das durchgängig in der Arbeit benutzt werden soll, wird im Abschnitt 4 entwickelt. Der Abschnitt 5 stellt den Hauptteil der Arbeit dar und zeigt die Integration verschiedener Methoden in einer *Workbench* zur Unterstützung der Informationsfusion. Weiterhin beschreibt dieser Abschnitt den entstandenen Prototypen. Eine Zusammenfassung und ein Ausblick auf weitere Forschungsschwerpunkte zur Informationsfusion beschliessen die Arbeit.

2 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 [BLN86]. Aktuell, hervorgerufen durch die Einführung des Data-Warehouse -Konzeptes, wird stärker auf die Integration und Aufbereitung der Inhalte Wert gelegt.

Einen Überblick über die Data-Warehouse(DW)-Architektur und den Ablauf des DW-Prozesses von der Extraktion aus den lokalen Quellen bis zur Auswertung der Daten wird in [CD97] gegeben. [DWI00] 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 DataMart Suite.

Weitere Forschungsprojekte zur interaktiven Datenaufbereitung und -integration sind unter anderem Clio [HMN+99] und Potter's Wheel [RH00]. Diese haben das Ziel einer interaktiven, datenorientierten und iterativen Aufbereitung der Daten für die weitere Analyse. In [HMN+99] 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 Möglichkeiten 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 [GLRS93], SchemaSQL [LSS96] oder auch FRAQL [SCS00].

Nach der Aufbereitung und Integration der Daten kann eine Analyse zur Gewinnung von Informationen erfolgen. 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.

Als Beispiele sind unter anderem Ableitung von Assoziationsregeln [AMS+96] oder auch Klassifizierungsalgorithmen zu nennen. 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 [STA98] durchgeführt. Das System DBMiner [Han98] 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 [RH00]. Verschiedene Techniken wurden entwickelt, um multidimensionale Daten dem Anwender leichter zugänglich aufzubereiten [Eic00], [Han98], [HK97]. Diese sind geprägt durch eine selektive Beschränkung der zu Grunde liegenden Dimensionalität zum Vorteil der besonders hervorgehobenen Darstellung einzelner 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 [Eic00,Han98]. 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 [MMC99]. 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.

3 Informationsfusion – Begriff und Anwendungen

Der Begriff der *Informationsfusion* beschreibt den Prozess der Integration und Interpretation von Daten aus heterogenen Quellen sowie die darauf aufbauende Konstruktion von Modellen für einen bestimmten Problembereich mit dem Ziel der Gewinnung von Informationen einer neuen, höheren Qualität. Diese Definition erklärt die Informationsfusion als ein interdisziplinäres Gebiet, das auf Methoden und Techniken verschiedener Bereiche, wie z.B. Datenbanken, Statistik, Maschinelles Lernen und Visualisierung zurückgreift.

Der Fusionsprozess beinhaltet dabei die verschiedenen Aspekte Datenzugriff, Datenintegration, Analyse und Verdichtung, Präsentation und Weiterverarbeitung sowie die Repräsentation der Metadateninformationen. Eine genaue Anforderungsanalyse und Beschreibung der einzelnen Bereiche sind in [CSS99] dargestellt.

Die verschiedenen Schritte der Integration und Analyse der Daten können durch einen Graphen modelliert werden. Hierbei beschreiben die Knoten des Graphen die

Datenquellen und die Operationen auf diesen Quellen. Die Kanten beschreiben die Aufeinanderfolge der Operationen. Dieser Graph dient im weiteren als Modell für ein *Worksheet*, welches die verschiedenen Sichten auf den Prozess beschreibt. Hierbei kann der Graph einmal direkt grafisch modelliert beziehungsweise implizit durch die Anwendung der verschiedenen Operationen auf die Daten in einer Spreadsheet-Ansicht erzeugt werden.

Es existieren vielfältige Anwendungsgebiete für die Informationsfusion. Exemplarisch sei die Untersuchung von Gensequenzen aus verschiedenen Gen- und Stoffwechseldatenbanken in der Bioinformatik genannt. Weiterhin ist der Produktionsvorbereitungsprozess in der Giesserei-Industrie ein Anwendungsgebiet. Ein weiteres Beispiel stellt die Analyse von Konto- und Kundendaten dar. Aus diesem Bereich wurde auch das Beispielszenario gewählt, welches durchgehend in der weiteren Arbeit benutzt werden soll.

4 Beispiel

Zur Verdeutlichung der Zielsetzung einer Workbench zur Unterstützung der Informationsfusion wird in diesem Abschnitt ein Beispiel konstruiert: In einem Kreditinstitut sollen potentielle Kreditnehmer in Bonitätsklassen eingeteilt werden, um eine differenzierte Preisstellung zu ermöglichen. Dazu werden vorliegende Kontrakte analysiert, die Kunden klassifiziert und deren mittlere historische Ausfallrate bestimmt. In Abhängigkeit dieser Ausfallrate werden sie Bonitätsklassen zugeordnet. Diese Daten dienen wiederum als Trainingsmenge für Methoden zur Bonitätseinstufung. Folgendes Beispiel skizziert diesen Vorgang:
Für jeden Kunden liege neben dem bisherigen Zahlungsverhalten das Einkommen vor. Für das Einkommen werden Klassen gebildet und die jeweiligen relativen Häufigkeiten der Ereignisse *Kreditausfall* und *Rückzahlung* bestimmt. Tabelle 1 zeigt eine solche Häufigkeitsverteilung. Anschliessend werden in Abhängigkeit der Ausfallhäufigkeiten Gruppen gebildet, die die Bonitätsklassen darstellen. Ein solches Ergebnis ist in Tabelle 2 dargestellt. Mit dieser Klassifizierung kann in Abhängigkeit vom Einkommen ab-

Einkommen	Rel. Häufigkeit Kreditausfall
$>$150.000	0,02%
100.000 bis 149.999	0,08%
80.000 bis 99.999	0,35%
60.000 bis 79.999	0,50%
40.000 bis 59.999	1,20%
bis 39.999	2,30%

Tabelle 1. Historische Ausfallraten

geschätzt werden, mit welcher Wahrscheinlichkeit ein Engagement ausfallen wird. Dieses kann bei der Angebotserstellung berücksichtigt werden, indem für Kunden schlechterer Bonität ein entsprechend höherer Zinsaufschlag (Credit Spread) verlangt wird.

Einkommen	Mittlere rel. Häufigkeit Kreditausfall	Bonitätsklasse
>100.000	0,06%	A
40.000 bis 99.999	0,42%	B
bis 39.999	2,30%	C

Tabelle 2. Ableitung der Bonitätsklassen

Das beispielhaft betrachtete Kreditinstitut verfügt einerseits über eine Kundendatenbank mit sozio-demographischen Daten, die mit den Kontodaten verknüpft werden soll, um eine geeignete Klassifizierung zu bestimmen. Dabei dienen alle abgelaufenen Kontrakte als Trainingsmenge. Für die restlichen Kunden werden die Bonitäten (und somit die zu erwartenden Ausfallwahrscheinlichkeiten) anhand des gefundenen Klassifizierers bestimmt. Mit einer Ex-Post-Analyse kann später die Klassifizierungsgüte für die Prognose ermittelt werden.

Die Kundendatenbank enthält die Relation

kunde(kundennummer,alter,ausbildung,einkommen,region,branche)

während die Kontodatenverwaltung die Relationen

kredit(kontonummer,kundennummer,saldo,zinssatz,restlaufzeit)

und

sicherheit(sicherheit_id,kontonummer,kundennummer, anrechnungsfaktor,betrag)

enthält. Der erste Integrationsschritt besteht in der Zusammenfassung der Relationen `kredit` und `sicherheit`. In SQL entspricht dieses Vorgehen der Definition von zwei Views.

```
create view kontosicherheit as
select kundennummer, kontonummer,
sum(anrechnungsfaktor * betrag) as volumen
from sicherheit
group by kundennummer, kontonummer

create view nettokredit as
select kredit.kundennummer, sum(saldo - volumen) as saldo
from kredit, kontosicherheit
where kredit.kundennummer = kontosicherheit.kundennummer
and kredit.kontonummer = kontosicherheit.kontonummer
group by kredit.kundennummer
```

Es kommen also zwei Operatoren zum Einsatz, die die Funktionalität von Sichten in relationalen Systemen abbilden. Die View `nettokredit` wird anschliessend über einen Join-Operator mit `kunde` aus der zweiten Datenbank zur Sicht

kunde_nettokredit(kundennummer, alter, ausbildung, einkommen, region, branche, saldo)

verknüpft. Um die gewünschte Klassifizierungsmethode durchführen zu können, müssen die einfließenden Attribute in diskrete Gruppen überführt werden. Diese werden zunächst vom Anwender fest vorgegeben und nicht automatisch (denkbar wäre hier eine automatische Klassenbildung nach Verteilungsgesichtspunkten) berechnet. Das Ergebnis dieser Operation wird mit den wie oben beschrieben bestimmten Bonitäten angereichert und bildet die Trainingsmenge

kunden_bonitaet(kundennummer, alter_kl, ausbildung, einkomm_kl, bonitaet, region, branche_kl, saldo_kl)

für den Klassifizierungsalgorithmus. Untersuchungen zum Themenbereich der Klassifizierung von Kreditnehmern finden sich u.a. in [BM99].

5 Werkzeugunterstützung für die Informationsfusion

Die Basis der Informationsfusion bildet eine enge Verzahnung der Integration heterogener Daten mit ihrer Aufbereitung sowie Analyse. Nur so kann eine interaktive Arbeitsweise unterstützt werden, die dem iterativen Charakter des Fusionsprozesses gerecht wird. Hierfür wird ein Vorrat an integrierten Werkzeugen für die einzelnen Schritte des Prozesses benötigt. Im folgenden werden erste Ergebnisse der Entwicklung einer solchen *Workbench* vorgestellt.

5.1 Architektur

Eine Workbench zur Unterstützung der Informationsfusion muss eine effiziente und interaktive Analyse großer, zum Teil heterogener Datenbestände ermöglichen. Dies umfasst die Definition und Ausführung von Anfragen, die Transformation von Daten sowie die Anwendung von Analyseoperationen und die Visualisierung der Ergebnisse. Vergleichbare Anforderungen sind auch in OLAP-Anwendungen zu finden, so dass für die Fusionsworkbench ein ähnlicher Architekturansatz gewählt wurde (Abb. 1).

Die Basis des Systems bildet die *Fusions-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 [SCS00]. Das Anfragesystem umfaßt weiterhin eine lokale Datenbank für temporäre Daten (z.B. Materialisierungen) und Ergebnisse sowie die eigentlichen Fusionsoperatoren, die ähnlich gespeicherten Prozeduren direkt auf den integrierten Datenbeständen ausgeführt werden können. Der Worksheet-Manager verwaltet komplette Fusionsprozesse, indem die Reihenfolge einzelner Aufbereitungs- und Analyseschritte berücksichtigt wird und so bei Daten- oder Parameteränderungen nur die betroffenen Schritte neu ausgeführt werden müssen.

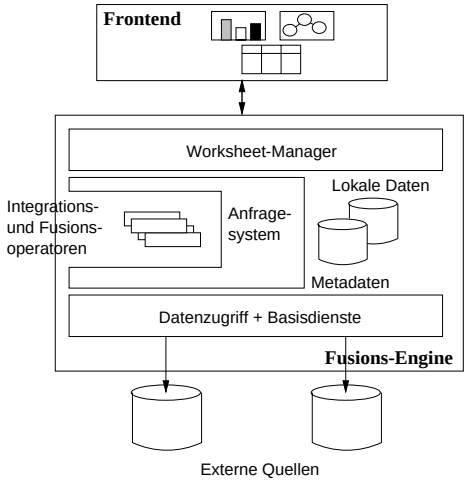


Abbildung 1. Architektur der Workbench

Die Benutzungsschnittstelle wird durch das Workbench-Frontend bereitgestellt. Mit diesem graphischen Analyse- und Definitionswerkzeug hat der Benutzer über die Fusions-Engine Zugriff auf die Daten der einzelnen Quellen. So können interaktiv Integrations- und Fusionsoperationen ausgeführt, Anfragen formuliert und die Ergebnisse visualisiert werden.

Die Architektur der Workbench ist mit der Trennung in Fusions-Engine und Frontend mit Ansätzen aus dem OLAP-Bereich vergleichbar. Ein wesentlicher Unterschied besteht jedoch darin, dass die zu analysierenden Daten nicht vorab extrahiert, transformiert, bereinigt und redundant in einem Warehouse abgelegt werden müssen. 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 erste Analysen durchgeführt werden, ohne dass zuvor Daten aufwendig migriert und transformiert werden müssen. 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.

5.2 Aufbereitung und Integration der Daten

Nach der Beschreibung der Architektur der Workbench soll nachfolgend auf die Grundlagen der Datenanalyse sowie die Aufbereitung und Integration der heterogenen Daten eingegangen werden. In den Aufbereitungs- und Integrationsschritten soll die Qualität der Daten gesichert werden, da sich nur aus qualitativ hochwertigen Daten relevante Analyseergebnisse ableiten lassen.

Im Data-Warehouse-Bereich werden Data-Cleaning- bzw. ETL-Werkzeuge zur Aufbereitung und Integration der Daten benutzt. Um diese Aktivitäten zu ermöglichen, muss ein Werkzeug folgende Eigenschaften aufweisen:

- **Schnelle Reaktionszeiten**: Die Erkennung und Lösung von Konflikten erfordert eine Interaktion mit dem Anwender. Um diese Arbeitsweise zu ermöglichen, müssen die Werkzeuge 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 möglich. Vielmehr sollten zunächst Stichproben untersucht werden, um anschließend die Ergebnisse auf den gesamten Datenbestand anzuwenden.
- **Integration der Werkzeuge**: Für die Aufbereitung der Daten müssen unterschiedliche Aktionen und Algorithmen zur Konfliktentdeckung und -lösung angewendet werden. 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, wodurch die Erkennung und Lösung dieser verschachtelten Konflikte erst ermöglicht wird.
- **Durchgehende grafische Benutzerführung**: Die Interaktion mit den Werkzeugen soll möglichst durch eine grafische Benutzerführung erleichtert werden. Hierdurch können einerseits Einarbeitungszeiten in komplexe Programmierumgebungen verringert werden und 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.

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 Daten auf dieses abgebildet werden müssen. Für diese Abbildung werden die Anfrage- und Integrationfähigkeiten der Multidatenbanksprache FRAQL eingesetzt. Diese ist mit ihren Eigenschaften in [SCS00] ausführlich beschrieben.

Zunächst erfolgt die Anwendung der Integrationsschritte auf einer Stichprobe des gesamten Datenbestandes. Diese kann durch bekannte Sampling-Verfahren wie z.B. [Vit87] erzeugt werden, die wie z.B. in [OR86] und [CMN99] beschrieben, in das Multidatenbanksystem integriert werden können. Die nachfolgende Beschreibung zeigt, wie die einzelnen Konfliktklassen behandelt werden. Vorausgesetzt wird 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.

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 oben beschriebenen Anpassung der Schemata werden Konflikte in den Datenwerten betrachtet. Diese sind allerdings nicht unabhängig von der Schemaanpassung zu sehen, da eine Abhängigkeit in beiden 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 Funktion 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 hier bereits bestimmte Zusammenhänge schnell erkannt werden können. Weiterhin 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.

Als grafische Unterstützung stehen neben der Tabellenansicht mit der Möglichkeit der Anwendung der Operationen weitere Ansichten zur Verfügung, die im Abschnitt 5.4 beschrieben werden.

Dieser Ansatz der datenorientierten Aufbereitung und Integration der heterogenen Quellen unterstützt die iterative und interaktive Lösung der auftretenden Konflikte. Durch die anfängliche Benutzung von Stichproben kann eine schnelle Reaktionszeit 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.

Hat der Anwender unter Verwendung der vorgestellten Integrationsmechanismen die Daten seinen Wünschen entsprechend 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.

5.3 Datenanalyse

In vielen Anwendungsfällen liefert die Integration verschiedener Datenquellen allein noch keinen Gewinn. 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 oder Regelmäßigkeiten aufzudecken. Zu diesem Zweck wurden in der Vergangenheit insbesondere auf dem Gebiet des Data Mining eine Vielzahl von Verfahren entwickelt [FPSSU96]. 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 benutzergesteuerte, navigierende Form darstellen – ist die unzureichende Kopplung zum Datenbanksystem. So arbeiten viele Data-Mining-Tools hauptspeicherbasiert und sind damit hinsichtlich der zu untersuchenden Datenmenge beschränkt. Andererseits bieten auch moderne Datenbankmanagementsysteme kaum Unterstützung in Form spezieller Operatoren oder Optimierungsstrategien. Ausnahmen sind hier u.a. [CDH+99]. Trotzdem bietet eine

enge Kopplung 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 [Cha98].

Vor diesem Hintergrund wird im Rahmen der hier vorgestellten Workbench eine enge Verbindung zwischen Analysetechniken und 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 [CFB99] und der Ableitung von Assoziationsregeln [STA98]. Dabei wurde deutlich, dass noch Performance-Probleme bestehen, die durch neue Datenbankprimitive und Optimierungstechniken zu lösen sind [Cha98].

Für erste Versuche im Rahmen der Workbench wurde daher ein einfacher Bayes'scher Klassifikator implementiert. Die Idee dieses Verfahrens ist die Vorhersage der Zuordnung eines Objektes zu einer diskreten Klasse C auf der Basis der diskreten Werte der Objektattribute $A_1 \ldots A_n$. Unter der Annahme der Unabhängigkeit der Attribute ist die optimale Vorhersage der Klassenwert c, für den $Pr(C = c|A_1 = a_1 \wedge \cdots \wedge A_n = a_n)$ maximal ist. Nach der Bayes'schen Regel ist einem neuen Objekt der Klassenwert c zuzuordnen, der das folgende Produkt maximiert [Elk97]:

$$\prod_{i=1\ldots n} Pr(A_i = a_i|C = c)$$

Die einzelnen Faktoren können jeweils aus Trainingsdaten ermittelt werden, für die die Zuordnung zu Klassen bekannt ist:

$$Pr(A_i = a_i|C = c) = \frac{count(A_i = a_i \wedge C = c)}{count(C = c)}$$

Die Anwendung dieses Prinzips erfordert zwei Operatoren: *BuildClassifier* und *ApplyClassifier*. *BuildClassifier* erzeugt anhand einer Relation mit Trainingsdaten eine Relation mit den Häufigkeiten für die Bestimmung der obigen Faktoren. Diese Relation wird anschließend vom Operator *ApplyClassifier* genutzt, um für die eigentlichen Eingangsdaten eine Klassenzuordnung zu bestimmen (Abb. 2).

Für das in Abschnitt 4 eingeführte Beispiel lassen sich die Häufigkeitsinformationen durch folgende SQL-Anfrage bestimmen [WZ99]:

```
insert into counts
select alter_kl, ausbildung, einkomm_kl, bonitaet as class,
       count (*) as cnt
from kunden_bonitaet
group by grouping sets ((alter_kl, bonitaet),
       (ausbildung, bonitaet), (einkomm_kl, bonitaet),
       (bonitaet))
```

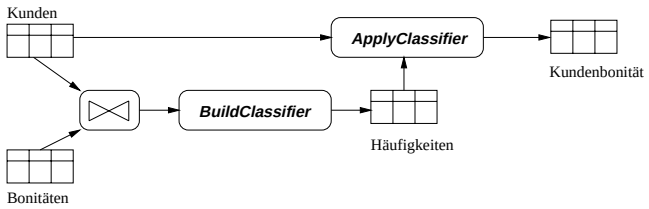


Abbildung 2. Klassifikationsprozeß

Das Ergebnis dieser Anfrage ist eine Relation, die die Häufigkeit der einzelnen Attributwerte in Verbindung mit der Klassenzuordnung enthält. Eine solche Datenstruktur wird in ähnlicher Form auch in [CFB99] als sogenannte CC-Tabelle für die Implementierung eines Entscheidungsbaumverfahrens genutzt.

Der obige Schritt wird durch den *BuildClassifier*-Operator implementiert. Dieser Operator erwartet als Parameter eine Relation mit den Trainingsdaten sowie die Beschreibung der Attribut- und Klassenspalten. Die Parameter können interaktiv vom Benutzer über des Workbench-Frontend festgelegt werden. Anhand dieser Informationen wird vom Operator eine entsprechende Anfrage generiert und ausgeführt. Das Ergebnis wird wiederum in einer Tabelle abgelegt.

Die eigentliche Klassifikation erfolgt nun durch Vergleich der aktuellen Attributwerte des zu klassifizierenden Objektes mit den Werten aus der `counts`-Tabelle. Anhand der Häufigkeiten kann die Wahrscheinlichkeit für die Zuordnung zu den einzelnen Klassen bestimmt werden. Die Anfrage für das obige Beispiel für ein neues Objekt mit den Attributwerten $a_1, \ldots, a_3$ ist dann wie folgt:

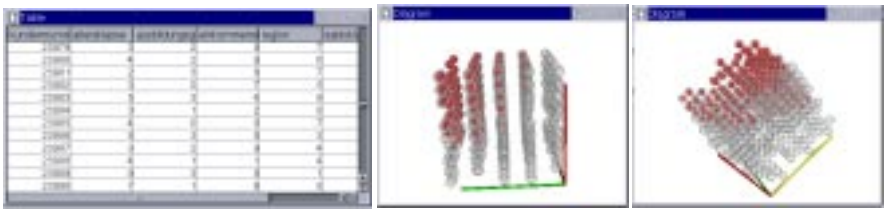
```
select c0.class, c1.cnt*c2.cnt*c3.cnt/power(c0.cnt,3) as prob
from counts c0, counts c1, counts c2, counts c3
where c0.class = c1.class and c1.class = c2.class
      and c2.class = c3.class
      and c1.alter_kl = a1 and c2.ausbildung = a2
      and c3.einkomm_kl = a3
      and c0.alter_kl is null and c0.ausbildung is null
      and c0.einkomm_kl is null
order by prob desc
```

Das bei einer Sortierung erste Ergebnistupel beinhaltet somit die Klassenzuordnung und die nicht-normalisierte Wahrscheinlichkeit. Diese Anfrage wird durch den *Apply-Classifier*-Operator ausgeführt. Auch dieser Operator generiert die eigentliche Anfrage anhand der aktuellen Parameter wie Eingangs- und Häufigkeitsrelation sowie Attributspalten.

Neben diesem einfachen Klassifikationsverfahren ist die Umsetzung weiterer Data-Mining-Operationen auf Basis von SQL geplant, so u.a. der Apriori-Algorithmus zur Aufdeckung von Assoziationsregeln und Clustering-Verfahren.

5.4 Visualisierung zur Benutzerunterstützung

Das Verständnis großer, meist multidimensionaler Datenbestände und die Extraktion von Informationen daraus setzt ihre umfassende Exploration voraus. Das Erkennen von Mustern und Trends, die Navigation im Datenbestand sowie das Auffinden von Beziehungen zwischen einzelnen Datensätzen erweist sich hierbei schnell als schwierig, oftmals als unmöglich [Eic00]. Es ist somit Aufgabe einer die Exploration unterstützenden Visualisierung, große Datenmengen handhabbar zu gestalten, die Betrachtung sowie Manipulation von Objekten in der Datenbasis zu ermöglichen und die Darstellung so aufzubereiten, dass Information durch den Benutzer leichter auffindbar wird. Diese Notwendigkeiten werden dadurch unterstrichen, dass die Interaktion gemäß den Abschnitten 5.1 und 5.2 einen wichtigen Aspekt im Prozess der Informationsfusion darstellt. Integrations- und Fusionsoperationen sind benutzergesteuert auszuführen beziehungsweise zu spezifizieren, Anfragen zu formulieren und die Ergebnisse auszuwerten, um gegebenenfalls den Fusionsprozess wieder zu revidieren.



(a) Tabellendarstellung

(b) Separierung einzelner Bonitätsklassen

(c) Einfluss selektiver Attribute auf die Bonität

Abbildung 3. Verschiedene Sichten eines exemplarischen Datenbestandes

Abbildung 3 stellt drei mögliche Sichten auf denselben Datenbestand dar. In Bild 3(a) erfolgt die Darstellung in Tabellenform während die Bilder 3(b) und 3(c) eine Filterung mittels Gaussian Splatting widerspiegeln. Beide Darstellungsformen ergänzen sich dabei. Eine Tabellenansicht für eine gegebene, beliebige Relation ist verhältnismäßig einfach zu erstellen. Ihre Spalten werden mit den Relationsattributen populiert und in den Zeilen respektive die zugehörigen Werte eingetragen. Ergebnisse sind auf diese Weise schnell dem Anwender anzuzeigen. Etwas aufwendiger gestaltet sich prinzipiell die Erzeugung der visuell reicheren Diagramme. Sie sind nicht immer vollständig automatisch aus einer gegebenen Relation abzuleiten. Einzelne Dimensionen der Eingabedaten sind untereinander variabel ins Verhältnis zu setzen. Sinnvolle Kombinationen in Hinsicht auf die Exploration und Interpretation durch den Anwender ergeben sich dabei oftmals erst aus dem Kontext und sind somit nur schwerlich zu generalisieren.

Die Motivation der hier dargestellten Diagramme ist ein besseres Verständnis der Beziehungen einiger Attribute von Kunden zu ihrer Bonität. Dargestellt wird der Ein-

fluss auf selbige von Seiten der Region (entlang der grün eingefärbten x-Achse), des Einkommens (entlang der rot gefärbten y-Achse) und des Ausbildungsgrades (entlang der gelb gefärbten z-Achse). Weitere Attribute fließen in die Betrachtung an dieser Stelle nicht mit ein, können und sollten allerdings Gegenstand weiterer Untersuchungen sein. Grau dargestellt ist der gesamte Bereich der Eingaberelation, welche die Trainingsdaten enthält. Rot und somit kontrastiv dunkler hervorgehoben ist eine Sicht auf den gleichen Eingabebereich skaliert entsprechend der Bonitätsklasse. Abbildung 3(c) offenbart den verhältnismäßig starken Einfluss der Region auf die Bonität, welcher sich darin ausdrückt, dass die rot (dunkler) dargestellten Bereiche sich mit größerem Abstand vom Koordinatenursprung maximieren. Dieser Effekt ist in abgeschwächter Form auch entlang der roten Achse zu beobachten, womit der stark gewichtete, wenn auch lineare Einfluss der Einkommensklasse auf die Bonität deutlich wird. Eine nahezu ausgeglichene Gleichverteilung ist für den Einfluss des Ausbildungsgrades abzulesen. Im Vergleich zu den anderen beiden Variablen wirkt er sich am schwächsten auf die Bonität eines Kunden aus. Abbildung 3(b) ist die Aufteilung der einzelnen Kundendatensätze auf die unterschiedlichen Bonitätsklassen zu entnehmen. Deutlich wird die starke Belegung der fünf am stärksten populierten Klassen. Die wenigen Ausreisser, die sich in den anderen Klassen wiederfinden, gehen in der Darstellung etwas unter.

Die Diagramme aus Abbildung 3(b) und 3(c) sind mit Methoden des Volumenrenderings erzeugt. Die Transformation von multidimensionalen Datensätzen mit verschiedenen Attributen für jeden einzelnen Datenwert in eine Volumenstruktur und deren anschließende Visualisierung ist eine verhältnismäßig einfache Möglichkeit, derart strukturierte Daten zur visuellen Exploration aufzubereiten. Abbildung 4 spiegelt das

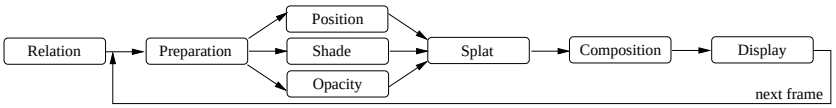


Abbildung 4. Visualisierungspipeline

der Volumendarstellung zu Grunde liegende Modell der Visualisierungspipeline wider. Die Relation als Ausgangspunkt repräsentiert hierbei generell in einer Datenbank gespeicherte Daten als Quelle für die weitere Verarbeitung. Diese werden für die Darstellung aufbereitet. In diesem Vorbereitungsschritt erfolgt die Zuordnung einzelner Attribute zu den Raumdimensionen, passende Farbgebungen sowie die Bestimmung der Durchlässigkeit darzustellender Voxel. Weitere Präsentationsvariablen können während dieser Phase einfach in die Darstellung eingebracht werden. In der mit Splat bezeichneten Phase wird den einzelnen Voxeln jeweils ein 3D-Gauss-Filter zugeordnet, der somit für die Abbildung der erzeugten Rauminformationen auf die Darstellungsfläche sorgt. Während der Komposition werden die unterschiedlichen Präsentationsinformationen zusammengefasst und respektive Grafikobjekte erzeugt, woraufhin das eigentliche Rendering und somit die Ausgabe auf dem Bildschirm angestoßen wird. Der Prozess wiederholt sich für den gesamten Eingabedatensatz.

Dieses Vorgehen basiert im Wesentlichen auf dem in [SML98] geschilderten Pipeline-Konzept von VTK. Dieses baut auf dem Prinzip der „lazy evaluation“ auf. Dabei wird ein Schritt der Pipeline nur dann ausgeführt, wenn von ihm generierte Daten zur weiteren Berechnung benötigt werden. Eine Modifikation der Eingaberelation bewirkt somit nur dann eine Neuberechnung der Szene, wenn eine Aktualisierung der Darstellung oder die Abfrage von Zwischenergebnissen der Pipeline erforderlich wird. Unnötige Berechnungen werden hierdurch effektiv vermieden.

5.5 Prototyp

Die Umsetzung der geschilderten Konzepte erfolgt auf Basis der in Abschnitt 5.1 geschilderten Architektur. Das System wird auf verschiedenen Unixplattformen implementiert. Die Verwendung einer standardisierten Datenbank-API , des Visualization Toolkits [SML98] 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.

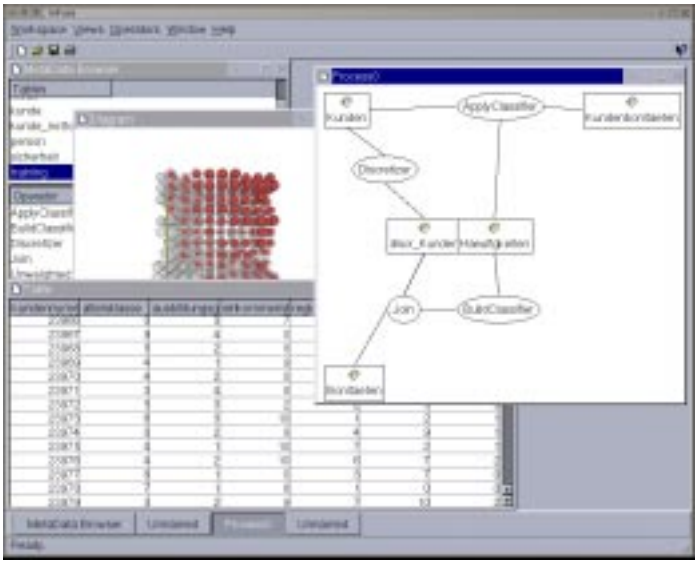


Abbildung 5. Prototyp mit Darstellung eines einfachen Prozessgraphen sowie verschiedenen Sichten auf einen exemplarischen Datensatz

Abbildung 5 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. Exem-

plarisch sind als zwei Sichten auf die Relation mit den Trainingsdaten eine Tabelle und ein einfaches Diagramm abgebildet (siehe dazu auch den vorherigen Abschnitt 5.4).

Weitere Entwicklungen sind in Arbeit. Zur Vervollständigung des Prototypen 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.

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 Anwendungsszenarien wird die praktische Relevanz der gebotenen Unterstützung evaluiert und erweitert. So wird der Satz von Präsentationsvariablen (Form, Farbe, Position, Transparenz) um Objektbewegungen [Bar98] angereichert. Außerdem werden nicht nur weitere Operatoren wie beispielsweise zusätzliche KDD-Verfahren implementiert, sondern auch die Basisdienste stetig 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

[AMS+96] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast Discovery of Association Rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 12, pages 307–328. AAAI Press / The MIT Press, Menlo Park, California, 1996.

[Bar98] Lyn Bartram. Enhancing Visualizations With Motion. In *Hot Topics: Information Visualization 1998*, North Carolina, USA, 1998.

[BLN86] 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.

[BM99] Jörg Baetge and Manolopoulous. Bilanz-ratings zur beurteilung der unternehmensbonität - entwicklung und einsatz des bbr baetge-bilanz-rating im rahmen des benchmarking. *Die Unternehmung*, (5):351–371, 1999.

[CD97] S. Chaudhuri and U. Dayal. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, 26(1), 1997.

[CDH+99] John Clear, Debbie Dunn, Brad Harvey, Michael L. Heytens, Peter Lohman, Abhay Mehta, Mark Melton, Lars Rohrberg, Ashok Savasere, and Robert M. Wehrmeister ans 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.

[CFB99] 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.

[Cha98] Surajit Chaudhuri. Data Mining and Database Systems: Where is the Intersection? *Data Engineering Bulletin*, 21(1):4–8, 1998.

[CMN99] 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.

[CSS99] 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.

[DWI00] Data Extraction, Transformation, and Loading Tools (ETL), http://www.dwinfocenter.org/clean.html, August 2000.

[Eic00] Stephen G. Eick. Visualizing Multi-Dimensional Data. *Computer Graphics*, pages 61–67, February 2000.

[Elk97] Charles Elkan. Boosting and Naive Bayesian Learning. Technical report, Dept. of Computer Science and Engineering, UCSD, September 1997.

[FPSSU96] 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.

[GLRS93] J. Grant, W. Litwin, N. Roussopoulos, and T. Sellis. Query Languages for Relational Multidatabases. *The VLDB Journal*, 2(2):153–171, April 1993.

[Han98] Jiawei Han. Towards On-Line Analytical Mining in Large Databases. *ACM SIGMOD Record*, (27):97–107, 1998.

[HK97] 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.

[HMN+99] 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.

[LSS96] L. V. S. Lakshmanan, F. Sadri, and I. N. Subramanian. SchemaSQL - A Language for Interoperability in Relational Multi-database Systems. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *Proc. of the 22nd Int. Conf. on Very Large Data Bases, VLDB'96, Bombay, India, September 3–6, 1996*, pages 239–250, San Francisco, CA, 1996. Morgan Kaufmann Publishers.

[MMC99] Klaus Mueller, Torsten Möller, and Roger Crawfis. Splatting Without The Blur. In *Proceedings of IEEE Conference on Visualization 1999*, pages 363–371, October 1999.

[OR86] 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.

[RH00] Vijayshankar Raman and Joseph M. Hellerstein. An Interactive Framework for Data Cleaning. http://control.cs.berkeley.edu/abc/, 2000. Working draft.

[SCS00] K.-U. 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.

[SML98] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit – An Object-Oriented Approach to 3D Graphics*. Prentice Hall PTR, 2. edition, 1998.

[STA98] 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.

[Vit87] J.S. Vitter. An Efficient Algorithm for Sequential Random Sampling. *ACM Transactions on Mathematical Software*, 13(1):58–67, March 1987.

[WZ99] Haixun Wang and Carlo Zaniolo. User-Defined Aggregates for Datamining. In *1999 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 1999.