Anfragetechniken für heterogene Datenbanksysteme

HABILITATIONSSCHRIFT

zur Erlangung der Venia legendi für

das Fach Informatik

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von: Dr.-Ing. Kai-Uwe Sattler
geb. am 26.09.1968 in Eisleben

Gutachter:
Prof. Dr. habil. Gunter Saake (Universität Magdeburg)
Prof. Dr. habil. Donald Kossmann (TU München)
Prof. Dr. habil. Gerhard Weikum (Universität des Saarlandes)

Magdeburg, 16. Mai 2003

Zusammenfassung

In vielen Anwendungsgebieten ist die Bereitstellung eines transparenten, integrierenden Zugriffs auf verteilte und möglicherweise auch autonome Datenquellen eine der wesentlichen Anforderungen an die Datenbanktechnologie. Diese seit vielen Jahren bestehende Aufgabe hat in der letzten Zeit durch die Verbreitung Web-basierter Informationssysteme eine noch größere Bedeutung erlangt. Eine wesentliche Herausforderung ist dabei die Überwindung von Heterogenitäten, die insbesondere aus der Autonomie der Quellen erwachsen. Im Kontext virtueller Integrationsansätze, wie sie mit Mediatoren, Multidatenbanksprachen und föderierten Datenbanksystemen verfolgt werden, nehmen somit Techniken der Anfrageverarbeitung ein zentrale Rolle ein. Sie bilden nicht nur die Basis für den eigentlichen Zugriff auf entfernte Datenquellen, sondern auch für die Definition integrierter Sichten und die Überwindung von Integrationskonflikten auf Schema- und Instanzebene.

Den Gegenstand dieser Arbeit bilden daher Anfragetechniken für heterogene Datenbanksysteme. Ausgehend von Anforderungen ausgewählter Anwendungsgebiete werden Anfrageoperationen im Kontext konkreter Anfragesysteme vorgestellt. Hierbei werden drei Bereiche betrachtet. Im ersten Teil werden Integrationsoperationen zur Behandlung von Konflikten vorgestellt, die durch die Heterogenität der Quellen auf Instanz-, Schema- und semantischer Ebene verursacht werden. Neben Operationen zur Restrukturierung von Schemata zählen hierzu auch Operationen, die Inkonsistenzen auf Datenebene behandeln können, die durch unterschiedliche Repräsentationen in den verschiedenen Quellen entstehen. Der zweite Bereich umfasst Operationen zur Unterstützung von Datenbereinigungs- und Datenanalyseaufgaben. Hierfür werden Primitive vorgestellt, die typische Teilaufgaben von Data Cleaning und – am Beispiel eines Klassifikationsverfahrens – von Data Mining lösen. Mit dem dritten Teil werden schließlich Operationen betrachtet, die durch die Einbeziehung von Metainformationen zur Semantik der Daten bestehende Heterogenitäten überwinden. Die Einsatzmöglichkeiten dieser Operationen werden am Beispiel von Anwendungen der Informationsfusion sowie der Föderierung digitaler Bibliotheken aufgezeigt.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Assistent in der Arbeitsgruppe Datenbanken am Institut für Technische und Betriebliche Informationssysteme der Otto-von-Guericke-Universität Magdeburg sowie zu einem kleineren Teil während eines Forschungsaufenthaltes an der University of California Davis. Ohne die Unterstützung von Kollegen an diesen beiden Einrichtungen wäre die Arbeit sicher nicht möglich gewesen.

Mein besonderer Dank gilt daher meinem Mentor Prof. Gunter Saake für die weitreichende Unterstützung in den vergangenen Jahren. Er hat mich nicht nur zu dieser Arbeit motiviert und in jeder erdenklichen Weise gefördert, sondern auch die Freiräume und das wissenschaftliche Umfeld geschaffen, die das Gelingen der Arbeit ermöglicht haben. Bedanken möchte ich mich auch bei Prof. Donald Kossmann und Prof. Gerhard Weikum für die Übernahme der Gutachten zu meiner Arbeit.

Ich danke weiterhin den Kolleginnen und Kollegen aus dem Institut für Technische und Betriebliche Informationssysteme für das angenehme Arbeitsumfeld, insbesondere Prof. Georg Paul, der mich speziell auf dem Weg zur Promotion gefördert hat, sowie Fred Kreuzmann, Steffen Thorhauer und Gerd Lange für die technische Unterstützung. Besonders erwähnen möchte ich auch Ingolf Geist, Hagen Höpfner und Eike Schallehn. Sie waren nicht nur Partner für viele konstruktive Diskussionen, sondern haben durch gemeinsame Papiere auch direkten Anteil an meiner Arbeit.

Mein Dank gilt ferner Prof. Stefan Conrad für viele hilfreiche Anregungen und Diskussionen sowie Michael Gertz für die angenehme und konstruktive Zusammenarbeit während meines Aufenthaltes an der UC Davis. Nicht vergessen möchte ich meine ehemaligen Kollegen Oliver Dunemann und Martin Endig, die ebenfalls in der Arbeitsgruppe Datenbanken mit mir zusammengearbeitet haben, sowie alle Studenten, die mich im Rahmen von Diplom- und Projektarbeiten unterstützt haben.

Schließlich danke ich noch dem dpunkt.verlag für die Erlaubnis, den Beitrag „Datenintegration & Mediatoren“ in diese Arbeit aufnehmen zu können.

Das größte Dankeschön gebührt jedoch Arved, Bennett und Britta für ihre Liebe, Unterstützung und Aufmunterung sowie ihr Verständnis für viele, viele Stunden am Schreibtisch, die leider zu häufig zulasten von Fußballspielen, Rollerfahren oder einfach nur Familienleben gegangen sind.

Kai-Uwe Sattler

Inhaltsverzeichnis

Gesamtdarstellung

1 Motivation und Abgrenzung

Die effiziente Verwaltung großer Datenbestände ist seit langem eine Kernaufgabe in vielen Anwendungsbereichen. Die Notwendigkeit, Daten dauerhaft zur Verfügung zu stellen, die Konsistenz auch bei gleichzeitigem Zugriff vieler Benutzer zu gewährleisten, vor missbräuchlichem Zugriff und Verlust durch Systemfehler zu schützen sowie effiziente Recherchemöglichkeiten anzubieten, besteht nicht nur in klassischen betriebswirtschaftlichen und administrativen Anwendungen, sondern auch in technischen und wissenschaftlichen Bereichen, etwa im Rahmen der Produktentwicklung oder bei der Sammlung und Auswertung von Messergebnissen.

Die Entwicklung des Internet und des World Wide Web haben die Bedeutung von Datenbanktechnologien in den letzten Jahren noch dramatisch verstärkt. So werden die Daten vieler Websites in Datenbanksystemen verwaltet, entweder indem die Web-Dokumente dynamisch mithilfe von Datenbankanfragen erzeugt werden oder durch die Generierung von Datenbankreports in Form von Web-Dokumenten.

Dies ist aber nur ein Aspekt des Einsatzes von Datenbanktechnologie. Die Vielzahl an verfügbaren Quellen im Web, deren scheinbar unkontrollierbares Veröffentlichen, Wachsen und Wiedereinstellen, die damit verbundenen Redundanzen, Inkonsistenzen und Qualitätsprobleme verstärken den Wunsch nach Diensten, die durch Kombination von Daten aus verschiedenen Quellen sowie deren Bereinigung und Aufbereitung einen „Mehrwert“ an Information anbieten. Eine solche Daten- oder Informationsintegration eröffnet eine Vielzahl neuer Anwendungen: beginnend bei einfachen „Metasuchmaschinen“ für Preisvergleiche oder Produktsuche bis hin zu entscheidungsunterstützenden Systemen, die Analyse- und Data-Mining-Techniken auf die integrierten Daten anwenden, oder Systeme, die Hintergrundwissen beispielsweise in Form von Ontologien zur Verbesserung der Anfrage- und Integrationstechniken nutzen. Web-Suchmaschinen wie Google oder Lycos sind nur scheinbar eine Alternative, da die hier verwendeten Information-Retrieval-Techniken im Wesentlichen auf eine Volltextsuche ausgerichtet sind und als Ergebnis Dokumente liefern, die die gesuchten Stichworte (Terme) enthalten. Strukturierte Anfragen nach einzelnen Elementen eines Objektes oder gar komplexere Operationen wie Verbund- oder Mengenoperationen, die weiterverarbeitbare strukturierte Objekte liefern, werden dagegen nicht oder nur sehr eingeschränkt unterstützt.

Eine wesentliche Aufgabe im Rahmen integrierter Informationssysteme ist somit die Beantwortung von strukturierten Anfragen in einer transparenten Weise, welche die Existenz mehrerer Quellen und somit die Herkunft der Daten weitgehend verbirgt. Anfragebearbeitung ist – ebenso wie die oben genannten Aufgaben der Datenverwaltung – eine Kernkomponente von Datenbanksystemen. Die Notwendigkeit ergibt sich aus dem deklarativen Charakter moderner Anfragesprachen: Ein Benutzer formuliert, „was“ gefunden werden soll und nicht, „wie“ dies erfolgt. Das Ableiten und Ausführen einer möglichst optimalen – in Bezug auf Kosten oder Antwortzeit – Folge von Operationen zum Auffinden der gesuchten Daten ist Aufgabe des Datenbanksystems und hier speziell des Anfrageoptimierers und der Ausführungskomponente.

Ein großes Problem bei der Integration ist, dass die Datensammlungen meist unabhängig voneinander entstanden sind und auch weitergepflegt werden. So besteht eine hohes Maß an Heterogenität der zu integrierenden Quellen auf unterschiedlichen Ebenen:

- auf Systemebene durch die Verwendung verschiedener Hardwareplattformen, Betriebssysteme, Datenbanksysteme, Protokolle und Anfrageschnittstellen,

- ❏ auf Datenmodellebene durch Nutzung unterschiedlicher Datenmodelle, wie z.B. relational, objektorientiert oder semistrukturiert,
- ❏ auf Schemaebene durch unterschiedliche Modellierung des Anwendungsbereichs, die sich schließlich in den Datenbankschemata widerspiegelt,
- ❏ sowie auf Instanzebene, indem verwandte Sachverhalte in verschiedener Weise in den Daten repräsentiert sind und auf diese Weise „Überlappungen“ zwischen den Extensionen bestehen.

Die Etablierung und Verbreitung von Standards wie SQL, ODBC, CORBA, XML, XQuery, Web Services oder auch von Referenzmodellen und Standardschemata etwa in Form von konkreten XML-Sprachen für spezifische Anwendungen vereinfacht zwar in vielen Fällen die Integration, kann aber letztlich nicht alle Unterschiede beseitigen. Die Autonomie der Quellen, die Weiterentwicklung im Interesse des technischen Fortschritts, mit dem Standardisierungsbestrebungen naturgemäß nicht Schritt halten können und insbesondere die unterschiedlichen Zielstellungen und Ausrichtungen der Datenanbieter stehen dem entgegen. Grundsätzlich ist eine Integration verschiedener Datenquellen auf zwei Wegen möglich:

- ❏ Beim Ansatz der *Materialisierung* werden die Daten periodisch aus den Quellen extrahiert und in einer zentralen Datenbank abgelegt. Anfragen können dadurch ausschließlich mithilfe der zentralen Datenbank beantwortet werden. Somit ist zum Anfragezeitpunkt kein Zugriff auf die Quellsysteme mehr notwendig. Das Ergebnis ist eine hohe Performanz und die Möglichkeit, eine umfassende Datenbereinigung zur Beseitigung von Inkonsistenzen und Fehlern in den Daten vornehmen zu können. Diese Vorteile werden jedoch mit Aktualitätsproblemen erkauft.
- ❏ Der *virtuelle Ansatz* basiert dagegen auf der Definition von Sichten über den Daten der Quellen. Anfragen über dem integrierten Bestand müssen daher in Anfragen an die Quellen transformiert, an die Quellen gesendet und dort ausgeführt werden. Die Vorteile dieses Verfahrens sind die hohe Aktualität der Daten und das Wegfallen einer redundanten Datenhaltung. Demgegenüber stehen die Nachteile einer aufwändigen Anfrageausführung und die Abhängigkeit von der Verfügbarkeit der Quellen.

Verfolgt man den Ansatz der virtuellen oder auch logischen Integration weiter und setzt ihn in Bezug zu Datenbanksystemen, so kommt man zum Begriff des „heterogenen Datenbanksystems“ (HDBS). Darunter soll in dieser Arbeit ein Softwaresystem verstanden werden, das

1. Daten aus mindestens zwei verschiedenen Datenbanken (sogenannte Quellen) kombiniert, d.h. in einer logisch integrierten Sicht kombiniert,
2. wobei diese Datenbanken wiederum heterogen bezüglich mindestens einer der oben genannten Ebenen (System, Datenmodell, Schema, Instanz) sind,
3. und das Datenbankoperationen (speziell Anfragen) durch Transformation und Delegierung an die Quellen realisiert.

Die beteiligten Datenbanken müssen nicht notwendigerweise vollständige Datenbanksysteme im Sinne der Codd’schen Regeln, d.h. mit kompletter Managementfunktionalität, sein, sondern können auch in Form von strukturierten Dateien, Websites oder Anwendungssystemen mit eingeschränkter Anfragefunktionalität vorliegen. Charakteristisch ist aber in jedem Fall, dass die Datenquellen meist durch ihre Autonomie verursachte Unterschiede auf verschiedenen Ebenen aufweisen, die durch eine Integration

zu überwinden sind, und dass Datenbankoperationen auf der globalen Ebene entsprechende Operationen auf den Quellen bewirken, d.h. eine virtuelle Integration verfolgt wird. Die Bandbreite möglicher Operationen reicht dabei von einfachen Anfragen, die an alle Quellen weitergeleitet und durch Kombination der Teilergebnisse beantwortet werden, über eine Zerlegung von Anfragen entsprechend der von den einzelnen Quellen exportierten Schemata bis hin zur Ausführung von Änderungsoperationen einschließlich der Realisierung einer Transaktionssemantik.

Gegenstand der vorliegenden Arbeit bilden Sprachkonzepte und Techniken der Anfrageverarbeitung in heterogenen Datenbanksystemen. Motiviert wird dies durch eine Reihe von Besonderheiten heterogener Datenbanksysteme, die eine einfache Übernahme von Anfragetechniken aus klassischen zentralen DBS sowie aus homogenen verteilten DBS erschweren. Zu diesen besonderen Eigenschaften gehören u.a.:

- Es ist eine meist größere Anzahl von Quellen zu integrieren, die weit verteilt vorliegen können und teilweise auch häufig wechseln.
- Die beteiligten Quellen sind durch Heterogenität auf System-, Schema- und Instanzebene gekennzeichnet.
- Die Quellen weisen eine hohe Autonomie hinsichtlich der Verfügbarkeit sowie der Erstellung und Änderung der Schemata auf.
- Bei den Quellen handelt es sich oft um keine voll funktionalen Datenbanksysteme oder die Datenbankfunktionalität ist bewusst hinter einer einschränkenden Schnittstelle verborgen, die nur bestimmte Anfragen zulässt. Beispiele hierfür sind Web-Schnittstellen über CGI-Programme oder vergleichbare Mechanismen und Quellen, die nur Funktionen exportieren, wie etwa über CORBA oder SOAP.
- Es liegen nur wenige Informationen über die Eigenschaften der Quellen vor, z.B. zum Inhalt, zu den operationalen Fähigkeiten oder zu den Kosten von Operationen.
- Die Qualität der Daten sowie die durch die Heterogenität der Quellen verursachten Integrationskonflikte machen eine Aufbereitung und Bereinigung der Daten erforderlich.

Aus diesen Merkmalen ergeben sich für die Anfrageverarbeitung konkrete Anforderungen, die im Interesse einer adäquaten Unterstützung der Anwendungen bzw. Nutzer sowie einer effizienten Ausführung spezielle Vorkehrungen und Techniken erfordern.

Im Weiteren werden eigene Arbeiten vorgestellt, die solche Techniken zum Gegenstand haben und folgende Aspekte schwerpunktmäßig behandeln:

- Anfrageoperationen zur Datenintegration
- Anfrageunterstützung für Datenbereinigung und -analyse
- Nutzung semantischer Metadaten zur Anfrageverarbeitung
- Anwendungen heterogener DBS

In den folgenden Abschnitten werden diese Aspekte weiter untersetzt und es wird der Bezug der Arbeiten zu diesen Schwerpunkten sowie untereinander dargestellt. Der Beitrag der Gesamtarbeit liegt somit in der Entwicklung von erweiterten Anfrageoperationen und -ausführungstechniken für die Integration,

Aufbereitung und Analyse heterogener Datenbestände. Diese Techniken werden am Beispiel konkreter Realisierungen von Anfragesprachen für heterogene DBS sowie in den Anwendungsbereichen der Informationsfusion und digitaler Bibliotheken betrachtet.

Ein Überblick zu Anfragesystemen für heterogene Datenbanken speziell im Kontext des World Wide Web wird in [SCS02, Anhang A.1, Seite 17] gegeben. Im Mittelpunkt stehen dabei Mediatorsysteme als eine Ausprägung heterogener Datenbanksysteme, die auf einem meist semistrukturierten Datenmodell basieren, das wiederum die Integration von Web-Quellen vereinfacht, und die den Problemen der starken Heterogenität und Autonomie durch die Beschränkung auf (lesende) Anfrageoperationen begegnen. In diesem Beitrag werden dazu ausgehend von einer Vorstellung der Basisarchitektur solcher Systeme die Grundprinzipien der Anfrageverarbeitung beschrieben und spezielle Anfragetechniken für die Integration von Web-Daten erläutert.

Die oben genannten Anforderungen an Anfragesysteme werden in [SS99, Anhang A.2, Seite 51] und [CSS99, Anhang A.3, Seite 61] für das Anwendungsszenario der Informationsfusion im Detail herausgearbeitet. Dabei wird unter Informationsfusion der 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 verstanden. Dieser – vom Charakter her interaktive und iterative – Prozess umfasst somit neben Kernaufgaben eines heterogenen Datenbanksystems wie Datenzugriff und Datenintegration auch Aspekte der Analyse, Aufbereitung und Verdichtung von Daten. Da auch diese Aspekte letztlich auf Datenbankanfragen und -manipulationen beruhen, setzt eine effiziente Ausführung dieser Aufgaben geeignete Funktionen eines heterogenen Datenbanksystems voraus, die über das klassische Spektrum an Funktionalität zur Anfrageverarbeitung hinausgehen. Ausgehend von den Herausforderungen des Gesamtprozesses werden die Anforderungen an die Datenbanktechnologie wie die Realisierung eines effizienten Datenzugriffs, die Integration externer Daten, die übergreifende, erweiterbare Optimierung auch komplexerer Fusionsoperationen oder die „On the fly“-Indexgenerierung, sowie Anforderungen an Knowledge-Discovery-Techniken wie u.a. die Effizienz und Skalierbarkeit der Verfahren formuliert. Basierend auf diesen Anforderungen wird die Eignung von Techniken föderierter (und somit heterogener) Datenbanksysteme diskutiert, die damit den Ausgangspunkt weiterer Arbeiten zur Unterstützung der Informationsfusion bilden.

2 Anfrageoperationen zur Datenintegration

Eine wichtige Aufgabe bei der Integration von Datenbeständen aus verschiedenen Quellen ist die Behandlung der durch die Heterogenität verursachten Konflikte. Beim Ansatz der virtuellen Integration muss dies durch geeignete *Anfrageoperationen* der verwendeten Anfragesprache realisiert werden. Ausgehend von einer an die Literatur angelehnten Klassifikation von Integrationskonflikten wird hierzu in [SCS00, Anhang A.4, Seite 67] ein erster Entwurf der Multidatenbanksprache FRAQL vorgestellt, die sich durch spezielle Sprachkonzepte zur Konfliktbehandlung auszeichnet. Konkret sind dies Konstrukte zur Definition von Import- und Integrationssichten, wobei Importsichten virtuelle Relationen von externen Tabellen (in Form von Datenbanktabellen oder auch strukturierten Dateien) darstellen, während Integrationssichten die Verknüpfung von Relationen erlauben. Die Importsichten unterstützen verschiedene Abbildungen von Attributen und Attributwerten wie Projektion, Umbenennung und Werttransformation unter Verwendung von benutzerdefinierten Funktionen als auch Abbildungstabellen. Als Integrationsoperationen, die beispielsweise im Rahmen der Definition von Integrationssichten angewendet werden können, stehen neben erweiterten Verbund- und Vereinigungsoperationen insbesondere Schemaoperationen zur Verfügung, welche die Verknüpfung von Daten und Metadaten (d.h.

Schemaelementen) erlauben. So lassen sich Relationen bzw. Attribute in einer Anfrage berechnen und auf diese Weise dynamisch Schemata festlegen.

Dieser Sprachentwurf, der in einem Multidatenbankanfragesystem auch implementiert wurde, wird in [SCS03, Anhang A.5, Seite 79] weiterentwickelt. Dabei erfolgt insbesondere die Definition der Semantik der Operationen als Erweiterung der Relationenalgebra. Konkret werden Operationen zur sogenannten Tabellen- bzw. Attributdereferenzierung sowie die Transposition vorgestellt. Diese Operationen ermöglichen Schemarestrukturierungen zur Überwindung struktureller Konflikte. Für die Auflösung von Instanzkonflikten, d.h. die unterschiedliche Repräsentation semantisch äquivalenter Objekte wird eine auf benutzerdefinierten Aggregatfunktionen basierende Technik diskutiert, wobei Standardfälle durch vordefinierte Funktionen abgedeckt werden können.

Weiterhin wird die beispielgetriebene Erkennung und Behandlung von Konflikten durch ein Softwarewerkzeug diskutiert. Dieses Werkzeug visualisiert potentielle Instanzkonflikte und gibt durch Anwendung von Heuristiken Vorschläge zu deren Auflösung. Ein beispielgetriebenes Vorgehen wird realisiert, indem dem Nutzer bei konfliktbehafteten Attributwerten eine Auswahlmöglichkeit angeboten und diese Auswahl vom System ausgewertet wird. Darauf aufbauend wird versucht, Auflösungsregeln in Form von vordefinierten Aggregatfunktionen abzuleiten, mit deren Hilfe schließlich eine konfliktfreie Repräsentation des Attributwertes gewonnen werden kann.

Weitere Arbeiten in diesem Kontext betreffen spezielle Operationen und deren Implementierung. Gegenstand des in [SDG+01, Anhang A.6, Seite 103] beschriebenen Ansatzes sind Operationen zur Reduzierung der Ergebnisgröße von Anfragen. Dies ist motiviert durch die Beobachtung, dass gerade bei der Exploration bzw. Analyse großer Datenbestände häufig nur Ausschnitte oder Stichproben untersucht werden. Hierfür bieten sich Operationen wie das „Abschneiden“ eines Anfrageergebnisses nach den ersten n Tupeln (`LIMIT FIRST`) oder zur Gewinnung einer repräsentativen Stichprobe (`LIMIT SAMPLE`) aus einer Relation bzw. einem Anfrageergebnis an, wobei die Ergebnisgröße absolut oder prozentual zur Größe der Ursprungsrelation vorgegeben werden kann. Im Rahmen eines heterogenen Datenbanksystems lassen sich diese „Beschränkungsoperationen“ zu den Quellsystemen propagieren, da insbesondere moderne DBMS aber auch Web-Quellen die Möglichkeit der Begrenzung der Ergebniskardinalität bieten. Notwendig ist hierfür eine Abschätzung der Kardinalität der Zwischenergebnisse unter Berücksichtigung der Selektivitäten der global auszuführenden Anfrageoperationen. In dieser Arbeit wird hierfür der Einsatz von Histogrammen auf globaler Ebene des Multidatenbanksystems untersucht, wodurch eine exaktere Abschätzung der Selektivitäten ermöglicht wird. Zur Ausführung von `LIMIT FIRST`-Anfragen werden jeweils zwei Beschränkungsoperationen in den Plan eingefügt: die erste Operation zur Begrenzung der Ergebnisgröße der Quellanfragen, die zweite Operation als Wurzel des Anfragegraphen mit einer Rückkopplung zur ersten Operation. Für `LIMIT SAMPLE`-Anfragen wird die Sample-Operation im Anfragebaum nach unten verschoben und die Stichprobengröße entsprechend der Selektivität der nachfolgenden Operationen angepasst. Diese Konstellation sichert einerseits die Einhaltung der Anfragevorgaben bezüglich der benötigten Tupel und reduziert andererseits die von der Quelle angeforderten Daten. In Experimenten konnte auch gezeigt werden, dass bereits mit relativ einfachen Histogrammen mit wenigen Buckets genügend exakte Abschätzungen möglich sind, die ein Restart von Quellanfragen weitgehend vermeiden.

In den in [SSS01, Anhang A.7, Seite 119] und [SSS02, Anhang A.8, Seite 135] dokumentierten Arbeiten wird das Problem der unterschiedlichen Repräsentation ein und der selben Real-Welt-Objekte in heterogenen Quellen adressiert. Dabei wird ausgehend von der Beobachtung, dass in realen Szenarien Tests auf Gleichheit von Attributwerten zur Entscheidung über die Äquivalenz von Objekten nur bedingt geeignet sind, eine ähnlichkeitsbasierte Gruppierungsoperation eingeführt. Mit dieser Operation werden Tupel gruppiert, die bezüglich eines nutzerdefinierten Ähnlichkeitsmaßes – das als Gruppierungsfunktion spezifiziert werden kann – ähnlich sind, d.h. einen gegebenen Schwellwert überschrei-

ten. Als Ähnlichkeitsmaß wird dabei insbesondere die Editierdistanz genutzt, die für häufig als identifizierende Attribute genutzte Zeichenketten kürzerer bis mittlerer Länge am besten geeignet ist. Weiterhin werden verschiedene Strategien zur Gruppierung diskutiert: die Betrachtung der transitiven Hülle als Gruppierungskriterium sowie eine strikte Ähnlichkeit, wonach alle Tupel einer Gruppe paarweise ähnlich sein müssen.

Neben der formalen Definition der Semantik der Gruppierungsoperation werden Optimierungs- und Implementierungsmöglichkeiten beschrieben. Hierbei wird sowohl die Einbindung nutzerdefinierter Gruppierungsfunktionen in FRAQL als auch die Implementierung auf Basis eines kommerziellen DBMS vorgestellt. Die effiziente Ausführung der Ähnlichkeitsoperationen wird durch einen Trie-basierten Index unterstützt, der „on the fly“ aufgebaut wird.

Da die Bildung von Gruppen ähnlicher Tupel nur der erste Schritt bei der Überwindung von Instanzkonflikten ist – konkret der Schritt der Entity-Identifikation, wird die Gruppierung im Weiteren mit den oben bereits erwähnten nutzerdefinierten Aggregatfunktionen zum Vereinheitlichen (*Reconciliation*) der Tupel jeder Gruppe kombiniert. Zusammen bieten diese beiden Konzepte somit die Möglichkeit, im Rahmen einer Anfrage potentielle „Duplikate“ in den Daten aufzudecken und die eventuell vorhandenen Inkonsistenzen durch Anwendung von Aggregatfunktionen zu beseitigen. Insbesondere das FRAQL-System unterstützt die dafür notwendigen Erweiterungsschnittstellen zur Definition eigener Ähnlichkeitsfunktionen (basierend auf vordefinierten Primitiven wie der Editierdistanz) und Auflösungsfunktionen in Form von Aggregaten.

In [SH99, Anhang A.9, Seite 153] wird schließlich das Problem der Integration von Nicht-DBS-Quellen untersucht, d.h. der Zugriff auf Websites mit einer (eingeschränkten) Anfragefunktionalität, die durch Aufrufe von Gateways wie CGI-Programmen oder Servlets realisiert ist. So wird ein Toolkit zur Entwicklung von Adaptern beschrieben, das als Schnittstelle eine SQL- bzw. JDBC-Schnittstelle unterstützt, Anfragen in entsprechende HTTP-Requests umsetzt und aus den Ergebnisdokumenten die relevanten Daten extrahiert. Die möglicherweise beschränkten Anfragefähigkeiten der Quelle können dabei vom Adapter durch zusätzliche Filteroperationen kompensiert werden. Das Toolkit basiert auf einem operationalen Extraktionsansatz, wobei eine Skriptsprache zusammen mit Primitiven eingesetzt wird. Das zur Implementierung eines konkreten Adapters notwendige Skript wird dabei semiautomatisch anhand vorgegebener Beispielseiten, in denen die zu extrahierenden Datenbereiche markiert sind, generiert. Speziell für datenbank-gestützte Websites, die meist Seiten mit einer festen Struktur aufweisen, lässt sich auf diese Weise die Implementierung von Adaptern vereinfachen.

3 Anfrageunterstützung zur Datenbereinigung und -analyse

Einige Anwendungen etwa aus dem Bereich Informationsfusion sind dadurch charakterisiert, dass sie umfangreiche Mechanismen zur Erhöhung der Datenqualität oder zur Unterstützung von Analyse- und Data-Mining-Aufgaben erfordern. Beispiele hierfür sind Anwendungen aus der Bioinformatik, die komplexe Verknüpfungen und Analysen über den verschiedenen Daten der einzelnen Quellen durchführen, aber auch typische Data-Mining-Anwendungen aus dem betriebswirtschaftlichen Bereich, die Muster und Zusammenhänge über mehrere Quellen hinweg aufdecken sollen.

Das Problem der Datenaufbereitung bzw. -bereinigung als notwendige Vorstufe der Datenanalyse wird in [SS01, Anhang A.10, Seite 159] untersucht. Typischerweise umfasst dieser Prozess die Schritte der Datenselektion, Integration, Transformation, Datenbereinigung (Data Cleaning) sowie Reduktion, wobei diese meist auf materialisierten Daten durchgeführt werden. Dies ist allerdings mit einem erhöhten Speicherplatzbedarf verbunden und kann zu Aktualitätsproblemen führen. In dem Bei-

trag wird daher ein Framework von Aufbereitungsoperationen vorgestellt, das als Teil von FRAQL implementiert wurde. Konkret werden Operationen zur Datennormalisierung, Duplikateliminierung, zur Unterdrückung von Rauschen in Daten, zur Erkennung von Ausreißern sowie zur Datenreduktion beschrieben. Ziel ist dabei, diese Operationen im Rahmen von Anfragen und damit auch in Sichtdefinitionen einsetzen zu können, so dass eine explorative, iterative Arbeit mit den Daten ohne notwendige explizite Materialisierung möglich ist. Es konnte gezeigt werden, dass wesentliche Operationen realisierbar sind. Dabei hat es sich als sinnvoll erwiesen, nicht komplette Aufgabenstellungen durch dedizierte Operationen zu unterstützen, sondern Basisoperationen oder Primitive bereitzustellen, mit deren Hilfe sich komplexere Operationen realisieren lassen. Als Implementierungstechnik für solche Primitive werden u.a. nutzerdefinierte Gruppierungs- und Aggregatfunktionen eingesetzt. Mit letzteren – und speziell mit einer besonderen Form davon, die auch Zwischenergebnisse zurückgeben kann (*Early-Return*-Funktionen) – lassen sich insbesondere Funktionen zur Konstruktion von Histogrammen formulieren, die wiederum für Diskretisierung und Rauschunterdrückung benötigt werden.

Der Ansatz der Anfrageprimitive wird in [SD01, Anhang A.11, Seite 169] in Richtung Data-Mining-Unterstützung weiterentwickelt. Am Beispiel von entscheidungsbaumbasierten Klassifikationsverfahren werden Datenbankprimitive entwickelt, die ein effizientes Mining unterstützen. Die Zielstellung ist hierbei nicht primär die Möglichkeit der Formulierung von deskriptiven „Data-Mining-Anfragen“ sondern die Ausnutzung von inhärenten Datenbankfunktionalitäten wie Optimierung, Indexunterstützung, Parallelisierung oder die effiziente Verwaltung von Daten im Gigabyte-Bereich. Auf diese Weise wird eine Verbesserung der Skalierbarkeit von Data-Mining-Verfahren angestrebt, da die aktuell vorherrschenden hauptspeicherbasierten Verfahren entweder eine Beschränkung der zu analysierenden Daten erfordern oder mit einer dramatische Verschlechterung des Laufzeitsverhaltens beim Überschreiten der Hauptspeichergrenzen verbunden sind.

Ausgehend von der Analyse typischer Klassifikationsverfahren aus der Literatur und der Identifizierung der berechnungsintensiven Teile werden drei Primitive zum Aufbau sowie zur Nutzung von Entscheidungsbäumen vorgestellt. Die Phase der Konstruktion von Entscheidungsbäumen aus einer Trainingsmenge von Daten wird dabei durch eine Filteroperation sowie eine Statistikoperation unterstützt. Die Filteroperation selektiert eine Partition von Datensätzen anhand einer konjunktiven Bedingung, die einen Pfad im Entscheidungsbaum repräsentiert. Die effiziente Berechnung solcher Partial-Match-Anfragen wird durch Einsatz eines multidimensionalen Hashverfahrens ermöglicht. Im Vergleich zu anderen Zugriffsverfahren wie Full Scan und Bitmap-Indexen erweist sich diese Realisierung speziell bei einer größeren Anzahl von Attributvergleichen effizienter, d.h. bei Knoten, die sich tiefer im Baum befinden, was auch entsprechend häufiger auftritt. Die Statistikoperation dient zur Auswahl eines Splitpunktes zu einem gegebenen Knoten im Baum, d.h. zur Bestimmung des Attributes, das an diesem Knoten den verbleibenden Teil der Daten am besten (im Sinne der Informationsentropie) in einzelne Klassen partitioniert. Hierzu werden zu einer zuvor bestimmten Partition die Ausprägungen der verbleibenden Attribute sowie die Klassenzuordnungen gezählt und darauf aufbauend wird mit Hilfe der Entropie das entsprechende Attribut ausgewählt. Da hierbei verschiedene Gruppierungskriterien kombiniert werden müssen, erfordert diese Operation mehrere Full Scans. In der beschriebenen Realisierung der Statistikoperation wird dagegen nur ein einziger Scan durchgeführt, indem die Häufigkeiten des Auftretens von Attributwerten zusammen mit Klassenzuordnungen in einem Feld gesammelt werden.

Als dritte Primitive wird schließlich der sogenannte *Prediction Join* beschrieben, mit dessen Hilfe neue Daten mit einem Entscheidungsbaum verknüpft und entsprechend die Klassenzuordnungen „vorausgesagt“ werden. Hierzu wird eine relationale Repräsentation eines Entscheidungsbaums vorgestellt, auf deren Basis der Join als eine Kombination von Kantenselektion und Baumtraversierung implementiert werden kann.

Zu den einzelnen Primitiven werden Implementierungen in Form von nutzerdefinierten Tabellen-Funktionen diskutiert und am Beispiel des Oracle-DBMS evaluiert. Die Experimente zeigen eine deutliche Performance-Verbesserung gegenüber reinen SQL-Implementierungen. Eine Implementierung als Tabellen-Funktion ermöglicht dabei den Einsatz im Rahmen von SQL-Anfragen, so dass Data-Mining-Tasks wie die Konstruktion eines Entscheidungsbaums als eine Folge von Anfragen realisiert werden können. Die Bereitstellung von Primitiven im Gegensatz zu einer dedizierten Data-Mining-Anfragesprache hat neben der prinzipiellen Portierbarkeit auf andere DBMS insbesondere den Vorteil, dass dadurch eine größere Klasse von Verfahren unterstützt werden kann. So wird in der Arbeit im Wesentlichen das ID3-Klassifikationsverfahren betrachtet, die Primitive sind aber auch in anderen Verfahren einsetzbar.

4 Nutzung semantischer Metadaten zur Anfrageverarbeitung

Ein weiterer Aspekt der Integration heterogener Daten und der darauf aufbauenden Anfrageverarbeitung ist der Zugriff auf Daten, die entweder keine expliziten lokalen oder globalen Schemastrukturen aufweisen oder für die sich aufgrund unterschiedlicher Diskursbereiche der Quellen nur schwer Korrespondenzen formulieren lassen. Ein Lösungsansatz für diese Problemklasse ist die Nutzung semantischer Metadaten zur Integration und Anfrageverarbeitung, d.h. von Informationen über die Bedeutung der Daten. Auf diese Weise wird das Anwendungs- oder Hintergrundwissen zu der betrachteten Domäne explizit modelliert.

Ein solcher Ansatz wird in [GSG+02, Anhang A.12, Seite 177] in Verbindung mit *Annotationen* zu heterogenen Daten entwickelt. Die Daten sind in dem untersuchten Anwendungsbereich der Neurowissenschaften im Wesentlichen Rasterbilder, wobei aber prinzipiell auch Texte oder weiter strukturierte Datensätze möglich sind. Diese Daten bzw. Bilder, die aus unterschiedlichen Quellen stammen und dabei auch verteilt vorliegen können, werden im Rahmen von Analysen und Experimenten von Wissenschaftlern gesichtet und annotiert, d.h. interessante Regionen in den Bildern werden markiert und mit zusätzlichen Informationen versehen. Die dadurch erzeugten Annotationen sind wiederum Instanzen sogenannter Konzepte. Ein Konzeptschema aus mehreren Konzepten, deren Eigenschaften und Beziehungen (wie Spezialisierung, aber auch räumliche und funktionale Beziehungen) repräsentiert somit das Hintergrundwissen in Form eines Vokabulars oder einer Taxonomie. Dabei muss das Konzeptschema nicht notwendigerweise vorab modelliert werden, sondern kann sukzessive während der Annotation neuer Daten erweitert werden. Dem Gesamtmodell – dem sogenannte *Annotation Graph Model* – liegt ein einfaches graphbasiertes Datenmodell zugrunde, das die einheitliche Repräsentation sowohl von Konzepten, Annotationen als auch Dokumenten bzw. Bildern sowie deren Beziehungen ermöglicht.

Die Recherche in den auf diese Weise integrierten Daten erfolgt mit Hilfe von Anfragen über der Metaebene, d.h. über Konzepte und Annotationen. Hierzu werden in der Arbeit geeignete Anfrageoperationen über Annotationsgraphen vorgestellt, wie die Selektion von Konzepten und Annotationen anhand von bestimmten Eigenschaften, die Traversierung von Beziehungen zwischen Konzepten aber auch zu Annotationen und Basisdaten, sowie die Bestimmung der transitiven Hülle bezüglich solcher Beziehungen. Ausgehend von der zunächst algebraischen Notation dieser Operationen wird eine einfache XPath-ähnliche Anfragesprache beschrieben. Die Implementierung des Anfragesystems für diese Sprache bildet den Kern eines Annotationsservers, der im Rahmen eines Teilprojektes des Human Brain Projects zur Annotation und Suche von Rasterbildern mit Hirnschnitten zum Einsatz kommt.

In einem zweiten Beitrag zu dieser Thematik [GS02, Anhang A.13, Seite 187] wird der Aspekt der Anfrageverarbeitung genauer behandelt. Da die Anfragen bei diesem Ansatz nur über den Metadaten und somit ausschließlich auf globaler Ebene ausgeführt werden, wird eine relationale Abbildung von

Annotaionsgraphen diskutiert. Konzeptdefinitionen werden ebenso wie Annotationen, Dokumentinformationen und Beziehungen in SQL-Tabellen gespeichert. Zur Ausführung von Anfragen über den Graphen müssen diese nach SQL transformiert werden. Die hierfür notwendigen Transformationsregeln werden beschrieben.

In [SGHS03, Anhang A.14, Seite 201] wird die Idee der Nutzung semantischer Metadaten zum Zugriff auf heterogene Daten im Kontext von Mediatorsytemen angewendet. Hierzu wird als Integrationsmodell ebenfalls ein Graphenmodell zur Definition von Konzepten und Beziehungen verwendet, das jedoch auf dem Klassenmodell von RDF Schema basiert. Das eigentliche Datenmodell zur Repräsentation der Basisdaten der Quellen aus Sicht des Mediators ist ein semistrukturiertes Modell in Form von XML. Die Zuordnung zwischen globalen Konzepten und den Basisdaten aus den Quellsystemen erfolgt dabei über Abbildungsvorschriften die angeben, wie eine Quelle ein gegebenes Konzept durch lokale Daten unterstützt. Da die Definition der Abbildung für jede Quelle getrennt und unabhängig von anderen erfolgt, wird somit ein Local-as-View-Ansatz verfolgt.

Als Anfragesprache kommt hier eine XQuery-Variante zum Einsatz, die neben den bereits oben erwähnten Operationen wie Konzeptselektion, Traversierung von Beziehungen und Bestimmung der transitiven Hülle auch Mengenoperationen und insbesondere Schemaoperationen anbietet, die den Zugriff und die „Berechnung“ von Schemaelementen (hier: von Konzepten und Eigenschaften) erlauben. Anfragen werden formuliert, indem zunächst eine Menge von Konzepten bestimmt wird, zu denen die Extension in Form der Vereinigung aller zugehörigen Basisdaten aus den relevanten Quellen berechnet wird. Für diese Extensionen können wiederum Selektionsbedingungen angegeben werden. Die Bestimmung der Extension ist demzufolge mit einer Menge von Quellenanfragen verbunden, wobei sowohl zur Identifizierung der relevanten Quellen als auch zur Transformation der Anfragen die Abbildungsvorschriften genutzt werden. Die Ergebnisse der Quellenanfragen werden mit Hilfe einer äußeren Vereinigung zusammengeführt. Entsprechend wird diese Operation auch als *extensionaler Union* bezeichnet. Die Semantik der Anfragesprache wird mithilfe von Algebraoperationen definiert. Weiterhin werden Übersetzungsregeln sowie Algorithmen zur Anfrageauswertung und zur Einbeziehung eines semantischen Caches für XML-Daten dargestellt. Das komplette Anfragesystem ist im Rahmen eines Mediators für kulturhistorische Datenbanken implementiert.

5 Anwendungen

Die beschriebenen Anfragetechniken für heterogene Datenbanksysteme wurden – neben den oben beschriebenen konkreten Anwendungen – für den Einsatz in zwei größeren Szenarien entwickelt und dort auch evaluiert. Konkret handelt es sich dabei um das vom BMBF geförderte Projekt „Föderierungsdienste für heterogene Dokumentenquellen“ sowie das Projekt der DFG-Forschergruppe „Workbench für die Informationsfusion“.

Digitale Bibliotheken. Speziell für den Bereich wissenschaftlicher Veröffentlichungen sind im Internet eine ganze Reihe digitaler Bibliotheken verfügbar. Diese Angebote, die sowohl Volltextinhalte als auch Metadaten umfassen, werden von Verlagen, Organisationen wie ACM oder IEEE Computer Society als auch von speziellen Forschungs-Communities (wie z.B. DBLP an der Universität Trier) und einzelnen Universitäten bereitgestellt. Die meisten digitalen Bibliotheken erlauben einfache Suchanfragen, etwa nach Titel oder Autor, teilweise auch eine Volltextsuche im Abstrakt oder dem gesamten Beitrag. Die Notwendigkeit einer Integration dieser autonomen Quellen ergibt sich aus dem Bedarf nach übergreifenden Recherchemöglichkeiten: Sollen etwa Beiträge verlags- oder fachgebietsübergreifend gesucht werden, so ist dies ohne integrierten Zugriff mit einer aufwändigen Einzelrecherche bei allen

relevanten Verlagen und Diensten verbunden. Eine weitere sinnvolle Anwendung ist z.B. das Citation Linking, d.h. das Verfolgen von Zitierungen (Welche Beiträge wurden am häufigsten zitiert? Gibt es Cluster von Zitierungen?).

Integrationsszenarien aus dem Bereich digitaler Bibliotheken sind meist durch einfache, aber heterogene Schemata charakterisiert. Zwar existieren eine Reihe von Standards wie Z39.50, MARC oder OAI, aber diese lassen bezüglich der Gestaltung der Schemata durchaus Freiräume offen. Weiterhin können zwischen den einzelnen Datenbeständen Überlappungen existieren. Inkompatibilitäten bzw. Heterogenitäten auf Instanzebene erschweren jedoch die Erkennung und Behandlung von potentiellen Duplikaten.

Ein Föderierungsdienst für digitale Bibliotheken auf Basis von FRAQL wird in [EHS+00, Anhang A.15, Seite 221] sowie [SES00b, Anhang A.16, Seite 229] vorgestellt. Hierzu werden zunächst verschiedene Formen von Adaptern für den Zugriff auf externe Quellen diskutiert. Adapter für kooperative Provider zeichnen sich durch die Ausnutzung der Quellenfähigkeiten und einen damit verbundenen effizienten Datentransfer sowie einen geringen Implementierungsaufwand auf Provider-Seite aus. Als Realisierungsvariante wird dabei eine Klasse von XML-basierten Adaptern beschrieben, die XML-Daten von der Quelle erwarten und durch ein XSLT-Stylesheet zur Transformation in das globale Schema parametrisiert werden können. Adapter für nicht-kooperative Provider werden dagegen durch Wrapper – wie etwa in der in [SH99] beschriebenen Form – implementiert. Die Quellenfähigkeiten werden auf globaler Ebene als sogenannte *Query Constraints* zu den Import-Sichten von FRAQL spezifiziert. Hierbei kann angegeben werden, welche Prädikate über welchen Attributen und in welchen Kombinationen in Quellenanfragen auftreten dürfen. Bei der Umformung und Zerlegung der Anfrage werden diese Constraints ausgewertet, um eine vom Quellsystem auswertbare Teilanfrage abzuleiten.

In [SES00a, Anhang A.17, Seite 239] wird das im Kontext der Integration digitaler Bibliotheken häufig auftretende Problem des *Citation Linking* untersucht. Hierbei geht es darum, über eine Quelle hinaus bibliographische Referenzen verfolgen zu können. Da die existierenden Quellsysteme meist ein eigenes Schlüsselsystem verwenden, muss eine Abbildung in ein globales Identifizierungsschema erfolgen. Im Beitrag wird eine solche Realisierung unter Verwendung von Abbildungstabellen als Teil der Import-Sichten von FRAQL beschrieben. Weiterhin wird gezeigt, wie diese Tabellen auch zur Auswahl der anzufragenden Quellen ausgewertet werden können.

Informationsfusion. Die Integration von Daten aus verschiedenen Quellen erlaubt nicht nur die Ausführung von Anfragen sondern ermöglicht auch eine weitergehende Analyse der Daten bis hin zum Data Mining – dem Aufdecken von Mustern und Zusammenhängen in den Daten. Dieser als Informationsfusion bezeichnete Ansatz basiert auf der Beobachtung, dass in Szenarien mit vielen potentiellen Datenquellen zu einer gegebenen Analyseaufgabe häufig nur Ausschnitte aus dem Gesamtdatenbestand relevant sind und sich daher eine vorab durchgeführte Materialisierung nicht lohnt bzw. gar nicht möglich ist.

Aufgrund der üblicherweise auftretenden Heterogenitäten sind wiederum Integrationsoperationen notwendig. Darüber hinaus muss auch die Qualität der zu analysierenden Daten gewährleistet werden, indem etwa Inkonsistenzen, Rauschen oder Ausreißer entfernt werden. Hierfür werden entsprechende Datenbereinigungsoperationen benötigt. Schließlich sind noch die (oft komplexen) Analyse- bzw. Data-Mining-Anfragen zu unterstützen, um auch bei einer virtuellen Integration eine effiziente Ausführung zu gewährleisten. Dies kann u.a. dadurch realisiert werden, dass Analyseschritte zunächst nur auf einem Ausschnitt der Daten (Stichproben) durchgeführt werden, um die Relevanz und Qualität der Daten zu prüfen sowie die Verfahren in geeigneter Weise zu parametrisieren. Für spätere vollständige Untersuchungen können dann die integrierten und bereinigten Daten materialisiert werden.

Als Basis für eine Software-Umgebung zur Informationsfusion bietet sich ein heterogenes Datenbanksystem an. Die Herausforderungen liegen hierbei in der Bereitstellung und effizienten Realisierung entsprechender Integrations-, Aufbereitungs- und Analyseoperationen. Hierzu wird in [DGJ+01, Anhang A.18, Seite 247] sowie in einer erweiterten und überarbeiteten Darstellung in [DGJ+02a, Anhang A.19, Seite 265] eine Software-Umgebung in Form einer Workbench beschrieben. Dieses Client/Server-System besteht aus einer „Fusion Engine“, die Fusionsdienste bereitstellt, sowie aus einer graphischen Benutzerschnittstelle. Die Fusion Engine basiert im Kern auf dem FRAQL-Anfragesystem, dass Dienste zum Zugriff auf externe Datenquellen, zur Integration sowie zur Ausführung globaler Anfragen bereitstellt. Weiterhin gehören zur Fusion Engine eine Komponente zur Verwaltung von Worksheets sowie eine Komponente zur Verwaltung von Operationen. Worksheets sind hierbei Beschreibungen von Fusionsabläufen in Form von Datenflussgraphen aus Datenquellen und Operatoren als Knoten sowie den Datenflüssen als Kanten. Operationen sind neben den Anfrage- und Integrationsoperationen insbesondere Analyse- und Aufbereitungsfunktionen wie Implementierungen von Klassifikationsverfahren, von Prediction Joins, zur Datentransformation und -normalisierung oder auch zur Konstruktion von Histogrammen. Diese Operationen sind als dynamisch nachladbare Module implementiert, die über FRAQL Zugriff auf den integrierten Datenbestand haben.

Der Client in dieser Umgebung ist eine graphische Benutzerschnittstelle zum Aufbau von Worksheets sowie zur Ausführung der damit repräsentierten Prozesse. Ein wesentliches Merkmal ist dabei die Unterstützung einer interaktiven und inkrementellen Arbeitsweise: Der Nutzer kann Datenquellen auswählen, darauf Operationen anwenden und sich die Ergebnisse in unterschiedlichen Sichten visualisieren lassen. Auf diesen Ergebnissen können wiederum neue Operationen angewendet werden bis schließlich das gewünschte Analyse- oder Fusionsergebnis erzielt wird. Die Abfolge der einzelnen Teilschritte ergibt einen Fusionsprozess der als Worksheet abgelegt und später (etwa für andere Daten) wieder aufgerufen sowie parameterisiert werden kann. Bei den Operationen wird zwischen materialisierenden und Streaming-Operationen unterschieden, wobei jedoch die Handhabung der Zwischenergebnisse für den Nutzer transparent ist. Änderungen einzelner Parameter von Operationen können so eine automatische Neuberechnung der abhängigen Teilschritte initiieren. Als hilfreich erweisen sich hierbei insbesondere die FRAQL-Basisoperationen zur Limitierung von Anfrageergebnissen wie etwa die Konstruktion von Stichproben. So kann zunächst auf einem kleinen repräsentativen Datenausschnitt der Gesamtablauf explorativ definiert und parameterisiert werden, bevor ein länger laufender Prozess zur Analyse des Gesamtdatenbestandes gestartet wird. In [DGJ+02a] wird dies anhand eines Beispiels aus dem Bank-Bereich dargestellt; in [DGJ+02b, Anhang A.20, Seite 277] dient eine Problemstellung aus dem Bereich Comparative Genomics als Anwendungsbeispiel.

6 Fazit und Ausblick

Gegenstand der vorliegenden Arbeit bilden Fragestellungen der virtuellen Integration von Daten aus heterogenen Quellen sowie deren Aufbereitung, Nutzbarmachung und Analyse für konkrete Anwendungen. Im Mittelpunkt stand dabei eine operationale Sichtweise, d.h. Anfragetechniken im Gegensatz zu Fragen des Entwurfs integrierter Datenbanken etwa mit Methoden zur Schemaintegration. Dieser Fokus wurde gewählt, da einerseits Anfragesprachen und -verarbeitung ein Kernelement virtueller Integrationsansätze bilden und andererseits der Umgang mit heterogenen, verteilten Datenbeständen besondere Anforderungen an Anfragetechniken stellt, die durch gegenwärtig verfügbare Lösungen nur bedingt erfüllt werden.

Der Beitrag der vorgestellten Arbeiten liegt in der Entwicklung von neuen Anfrageoperationen für heterogene Datenbanksysteme, welche

- ❏ die Schemaebene durch Operationen über Schemaelemente speziell zur Restrukturierung,
- ❏ die Instanzebene mit Operationen zur Datentransformation und -bereinigung sowie
- ❏ die semantische Ebene mit Operationen über Konzepten und deren Beziehungen

berücksichtigen. Die Realisierung erfolgte dabei sowohl durch dedizierte Operationen als auch in Form von Primitiven, die als Bausteine für komplexere Operationen angesehen werden können. Die hierfür notwendigen Erweiterungsmechanismen für Anfragesysteme wie nutzerdefinierte Aggregat- und Gruppierungsfunktionen wurden ebenfalls vorgestellt. Als Ergebnis entstanden mit FRAQL, dem Annotationssystem sowie dem konzeptbasierten Mediator drei Anfragesysteme für verschiedene Anwendungsfälle. Somit versteht sich die Arbeit nicht als Vorschlag zu einer komplexen, alle Features umfassenden Anfragesprache sondern als Framework von Operationen, die unterschiedlichen Anforderungen gerecht werden können.

Aus den dargestellten Ergebnissen lassen sich einige Anschlussfragestellungen ableiten, die gleichzeitig auch den Forschungsbedarf in dem Gebiet charakterisieren. Ein wesentlicher Bereich, der in der vorliegenden Arbeit nur am Rande betrachtet wird, ist die Optimierung von Anfragen in heterogenen Datenbanksystemen. Die Besonderheiten dieses Szenarios wie etwa Probleme bei der Vorhersagbarkeit von Kosten durch fehlende Information über die Quelldaten und -systeme sowie von Transferzeiten und -raten können durch klassische Optimierungstechniken nur unzureichend berücksichtigt werden. Stattdessen werden Techniken benötigt, die eine Anpassung der Anfragepläne an veränderte Ausführungsbedingungen vornehmen können. Solche Techniken sind auch inbesondere dann geeignet, wenn wie bei den in FRAQL realisierten Schemaoperationen die in einer Anfrage einbezogenen Datenbankobjekte erst zur Ausführungszeit berechnet werden. Ein weiterer Aspekt ist die Unterstützung von Analyseaufgaben, die bei großen und unter Umständen weit verteilten Datenbeständen mit langen Anfragezeiten verbunden sind, für die jedoch häufig keine exakten Ergebnisse benötigt werden. So lassen sich Aggregationen approximieren oder Anfrageergebnisse inkrementell verfeinern, indem zunächst nur ein grobes Ergebnis auf der Basis weniger Daten (etwa einer Stichprobe) geliefert wird, das im weiteren Verlauf durch Hinzunahme weiterer Daten hinsichtlich der Genauigkeit verbessert wird. Sowohl für eine adaptive Anfrageverarbeitung als auch die sogenannte Online-Ausführung existieren bereits Vorschläge. Eine Verbindung mit den hier behandelten Techniken stellt jedoch noch eine große Herausforderung dar.

Literatur

[CSS99] S. Conrad, G. Saake, und K. Sattler. Informationsfusion - Herausforderungen an die Datenbanktechnologie. In A.P. Buchmann (Herausgeber), *Datenbanksysteme in Büro, Technik und Wissenschaft, BTW'99, GI-Fachtagung, Freiburg*, Informatik aktuell, Seiten 307–316. Springer-Verlag, 1999.

Der Anteil an dieser Arbeit betrifft die Ableitung und Formulierung der konkreten Anforderungen von Techniken der Informationsfusion.

[DGJ+01] O. Dunemann, I. Geist, R. Jesse, G. Saake, und K. Sattler. INFUSE – Eine datenbankbasierte Plattform für die Informationsfusion. In A. Heuer, F. Leymann, und D. Priebe (Herausgeber), *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW 2001), GI-Fachtagung, Oldenburg*, Reihe Informatik aktuell, Seiten 9–25. Springer-Verlag, 2001.

Der Beitrag zu dieser Arbeit besteht in der Konzeption und Entwicklung des Kernsystems der Fusion Engine sowie der darauf aufbauenden Konzepte der Fusionsoperationen und Worksheets.

[DGJ+02a] O. Dunemann, I. Geist, R. Jesse, G. Saake, und K. Sattler. Informationsfusion auf heterogenen Datenbeständen. *Informatik - Forschung und Entwicklung*, 17(3):112–122, 2002.

Siehe [DGJ+01]

[DGJ+02b] O. Dunemann, I. Geist, R. Jesse, K. Sattler, und A. Stephanik. A Database-Supported Workbench for Information Fusion: INFUSE (Software Demonstration). In C.S. Jensen, K.G. Jeffery, J. Pokorný, S. Saltenis, E. Bertino, K. Böhm, und M. Jarke (Herausgeber), *Proc. 8th Conf. on Extending Database Technology (EDBT 2002), Prague*, LNCS, Seiten 756–758. Springer-Verlag, 2002.

Siehe [DGJ+01]

[EHS+00] M. Endig, M. Höding, G. Saake, K. Sattler, und E. Schallehn. Federation Services for Heterogeneous Digital Libraries Accessing Cooperative and Non-cooperative Sources. In Y. Kambayashi, G. Wiederhold, J. Klavans, W. Winiwarter, und H. Tarumi (Herausgeber), *Proc. of Kyoto Int. Conf. on Digital Libraries: Research and Practice*, Seiten 314–321. IEEE Computer Society Press, 2000.

Der eigene Beitrag zu diesem Anwendungsbereich umfasst die Entwicklung und Bereitstellung des Anfragesystems als Basis des Föderierungsdienstes.

[GS02] M. Gertz und K. Sattler. Integrating Scientific Data through External Concept-based Annotations. In Z. Lacroix (Herausgeber), *Proc. 2nd Int. Workshop on Data Integration over the Web (DIWeb 2002), Toronto, Canada*, Seiten 87–101, 2002.

Die Entwicklung des Annotationsmodells und der darauf aufbauenden Anfrageoperationen sowie die Realisierung des Annotationsservers bilden den eigenen Anteil an dieser Arbeit.

[GSG+02] M. Gertz, K. Sattler, F. Gorin, M. Hogarth, und J. Stone. Annotating Scientific Images: A Concept-based Approach. In J. Kennedy (Herausgeber), *Proc. 14th Int. Conf. on Scientific and Statistical Database Management (SSDBM 2002), Edinburgh, Scotland*, Seiten 59–68. IEEE Computer Society, 2002.

Siehe [GS02]

[SCS00] K. Sattler, S. Conrad, und G. Saake. Adding Conflict Resolution Features to a Query Language for Database Federations. *Australian Journal of Information Systems*, 8(1):116–125, 2000.

Der eigene Anteil dieser Arbeit umfasst Sprachentwurf und Implementierung von FRAQL sowie die Anwendung der Sprachkonzepte zur Behandlung der Integrationskonflikte.

[SCS02] K. Sattler, S. Conrad, und G. Saake. Datenintegration und Mediatoren. In E. Rahm und G. Vossen (Herausgeber), *Web-Datenbanken*. dpunkt.verlag, Heidelberg, 2002.

Der eigene Beitrag zu dieser Arbeit besteht in der Aufarbeitung und Diskussion der Architektur von Mediatorsystemen sowie den Aspekten der Anfrageverarbeitung.

[SCS03] K. Sattler, S. Conrad, und G. Saake. Interactive Example-driven Integration and Reconciliation for Accessing Database Federations. *Information Systems*, 28:393–414, 2003.

Der eigene Beitrag zu dieser Arbeit betrifft neben den Spracherweiterungen insbesondere die Semantikdefinitionen der Schemaoperationen sowie die Idee des beispielgetriebenen Vorgehens einschließlich der Werkzeugunterstützung.

[SD01] K. Sattler und O. Dunemann. SQL Database Primitives for Decision Tree Classifiers. In H. Paques, L. Liu, und D. Grossman (Herausgeber), *Proc. of the 10th ACM CIKM Int. Conf. on Information and Knowledge Management, Atlanta, Georgia, USA*, Seiten 379–386, 2001.

Der eigene Beitrag betrifft hier die Entwicklung der Primitive sowie deren Einbindung in das konkrete Klassifikationsverfahren.

[SDG+01] K. Sattler, O. Dunemann, I. Geist, G. Saake, und S. Conrad. Limiting Result Cardinalities for Multidatabase Queries using Histograms. In B.J. Read (Herausgeber), *Proc. of 18th British National Conf. on Databases (BNCOD 2001), Oxford, U.K.*, LNCS 2097, Seiten 152–167. Springer-Verlag, 2001.

Der Beitrag zu dieser Arbeit besteht in der Konzeption der Histogrammbasierten Realisierung der Operatoren.

[SES00a] E. Schallehn, M. Endig, und K. Sattler. Citation Linking in Federated Digital Libraries. In M. Roantree, W. Hasselbring, und S. Conrad (Herausgeber), *Proc. 3rd Int. Workshop on Engineering Federated Information Systems (EFIS'00), Dublin, Ireland*, Seiten 53–60. Akadem. Verlagsgesellschaft Berlin, 2000.

Siehe [EHS+00]

[SES00b] E. Schallehn, M. Endig, und K. Sattler. Integrating Bibliographical Data from Heterogeneous Digital Libraries. In Y. Masunaga, J. Pokorný, J. Stuller, und B. Thalheim (Herausgeber), *Proc. of Challenges – Symposium on Advances in Databases and Information Systems (ADBIS-DASFAA 2000), Prague, Czech Republic*, Seiten 161–170, 2000.

Siehe [EHS+00]

[SGHS03] K. Sattler, I. Geist, R. Habrecht, und E. Schallehn. Konzeptbasierte Anfrageverarbeitung in Mediatorsystemen. In G. Weikum, H. Schöning, und E. Rahm (Herausgeber), *Proc. BTW 2003 – Datenbanksysteme für Business, Technologie und Web, Leipzig*, GI-Edition Lecture Notes in Informatics, Seiten 78–97, 2003.

Der Beitrag zu dieser Arbeit besteht in Entwurf und Realisierung der Anfragesprache sowie der Entwicklung der Anfrageprozessors.

[SH99] K. Sattler und M. Höding. Adapter Generation for Extraction and Querying Data from Web Sources. In S. Cluet und T. Milo (Herausgeber), *Proc. of 2nd ACM SIGMOD Workshop WebDB'99 (Informal Proceedings), Philadelphia*, Seiten 49–54, 1999. http://www-rocq.inria.fr/∼cluet/WEBDB/.

Der eigene Beitrag zu dieser Arbeit besteht in der Konzeption und Entwicklung des Toolkits einschließlich der Extraktionsprimitive.

[SS99] K. Sattler und G. Saake. Supporting Information Fusion with Federated Database Technologies. In S. Conrad, W. Hasselbring, und G. Saake (Herausgeber), *Proc. 2nd Int. Workshop on Engineering Federated Information Systems (EFIS'99), Kühlungsborn, Germany*, Seiten 179–184. infix-Verlag, Sankt Augustin, 1999.

Der Anteil an dieser Arbeit betrifft die Ableitung und Formulierung der konkreten Anforderungen von Techniken der Informationsfusion.

[SS01] K. Sattler und E. Schallehn. A Data Preparation Framework based on a Multidatabase Language. In M.E. Adiba, C. Collet, und B.C. Desai (Herausgeber), *Proc. of Int. Database Engineering and Applications Symposium (IDEAS 2001), Grenoble, France*, Seiten 219–228. IEEE Computer Society, 2001.

Die Transformations- und Bereinigungsoperationen sowie deren Einsatz zur Bearbeitung konkreter Aufgaben der Datenaufbereitung bilden den eigenen Beitrag zu dieser Arbeit.

[SSS01] E. Schallehn, K. Sattler, und G. Saake. Advanced Grouping and Aggregation for Data Integration. In H. Paques, L. Liu, und D. Grossman (Herausgeber), *Proc. of the 10th ACM CIKM Int. Conf. on Information and Knowledge Management, Atlanta, Georgia, USA*, Seiten 547–549, 2001. *Short Paper*.

Der Anteil an dieser Arbeit umfasst die Definition der Semantik der Gruppierungs- und Aggregatoperationen, die Entwicklung der zur Einbindung in FRAQL notwendigen Erweiterungsmechanismen sowie die Trie-basierte Realisierung der Indexstruktur.

[SSS02] E. Schallehn, K. Sattler, und G. Saake. Extensible and Similarity-based Grouping for Data Integration. In *Proc. of 18th Int. Conf. on Data Engineering (ICDE'02), San Jose, CA*, 2002. *Poster Paper*.

Siehe [SSS01]