Informationsflut und Datenbanktechnologie — Defizite und Herausforderungen [1]

Gunter Saake

Zusammenfassung Der Umfang der elektronisch gespeicherten Information wächst weltweit in immer weiter steigender Geschwindigkeit. In diesem Zusammenhang kann durchaus von einer *Informationsflut* gesprochen werden, deren Auswirkungen zum Teil unvorhersehbar sind — die Beherrschung dieser Informationsflut wird zukünftig ein wesentlicher Wettbewerbsvorteil sein.

Die Datenbanktechnologie wurde zur Beherrschung großer Datenbestände entwickelt. Welche Herausforderungen stellen sich der Datenbankforschung im Zusammenhang mit den beschriebenen Entwicklungen? Der vorliegende Text diskutiert Defizite und Herausforderungen der aktuellen Datenbanktechnologie und versucht einen Blick auf zukünftige Entwicklungen. Ein aktueller Teilaspekt, die Entwicklung *föderierter Datenbanken*, wird dabei detaillierter vorgestellt.

1 Einleitung

Die Datenbanktechnik und ihre Anwendungen befinden sich zur Zeit in einer Umbruchphase. Die klassische Datenbank-Welt bezieht sich auf einen stark *zentralistischen Ansatz*: Betrachtet wird *ein* integrierter Datenbestand, der den gesamten Informationsbedarf eines Unternehmens beschreibt. Dieser Datenbestand wird in *einem* Datenmodell entworfen und führt zu einer umfassenden Beschreibung, die dann folgend dieser globalen Sichtweise zum Teil sogar *unternehmensweites Datenmodell* genannt wird. Als Technologie wird ebenfalls eine einheitliche Plattform angestrebt, etwa eine Realisierung mittels relationaler Datenbanksysteme (RDBS).

Im Extremfall wäre die Idealsituation für jedes Unternehmen genau ein Datenbankschema, das die *eine* verwaltete Datenbank beschreibt. Folgend diesem Ansatz sollte selbst eine verteilte Realisierung dieser Datenbank dem Anwender verborgen bleiben: Verteilungsaspekte sind in dieser Sichtweise *transparent*, also nicht sichtbar [Kus96].

Die Grenzen dieses idealisierten Ansatzes sind allerdings seit längerer Zeit sichtbar geworden:

- Lokale Organisationseinheiten arbeiten oft unter einem hohen Grad an *Autonomie* bezüglich der zu verwaltenden Informationen.
- Sowohl in Bezug auf die Datenverwaltungs-Software als auch auf die Strukturierung der Datenbestände besteht eine *Vielfalt* an sich zum Teil widersprechenden Anforderungen.

[1] Ausarbeitung der Antrittsvorlesung von Gunter Saake an der Otto-von-Guericke-Universität Magdeburg, gehalten am 25. 1. 1996. Aufgrund der späten Drucklegung wurden aktuelle Literaturreferenzen ergänzt, die die Ergebenisse einiger der in der Antrittsvorlesung vorgestellten Forschungsvorhaben dokumentieren.

- Im Zeitalter des Internets kann keine isolierte und abgeschlossene Informationsmodellierung mehr ausreichen — die *Außenwelt* mit ihren nicht oder nur wenig beeinflußbaren Informationsquellen fordert offene Systeme auch in der Datenhaltung.
- Mehr denn je befinden sich Organisationsstrukturen, Geschäftsregeln und auch Informationsdienste in einem Zustand des *ständigen Wandels*, auf den die Datenhaltungs-Infrastruktur flexibel reagieren muß.

Ziel dieses Beitrags ist eine Einschätzung des Selbstverständnisses der Datenbank-Forschung. Verbunden damit ist ein vorsichtiger Ausblick in die Zukunft und ein exemplarisches Beschreiben neuer Ansätze, die sich speziell den beschriebenen Herausforderungen stellen.

Der Beitrag ist wie folgt gegliedert. Der Aspekt der Informationsflut und die Konzepte der klassischen Datenbanktechnologie werden jeweils in einem kurzen Kapitel beschrieben. Die Probleme und Defizite der relationalen Datenbanktechnologie und mögliche Antworten darauf sind Inhalt von Abschnitt 3.3. Der Ansatz der *föderierten Datenbanken* als eine Reaktion auf die geänderten Anforderungen wird in Abschnitt 4 ausführlicher diskutiert. Weitere neue Ansätze und Herausforderungen an die Datenbanktechnologie werden anschließend in Abschnitt 5 beschrieben.

2 Motivation: Informationsflut

In der aktuellen Situation ist die Datenhaltung auf elektronischen Medien geprägt durch die Informations- bzw. Datenflut sowie durch einen Wechsel der Computing-Infrastruktur im Vergleich zu klassischen Anwendungsszenarien. Dieses hat für die Datenbanktechnologie Konsequenzen verschiedenster Art:

- Eine *quantitative* Veränderung betrifft das Aufkommen an zu verwaltenden Daten, die Größe einzelner Datenobjekte sowie verbunden hiermit Effizienzforderungen für den Zugriff auf gespeicherte Daten.
- *Qualitativ* ist ein Trend zu mehr Semantik als Teil der zu speichernden Daten zu erkennen. Stichworte hierfür sind der Übergang von Datenbanken zu Objektbanken und Wissensbanken, intelligente Suchverfahren und differenzierte Zugriffsprofile als Teil 'lernender', adaptiver Systeme.
- *Pragmatische* Konsequenzen ergeben sich aus der sich ändernden Datenverarbeitungs-Infrastruktur in Unternehmen, neuen Standards und dem Einfluß der Telekommunikationstechnik auf die Datenhaltung.

Diese drei Aspekte werden in den folgenden Abschnitten detaillierter betrachtet.

2.1 Quantitative Veränderungen

Die quantitativen Veränderungen betreffen insbesondere die Anzahl und Größe zu speichernder Informationseinheiten und die Effizienzanforderungen beim Zugriff auf die Daten.

- Das bekannteste Beispiel für einen unüberschaubar wachsenden Datenbestand ist zur Zeit das *World Wide Web* im Internet. Es handelt sich aus Datenverwaltungssicht um "unstrukturierte" riesige Dokumentenbestände, die durch Verweise miteinander verknüpft sind — ein unüberschaubarer gerichteter Graph mit unzähligen Einstiegspunkten.

- In der eher klassischen Datenverarbeitung etwa in Unternehmen ist ein Trend erkennbar, eine auch langfristige Archivierung möglichst *aller* elektronisch gesteuerten Vorgänge (elektronische Kassen, Lagerverwaltung, ...) vorzunehmen. Derartige Datenbestände können in einem *Data Warehouse* zusammengefaßt werden und bilden eine Quelle für mit statistischen Methoden (*Data Mining*) und regelbasierten Ansätzen (*Knowledge Discovery*) identifizierbare geschäftskritische Zusammenhänge.

- Die Verbreitung von Personal-Computern hat die Methoden der Dokumentverarbeitung stark verändert. Neben der Aufbewahrung der eigentlichen Dokumente werden verschiedene Versionen archiviert und eine redundante Speicherung von Dokumenten ist weit verbreitet. Auch werden deutlich mehr Dokumente elektronisch archiviert — die schöne Idee des papierarmen Büros zeigt ihre Schattenseiten (ohne den Papierberg abzutragen).

  Von der Datenhaltung her gesehen sind Dokumente oft mehr oder weniger stark strukturierte große Datenobjekte mit heterogenen Bestandteilen, etwa eingebetteten Bildern und Graphiken.

- Auch der Speicherbedarf für wissenschaftliche Daten steigt ständig. Zu nennen sind hier Satellitendaten, Bildarchive, Simulationsergebnisse, chemische Datenbanken oder das Genom-Projekt. Neben dem enormen Umfang des Speicherbedarfs zeichnen sich wissenschaftliche Daten oft durch spezielle Suchbedingungen aus, die eher auf Mustererkennung und Ähnlichkeitssuche als auf dem von Datenbanksprachen primär unterstützten Identitätsvergleich basieren.

- Bereits im Zusammenhang mit Dokumenten ist die Speicherung von Bildern und Animationen angesprochen worden, die sich neben einem großen Speicherbedarf auch durch Probleme der Inhaltssuche auf derartigen Objekten auszeichnen.

- Als letzte und besonders anspruchsvolle Aufgabe ist die Speicherung von digitalen Videosequenzen, etwa im Zusammenhang mit dem *Video on Demand*, zu nennen. Neben dem Speicherbedarf ist hier insbesondere der Echtzeitaspekt relevant — der Transfer vom Hintergrundspeicher zum Endgerät sollte ein direktes Abspielen der Videosequenz inklusive schnellem Vorlauf und ähnlichen Funktionen ermöglichen.

Zusammengefaßt droht das Bedarfswachstum sogar die aktuellen Kapazitätssteigerungen der Datenträgerentwicklung zu übersteigen — die Daten- und Informationsflut droht uns zu erschlagen.

2.2 Qualitative Veränderungen

Neben den quantitativen Veränderungen sind auch eher qualitative Veränderungen in der Art und Struktur der gespeicherten Daten zu erkennen. Grob gesagt, wird mehr "Semantik" in die Datenbank gepackt — neben den reinen Daten werden zunehmend auch zugehörige Regeln, Funktionen und Anwendungswissen erfaßt.

- Neuere Datenbankmodelle nehmen mehr Anwendungssemantik in die Datenbankbeschreibung auf, die vom Datenbank-Management-System etwa für Optimierungszwecke genutzt werden kann. Aktuelle kommerzielle relationale Datenbanken verwalten bereits einfache Regeln und Bedingungen, und Forschungsprototypen streben eine volle Integration von Regeln in das DBMS an.

  Im Büro-Bereich werden komplette Dokumente statt einfacher "Datensätze" verwaltet. Allgemein werden in Zukunft komplexe Strukturen und anwendungsspezifische Datentypen von DBMS unterstützt.

- Gerade im Bereich der Bilddatenbanken und in der Dokumentverwaltung besteht ein Bedarf für eine Inhaltssuche anstelle eines einfachen Attribut-"Matchings".

- Die Entwicklungen im Bereich der *deduktiven* und *aktiven Datenbanken* zielen ebenfalls auf die Integration von mehr Anwendungswissen in die Datenbankbeschreibung und der vom DBMS angebotenen Funktionalität. In diesen Ansätzen überwachen Regelsysteme die Qualität der Datenbestände und dieses Regel*'wissen'* wird als Teil der Datenbank behandelt.

Einer der auf diese neuen Anforderungen reagierenden Ansätze ist die Entwicklung von *Objektdatenbanken* [SST97], auf die wir später näher eingehen werden.

2.3 Pragmatische Veränderungen

Neben den quantitativen und qualitativen Anforderungen an Datenbestände hat sich auch das Umfeld geändert, in dem Datenbanken verwaltet werden. Wir bezeichnen diese veränderten Bedingungen als pragmatisch motivierte Anforderungen.

- Die Entwicklungen der Rechnerarchitektur und der Kommunikationstechnologie haben zu neuen Architekturkonzepten auch für die Datenverwaltung geführt. Der Preisverfall für Speicher hat zur Entwicklung von sogenannten Hauptspeicherdatenbanken geführt, bei denen der Datentransfer vom Hintergrundspeicher nicht mehr der primäre Leistungsparameter ist. Verteilte Datenhaltung und Client-Server-Architekturen sind durch den steigenden Einsatz von Workstations und PCs verbreitete Ansätze geworden [Dad96].

  Andere aktuelle Entwicklungen betreffen Parallelrechner-DBMS [Rah94], Datenbanken in Workstation-Clustern und Datenhaltungsprobleme beim *Mobile Computing*.

- Neben den informationstechnischen Rahmenbedingungen haben sich auch die organisatorischen Konzepte weiterentwickelt. An Stelle einer zentralen DV-Abteilung tritt eine dezentralisierte Datenverarbeitung in den Unternehmen.

  Als Resultat dieser organisatorischen Änderungen werden lokale, autonom bearbeitete Datenbestände als Teile einer zentral verwalteten Datenbank aufgefaßt. Dieser Ansatz führt zum Konzept der *föderierten Datenbanken*, das in dieser Ausarbeitung später noch ausführlicher behandelt wird.

- Ein weiterer Trend betrifft das Verschmelzen der Datenhaltung mit anderen, vormals getrennten Bereichen: Betriebssysteme, Benutzerschnittstellen und Kommunikations-Software.

- Die Verbreitung des Zugangs zu Datenbanken über PCs und das Internet hat zu neuen Benutzerklassen geführt. Stichwort ist hier der Datenbank-Zugang für "Laien", der ohne eine aufwendig zu erlernende Kunstsprache wie SQL erfolgen muß.
- Als letzter Punkt seien hier aktuelle Standardisierungsbestrebungen wie SQL3, ODMG-93, OLE oder CORBA genannt, die die Entwicklungen im Datenbankbereich beeinflußen.

Die genannten neuen Randbedingungen haben einen starken Einfluß auf die Entwicklung der Datenbanktechnologie genommen. Die folgenden Ausführungen sollen die resultierenden Entwicklungen beispielhaft verdeutlichen.

3 Klassische Datenbanktechnologie

Um Defizite der heutigen Datenbanktechnologie und Antworten auf neue Anforderungen einschätzen zu können, müssen wir uns die klassischen Konzepte und Techniken dieses Gebietes genauer ansehen. Der Begriff *Datenbank* bezeichnet einen im allgemeinen von mehreren Anwendungen bzw. Anwendern langfristig *gemeinsam genutzten* Datenbestand [HS95]. Diese Definition induziert die im folgenden betrachteten Konzepte der Datenunabhängigkeit, der Trennung der Datenbeschreibung von den eigentlichen Daten, und die zentralen Funktionen eines Datenbank-Management-Systems.

3.1 Datenbanksysteme

Eine Datenbank wird durch eine Spezial-Software verwaltet, dem sogenannten *Datenbank-Management-System* (kurz DBMS). In der Regel handelt es sich um große Datenbestände, die spezielle effiziente Zugriffsmethoden erfordern. Die Verwendung in mehreren Anwendungen erzwingt eine anwendungsunabhängige, *abstrakte* Datenbeschreibung, die weder Implementierungsdetails noch Anwendungsspezifika widerspiegeln darf.

Ein DBMS ermöglicht den Zugriff im Mehrbenutzerbetrieb und sorgt hierbei insbesondere für die notwendige Synchronisation. Die integrierte Datenhaltung im Mehrbenutzerbetrieb sollte *anwendungsunabhängig*, *programmiersprachenunabhängig* und unabhängig von Betriebssystemen und Hardware-Plattformen erfolgen. Das DBMS stellt eine genormte Schnittstelle zu den Anwendungen bereit. Es löst die effizienzkritische Aspekte des Datenzugriffs intern, um Abhängigkeiten der Anwendungen von Interna der Datenhaltung (die aus Performance-Gründen geändert werden können) zu vermeiden: Das DBMS *optimiert* Anfragen und andere Datenzugriffe. Weitere Aufgaben eines DBMS umfassen Datenschutz, Datensicherheit und die Sicherstellung der Datenintegrität.

Der prinzipielle Aufbau von Datenbanksystemen wird in Abbildung 1 verdeutlicht. Ein Datenbanksystem besteht aus einem (oder mehreren) Datenbeständen, den *Datenbanken*, die von der Spezial-Software DBMS integriert verwaltet werden. Der Zugriff der Anwendungen auf die Daten erfolgt ausschließlich über die Schnittstellen des DBMS.

Datenunabhängigkeit

Eine der zentralen Forderung an Datenbanksysteme ist die nach *Datenunabhängigkeit*. Die Datenunabhängigkeit fordert grob gesagt die Stabilität der Schnittstellen gegenüber Ände-

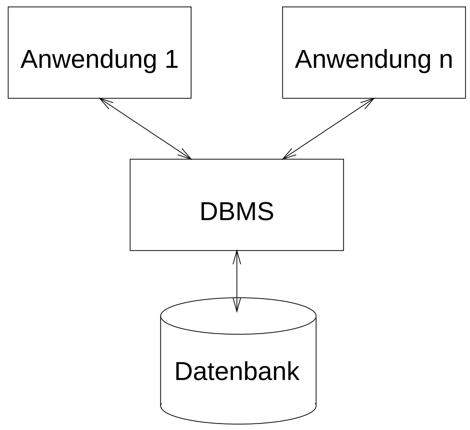

Abbildung 1: Aufbau von Datenbanksystemen

rungen sowohl der Anwendungen als auch der internen Datenspeicherung.

Die Datenunabhängigkeit wird durch die in Abbildung 2 skizzierte *Drei-Ebenen-Architektur* verwirklicht. Die *physische* Datenunabhängigkeit fordert die Trennung der konzeptionellen Datenbeschreibung von der internen Datendarstellung. Die *logische* Datenunabhängigkeit wiederum trennt die externen, auf Anwendungen zugeschnittenen Darstellungen von der konzeptionellen Gesamtsicht. Das Ergebnis sind die drei dargestellten Schemaebenen.

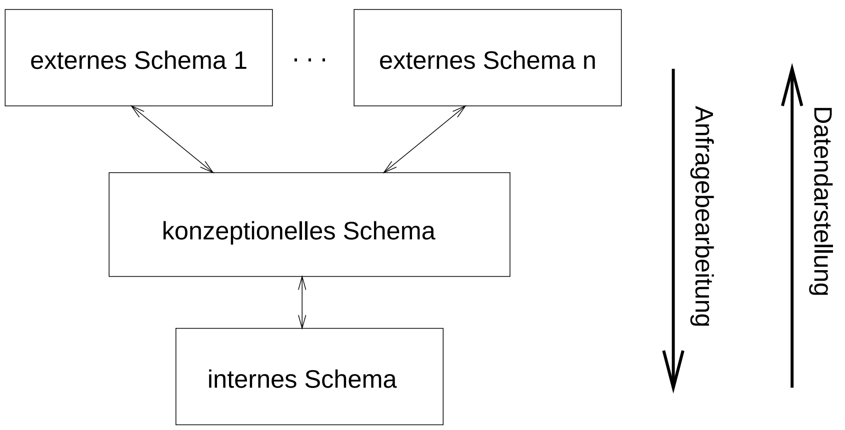

Abbildung 2: Drei-Ebenen-Architektur von Datenbanksystemen

Aufgaben und Funktionen eines DBMS

Aus den beschriebenen Konzepten ergeben sich insbesondere die folgenden neun Anforderungen an die Funktionalität von DBMS, die auch als die Codd'schen Regeln bekannt sind [Cod82]:

1. *Integration: Datenbankmodell.* Ein DBMS muß den Gesamtdatenbestand integriert und redundanzfrei verwalten.

2. *Operationen: Manipulationssprachen.* Die Schnittstelle des DBMS muß Operationen zur Datenmanipulation und Suchfunktionen bereitstellen.

3. *Katalog / Data Dictionary.* Die Datenbankbeschreibung ist explizit zugreifbar.

4. *Benutzersichten.* Die logische Datenunabhängigkeit wird durch externe, anwendungsspezifische Datenrepräsentationen unterstützt.

5. *Konsistenzüberwachung.* Die Integrität des Datenbestandes wird kontrolliert.

6. *Datenschutz.* Unbefugter Zugriff auf die Datenbank wird verhindert.

7. *Transaktionen.* Operationen auf den Daten können zu logischen Einheiten zusammengefaßt werden.

8. *Synchronisation.* Der Mehrbenutzerbetrieb führt zu keinen Anomalien.

9. *Datensicherung.* Unbeabsichtigter Datenverlust aufgrund von Bedienungsfehlern, Software- oder Hardwareversagens wird verhindert.

Die Historie der Datenbanktechnologie zeigt einen Weg von unkontrollierter Datenhaltung hin zu einer zentralen, integrierten Datenhaltung durch Datenbanksysteme. Den bisherigen Höhepunkt dieser Entwicklung stellen die *relationalen Datenbanksysteme* dar, die die meisten beschriebenen Konzepte in der Reinform unterstützen und die wir im nächsten Abschnitt genauer charakterisieren werden.

3.2 Relationale Datenbanken

Relationale Datenbanken bieten als einziges Modellierungskonzept *Relationen* über elementaren Wertebereichen an. Relationen werden als *Tabellen* notiert. Abbildung 3 zeigt zwei Relationen, die Daten über Bücher und Ausleihvorgänge speichern. Beziehungen zwischen Einträgen in unterschiedlichen Tabellen werden ausschließlich durch Wertevergleiche hergestellt. Im Beispiel werden die beiden Tabellen durch die Einträge in der Spalte `INVENTRANR` in Beziehung gesetzt.

`AUSLEIH`

INVENTARNR	NAME
4711	Meyer
1201	Schulz
0007	Müller
4712	Meyer

`BUCH`

INVENTARNR	TITEL	ISBN	AUTOR
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth

Abbildung 3: Zwei Beispielrelationen

Die theoretischen Grundlagen des relationalen Datenbankmodells basieren auf Mengen und Funktionen. Relationen sind Mengen von Tupeln; eine Reihenfolge der Tupel ist nicht vorgegeben, und es gibt keine Duplikate. Mit der relationalen Algebra existiert eine theoretische Grundlage für Anfragen. Für die implementierten Systeme wird allerdings von der strengen formalen Grundlage abgewichen, um etwa Duplikate in Tabellen und Sortierung behandeln zu können.

Eng verbunden mit dem relationalen Modell ist die standardisierte Datenbanksprache *SQL* [DD93, HS95]. Das folgende Beispiel zeigt die Grundprinzipien von Anfragen in SQL:

```
select BUCH.INVENTARNR, TITEL, NAME
from   BUCH, AUSLEIH
where  NAME = 'Meyer' and
       BUCH.INVENTARNR = AUSLEIH.INVENTARNR
```

Diese Anfrage an die beiden Beispielrelationen aus Abbildung 3 verbindet die Einträge beider Tabellen über die Inventarnummer und gibt die von 'Meyer' ausgeliehenen Bücher aus. SQL-Anfragen sind deskriptiv: Die Verarbeitungsreihenfolge und die für die Anfragebeantwortung notwendigen Algorithmen werden nicht vorgegeben.

Charakteristika Relationaler Datenbanksysteme

Die existierende relationale Datenbanktechnologie wurde für ein bestimmtes Anwendungsszenario optimiert, das aus kommerziellen Anwendungen motiviert ist. Die daraus entstehenden Eigenschaften von relationalen Datenbank-Management-Systemen RDBMS sind Ausgangspunkt für Neu- und Weiterentwicklungen im Datenbankbereich und sollen daher hier kurz skizziert werden.

Ein typischer Anwendungsbereich für relationale Datenbanken, etwa im Bereich Datenhaltung für eine Bankanwendung, ist durch bestimmte Eigenschaften gekennzeichnet. Die klassische relationale Technologie ist auf diese Anforderungen hin optimiert:

- Gespeichert werden *kleine Datensätze fester Struktur*, die zum Beispiel Einträgen in einem Formular oder einer Tabelle entsprechen. Mehrere dieser Datensätze passen auf eine Seite des Hintergrundspeichers.

- Es existieren verhältnismäßig *wenige Typen von Datensätzen*. Diese Aussage ist relativ zu sehen: Für jeden Datensatztyp gibt es sehr viele tatsächlich gespeicherte einzelne Datensätze. Insgesamt ist die Anzahl der Datensatztypen, also der Relationenschemata, auch in kommerziellen Anwendungen oft groß und nicht zu vernachlässigen.

- Das wesentliche Qualitätsmerkmal für einen Datenbankentwurf ist die Vermeidung von Redundanz und den sich aus redundanter Speicherung ergebenden Änderungsanomalien. Die ausgereifte *Entwurfstheorie bezüglich Datenredundanz* ist eine Stärke des relationalen Modells.

- Relationale Datenbanken unterstützen nur *einfache Änderungsoperationen* als Basisoperationen: Einfügen von Tupeln, Attributänderungen, und Löschen von Tupeln.

- Aus den Basisänderungen werden *einfache, kurze Transaktionen* gebildet. Typisches Beispiel einer derartigen Transaktion ist eine Buchung zwischen zwei Bankkonten.

- Das DBMS unterstützt das *isolierte Arbeiten* vieler Anwender. Kooperierendes Arbeiten, wie es etwa in Entwurfsprozessen notwendig ist, wird nicht unterstützt.

Die Hauptziele der relationalen Datenbanktechnologie lassen sich mit den Schlagworten *Redundanzvermeidung und zentrale Kontrolle* des Datenbankzugriffs zusammenfassen.

Pluspunkte Relationaler Datenbanksysteme

In der aktuellen Diskussion betreffend neuer Datenbankmodelle, etwa der objektorientierten Ansätze, werden oft die unbestreitbaren Vorteile der relationalen Datenbanktechnologie nicht angemessen gewürdigt.

- Relationale Datenbanksysteme realisieren die Trennung von konzeptioneller Darstellung und der internen Speicherung, die vom Prinzip der physischen Datenunabhängigkeit gefordert wird, in vorbildlicher Weise.
- Sie bieten deskriptive Sprachen mit effizienter Optimierung an, die trotz der verborgenen internen Realisierung einen effizienten Zugriff ermöglichen.
- Mit der Normalisierungstheorie werden Qualitätsmaße für relationale Schemata vorgegeben, mit deren Hilfe der Grad der Redundanz in der Speicherung minimiert werden kann.
- Mit SQL existiert ein verbreiteter Standard zum Zugriff auf relationale Datenbanken.
- Heutige relationale Datenbanksysteme bieten ausgereifte Techniken für interne Speicherstrukturen, Datensicherung und Transaktionen im Mehrbenutzerbetrieb.

Neue Datenbankansätze müssen sich an diesen Eigenschaften messen lassen, wenn sie die relationale Technologie ablösen wollen.

3.3 Probleme und Defizite

Neben den erwähnten unbestrittenen Vorzügen der relationalen Datenbanktechnologie existieren eine Reihe von Schwachpunkten, die sich vor allem aus der Einfachheit des zugrundeliegenden Modells ergeben.

- Relationale Datenbanken bieten nur *eingeschränkte Modellierungsmöglichkeiten* an: Relationen sind das einzige Modellierungskonzept. Andere Strukturen, etwa Hierarchien oder Listenstrukturen müssen auf das Relationenmodell abgebildet werden. Problematisch ist auch die Beschränkung auf elementare Wertebereiche für Attribute, die in einigen Anwendungsbereichen nicht adäquat sind.
- Die Abbildung auf Relationen bedeutet für komplexe Strukturen zwangsweise einen *Effizienzverlust durch die notwendige Transformationen.*
- In der Anwendungsprogrammierung wird bei relationalen Datenbanken vom sogenannten *Impedance Mismatch* gesprochen. Gemeint ist hier, daß die Datenstrukturen von SQL (Mengen von Tupeln) kein direktes Gegenstück in prozeduralen Sprachen wie Pascal oder C haben, und der Anwendungsprogrammierer sich daher in zwei Verarbeitungsparadigmen zurechtfinden muß.

- Die aktuelle Arbeits- und Unternehmenskultur widerspricht dem zentralisierten Ansatz, in dem *alle* Daten zentral und redundanzfrei verwaltet werden.

- Ebenfalls widerspricht die heutige Informations- und Kommunikations-Infrastruktur einer ausschließlich zentralen Datenhaltung.

Diese Probleme wurden besonders in den Bemühungen deutlich, die klassische Datenbanktechnologie für sogenannte Nicht-Standard-Anwendungen im Ingenieurbereich einzusetzen. Eine ausführliche Diskussion der Probleme kann insbesondere in Büchern über Objektdatenbanken gefunden werden, die mit dieser Diskussion die Einführung neuer Datenmodellkonzepte motivieren [Heu97, SST97].

Mögliche Antworten auf Probleme

Die erwähnten Defizite haben in der Datenbankforschung aktuell zu mehreren Forschungsrichtungen geführt, die neue Techniken entwickeln oder klassische Ansätze erweitern. Insbesondere drei Richtungen werden derzeit intensiv untersucht: *Objektdatenbanken*, *verteilte und föderierte Datenbanken* und *aktive Datenbanken*.

- **Objektdatenbanken**

  Objektdatenbanken oder auch *objektorientierte Datenbanken* unterstützen ein objektorientiertes Datenbankmodell, das an Modelle objektorientierter Entwurfsverfahren oder die Strukturen objektorientierter Programmiersprachen wie Smalltalk oder C++ angelehnt ist. Die Konzepte von Objektdatenbanken werden zum Beispiel in [Heu97, SST97] ausführlich diskutiert.

  Im Vergleich zu relationalen Datenbanken bieten Objektdatenbanken insbesondere die folgenden Vorteile:

  - Das Datenbankmodell bietet reichere Modellierungskonzepte.
  - Die Kopplung an objektorientierte Programmiersprachen verspricht die Probleme des "Impedance Mismatch" zu beseitigen.

  Aktuell wird besonders an der Entwicklung *objektrelationaler* Datenbanksysteme gearbeitet, die die Konzepte von relationalen und Objektdatenbanken vereinen sollen.

- **Verteilte und Föderierte Datenbanken**

  Föderierte Datenbanken sind ein Spezialfall der verteilten Datenhaltung [Rah94, Dad96], in dem verteilte Datenbanksysteme unter weitgehender Beibehaltung ihrer Autonomie föderiert werden [Con97]. Ein föderiertes Datenbanksystem FDBS stellt eine homogene Schnittstelle zu verteilten, heterogenen Datenbeständen zur Verfügung.

  Föderierte Datenbanken bieten folgende Vorteile:

  - Die Unterstützung einer verteilten Informations-Infrastruktur im Gegensatz zum zentralisierten Ansatz entspricht Konzepten neuer Organisationsstrukturen [UH97].
  - Die (teil-)autonome Verwaltung lokaler Datenbestände ermöglicht insbesondere die Einbindung existierender Datenbanken ("legacy systems").

- **Aktive und Deduktive Datenbanken**

  Aktive Datenbanken integrieren ein Regelsystem in das Datenbanksystem [DG96, WC96]. Im Gegensatz zu deduktiven Datenbanken, die einen an Logikprogrammierung angelehnten Ansatz verfolgen, unterstützen aktive Datenbanken eine operationale Regelsemantik, in der Trigger durch elementare Datenbankänderungen aktiviert werden.

  Aktive Datenbanksyteme erweitern die Datenbanktechnologie in folgenden Punkten:

  – Die Konzepte aktiver Datenbanken ermöglichen die enge Kopplung wissensbasierter Ansätze mit Datenbanken. Als Resultat kann mehr Anwendungssemantik durch das DBMS unterstützt werden.

  – Die in klassischen Datenbanken eher vernachlässigte Integritätsüberwachung für komplexere Bedingungen kann durch aktive Regeln realisiert werden.

Von besonderem Interesse sind natürlich die Versuche, diese drei Richtungen zu integrieren, um alle angesprochenen Problembereiche gleichzeitig in den Griff zu bekommen.

Im folgenden wird als eine dieser drei Richtungen die Entwicklung *föderierter Datenbanken* vorgestellt, die einen Forschungsschwerpunkt der Magdeburger Datenbankgruppe darstellt.

4 Föderierte Datenbanken

DerAnsatz der *föderierten Datenbanksysteme* realisiert einen bereichsübergreifenden Zugriff auf lokal gehaltene Datenbestände. Im Gegensatz zur klassischen Idee der verteilten Datenbanksysteme behalten die Komponentensysteme (teilweise) ihre Autonomie.

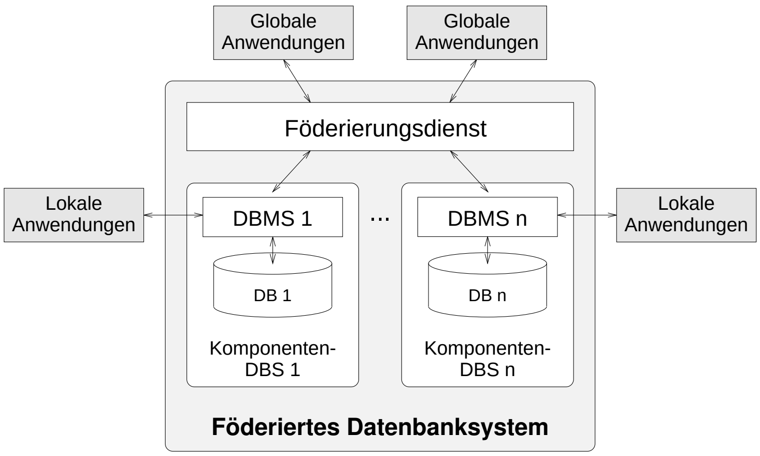

Abbildung 4: Aufbau eines föderierten Datenbanksystems

Die Abbildung 4 zeigt den prinzipiellen Aufbau eines föderierten Datenbanksystems. Der Föderierungsdienst kapselt die lokalen Datenbanken weitgehend unverändert ein und ermöglicht dadurch globale Anwendungen auf einer virtuellen Schnittstelle, die eine integrierte Sicht auf die lokalen Datenbestände bietet.

Lokale Anwendungen arbeiten weiterhin auf den lokalen Datenbeständen. Dies vermeidet den Aufwand der Re-Implementierung dieser Anwendungen auf der globalen Ebene und kann auch Effizienzvorteile bringen, da die Fähigkeiten der lokalen Systeme direkt ausgenutzt werden können.

4.1 Aufbau föderierter Datenbanksysteme

Ein föderiertes Datenbanksystem (kurz FDBS) kann als ein Datenbanksystem mit speziellen Eigenschaften charakterisiert werden. Ein FDBS erfüllt die allgemeinen Anforderungen an ein Datenbanksystem, stellt dabei eine virtuelle, homogene Schnittstelle für Zugriff auf unterschiedliche Datenquellen zur Verfügung, kapselt zu diesem Zweck die Heterogenität der Datenquellen ein und bewahrt dabei die *Autonomie* der Komponenten-Systeme [SL90, Con97]. Die lokale Anwendungen laufen aufgrund der Autonomie unverändert weiter.

FDBS ermöglichen prinzipiell eine globale Redundanzkontrolle bzw. Integritätssicherung über den Föderierungsdienst. Die lokale Autonomie muß allerdings teilweise eingeschränkt werden, um die Integritätserzwingung zu ermöglichen. Wir werden auf diesen Aspekt später noch eingehen.

Ein weiterer, praktisch relevanter Aspekt des Föderierungsdienstes ist eine schrittweise Daten-Migration zwischen Komponenten-DBMS, ohne daß laufende Anwendungen auf der globalen Ebene davon betroffen werden [RS94].

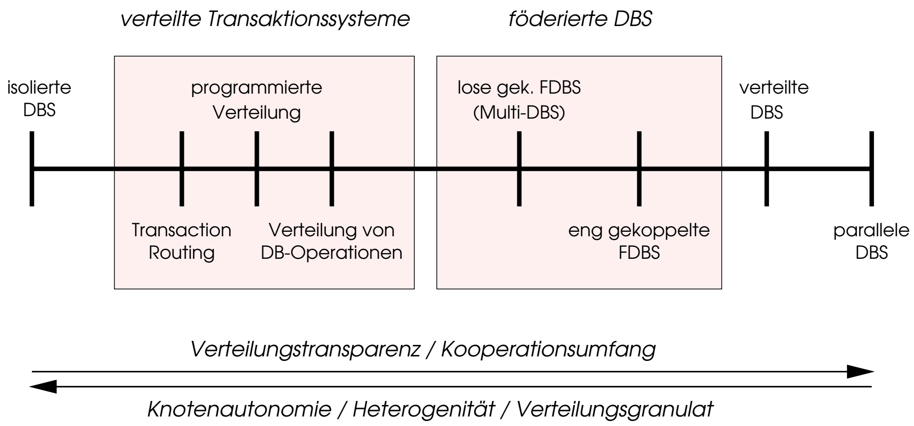

Abbildung 5: Spektrum bei verteilten Datenbanksystemen

Die Abbildung 5 aus [Rah94] zeigt das Spektrum der möglichen Realisierungen verteilter Datenverwaltung. Diesem Schema nach sind Knotenautonomie und Heterogenität gegenläufig zu den Eigenschaften Verteilungstransparenz bzw. Kooperation. Föderierte Datenbanken stellen in diesem Spektrum folglich einen Kompromiß dar: Ziel ist eine integrierte Zugriffsschicht ($\rightarrow$ Verteilungstransparenz) unter weitgehender Beibehaltung der lokalen Autonomie inklusive der Heterogenität lokaler Datenbeschreibungen.

Die Datenbeschreibungen eines FDBS sind in einer Erweiterung der klassischen Drei-Ebenen-Schema-Architektur organisiert. Abbildung 6 zeigt die sogenannte *Fünf-Ebenen-Architektur* nach Sheth und Larson [SL90], die sich als Referenzmodell für FDBS-Entwicklungen etabliert hat.

Die *fünf* für den Aufbau eines FDBS relevanten Schema-Ebenen haben die folgenden Aufgaben:

1. Die Komponentendatenbanksysteme KDBS stellen als Ausgangs-Schemata ihr jeweiliges *lokales konzeptionelles Schema* zur Verfügung. Die anderen beiden Schema-Ebenen

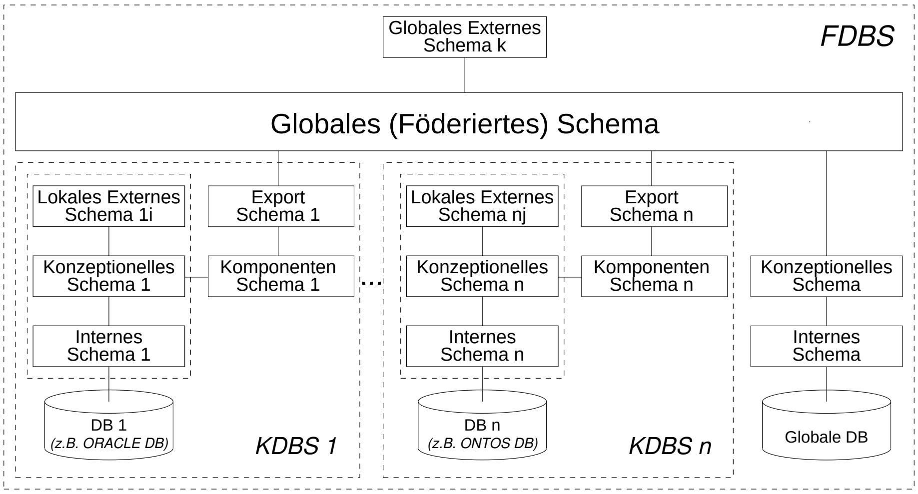

Abbildung 6: Schemaarchitektur eines föderierten Datenbanksystems

der Drei-Ebenen-Schema-Architektur der KDBSe — internes Schema und diverse (lokale) externe Schemata — sind weiterhin vorhanden, werden aber in die Fünf-Ebenen-Architektur nicht einbezogen.

2. Für jedes KDBS wird ein *Komponenten-Schema* aus dem konzeptionellen Schema abgeleitet. Das Komponentenschema überführt die lokalen Datenbankbeschreibungen in das global verwendete Datenbankmodell.

3. Das jeweilige *Export-Schema* blendet diejenigen Daten aus, die nicht föderiert werden sollen.

4. Das *globale Schema* oder auch *Föderierungs-Schema* integriert die einzelnen Export-Schemata zu einer homogenen Datenbeschreibung des Gesamtdatenbestands.

   Auf die Integrationsproblematik werden wir noch gesondert eingehen.

5. Die (globalen) *externen Schemata* haben — wie auch in der klassischen Drei-Ebenen-Schema-Architektur — die Aufgabe, anwendungsspezifische Sichten auf den Gesamtdatenbestand zur Verfügung zu stellen.

Nach der allgemeinen Diskussion der Architektur von FDBS werden wir zwei besonders wichtige Aspekte im Detail betrachten: Die Integration von Datenbanken und die Sicherung der Integrität in FDBS.

4.2 Integration heterogener Datenbestände

Das Problem der Integration heterogener Datenbestände stellt sich bei FDBS in besonderem Maße, da die lokalen Datenbanksysteme ihre Autonomie (und damit ihre Heterogenität) bewahren [Sch98]. Die zur Integration notwendigen Transformationen müssen daher durch das FDBS selbst vorgenommen werden.

Ziel der Integration ist ein widerspruchsfreies und homogenes *globales Schema*, das die Basis für anwendungsspezifische Sichten bilden kann. Die Heterogenität der lokalen Datenbanken manifestiert sich auf verschiedenen Ebenen: Die KDBS können heterogene Datenbankmodelle haben, selbst homogene Datenbankmodelle können zu heterogenen Datenbank-Schemata führen, und selbst bei identischem Schema kann Heterogenität auf der Datenebene auftreten. Wir werden diese drei Ebenen im folgenden kurz diskutieren.

Ein weiteres Problem ist die Wahl des Datenmodells für das globale Schema. Wird für dieses Modell eine große Ausdrucksfähigkeit erreicht, so spricht man von einem *semantisch reichen* Datenbankmodell. Typische semantisch reiche Datenbankmodelle sind Objektmodelle und erweiterte ER-Modelle. Für die Handhabbarkeit hingegen ist eine minimale Anzahl orthogonaler Konzepte vorzuziehen, die zu *semantisch armen* Datenbankmodellen führen. Das in [SS96a, Sch98] verwendete Datenbankmodell GIM ist ein semantisch armes Datenbankmodell, das nur wenige Konzepte unterstützt, aber dadurch leichter algorithmische Integrationsschritte ermöglicht.

4.2.1 Heterogenität der Datenmodelle

Die Heterogenität der Datenmodelle wird bei der Transformation in die Komponenten-Schemata beseitigt. Abbildung 7 zeigt zwei Datenbankschemata in heterogenen Datenmodellen, die denselben Weltausschnitt beschreiben. Die erste Datenbankbeschreibung zeigt ein Schema einer objektorientierten Datenbank in der in [Heu97] benutzten graphischen Notation. Die rechte Abbildung zeigt zwei Relationen im relationalen Datenmodell, die dieselbe Information speichern können. Die Aggregierung der objektorientierten Darstellung wird hier durch Fremdschlüsselbedingungen modelliert.

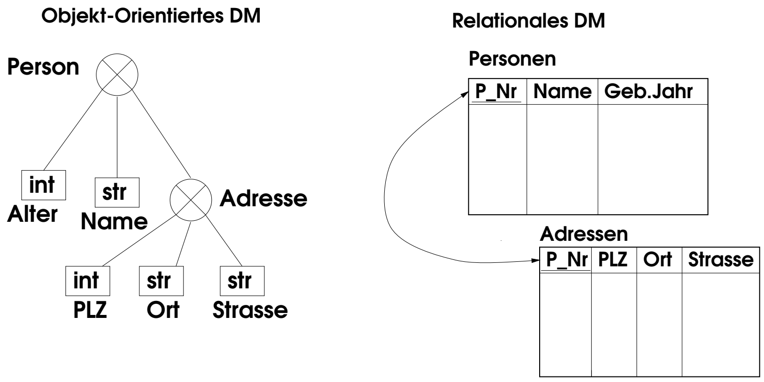

Abbildung 7: Heterogenität von Datenbankmodellen

Die Auflösung der Heterogenität auf der Datenmodellebene erfolgt durch eine Transformation in ein gemeinsames "Föderierungsdatenmodell". Verbreitet sind zur Zeit Objektmodelle (folgend dem ODMG-Standard [Cat94]) oder semantisch arme Datenbankmodelle, die formale Integrationsschritte ermöglichen [CHS+97].

4.2.2 Heterogene Datenbank-Schemata

Auch wenn die Datenmodelle zweier Komponenten-DBMS übereinstimmen, kann die Modellierung unterschiedlich sein. Abbildung 8 zeigt zwei Schemaausschnitte im erweiterten ER-Modell [EGH+92, HS95]. In einem Schema wird das Geschlecht von Personen durch ein

Attribut `Geschlecht` modelliert. Im Schema B wird dieselbe Information dadurch modelliert, daß die Klasse `Person` in die beiden Klassen `Frau` und `Mann` partitioniert wird.

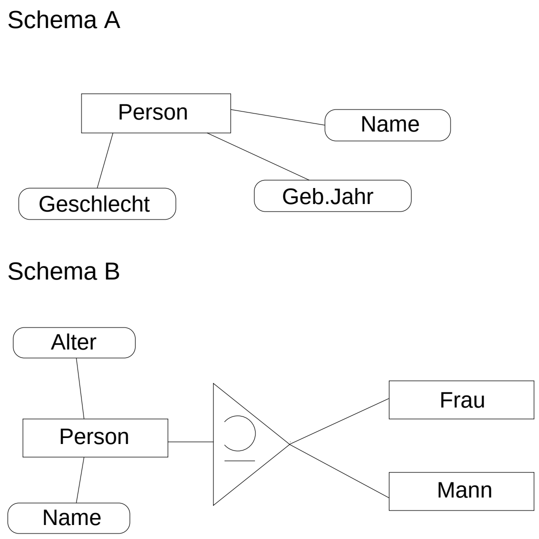

Abbildung 8: Heterogenität von Datenbankschemata

Der Prozeß der Auflösung der Heterogenität auf Schema-Ebene wird als *Schema-Integration* bezeichnet. Schema-Integration in erweiterten ER-Modellen wird z.B. von Spaccapietra und Koautoren in [SPD92, SP94] ausführlich diskutiert. In der Magdeburger Gruppe werden speziell die Möglichkeiten der Integration im Föderierungsdatenmodell GIM untersucht [SS96a, SS96b, SS98, Sch98, ST98].

4.2.3 Heterogenität auf der Datenebene

Auch auf der Datenebene selber tritt Heterogenität auf, wie Abbildung 9 verdeutlicht. In diesem Beispiel weisen zwei relationale Komponenten-DBMS dasselbe relationale Schema auf, verwenden aber zum Beispiel unterschiedliche Konventionen für die tatsächliche Speicherung von Namen und Geburtsjahren. Während dieses auf Probleme des Datenbankentwurfs zurückgeführt werden kann (allerdings sind Komponenten-DBMS auch diesbezüglich autonom), zeigt das Beipiel weitere Problem auf: Für einige Informationen, etwa für Berufsbezeichnungen, gelten keine allgemeingültigen Konventionen, und Tippfehler (hier im Vornamen) können ebenfalls auftreten.

Die Auflösung der Heterogenität auf der Datenebene muß auf vielfältige Methoden der Datenanalyse und Ähnlichkeitssuche zugreifen. *Explizit gespeichertes semantisches Föderierungswissen* kann unterschiedliche Konventionen bei Einträgen explizit darstellen, so daß eine Datenintegration weitgehend automatisch erfolgen kann.

4.2.4 Schema-Konflikte bei der Integration

Abbildung 10 klassifiziert die bei der Integration heterogener Datenquellen auftretenden Konflikte. *Namenskonflikte* können mittels linguistischem Wissen und Begriffswörterbüchern aufgelöst werden.

Personen

Name	Geb.Jahr	Beruf
Peter Meier	1962	Dipl.-Inform.
Johannes Conrad	1928	Dichter
...	...	...

Personen

Name	Geb.Jahr	Beruf
Meier, Peter	62	Informatiker
Johanes Conrad	28	Dichter
...	...	...

Abbildung 9: Heterogenität von Datenbankinstanzen

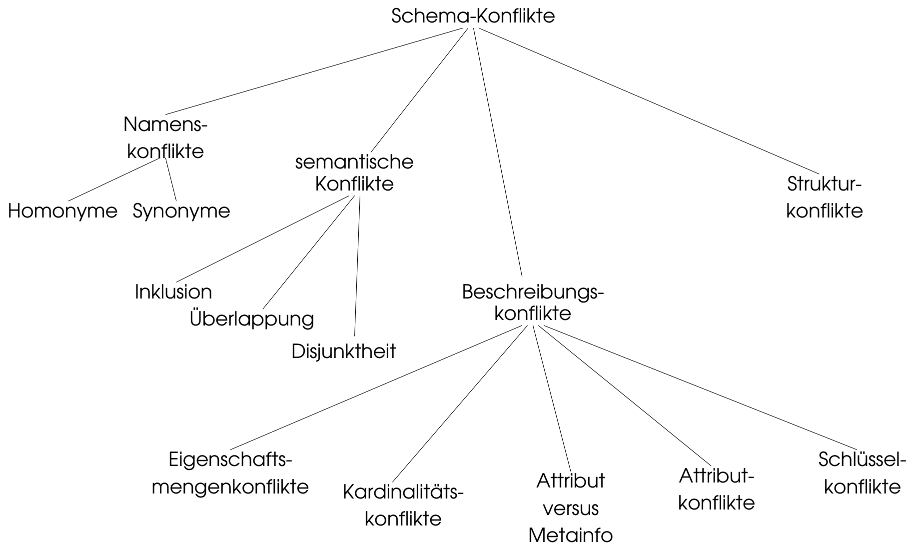

Abbildung 10: Klassifikation von Konflikten

Semantische Konflikte betreffen die Extensionen von Klassen — etwa können die Personen einer Klasse `Person` eines DBMS eine Obermenge der Klasse `Mitarbeiter` eines anderen DBMS sein. Derartige Beziehungen müssen erkannt und explizit gemacht werden, um eine korrekte Integration zu erreichen.

Der Bereich der *Beschreibungskonflikte* faßt eine Reihe von Varianten der Beschreibung von Eigenschaftscharakterisierungen und deren Modellierung zusammen, die bei der Integration aufeinander abgestimmt werden müssen. So können in verschiedenen Datenbanksystemen unterschiedliche Kardinalitäten für Beziehungen oder Schlüsselbedingungen gewählt worden sein.

Als *Strukturkonflikte* bezeichnet man schließlich unterschiedliche Modellierungen desselben Sachverhalts, wie sie z.B. in Abbildung 8 aufgetreten war.

4.3 Integritätssicherung

Die Integritätssicherung in föderierten Datenbanken stellt im Vergleich zu konventionellen Datenbanken einige neue Anforderungen, die Inhalt aktueller Forschungaufgaben sind.

Die Berücksichtigung von Integritätsbedingungen während der Transformationsschritte und der Integration ist ein bisher wenig untersuchter Aspekt beim Entwurf von FDBS [CST97, CST98]. Hierbei ist das Wissen über die Beziehungen zwischen Integritätsbedingungen nützlich [TS98]. In einer Prädikatenlogik formulierte Bedingungen müssen während der Schematransformation ebenfalls umgeformt und in der Integrationsphase verschmolzen bzw. aufgeteilt werden.

Ein weiterer relevanter Aspekt ist die Formulierung und Erzwingung *globaler Integritätsbedingungen*. Globale Bedingungen werden auf jeden Fall benötigt, wenn die Datenbestände der KDBS überlappend sind, also etwa Personendaten in mehreren lokalen Datenbanken gespeichert sind. Hier müssen globale Bedingungen für die Erhaltung der Konsistenz bei Änderungen sorgen.

Eine Möglichkeit der Überwachung von deklarativen Integritätsbedingungen ist ihre Umformung in *aktive Regeln*. Übertragen auf FDBS, können *globale Regeln* auf der Föderierungsebene globale Integritätsbedingungen überwachen. Derartige globale Regeln sind nach folgendem Muster aufgebaut:

```
define rule GlobaleRegel
on potentiell integritätsverletzendes Ereignis
if globale Bedingung (deskriptiv)
do globale Aktions(folge) zur Integritätserzwingung
```

Natürlich ist eine Optimierung derartiger Regeln notwendig, um eine Ausnutzung lokaler Integritätsmechanismen zu ermöglichen. Kritischer Punkt bei diesem Ansatz ist die lokale Erkennung relevanter Ereignisse, die eine globale Bedingung verletzen könnten.

Abbildung 11 zeigt einen Vorschlag für eine Architektur der aktiven Integritätssicherung, in der die Varianten der Erkennung potentiell integritätsverletzender Ereignisse aufzeigt [TC97]. Datenbanksysteme, die selber aktive Mechanismen wie Trigger oder ECA-Regeln unterstützen, können diese Mechanismen zur Alarmierung der Föderierungsebene ausnutzen.

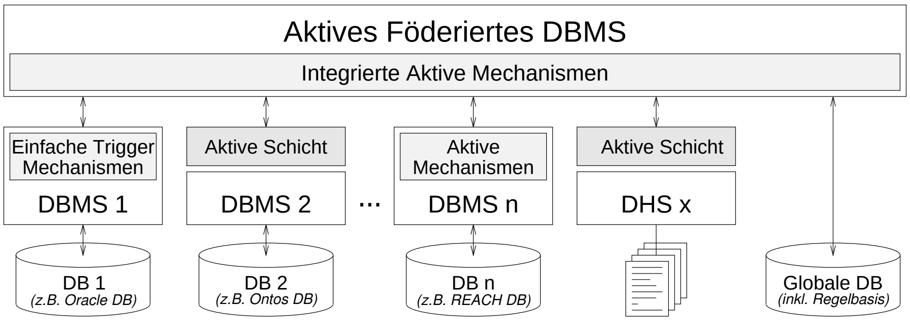

Abbildung 11: Architektur der aktiven Integritätssicherung

Besitzt ein DBMS keine aktiven Mechanismen, oder handelt es sich um ein einfaches Datenhaltungssystem DHS ohne volle Datenbank-Funktionalität, muß dieses Komponentensystem durch eine aktive Schicht eingekapselt werden, die relevante Ereignisse erkennt.

Diese Architektur verdeutlicht, daß eine globale Integritätssicherung ohne Eingriffe in die Autonomie der Komponenten-Datenbanksysteme nicht möglich ist.

4.4 Weitere Aspekte

Föderierte Datenbanksysteme sind ein Forschungsschwerpunkt der Magdeburger Datenbankgruppe. Neben den bisher erläuterten Aspekten gibt es eine Reihe weiterer Forschungsgebiete im Zusammenhang mit FDBS, die zum Teil auch in Magdeburg bearbeitet werden:

- Aufgrund der Autonomie der Komponentensysteme müssen spezielle Transaktionsmodelle entwickelt werden [ST97, STS98a, STS98b, STS98c].
- Durch die Mischung von lokalen und globalen Zugriffen wird ein spezielles Datensicherheitsmodell notwendig [Hil97, HS98].
- Anfragen und Änderungen auf der globalen Ebene müssen in lokale Anfragen umgesetzt und optimiert werden.

Eines der Magdeburger Projekte betrifft den Einsatz von Föderierungsdiensten im Anwendungsbereich der Fabrikplanung [SCC+97, HSC97, Köl97]. Hier hat sich gezeigt, daß in vielen Anwendungsbereichen der Integration von Dateisystemen eine besondere Rolle zukommt [Höd96].

5 Herausforderungen an die Datenbanktechnologie

Eines der Ziele dieser Ausarbeitung ist es, einen Blick in die Zukunft zu wagen und künftige Entwicklungen im Datenbankbereich abzuschätzen — auch wenn derartige Versuche naturgemäß spekulativ sind. Welches sind die Herausforderungen an die Datenbanktechnologie, die die nächsten Jahre prägen werden?

Beginnen wir mit den Herausforderungen aus der Sicht der aktuellen Datenbanktechnologie. Noch immer sind "echte" Objektdatenbanksysteme weit vom Reifestadium entfernt, das relationale Datenbanksysteme erreicht haben. Unter echten ODBMS verstehen wir hierbei Datenbanksysteme, die das Objektparadigma unterstützen, aber auch Datenbankeigenschaften wie logische und physische Datenunabhängigkeit, deskriptive Anfragesprachen, Transaktionsverwaltung im Mehrbenutzerbetrieb, Datenschutz und Datensicherheit unterstützen [SST97, Heu97].

Ein aktuelles Gebiet ist die Integration von *Workflow-Unterstützung* und anderer Ablaufsteuerung in Datenbanksysteme. *Historische und temporale Datenbanken* sind eine Entwicklungsrichting, in der noch viele Punkte offen sind [TCG+93]. Die bereits erwähnten *aktiven Datenbanksysteme* sind ein weiteres aktuelles Forschungsgebiet.

Diese Gebiete zeigen den aktuellen Trend auf, *mehr Semantik und Wissen über die Daten* in die Datenbankbeschreibung zu integrieren. Neben der technischen Realisierung muß gesagt werden, daß die zugehörige Entwurfsmethodik bei weitem nicht ausgereift ist — für derartige Datenbanken fehlen Qualitätsmaße für Datenbankschemata, die für relationale Datenbanken existieren und selbstverständlich eingesetzt werden.

Die konzeptionelle Modellierung von komplexen Informationssystemen ist ebenfalls immer noch nicht in einem Maße ausgereift, das den Einsatz von formalen Methoden und Werkzeugunterstützung beim integrierten Entwurf von Datenbank und Anwendungsfunktionen ermöglicht, wie er bei semantisch reichhaltigen Datenbeschreibungen notwendig wird. Die objektorientierte Spezifikation in Sprachen wie TROLL [Saa93, JSHS96] ermöglicht derartig integrierte konzeptionelle Beschreibungen, ist aber als verhältnismäßig junge Technologie sicher noch nicht für den kommerziellen Einsatz ausgereift.

Weitere Herausforderungen an die Datenbanktechnologie ergeben sich aus Entwicklungen im Datenaufkommen, das in Datenbanksystemen verwaltet werden sollte. Diese Entwicklungen wurden im Kapitel 2 bereits skizziert, so daß an dieser Stelle nur auf die neuen Anforderungen an die Datenbanktechnologie hingewiesen sei.

Die existierenden und noch entstehenden weltweit verteilten Informationsbestände erfordern eine ausgereifte Datenbanktechnologie auf den Gebieten Verteilung und Konsistenzhaltung. Föderierte Datenbanksysteme sind ein Ansatz zur Verwaltung heterogener Datenbestände unter Beibehaltung autonomer Komponenten. Die in diesem Zusammenhang oft genannten *Data Warehouses* verfolgen ein ähnliches Ziel, indem sie verteilte Datenbestände zusammenführen, um übergreifende Analysen zu ermöglichen.

Eine große Herausfordung ist sicher die unabsehbar wachsende Informationsflut im Internet: Das *World Wide Web*. Bereits jetzt bildet das WWW den größten zusammenhängenden Informationsbereich, der jemals verwaltet wurde. Das WWW ist wild gewachsen und dementsprechend heterogen und mit zum Teil inkonsistenten, oft redundanten Informationen gefüllt. Noch ist offen, inwieweit hier bekannte und neu zu entwickelnde Datenbanktechniken zu einer Verbesserung der Nutzung dieses Informationsbestands beitragen können.

Die Kombination Datenbanken und *Mobile Computing* führt ebenfalls zu neuen Problemfeldern. Im Bereich Mobile Computing sind teilnehmende Rechner oft nicht erreichbar, ausgelagerte Datenbestände somit oft längere Zeit nicht verfügbar oder können gar endgültig verloren gehen.

Neben diesen eher allgemeinen Problemstellungen existieren eine Reihe neuer Anwendungen, die spezielle Anforderungen haben, und denen sich die Datenbanktechnologie stellen muß:

- Bildarchive und Multi-Media-Daten stellen eine Herausforderung an Datenbanksysteme bezüglich der Größe der einzelnen Datenobjekte dar.

- Insbesondere Satellitendaten werden in einer konstant hohen Rate ununterbrochen produziert und müssen archiviert und ausgewertet werden.

- Das Genom-Projekt stellt neben hohem Speicherbedarf neue Anforderungen an Ähnlichkeitssuche für Genom-Daten.

- Internet-Server, insbesondere Video-Server, kombinieren die Aspekt der Client-Server-Architektur mit großen Datenobjekten und einem garantierten Quality-of-Service.

- Anwendungen des Data Mining kombinieren statistische Algorithmen mit den datenbankspezifischen Datenstrukturen.

Danksagung

Eine Antrittsvorlesung hat nicht nur einen einzelnen Aspekt zum Inhalt wie übliche Veröffentlichungen. Vielmehr bietet sie einen Überblick über aktuelle und zukünftige Forschungsgebiete einer Professur, und damit der Arbeit einer ganzen Gruppe von Wissenschaftlern. Genannt seien hier die jetzigen und ehemaligen Mitarbeiterinnen und Mitarbeiter, die wesentlichen Einfluß auf die Arbeitsgebiete hatten und haben: Andreas Christiansen, Stefan Conrad, Steffen Geschke, Thorsten Hartmann, Eyk Hildebrandt, Michael Höding, Meike Hollatz, Jan Kusch, Ingo Schmitt, Kerstin Schwarz und Can Türker.

Literatur

[Cat94] R. G. G. Cattell, Herausgeber. *The Object Database Standard: ODMG-93.* Morgan Kaufmann Publishers, San Mateo, CA, 1994.

[CHS+97] S. Conrad, M. Höding, G. Saake, I. Schmitt und C. Türker. Schema Integration with Integrity Constraints. In C. Small, P. Douglas, R. Johnson, P. King und N. Martin, Herausgeber, *Advances in Databases, 15th British National Conf. on Databases, BNCOD 15, London, UK, July 1997*, S. 200–214, Lecture Notes in Computer Science, Band 1271, Springer-Verlag, Berlin, 1997.

[Cod82] E. F. Codd. Relational Database: A Practical Foundation for Productivity. *Communications of the ACM*, 25(2):109–117, Februar 1982.

[Con97] S. Conrad. *Föderierte Datenbanksysteme: Konzepte der Datenintegration.* Springer-Verlag, Berlin/Heidelberg, 1997.

[CST97] S. Conrad, I. Schmitt und C. Türker. Behandlung von Integritätsbedingungen bei Schemarestrukturierung und Schemaintegration. In K. R. Dittrich und A. Geppert, Herausgeber, *Datenbanksysteme in Büro, Technik und Wissenschaft, BTW'97, GI-Fachtagung, Ulm, März 1997*, S. 352–369, Informatik aktuell, Springer-Verlag, Berlin, 1997.

[CST98] S. Conrad, I. Schmitt und C. Türker. Considering Integrity Constraints During Federated Database Design. In S. M. Embury, N. J. Fiddian, A. W. Gray und A. C. Jones, Herausgeber, *Advances in Databases, 16th British National Conf. on Databases, BNCOD 16, Cardiff, Wales, July 1998*, S. 119–133, Lecture Notes in Computer Science, Band 1405, Springer-Verlag, Berlin, 1998.

[Dad96] P. Dadam. *Verteilte Datenbanken und Client/Server-Systeme — Grundlagen, Konzepte, Realisierungsstrukturen.* Springer-Verlag, Berlin, 1996.

[DD93] C. J. Date und H. Darwen. *A Guide to the SQL Standard.* Addison-Wesley, Reading, MA, 3 Auflage, 1993.

[DG96] K. R. Dittrich und S. Gatziu. *Aktive Datenbanksysteme — Konzepte und Mechanismen*, Thomson's Aktuelle Tutorien, Band 13. International Thomson Publishing, Bonn, 1996.

[EGH+92] G. Engels, M. Gogolla, U. Hohenstein, K. Hülsmann, P. Löhr-Richter, G. Saake und H.-D. Ehrich. Conceptual Modelling of Database Applications Using an Extended ER Model. *Data & Knowledge Engineering*, 9(2):157–204, 1992.

[Heu97] A. Heuer. *Objektorientierte Datenbanken: Konzepte, Modelle, Standards und Systeme*. Addison-Wesley, Bonn, 2 Auflage, 1997.

[Hil97] E. Hildebrandt. Sicherheitsaspekte beim Schemaebenenaufbau in Föderierten Datenbanksystemen. In C. Eckert, T. Polle und T. Stülten, Herausgeber, *Kurzfassungen — 9. Workshop "Grundlagen von Datenbanken", Königslutter (20.5.–23.5.1997)*, S. 36–40, Nr. 643. Fachbereich Informatik, Universität Dortmund, 1997.

[Höd96] M. Höding. An Approach to Integration of File Based Systems into Database Federations. In *Heterogeneous Information Management, Prague, Czech Republic, 4–5 November 1996, Proc. of the 10th ERCIM Database Research Group Workshop*, S. 61–71. ERCIM-96-W003, European Research Consortium for Informatics and Mathematics, 1996.

[HS95] A. Heuer und G. Saake. *Datenbanken — Konzepte und Sprachen*. International Thomson Publishing, Bonn, 1995.

[HS98] E. Hildebrandt und G. Saake. User Authentication in Multidatabase Systems. In *9th International Workshop on Database and Expert Systems Applications (DEXA'98), Wien, Austria, August 1998*. IEEE Computer Society Press, 1998. *To appear*.

[HSC97] M. Höding, G. Saake und S. Conrad. Integration heterogener Informationssysteme durch föderierte Datenbanksysteme. *Wissenschaftsjournal der Otto-von-Guericke Universität Magdeburg*, 3(1):29–39, 1997.

[JSHS96] R. Jungclaus, G. Saake, T. Hartmann und C. Sernadas. TROLL – A Language for Object-Oriented Specification of Information Systems. *ACM Transactions on Information Systems*, 14(2):175–211, April 1996.

[Köl97] A. Köller. Dateien in föderierten Datenbanksystemen - Untersuchungen an Systemen zur Fabrikplanung. Preprint Nr. 4, Fakultät für Informatik, Universität Magdeburg, 1997.

[Kus96] J. Kusch. *Entwicklung verteilter Objektstrukturen für skalierbare und hochgradig verfügbare Informationssysteme*, Dissertationen zu Datenbanken und Informationssystemen, Band 9. infix-Verlag, Sankt Augustin, 1996.

[Rah94] E. Rahm. *Mehrrechner-Datenbanksysteme*. Addison-Wesley, Bonn, 1994.

[RS94] E. Radeke und M. H. Scholl. Federation and Stepwise Reduction of Database Systems. In W. Litwin und T. Risch, Herausgeber, *Applications of Databases, Proc. of the 1st Int. Conf., ADB-94, Vadstena, Sweden, June 1994*, S. 381–399, Lecture Notes in Computer Science, Band 819, Springer-Verlag, Berlin, 1994.

[Saa93] G. Saake. *Objektorientierte Spezifikation von Informationssystemen*, Teubner-Texte zur Informatik, Band 6. Teubner-Verlag, Stuttgart, Leipzig, 1993.

[SCC+97] G. Saake, A. Christiansen, S. Conrad, M. Höding, I. Schmitt und C. Türker. Föderierung heterogener Datenbanksysteme und lokaler Datenhaltungskomponenten zur systemübergreifenden Integritätssicherung — Kurzvorstellung des Projekts $\mathbf{SIGMA}_{FDB}$. In K. R. Dittrich und A. Geppert, Herausgeber, *Datenbanksysteme in Büro, Technik und Wissenschaft, BTW'97, GI-Fachtagung, Ulm, März 1997*, S. 322–331, Informatik aktuell, Springer-Verlag, Berlin, 1997.

[Sch98] I. Schmitt. *Schemaintegration für den Entwurf Föderierter Datenbanken*, Dissertationen zu Datenbanken und Informationssystemen, Band 43. infix-Verlag, Sankt Augustin, 1998.

[SL90] A. P. Sheth und J. A. Larson. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, 22(3):183–236, September 1990.

[SP94] S. Spaccapietra und C. Parent. View Integration: A Step Forward in Solving Structural Conflicts. *IEEE Transactions on Knowledge and Data Engineering*, 6(2):258–274, April 1994.

[SPD92] S. Spaccapietra, C. Parent und Y. Dupont. Model Independent Assertions for Integration of Heterogeneous Schemas. *The VLDB Journal*, 1(1):81–126, Juli 1992.

[SS96a] I. Schmitt und G. Saake. Integration of Inheritance Trees as Part of View Generation for Database Federations. In B. Thalheim, Herausgeber, *Conceptual Modelling — ER'96, Proc. of the 15th Int. Conf., Cottbus, Germany, October 1996*, S. 195–210, Lecture Notes in Computer Science, Band 1157, Springer-Verlag, Berlin, 1996.

[SS96b] I. Schmitt und G. Saake. Schema Integration and View Generation by Resolving Intensional and Extensional Overlappings. In K. Yetongnon und S. Hariri, Herausgeber, *Proc. of the 9th ISCA Int. Conf. on Parallel and Distributed Computing Systems (PDCS'96), Dijon, France, September 1996*, S. 751–758, International Society for Computers and Their Application, Six Forks Road, Releigh, NC, 1996.

[SS98] I. Schmitt und G. Saake. Merging Inheritance Hierarchies for Database Integration. In *Proc. of the 3rd IFCIS Int. Conf. on Cooperative Information Systems, CoopIS'98, August 20–22, 1998, New York, USA*, IEEE Computer Society Press, Los Alamitos, CA, 1998.

[SST97] G. Saake, I. Schmitt und C. Türker. *Objektdatenbanken — Konzepte, Sprachen, Architekturen*. International Thomson Publishing, Bonn, 1997.

[ST97] K. Schwarz und C. Türker. Investigating Advanced Transaction Models for Federated Database Systems. Preprint Nr. 7, Fakultät für Informatik, Universität Magdeburg, 1997.

[ST98] I. Schmitt und C. Türker. Refining Extensional Relationships and Existence Requirements for Incremental Schema Integration. In *Proc. of the 7th Conf. on Information and Knowledge Management (CIKM'98), Washington D.C., USA*. ACM Press, 1998. *To appear.*

[STS98a] K. Schwarz, C. Türker und G. Saake. Execution Dependencies in Transaction Closures. In *Proc. of the 3rd Int. Conf. on Cooperative Information Systems (CoopIS'98), August 20–22, 1998, New York City, USA*. IEEE Computer Science Press, 1998.

[STS98b] K. Schwarz, C. Türker und G. Saake. Extending Transaction Closures by N-ary Termination Dependencies. In W. Litwin, T. Morzy und G. Vossen, Herausgeber, *Second East-European Symposium on Advances in Databases and Information Systems (ADBIS'98), September 8-11, 1998, Poznan, Poland*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1998. *To appear.*

[STS98c] K. Schwarz, C. Türker und G. Saake. Transitive Dependencies in Transaction Closures. In B. Eaglestone, B. C. Desai und J. Shao, Herausgeber, *Proc. of the 1998 Int. Database Engineering and Applications Symposium (IDEAS'98), July 8–10, 1998, Cardiff, Wales, UK*, S. 34–43, IEEE Computer Science Press, Los Alamitos, CA, 1998.

[TC97] C. Türker und S. Conrad. Towards Maintaining Integrity of Federated Databases. In *Data Management Systems, Proc. of the 3rd Int. Workshop on Information Technology, BIWIT'97, July 2–4, 1997, Biarritz, France*, S. 93–100, IEEE Computer Society Press, Los Alamitos, CA, 1997.

[TCG+93] A. U. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev und R. Snodgrass. *Temporal Databases — Theory, Design and Implementation.* Benjamin/Cummings, Redwood City, CA, 1993.

[TS98] C. Türker und G. Saake. Deriving Relationships between Integrity Constraints for Schema Comparison. In W. Litwin, T. Morzy und G. Vossen, Herausgeber, *Second East-European Symposium on Advances in Databases and Information Systems (ADBIS'98), September 8-11, 1998, Poznan, Poland*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1998. *To appear.*

[UH97] T. Unger und M. Höding. A Practitioner's View to the Integration of Virtual Enterprises Database Systems by Federation Techniques. In *First East-European Symposium on Advances in Databases and Information Systems, ADBIS'97, St.-Petersburg, September 1997*, S. 356–363, Band 1. Moscow ACM SIGMOD Chapter, 1997.

[WC96] J. Widom und S. Ceri, Herausgeber. *Active Database Systems — Triggers and Rules for Advanced Database Processing.* Morgan Kaufmann Publishers, San Francisco, CA, 1996.