

Hinweis

Die Rechte an der nachfolgenden Arbeit liegen ausschließlich beim Autor.

Die unberechtigte Vervielfältigung, Transformation, Verbreitung und Weitergabe ist, auch in Teilen, unabhängig vom Medium ohne Zustimmung des Autors unzulässig. Korrekte Zitierungen sind zulässig.

Die nachfolgende Arbeit ist als Hausarbeit an der Berufsakademie Berlin erstellt worden. Trotz sorgfältiger Recherche und Überprüfung übernimmt der Autor keinerlei Gewährleistung für die Korrektheit des Inhalts oder für aus der Anwendung evtl. resultierende Schäden.

Die nachfolgende PDF-Datei ist mit Sicherheitsmechanismen geschützt.

Die Umgehung ist nach UrhG und DMCA illegal. Benötigen sie eine andere Version, wenden sie sich bitte an den Autor.

Christoph Hanser

chanser@web.de

22.12.2003

In welche Richtungen werden sich Datenbanksysteme entwickeln?

Hausarbeit

vorgelegt am 08.12.2003

an der

Berufsakademie in der Fachhochschule für Wirtschaft Berlin

Bereich:	<i>Wirtschaft</i>
Fachrichtung:	<i>Wirtschaftsinformatik</i>
Studienjahrgang:	<i>2001</i>
Studienhalbjahr:	<i>5. Semester</i>
von	<i>Christoph Hanser</i>
Ausbildungsbetrieb	<i>Deutsche Post ITSolutions GmbH</i>
Betreuender Dozent:	<i>Prof. Dr. Dorle Linz</i>

INHALTSVERZEICHNIS

Inhaltsverzeichnis.....	III
Abkürzungsverzeichnis.....	IV
1. Einleitung.....	1
2. Trendforschung.....	2
2.1 Trends in der Informationstechnologie.....	2
2.2 Bewertung von Trendaussagen in der Informationstechnologie.....	2
3. Trends im Bereich Datenbanksysteme.....	3
3.1 Datenbankmodelle.....	3
3.1.1 Beschreibung der Modelle.....	4
3.1.2 Ausblick.....	6
3.2 Vernetzung von Datenbanksystemen.....	8
3.2.1 Formen der Datenverteilung.....	8
3.2.2 Notwendigkeit von Interoperabilität.....	9
3.2.3 Datenbankschnittstellen.....	10
3.2.4 Abfragesprachen.....	11
3.2.5 Zusammenfassung.....	13
3.3 Integration contra Diversifikation.....	14
3.3.1 Einbinden von Funktionalität.....	14
3.3.2 Klare Schnittstellen.....	16
3.3.3 Ausblick.....	17
3.4 Weitere Trends.....	18
3.4.1 Data Warehouse und Data Mining.....	18
3.4.2 Deduktive Datenbanksysteme.....	20
3.4.3 Parallele Datenbanksysteme.....	21
4. Zusammenfassung.....	22
Literaturverzeichnis.....	V
Internetverzeichnis.....	VIII
Ehrenwörtliche Erklärung.....	X

ABKÜRZUNGSVERZEICHNIS

BLOB	Binary Large Objects
BSI	Bundesamt für Sicherheit in der Informationstechnik
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CASE	Computer Aided Software Engineering
CTI	Computer Telephony Integration
ebXML	electronic business XML
EDI	Electronic Data Interchange
GIS	Geographic Information System
GQL	Graphical Query Language
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
ODMG	Object Database Management Group
OLAP	Online Analytic Processing
OLTP	Online Transaction Processing
OQL	Object Query Language
QBE	Query By Example
SQL	Structured Query Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XQL	XML Query Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations
XSL-FO	XSL Formatting Objects

1. EINLEITUNG

„Datenbanksysteme spielen eine immer größere Rolle.“¹

„[Datenbanken haben] sicher einen Platz unter den ersten drei ... mit beinahe religiösem Eifer diskutierten Software-Kategorien.“²

„One of the most important application of computers.“³

Das Thema Datenbanksysteme ist vier Jahrzehnte nach der Entwicklung der ersten Systeme aktueller denn je. Datenbanken werden heutzutage im Supermarkt, in Bibliotheken, in Unternehmen und zuhause eingesetzt und reichen von einer Installation auf einem Einzelplatzrechner bis hin zu riesigen verteilten Informationssystemen. Entsprechend wichtig ist die Klärung der Frage, in welche Richtungen sich Datenbanksysteme entwickeln. Um dies zu beantworten, zeigt die vorliegende Hausarbeit Trends auf, die laut Experten in den nächsten Jahren eine große Rolle spielen werden.

Bei der Recherche sehr hilfreich war eine Studie des Bundesamts für Sicherheit in der Informationstechnik, die im Mai 2003 erschienen ist. Sie trägt den Namen „Kommunikations- und Informationstechnik 2010+3“ und untersucht aktuelle Trends in den besagten Bereichen. Bücher, Zeitschriften und Internet unterstützten und erweiterten Inhalte der Studie umfangreich.

Inhaltlich gliedert sich die Hausarbeit wie folgt:

- In Kapitel 2 wird zum besseren Verständnis der Thematik kurz erklärt, was eine Tendenz zu einem Trend macht. Es folgt eine Auflistung von Trends in der Informationstechnologie und eine Untersuchung, mit welchem Horizont hier überhaupt seriöse Aussagen getroffen werden können.
- Kapitel 3 umfasst den Hauptteil der Arbeit. Hier werden die Trends in den Technologiegebieten Datenbankmodelle, Vernetzung von Datenbanksystemen und Integration bzw. Diversifikation aufgezeigt. Es schließt sich ein Überblick über weitere wichtige Trends an.
- Abschließend folgt in Kapitel 4 eine Zusammenfassung der Thematik.

¹ Vgl. Eickler, A. / Kemper, A. (2001), S. 15.

² Schulz, H. / Siering, P. (2003), S. 142.

³ Vgl. Blaźewicz, J. / Morzy, T. (2003), S. 2.

2. TRENDFORSCHUNG

Aufgabe der Trendforschung ist die Identifizierung von Dynamiken, welche die bisherigen Strukturen in Frage stellen.⁴ Diese Dynamiken werden „Trends“ genannt. Sie lassen sich darin charakterisieren, dass sie evolutionär, langfristig und nachhaltig wirken.⁵ Um als solcher anerkannt zu werden, muss ein Trend daher eine Entwicklungsgeschichte aufweisen und seine Wirkung darf nicht von kurzer Dauer sein.

2.1 Trends in der Informationstechnologie

Zur Informationstechnologie gehören alle Vorgänge, die mit der Erfassung, Speicherung, Übertragung und Transformation von Daten zusammenhängen.⁶ Neben Datenbanken zählt man auch Softwareentwicklung, Hardware und Netzwerke zu den Teilgebieten der Informationstechnologie. Als übergreifende Trends in diesen Bereichen werden folgende Tendenzen ausgemacht, die zwar nicht alle im Bereich Datenbanksysteme entscheidend sind, allerdings einen guten Überblick über die Thematik bieten:⁷

- *Automatisierung und Vereinfachung* zur Effizienzsteigerung,
- *Integration und Standardisierung* zur Verzahnung von Netzen und Daten,
- *Kapazitäts- und Leistungssteigerung*, um die wachsenden Anforderungen zu bewältigen,
- *Vernetzung und Flexibilisierung* dienen der Verknüpfung verschiedener Plattformen,
- *Verteilung und Dezentralisierung* zur Förderung von Sicherheit und Skalierbarkeit,
- *Virtualisierung* zur Nutzung von Methoden nicht realer Objekte, z. B. im Internet.

2.2 Bewertung von Trendaussagen in der Informationstechnologie

Trendforschung in der Informationstechnologie ist ein relativ unsicheres Gebiet, da sich hier das Wissen besonders schnell erneuert. Im Gegensatz zu traditionellen Bereichen ist die Welt der Informationstechnologie sehr schnelllebig. Als Beispiel soll hier das Internet gelten, das sich seit

⁴ Vgl. Horx, M. / Wippermann, P. (1996), S. 19.

⁵ Vgl. Haderer, T. / Arentzen, U. (2000), Stichwort „Trend“.

⁶ Vgl. Hansen, H. R. / Neumann, G. (2001), S. 11.

⁷ Vgl. Alkassar, A. et al. (2003), S. 328.

1992 in Europa etablierte.⁸ Mittlerweile ist das Internet kaum aus dem Alltag wegzudenken, und nach neusten Umfragen ist jeder Deutsche täglich durchschnittlich 49 Minuten online.⁹

Dies zeigt, dass seriöse Trendaussagen im IT-Bereich nur mit Vorsicht gemacht werden können. Über das nächste Jahrzehnt hinaus kann nur spekuliert werden. Entsprechend wird diese Hausarbeit sich auf die Entwicklung in den nächsten Jahren beschränken und dabei sorgfältig die verschiedenen Expertenmeinungen vergleichen.

3. TRENDS IM BEREICH DATENBANKSYSTEME

Die Thematik der Datenbanksysteme gliedert sich in verschiedene Technologiefelder wie z. B. Datenbankmodelle, Vernetzung und Integration. Nach einer kurzen Einführung in diese Sachgebiete und eventuell konkurrierende Verfahren erfolgt eine Argumentation, welche Trends sich warum durchzusetzen scheinen.

3.1 Datenbankmodelle

Jedes Datenbankmanagementsystem basiert auf einem Datenbankmodell. Dieses Modell besteht aus Konzepten, mit deren Hilfe ein Datenmodell für eine Anwendung erzeugt wird.¹⁰ Die Datenbankmodelle stellen wohl eines der zur Zeit am meisten diskutierten Themen der Datenbankforschung dar.

In den 70er Jahren wurden die hierarchischen und Netzwerkdatenmodelle entwickelt, diese wurden ein Jahrzehnt später von den relationalen Datenbankmodellen abgelöst. Seit Mitte der 90er Jahre existieren die objektorientierten und objektrelationalen Datenbankmodelle.¹¹ Jüngere so genannte postrelationale Modelle¹² wurden entworfen und werden im Folgenden erklärt. Dabei werden die so genannten Legacy-Modelle¹³ – hierarchische und Netzwerkdatenbanken – ausgeklammert, da diese nur noch in Randbereichen eingesetzt werden.

⁸ Vgl. Münz, S. (2001), Abschnitt „Das Netz der Netze“.

⁹ Vgl. IDG Interactive (2003).

¹⁰ Vgl. Pernul, G. / Unland, R. (2003), S. 11.

¹¹ Vgl. Vossen, G. (2000), S. 6.

¹² Vgl. Bager, J. (1999), Glossarstichwort „Postrelational“.

¹³ Vgl. Elmasri, R. / Navathe, S. B. (2002), S. 468.

3.1.1 Beschreibung der Modelle

Das **relationale Modell** repräsentiert die Datenbank als Sammlung von Relationen. Diese werden als Tabellen abgebildet, so dass gespeicherte Daten übersichtlich dargestellt werden können.¹⁴ Dieses Modell genügt bisher den meisten Anwendungen – laut Halpin 90 Prozent¹⁵ –, so dass die relationalen Datenbanksysteme das Technologiefeld als etabliertes Konzept dominieren. Durch das Aufkommen neuer Anforderung, wie z. B. die Speicherung von Multimedia, Dokumenten oder graphischen Formaten, stößt das relationale Modell allerdings an seine Grenzen.¹⁶ Diese Formate können nicht oder nur umständlich segmentiert und atomar dargestellt werden, weshalb sich relationale Datenmodelle für die Speicherung von komplexen Objekten disqualifizieren.¹⁷

Das **objektorientierte Modell** vereint die Paradigmen von Objektorientierung und Datenbanksystemen. Eigenschaften wie Komplexität, Objektidentität, Datenkapselung, Typen und Klassen, Vererbung, Polymorphismus, Vollständigkeit und Erweiterbarkeit werden mit den Prinzipien Dauerhaftigkeit, Verwaltung großer Datenbestände, Mehrbenutzerbetrieb, Rekonstruierbarkeit und Ad-hoc-Abfragemöglichkeit verknüpft.¹⁸ Dadurch können komplexe Objekte direkt in einer Datenbank gespeichert werden und müssen nicht mehr in flache Tabellen „gepresst“ werden, was die Zusammenarbeit mit objektorientierten Programmiersprachen sehr vereinfacht.¹⁹

Anwendungsgebiete sind demnach Multimedia-Datenbanken mit Ton-, Bild- und Videodaten, geographische Informationssysteme und Entwurfsdatenbanken.²⁰ Als Nachteil zeigt sich, dass die objektorientierten Produkte keinem universellen Datenmodell folgen und durch ihre Komplexität sehr kostspielig sind. Fehlende Standards und Erfahrungswerte werden als weitere Mängel genannt.²¹

Das **objektrelationale Datenmodell** will die Vorteile beider Datenmodelle kombinieren. Dabei ist der Ansatz, relationale Datenbanksysteme um Objekte zu erweitern, erfolgreicher als die Ergänzung von relationalen Prinzipien in objektorientierte Datenbanksysteme.²² Mit dem

¹⁴ Vgl. Elmasri, R. / Navathe, S. B. (2002), S. 226.

¹⁵ Vgl. Halpin, T. (2001), S. 692.

¹⁶ Vgl. Kemper, A. / Eickler, A. (2001), S. 354.

¹⁷ Vgl. Erbs, H.-E. et al. (2003), S. 133.

¹⁸ Vgl. Meier, A. / Wüst, T. (2000), S. 5.

¹⁹ Vgl. Zhang, W. / Ritter, N. (2001), S. 228.

²⁰ Vgl. Rolland, F. (2003), S. 154.

²¹ Vgl. Begg, C. et al. (1999), S. 783f.

²² Vgl. Silberschatz, A. et al. (1999), S. 272.

SQL3-Standard wurde das relationale Datenmodell u. a. um große Objekte (z. B. für Multimedia), geschachtelte Relationen, Referenzen und Objektidentität erweitert, so dass es möglich ist, Objekte innerhalb einer Relation zu speichern.²³

Entsprechend bieten sich objektrelationale Datenmodelle durchaus als Alternative für relationale Datenmodelle an und können in den gleichen Bereichen eingesetzt werden. Gegenüber den objektorientierten Datenbanksystemen haben Sie den Vorteil, auch in Tabellenform vorliegende Daten effizient speichern zu können.

Geschäftsdaten erstrecken sich meistens über mehrere Dimensionen, wie z. B. die Zeit, das Produkt und die Abteilung. Zur Speicherung wurde deshalb das **multidimensionale Datenmodell** entworfen, welches die Daten in mehrdimensionalen Matrizen und Quadern – sogenannten „Cubes“ – ablegt.²⁴ Es wird in Data Warehouse-Produkten eingesetzt und vereinfacht das Abfragen von Daten, das Generieren von Berichten und das Durchführen von Analysen. Somit können Geschäftsdaten effizienter als mit anderen Datenmodellen gespeichert werden.²⁵

Das **semistrukturierte Datenmodell** ist auf die Speicherung von semistrukturierten Daten ausgerichtet. Darunter versteht man Daten mit einer Struktur, die variieren kann²⁶, selbstbeschreibend und indikativ ist und auch nur partiell vorhanden sein muss. Damit grenzt sich das semistrukturierte Datenmodell von den anderen Datenmodellen ab, die streng strukturiert sind.²⁷ Dies prädestiniert das Modell für die Speicherung von XML-Dokumenten.

Für die Datenbank-Integration von XML gibt es zwei Ansätze. Die führenden Datenbankhersteller integrieren eine *logische XML-Schnittstelle*, um XML zu lesen und auszugeben.²⁸ Stark strukturierte XML-Dokumente können auch relativ einfach übernommen werden. Sie werden entweder als BLOB²⁹ gespeichert oder durch so genannte Mapping-Tools integriert.³⁰ Dem gegenüber steht der Ansatz der *nativen XML-Datenbanken*. Diese werden immer beliebter, da sie XML nicht nur lesen und schreiben, sondern auch verwalten. Somit gibt es beim Integrieren von XML-Daten keinen Verlust. Native XML-Datenbanken nutzen die Dokumenttypdefinitionen als Schema und bilden davon eine Instanz, so dass die Verwaltung von XML viel einfacher als

²³ Vgl. Kemper, A. / Eickler, A. (2001), S. 392.

²⁴ Vgl. Dorndorf, U. / Pesch, E. (2003), S. 416.

²⁵ Vgl. Begg, C. et al. (1999), S. 910.

²⁶ Die Variation erklärt sich dadurch, dass XML-Elemente fehlen oder hinzukommen können.

²⁷ Vgl. Kazakos, W. et al. (2002), S. 28.

²⁸ Ebenda, S: 284.

²⁹ Eine Speicherung als BLOB kann einfach implementiert werden, erschwert allerdings eine Durchsuchung.

³⁰ Vgl. Mittermeier, L. (2003), S. 43.

bei herkömmlichen Datenbanksystemen möglich ist.³¹ Darüber hinaus eignet sich XML ebenfalls zur Speicherung stark strukturierter Daten, so dass XML-Datenbanksysteme durchaus Potential haben.³² Die Nachteile von nativen XML-Datenbanken liegen darin, dass sie sich noch im Entwicklungsstadium befinden und noch nicht auf die Erfahrungswerte und das Marktgewicht von bestehenden Datenbanksystemen zurückgreifen können.³³ Auch sollen sehr große Datenmengen einen unverhältnismäßig hohen Rechenaufwand erzeugen und erhöhen so die Hardwarekosten.³⁴

3.1.2 Ausblick

Die Studie des BSI bekräftigt, dass sich die Bedeutung der Datenmodelle im Wandel befindet. Die folgende Abbildung stellt das Umfrageergebnis bezüglich der Bedeutung der Datenmodelle dar:

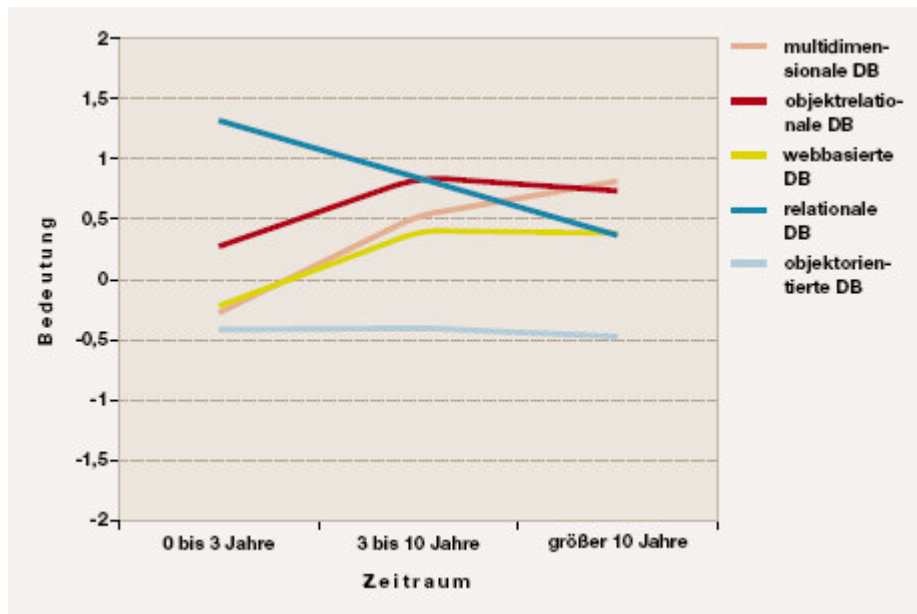


Abbildung 1: Bedeutung unterschiedlicher Datenmodelle

Quelle: Kommunikations- und Informationstechnik 2010+3 (2003), S. 220.

Laut der Studie werden die **relationalen Datenbanken** zwar in den nächsten Jahren noch dominieren, langfristig aber von modernen Datenbanksystemen abgelöst.

³¹ Vgl. Hansen, H. R. / Neumann, G. (2001), S. 1093.

³² Vgl. Kazakos, W. et al. (2002), S. 31.

³³ Vgl. Mittermeier, L. (2003), S. 52.

³⁴ Vgl. Kramer, Christian (2001), S. 74.

Nur einige Experten wie z. B. Meier³⁵ sehen in den rein **objektorientierten Systemen** einen Trend. Indes werden sie sich wohl eher in wissenschaftlichen Spezialgebieten als Nischenprodukt platzieren können.³⁶

Der Erfolg der **objektrelationalen Datenbanken** begründet sich damit, dass sie in der Lage sind, sowohl stark strukturierte Daten als auch komplexe Objekte zu speichern. Fast alle Experten – z. B. Elmasri³⁷ und Stonebreaker³⁸ – sind der Meinung, dass gerade die Verwaltung dieser komplexen Objekte entscheidend für den Erfolg eines Datenmodells ist. Immer wichtiger wird die Speicherung von Multimedia-Inhalten, geographischen Informationen, Konstruktionszeichnungen und anderen Formaten. Somit scheint die Ablösung relationaler Systeme eine Frage der Zeit zu sein.³⁹

Die **multidimensionalen Datenmodelle** werden weiterhin in Spezialanwendungen mit großen Datenmengen dominieren. Da ihre Entwicklung allerdings noch nicht so weit fortgeschritten ist, werden ihnen nur langfristig große Chancen eingeräumt.⁴⁰ Da Data Warehouse und Data Mining allgemein als Trend anerkannt sind,⁴¹ wird auch die Bedeutung des multidimensionalen Datenmodells mit Sicherheit wachsen.

Laut der BSI-Studie werden die **semistrukturierten Datenmodelle**, in der Abbildung „webbasierte Datenbanksysteme“ genannt, an Gewicht zunehmen.⁴² Hier gibt es allerdings widersprüchliche Expertenmeinungen, inwieweit dieses neue Datenmodell das Technologiefeld der Datenbanksysteme verändern wird. Sicher ist, dass es sich hierbei ebenso wie bei dem objektrelationalen Datenmodell um einen Trend handelt. Dies zeigt sich darin, dass in diesem Jahr viele Publikationen zum Thema XML und Datenbanken erschienen sind. Und einige Experten behaupten sogar, dass der Einfluss von XML auf die Datenbanktechnologie größer sein wird als der von SQL⁴³ und dem objektorientierten Datenmodell⁴⁴. Unsicher ist allerdings, ob sich das semistrukturierte Datenmodell nur als logische Schnittstelle oder als natives System durchsetzen wird. Ei-

³⁵ Vgl. Meier, A. / Wüst, T. (2000), S. VII.

³⁶ Vgl. Alkassar, A. et al. (2003), S. 201.

³⁷ Vgl. Elmasri, R. / Navathe, S. B. (2002), S. 469.

³⁸ Vgl. Devarakonda, R. S. (2002).

³⁹ Vgl. Meier, A. / Wüst, T. (2000), S. VII.

⁴⁰ Vgl. Alkassar, A. et al. (2003), S. 201.

⁴¹ siehe Kapitel 3.4.1 „Data Warehouse und Data Mining“.

⁴² Vgl. Alkassar, A. et al. (2003), S. 220.

⁴³ Vgl. Kazakos, W. et al. (2002), S. VII.

⁴⁴ Vgl. Udell, J. (2003), S. 1, Einleitung.

nige Experten wie Udell⁴⁵ oder Helmer⁴⁶ befürworten native Systeme, weil sie zur XML-Speicherung besser geeignet sind. Zur gleichen Zeit sind die großen Datenbankhersteller Oracle, IBM und Microsoft bemüht, XML so gut wie möglich zu integrieren.⁴⁷ Zusammenfassend lässt sich sagen, dass native XML-Datenbanken den renommierten Systemen voraussichtlich in Teilgebieten Marktanteile abnehmen, aber sie nicht ablösen werden.

3.2 Vernetzung von Datenbanksystemen

Ein weiteres brisantes Thema sind vernetzte Datenbanksysteme. Diese bestehen aus Informationseinheiten, die auf mehreren Rechnern verteilt sind, welche über ein Kommunikationsnetzwerk miteinander verbunden sind. Dabei sind die einzelnen Rechner autonom, haben aber die Aufgabe, andere Rechner mit Daten zu versorgen.⁴⁸ Die Idee der dezentralen Speicherung existiert schon seit Mitte der 70er Jahre, allerdings handelte es sich hierbei meist um gleichartige Datenbanksysteme.⁴⁹ Für unterschiedliche Datenbanksysteme fällt die Ist-Analyse eher negativ aus: es gibt bisher kaum Systeme, die Daten aus verschiedenartigen Quellen optimal integrieren.⁵⁰ Das Schlagwort „Interoperabilität“ steht in diesem Zusammenhang für die Fähigkeit unterschiedlicher Datenbanksysteme, miteinander zu kommunizieren.⁵¹

Im Folgenden wird untersucht, welche Formen der Datenverteilung es gibt, warum die Zusammenführung verschiedenartiger Datenbanken einen Trend darstellt und welche Voraussetzungen notwendig sind, um Interoperabilität zu ermöglichen.

3.2.1 Formen der Datenverteilung

Für die Verteilung der Daten gibt es verschiedene Formen, die vom Homogenitätsgrad abhängig sind. Zum einen gibt es **verteilte Datenbanksysteme**, die homogen und heterogen vorliegen können. In *homogenen Datenbanksystemen* verwenden Clients und Server das gleiche Datenbankmanagementsystem. Dabei werden die Inhalte auf verschiedene Systeme fragmentiert. In *heterogenen Datenbanksystemen* werden Datenbanken verschiedener Hersteller über

⁴⁵ Vgl. Udell, J. (2003), S. 2, Abschnitt „The Future of Documents“.

⁴⁶ Vgl. Helmer, S. et al. (2003), S. 142.

⁴⁷ Vgl. Schmitz, L. (2002).

⁴⁸ Vgl. Kemper, A. / Eickler, A. (2001), S. 443.

⁴⁹ Vgl. Hansen, H. R. / Neumann, G. (2001), S. 1082.

⁵⁰ Vgl. Alkassar, A. et al. (2003), S. 207.

⁵¹ Vgl. Elmasri, R. / Navathe, S. B. (2002), S. 217.

Gateways miteinander verbunden.⁵² Von verteilten Datenbanksystemen erwartet man eine höhere Zuverlässigkeit und Verfügbarkeit der Daten. Sollte ein Datenknoten ausfallen, können weitere Knoten den Anfrager mit Daten versorgen. Des Weiteren bieten sie eine bessere Performance, höhere Skalierbarkeit und große Transparenz.⁵³ In heterogenen Umgebungen können die Stärken der einzelnen Datenbanksysteme je nach Anwendung kombiniert werden.⁵⁴ Auch wird in der Datenbankforschung oft von dem Problem der „Wissensinseln“ gesprochen. Durch die lokale Speicherung des Wissens und deren Vernetzung sollen diese zusammengeführt werden.⁵⁵ Die Nachteile liegen in den höheren Kosten und vor allem in der Komplexität, wodurch leicht Fehler entstehen können.⁵⁶

Neben den verteilten Datenbanksystemen gibt es die **föderierten Datenbanksysteme**. Sollte bei den verbundenen Datenbanken das Datenmodell abweichen, z. B. wenn hierarchische und relationale Datenbanken miteinander verbunden werden, muss ein föderiertes Datenbankschema erstellt werden, welches auf einer abstrakten Ebene die zugänglichen Knoten beschreibt.⁵⁷ Die Verknüpfung findet hierbei über eine zusätzliche Softwareschicht statt, über welche die Datenbanksysteme miteinander kommunizieren.⁵⁸ Aufgrund ihrer lokalen Autonomie werden die föderierten Datenbanken oft als Mischform zwischen zentralen und verteilten Datenbanken gesehen.⁵⁹ In föderierten Datenbanksystemen geht es weniger um Vorteile wie Performancesteigerung oder Verfügbarkeit, als viel mehr um die grundlegende Notwendigkeit, Datenbanken miteinander zu verbinden.

3.2.2 Notwendigkeit von Interoperabilität

Das Thema Interoperabilität wird immer bedeutender. In den letzten Jahren wurden zunehmend verschiedenartige Datenbanksysteme miteinander verbunden. Dabei ist keine Trendwende abzusehen, im Gegenteil: die reibungslose Vernetzung von Datenbanken über ein föderiertes Schema wird immer wichtiger. Das zeigt sich am deutlichsten im Internet, wo mehr und mehr Datenbanken verknüpft werden. Selbst in einem Großunternehmen bleibt es eine Illusion, alle Daten in

⁵² Vgl. Rolland, F. (2003), S. 223f.

⁵³ Vgl. Elmasri, R. / Navathe, S. B. (2002), S. 825.

⁵⁴ Vgl. Alkassar, A. et al. (2003), S. 204.

⁵⁵ Vgl. Begg, C. et al. (1999), S. 652.

⁵⁶ Vgl. Silberschatz, A. et al. (1999), S. 626.

⁵⁷ Vgl. Rolland, F. (2003), S. 223f.

⁵⁸ Vgl. Silberschatz, A. et al. (1999), S. 622.

⁵⁹ Vgl. Begg, C. et al. (1999), S. 662.

einem einzigen System integrieren zu können.⁶⁰ Experten wie Halpin⁶¹, Carr⁶² und Sattler⁶³ schließen sich deshalb der Meinung an, dass heterogene Datenbanken im Mittelpunkt aktueller Forschungen stehen und somit einen Trend darstellen.

Für die reibungslose Kommunikation zwischen Datenbanksystemen sind folgende wesentlichen Voraussetzungen notwendig:⁶⁴

- Die Kommunikation muss über einen *Kanal* und ein *Protokoll* gesichert sein.
- *Anfragen* müssen generiert werden können.
- Ein bekanntes *Datenformat* muss eingesetzt werden.
- Die Daten müssen *semantisch verstanden* werden können.

Im Folgenden werden Entwicklungen in den Bereichen Datenbankschnittstellen und Datenbanksprachen untersucht, weil diese für die Erfüllung dieser Voraussetzungen am Wichtigsten sind. Ein Ausblick erfolgt direkt in den Kapiteln und eine Zusammenfassung schließt sich an.

3.2.3 Datenbankschnittstellen

Noch hat sich keine der standardisierten Datenbankschnittstellen auf dem Markt vollständig durchgesetzt. Zwar wird **EDI** bisher überwiegend zum Datentransfer genutzt, und die ISO hat Syntax-Regeln, Nachrichtentypen, deren Datenelemente und Dienste für Nachrichten normiert.⁶⁵ Allerdings ist die Implementierung von EDI mit hohen Installationskosten aufgrund von Software- und Hardwareanpassungen und mit Online-Gebühren verbunden, weshalb nur große Firmen EDI einsetzen.⁶⁶

So geht auch im Bereich der Interoperabilität der Trend in Richtung XML. Am meisten wird hierbei über **ebXML** diskutiert. Diese Schnittstelle ist wesentlich preiswerter zu integrieren und somit auch für kleine und mittelständische Unternehmen finanzierbar. Es müssen nur geringe Umprogrammierungen von Applikationen stattfinden, weil die kommunizierenden Geschäftsprozesse der Partner auf einfache Weise veröffentlicht werden.⁶⁷ Im Gegensatz zu EDI ermöglicht

⁶⁰ Vgl. Härder, T. / Rahm, E. (1999), S. 535.

⁶¹ Vgl. Halpin, T. (2001), S. 702.

⁶² Vgl. Carr, E. (2001), Abschnitt „Trend 1: More data in more places“.

⁶³ Vgl. Sattler, K.-U. (2003), S. 3.

⁶⁴ Vgl. Klischewski, R. (2003), S. 3.

⁶⁵ Vgl. Stickel, E. (1997), S. 216.

⁶⁶ Vgl. Oebbeke, Alfons (2003), Abschnitt „EDI wird durch Internet und XML erschwinglich“.

⁶⁷ Vgl. Jenz, D. E. (2002), Absatz 1.

XML eine flexible Spezifikation der Austauschformate und -bedingungen, so dass die Prozesse leicht angepasst und verbunden werden können.⁶⁸

Derzeit gibt es Bemühungen der Standardisierungsgruppen hinter ebXML und EDI, welche darauf abzielen, beide Protokolle zu ergänzen, damit existierende EDI-Applikationen nicht umprogrammiert werden müssen.⁶⁹ Trotzdem gehen die in der BSI-Studie befragten Experten davon aus, dass bereits 2006 EDI durch ebXML abgelöst wird.⁷⁰ Und auch viele Autoren – Jenz⁷¹ und Halpin⁷² u. a. – sehen das genauso.

3.2.4 Abfragesprachen

Eine weitere wichtige Voraussetzung für Interoperabilität sind die Abfragesprachen. Momentan dominiert SQL als Sprache für relationale und objektrelationale Datenbanksysteme das Technologiefeld. Daneben gibt es noch weitere Sprachen wie z. B. XSL, XQL, OQL, GQL und QBE.

Mit Hilfe von **SQL** können Tabellen eingerichtet werden und Werte eingegeben, verändert und ausgegeben werden. Für die relationalen Systeme ist SQL de facto zum Industriestandard geworden.⁷³ Neben SQL existieren für relationale Datenbanken zwar Sprachen wie QBE, Quel und Datalog, die allerdings kaum eingesetzt werden.⁷⁴ Seit 1986 wird SQL permanent weiter entwickelt: 1998 wurde SQL in der Version 3 um objektorientierte Elemente erweitert, so dass SQL auch in objektrelationalen Datenbanksystemen eingesetzt werden kann. In SQL4, welches Ende 2003 erscheinen soll, wird neben anderen Funktionen auch die Verwaltung von XML und externen Daten integriert sein. In bezug auf XML werden Funktionen zur Generierung von XML, ein XML-Datentyp und Abbildungen von SQL zu XML-Dokumenten angeboten, so dass SQL nun XML lesen und erzeugen kann.⁷⁵ Diese in den letzten fünf Jahren integrierten Funktionalitäten – Objekte, XML und externe Daten – haben dafür gesorgt, dass SQL nun einfacher in objekt- und XML-orientierten Umgebungen eingesetzt werden kann.

⁶⁸ Vgl. Alkassar, A. et al. (2003), S. 208.

⁶⁹ Vgl. Meehan, Michael (2001).

⁷⁰ Vgl. Alkassar, A. et al. (2003), S. 224.

⁷¹ Vgl. Jenz, D. E. (2002), Absatz 4.

⁷² Vgl. Halpin, T. (2001), S. 702.

⁷³ Vgl. Rolland, F. (2003), S. 99.

⁷⁴ Vgl. Silberschatz, A. et al. (1999), S. 188.

⁷⁵ Vgl. Michels, J.-E. et al. (2003), S. 32 ff.

Als nachteilig gilt, dass der SQL-Standard XML nicht gut genug einbindet, wobei an einer Lösung weiter gearbeitet wird.⁷⁶ Auch ist zweifelhaft, ob sich die Objektorientierung aus SQL3 durchsetzen kann, da SQL als einfache relationale Sprache kaum performant komplexe Objekte verwalten kann.⁷⁷

Von der ODMG wurde die Abfragesprache **OQL** standardisiert, die das Objektmodell voll unterstützt.⁷⁸ Seit SQL3 um Objekte erweitert wurde, wird OQL allerdings nur in rein objektorientierten Datenbankmanagementsystemen eingesetzt.⁷⁹ Um Interoperabilität zu unterstützen, wird OQL stark an SQL angelehnt. So ist es durchaus möglich, dass dieses SQL nur erweitern wird, anstatt sich als eigenständige Sprache durchzusetzen. Die Verschmelzung gestaltet sich allerdings aufgrund der verschiedenartigen Ausrichtung auf Tabellen bzw. Objekten als kompliziert.⁸⁰

Im Bereich der XML-Abfragesprachen wurden verschiedene Varianten entwickelt. Dabei wird **XQL** bereits kommerziell eingesetzt.⁸¹ Als reine Abfragesprache arbeitet es bei der Weiterverarbeitung und Präsentation der Anfragen mit **XSL** zusammen. XSL gliedert sich in die Teilbereiche XSLT für die Transformation, XPath, mit dessen Hilfe auf Teile eines XML-Baums zugegriffen wird, und XSL-FO für das Layout.⁸² XPath als Teilaspekt von XSL und damit XQL weist allerdings Defizite bei der XML-Anfrage auf.

Deshalb wurde vom W3C die Abfragesprache **XQuery** entwickelt, die sich sowohl um Anfragen als auch um die Präsentation kümmert. XQuery soll darüber hinaus nicht nur im Datenbank-Sektor eingesetzt werden, sondern auch zur Datenintegration und Dokumentenverarbeitung, z. B. bei der Volltextsuche.⁸³ Es wird davon ausgegangen, dass XPath die Techniken XQL und XSL bald in nativen XML-Datenbanksystemen und als logische Schnittstelle ablöst⁸⁴. Entsprechend wird XPath als zukunftssträchtige Abfragesprache gehandelt.⁸⁵

Die BSI-Studie kam bei der Frage, welche Abfragesprache sich in den nächsten Jahren durchsetzen wird, auf folgende Antwort:

⁷⁶ Vgl. Kazakos, W. et al. (2002), S. 101.

⁷⁷ Vgl. Pernul, G. / Unland, R. (2003), S. 258.

⁷⁸ Vgl. Meier, A. / Wüst, T. (2000), S. 76.

⁷⁹ Ebenda, S. 172.

⁸⁰ Vgl. Kemper, A. / Moerkotte, G. (2003), S. 106.

⁸¹ Vgl. Kazakos, W. et al. (2002), S. 79.

⁸² Ebenda, S. 107.

⁸³ Ebenda, S. 80.

⁸⁴ Vgl. Deutsche Informatik Akademie (2003), S. 2, Abschnitt „Zugriff auf XML in Datenbanken“.

⁸⁵ Vgl. Udell, J. (2003), S. 2, Abschnitt „Query Strategies“.

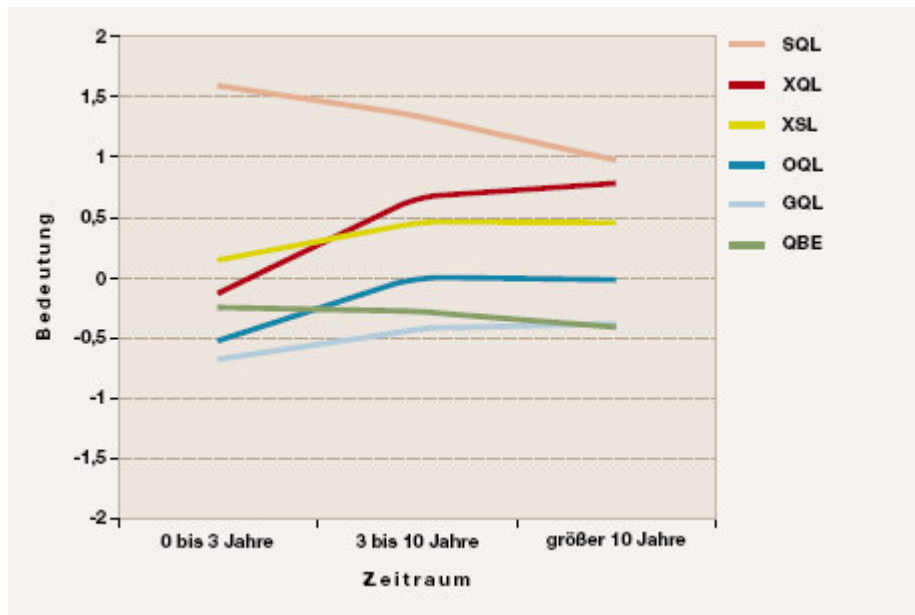


Abbildung 2: Bedeutung unterschiedlicher Datenbank-Abfragesprachen

Quelle: Kommunikations- und Informationstechnik 2010+3 (2003), S. 212.

Es ist demnach davon auszugehen, dass SQL seine Vormachtstellung behalten wird, während die XML-Sprachen gleichzeitig an Gewicht gewinnen. SQL wurde schon früh von Stonebreaker als „Intergalactic Dataspeak“⁸⁶ bezeichnet, und dies scheint sich zu bewahrheiten: Durch die Anpassungen der letzten Jahre an neue Trends wird SQL weiter die meistgenutzte Sprache bleiben.⁸⁷

Die XML-Sprachen erfüllen zwar noch nicht alle Anforderungen und es fehlen praktische Erfahrungen, dennoch werden diese Sprachen im Zuge der immer bedeutenderen Stellung von XML als Datenaustausch- und Datenbeschreibungsformat wichtiger.⁸⁸

Auch OQL scheint an Bedeutung zu gewinnen, v. a. weil die Objektivimplementierung in SQL noch nicht optimal umgesetzt ist.⁸⁹ Allerdings wird sich die Verbreitung durch die Weiterentwicklung von SQL und XQuery auf die objektorientierte Datenwelt beschränken.

3.2.5 Zusammenfassung

Bei der Vernetzung von Datenbanksystemen werden die heterogenen und föderierte Datenbanksysteme eine immer größere Rolle spielen. Um für die notwendige Interoperabilität zur Kommunikation zwischen den verschiedenartigen Datenbanksystemen zu sorgen, wird ebXML als

⁸⁶ Stonebreaker, Michael, zitiert nach Michels, J.-E. et al. (2003), S. 38.

⁸⁷ Vgl. Michels, J.-E. et al. (2003), S. 38.

⁸⁸ Vgl. Kazakos, W. et al. (2002), S. 105.

⁸⁹ Vgl. Pernul, G. / Unland, R. (2003), S. 258.

Schnittstelle EDI ablösen. Bei den Abfragesprachen wird die Weiterentwicklung von SQL und XQuery forciert. Wegen der großflächigen Verbreitung von SQL werden sich die XML-Sprachen allerdings nur langfristig durchsetzen.

Im Zusammenhang mit Interoperabilität gibt es über die genannten Bereiche hinaus Bestrebungen, die Kommunikation in vernetzten Datenbanken zu verbessern: dabei werden Web Services⁹⁰ und Webserver-Fähigkeiten der Datenbanken⁹¹ wichtiger.

3.3 Integration contra Diversifikation

Geht es um die Integration von Anwendungen, gibt es zwei gegenläufige Überzeugungen: **Diversifikation** steht im Bezug auf Datenbanken für klare Schnittstellen und eine getrennte Bereitstellung von Applikationen, während der Begriff der **Integration** stellvertretend für ein Einbinden von möglichst viel Funktionalität in ein Datenbanksystem benutzt wird.⁹²

Im Folgenden wird geklärt, welche einzubindenden Funktionalitäten einen Trend darstellen, und ob sich langfristig eher der Ansatz der Integration oder der Diversifikation durchsetzen wird.

3.3.1 Einbinden von Funktionalität

Bisher versuchen die großen Datenbankhersteller, möglichst viel Anwendung zu integrieren, damit zum einen ihre Produkte benutzerfreundlicher werden⁹³ und zum anderen die immer anspruchsvoller werdenden Applikationen leichter angebunden werden können.⁹⁴ Diese neuen Anforderungen verlangen von Datenbanksystemen einen Technologiebedarf, der von der Speicherung als BLOB bis zur Listenverwaltung komplexer Elementtypen, deren versionierter Speicherung sowie benutzerdefinierten Funktionen reicht.⁹⁵ Relationale Datenmodelle konnten dies bisher nicht ausreichend ermöglichen, weil sie einen zu einfachen Aufbau besitzen.

⁹⁰ Web Services sind im Internet bereitliegende Funktionen, die für den Datenzugriff in eigene Applikationen eingebunden werden können. Vgl. Klischewski, R. (2003), S. 3.

⁹¹ Webserver-Fähigkeiten in DBS ermöglichen, dass Datenbankoperationen über HTTP gebündelt werden. Vgl. Alkassar, A. et al. (2003), S. 206.

⁹² Vgl. Alkassar, A. et al. (2003), S. 209.

⁹³ Vgl. Carr, E. (2001), Abschnitt „Trend 6: Why more is less“.

⁹⁴ Vgl. Elmasri, R. / Navathe, S. B. (2002), S. 787.

⁹⁵ Vgl. Härder, T. / Rahm, E. (1999), S. 169.

Im Zuge der voranschreitenden Objektorientierung im Datenbankbereich⁹⁶ wird die Erfüllung der wachsenden Ansprüche möglich, und so rechnen Datenbankexperten damit, dass folgende Formate in Zukunft in fast jedem Datenbankmanagementsystem unterstützt werden:

- Mit **CAD-Programmen** werden Entwürfe von mechanischen und elektrischen Objekten gezeichnet. Eine CAD-Datenbank setzt voraus, dass sie CAD-Objekte mit Millionen von Teilen und Beziehungen verwalten kann und diese versionsgerecht speichert.⁹⁷
- **CAM-Systeme** sind ähnlich wie CAD-Programme aufgebaut, speichern allerdings nicht nur einen Entwurf, sondern auch Daten der Fertigung. Entsprechend werden Echtzeit-Reaktionen, eine schnelle Navigation und weitere Anforderungen erfüllt.⁹⁸
- Eine **CASE-Anwendung** unterstützt Mithilfe einer Entwicklungsumgebung den Zyklus einer Softwareentwicklung von Planung, Anforderung, Design, Entwicklung bis zu Tests, Wartung und Dokumentation. Auch hier werden besondere Anforderungen an die Versionierung und die Verwaltung sehr großer Entwürfe gestellt.⁹⁹ In einer CASE-Datenbank werden entsprechend Quellcode, Beziehungen von Softwaremodulen, Variablendefinitionen etc. gespeichert.¹⁰⁰
- **Multimedia-Formate** sind z. B. Bilder, räumliche Daten, Audio- und Videosequenzen.¹⁰¹ Multimedia-Datenbanken müssen nicht nur Multimedia effizient verwalten, sondern vor allem mit anderen Multimediaquellen zusammenarbeiten können.¹⁰²
- Ein **Office Information System** beinhaltet Geschäftsinformationen wie E-Mails, Dokumente und Rechnungen. Diese benötigen oft komplexe Datentypen wie Diagramme, formatierte Texte und Multimedia-Formate.¹⁰³
- **GIS** modellieren und verwalten raumbezogene zwei- oder dreidimensionale Informationen, und werden in der Geographie, Kartographie, Architektur und in Navigationssystemen eingesetzt.¹⁰⁴ Dabei bieten sich objektrelationale¹⁰⁵ und XML-Modelle¹⁰⁶ als Basis zur Speicherung der räumlichen Daten an.

⁹⁶ siehe Kapitel 3.1.1 „Beschreibung der Modelle“.

⁹⁷ Vgl. Begg, C. et al. (1999), S. 733.

⁹⁸ Ebenda, S. 733.

⁹⁹ Ebenda, S. 734.

¹⁰⁰ Vgl. Silberschatz, A. et al. (1999), S. 252.

¹⁰¹ Ebenda, S. 252.

¹⁰² Vgl. Blażewicz, J. et al. (2003), S. 5.

¹⁰³ Vgl. Begg, C. et al. (1999), S. 734.

¹⁰⁴ Vgl. Vossen, G. (2000), S. 725.

Im Bereich der Anwendungsintegration wird auch an speziellen Verfahren zur Lösung statistischer Probleme gearbeitet. Mit Hilfe von **evolutionären Algorithmen** sollen schwierige Probleme gelöst werden, z. B. eine optimale Ressourcenverteilung oder Probleme der Bioinformatik. **Neuronale Netze** sollen mit Hilfe von paralleler Informationsverarbeitung und Lernfähigkeit bei der Wissensgenerierung aus einer Datenbank und bei Grafik- und Sprachanalysen helfen. Diese Verfahren werden allerdings als nicht so wichtig wie die vorher genannten Anwendungen erachtet und deshalb im folgenden nicht weiter betrachtet.¹⁰⁷

3.3.2 Klare Schnittstellen

Die Alternative zum Einbinden immer neuer Formate und Anwendungen in Datenbanksysteme heißt Diversifikation. Statt neue Formate wie räumliche Daten und Multimedia in die Datenbanksysteme zu integrieren, sollen klare Schnittstellen dafür sorgen, dass Funktionalitäten von externen Applikationen bereitgestellt werden können. Welches Maß die Integration annehmen darf, wird unter Datenbankexperten heftig diskutiert.¹⁰⁸

Nachteile einer immer weitergehenden Integration liegen in der steigenden Komplexität und im Performancerückgang, welche allerdings bisher durch technologische Weiterentwicklungen aufgefangen werden konnten.¹⁰⁹ Auch kommt es zu finanziellen Verlusten, weil meist mehr Merkmale eingebaut als verkauft werden, und zudem immer Daten vorliegen, deren Integration eine Umprogrammierung erfordert.¹¹⁰

Klare Schnittstellen werden deshalb immer wichtiger, um Funktionalität außerhalb des Datenbanksystems anzusiedeln. Auch hier wird XML eine große Rolle spielen: als einheitliche Programmierschnittstelle soll es dem Austausch zwischen Datenbanksystemen und Applikationen dienen.

¹⁰⁵ Vgl. Kleiner, C. / Lipeck, U. W. (2001), S. 127.

¹⁰⁶ Vgl. Brinkhoff, T. / Weitkämper, J. (2001), S. 144.

¹⁰⁷ Vgl. Alkassar, A. et al. (2003), S. 205.

¹⁰⁸ Ebenda, S. 204.

¹⁰⁹ Ebenda, S. 209.

¹¹⁰ Vgl. Carr, E. (2001), Abschnitt „Trend 6: Why more is less“.

3.3.3 Ausblick

Auch wenn einige Experten weniger Anwendungsintegration fordern, wird der Einbau von modernen Funktionalitäten vorerst sicher wichtig bleiben. Folgende Grafik zeigt, welche Bedeutung die vom BSI befragten Experten den einzelnen Funktionalitäten beimessen:

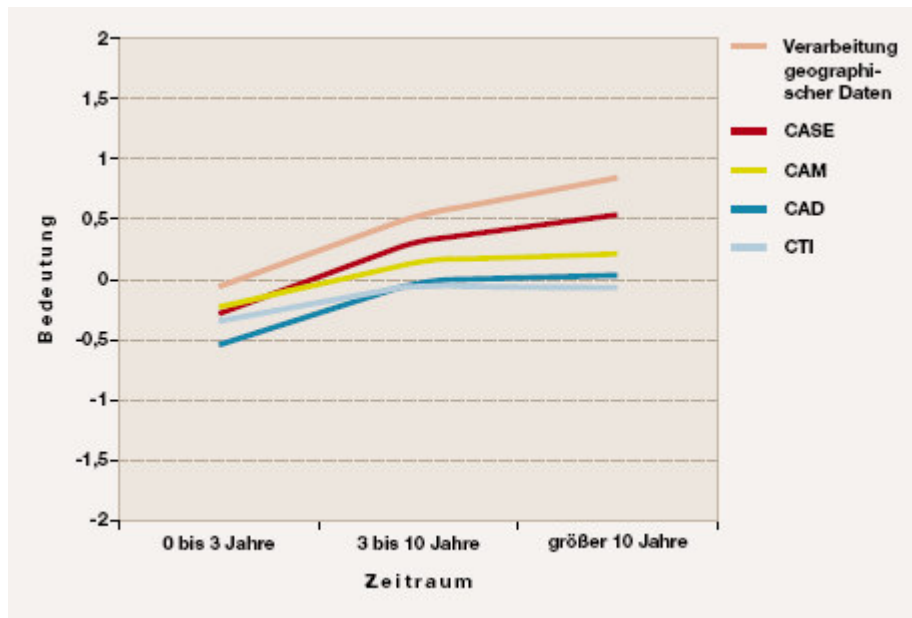


Abbildung 3: Bedeutung der Integration unterschiedlicher Funktionen in Datenbanksystemen

Quelle: Kommunikations- und Informationstechnik 2010+3 (2003), S. 205.

Vor allem geographische Daten und CASE werden dementsprechend eine große Rolle spielen, und auch CAM, CAD und CTI¹¹¹ werden in ihrer Bedeutung steigen. Der Einbau von Multimedia wurde in diesem Zusammenhang zwar nicht erörtert, allerdings wird die Integration von räumlichen und multimedialen Daten in der neuen Literatur allerdings meist analog als „neue Entwicklung“ eingestuft: U. a. sehen Halpin¹¹², Vossen¹¹³ und Silberschatz¹¹⁴ GIS und Multimedia-verwaltende Datenbanken als wichtige neue Technologien an. Elmasri spricht sogar von einer Vormachtstellung von Multimedia-Informationssystemen¹¹⁵. Blażewicz unterstützt ihn darin.¹¹⁶

¹¹¹ CTI steht für die Integration von Telefonie in die Computerwelt, Vgl. Siemens (2003).

¹¹² Vgl. Halpin, T. (2001), S. 700.

¹¹³ Vgl. Vossen, G. (2000), S. 725 ff.

¹¹⁴ Vgl. Silberschatz, A. et al. (1999), S. 710 und 719.

¹¹⁵ Vgl. Elmasri, R. / Navathe, S. B. (2002), S. 944.

¹¹⁶ Vgl. Blażewicz, J. / Morzy, T. (2003), S. 5.

Laut der BSI-Studie erachten Experten Diversifikation für mindestens genauso wichtig wie Integration.¹¹⁷ Bis 2006 sollen Mithilfe von XML definierte und einheitliche Schnittstellen für Programmierer vorliegen, so dass diese leicht von außerhalb Funktionen integrieren können.¹¹⁸

Zusammenfassend lässt sich folgern, dass man nicht klar ausmachen kann, ob nun Diversifikation oder Integration wichtiger wird. Sicher ist, dass es einige Funktionalitäten gibt, deren Einbau direkt in das Datenbanksystem notwendig ist, um komplizierte Anforderungen zu erfüllen. Und offenkundig ist auch, dass Datenbanken und Applikationen auf Dauer nur effizient zusammenarbeiten können, wenn klare Schnittstellen geschaffen werden.

3.4 Weitere Trends

Bisher wurden die drei Sachgebiete der Datenbankmodelle, der Vernetzung von Datenbanken und der Integration bzw. Diversifikation untersucht. Es gibt allerdings in der Datenbankforschung noch eine Vielzahl weiterer Entwicklungen, die durchaus einen Trend darstellen, z. B. im Bereich Sicherheit oder bei den aktiven, temporalen oder webbasierten Datenbanken. Diese zu beschreiben würde den Rahmen dieser Arbeit sprengen, und so beschränken sich die folgenden Ausführungen auf die Trends Data Warehouse, Data Mining, deduktive und parallele Datenbanken.

3.4.1 Data Warehouse und Data Mining

Data Warehouses speichern historische und aktuelle Geschäftsdaten, um bei allen Entscheidungen zu unterstützen, die eine Unternehmung treffen muss. Die benötigten Daten liegen meist in unterschiedlichen Quellen vor und deren Integration und Verwaltung sowie die Analyse sind die Hauptaufgaben eines Data Warehouses.¹¹⁹ Die verwalteten Daten eines Data Warehouse müssen folgenden Anforderungen genügen:¹²⁰

- Um Entscheidungen zu unterstützen, müssen sie *subjektorientiert* sein.
- Die Datenquellen müssen *konsistent* integriert werden.
- Die Daten müssen *zeitbezogen* gespeichert werden.
- Sie müssen *beständig* sein, d.h. sie dürfen nicht mehr ersetzt werden.

¹¹⁷ Vgl. Alkassar, A. et al. (2003), S. 210.

¹¹⁸ Ebenda, S. 205.

¹¹⁹ Vgl. Dorndorf, U. / Pesch, E. (2003), S. 387.

¹²⁰ Vgl. Begg, C. et al. (1999), S. 909.

Die Geschäftsdaten werden in multidimensionalen Datenmodellen gespeichert und mit Hilfe von Techniken wie OLAP und Data Mining analysiert. Traditionelle Datenbanksysteme unterstützen dagegen OLTP, welches aktuelle Informationen verwaltet, und reichern Data Warehouses mit zuvor verdichteten Informationen an.¹²¹ OLAP basiert auf diesen verdichteten Daten und analysiert diese, um kritische Daten für Entscheidungsträger aufzubereiten.¹²²

Data Mining kümmert sich um die Auswertung der Daten zur Generierung von neuem Wissen. Die Grenzen zu anderen Wissenschaften wie Statistik und Maschinelles Lernen sind hierbei fließend.¹²³ Mit Hilfe von verschiedenen Regeln sollen zukünftige Eigenschaften der Daten aufgezeigt, unbekannte Sachverhalte identifiziert (z. B. ein Hackerangriff oder ein Gen in einem DNA-Strang), Klassifikationen von Daten erstellt und die Verteilung begrenzter Ressourcen optimiert werden.¹²⁴

Laut der BSI-Studie wird die Bedeutung von Datenauswertung und Datenanalyse durch Data Warehouses und Data Mining in den nächsten Jahren weiter wachsen.¹²⁵ Auch andere Systeme wie Knowledge Management und Decision Support Systems werden von diesen erweiterten Fähigkeiten profitieren und teilweise traditionelle Datenbanksysteme ablösen. Insbesondere den so genannten Business Intelligence-Anwendungen, die operative Daten in entscheidungsrelevante Informationen transferieren, wird insgesamt ein stabiler Markt zugetraut.¹²⁶ Bei der Datenanalyse werden vor allem multidimensionales und hybrides OLAP¹²⁷ eine Rolle spielen, allerdings auch das noch eher unbekannt OLCP. Mit Hilfe von OLCP können Datenbanken mit komplexen Objekten wie z. B. Multimedia oder Grafiken nach Inhalten durchsucht werden.¹²⁸ Zur Weiterentwicklung von Data Mining wird vor allem die Forschung an den Techniken der Ähnlichkeitssuche und der Klassifikation von Objekten zunehmen.¹²⁹

Die Bedeutung von Data Warehouse und Data Mining wird auch von anderen Experten hervor gehoben: Elmasri traut diesen Datenbanktechniken eine große Rolle im nächsten Jahrzehnt zu¹³⁰.

¹²¹ Vgl. Vossen, G. (2000), S. 674.

¹²² Ebenda, S. 682.

¹²³ Vgl. Ester, M. / Sander, J. (2000), S. 15.

¹²⁴ Vgl. Elmasri, R. / Navathe, S. B. (2002), S. 916 ff.

¹²⁵ Vgl. Alkassar, A. et al. (2003), S. 214 ff.

¹²⁶ Vgl. IT-Director (2002), S. 57.

¹²⁷ Hybrides OLAP ist eine Mischung aus relationalem und multidimensionalem OLAP.

¹²⁸ Vgl. North, K. (1998), Abschnitt „OLCP“.

¹²⁹ Vgl. Alkassar, A. et al. (2003), S. 217.

¹³⁰ Vgl. Elmasri, R. / Navathe, S. B. (2002), S. 932.

Und fast alle gängigen Datenbank-Publikationen nennen sie als Trend, u. a. Silberschatz¹³¹ und Halpin¹³².

3.4.2 Deduktive Datenbanksysteme

Eine weitere Methode, um Wissen aus Fakten einer Datenbasis zu gewinnen, ist das Prädikatenkalkül der Logikprogrammierung. Wird diese Logik in ein Datenbanksystem integriert, spricht man von einem deduktiven Datenbanksystem.¹³³ Im Gegensatz zum Data Mining, das ebenso der Wissensgenerierung dient, bearbeitet es keine multidimensionalen, sondern relationale Datenbestände, und eignet sich so auch für herkömmliche Datenbanksysteme. Eingesetzt werden deduktive Datenbanksysteme z. B. in Verkehrsauskunfts-, wissensbasierten- und Expertensystemen.¹³⁴

Ein deduktives Datenbanksystem basiert entsprechend der Logikprogrammierung auf Regeln und Fakten, welches die Tupel der Datenbasis sind. Im Gegensatz zu den traditionellen Datenbanksystemen ist das deduktive Konzept somit turingvollständig, d. h. dass sie jedes mathematisch berechenbare Problem lösen können.¹³⁵ Tatsächlich wurde die Bedeutung von Deduktion mit der Integration rekursiver Sichten in SQL3 untermauert.¹³⁶

Momentan sind die deduktiven Datenbanktechniken allerdings nicht ausgereift genug und stoßen bei Anwendern auf wenig Akzeptanz.¹³⁷ Auch wenn diese Technik als elegant bezeichnet wird, sind noch einige Probleme zu lösen, z. B. wird Deduktion als zu langsam eingestuft.¹³⁸ Schon 1996 versprachen sich Spezialisten von deduktiven Datenbanken langfristig eine zukunftsweisende Technologie.¹³⁹ Tatsächlich sehen die von dem BSI befragten Experten heute erst recht eine Zukunft für deduktive Techniken und schätzen, dass diese 2007 in den gängigen Datenbanksystemen verfügbar sein werden.¹⁴⁰

¹³¹ Vgl. Silberschatz, A. et al. (1999), S. 708.

¹³² Vgl. Halpin, T. (2001), S. 660.

¹³³ Vgl. Kemper, A. / Eickler, A. (2001), S. 415.

¹³⁴ Vgl. Disterer, Georg et al. (2003), S. 236.

¹³⁵ Vgl. Fredegar, Florian (2003), Absatz 8.

¹³⁶ Vgl. Bry, F. / Seipel, D. (1996), Absatz 1.

¹³⁷ Vgl. Alkassar, A. et al. (2003), S. 201.

¹³⁸ Vgl. Halpin, T. (2001), S. 700.

¹³⁹ Vgl. Bry, F. / Seipel, D. (1996), letzter Absatz.

¹⁴⁰ Vgl. Alkassar, A. et al. (2003), S. 202.

3.4.3 Parallele Datenbanksysteme

Mit den zu verarbeitenden Daten wachsen auch die Anforderungen an Datenbanksysteme. Einprozessor-Server werden in Zukunft kaum mehr ausreichen, um Anfragen an Informationssysteme zu beantworten. Parallelität stellt hierbei eine Schlüsseltechnologie zur Steigerung von Performance und Skalierbarkeit dar, und entsprechend aktiv wird im Bereich der parallelen Datenbanksysteme entwickelt.¹⁴¹ Ähnlich wie beim Architekturkonzept der verteilten Datenbanksysteme sind auch hier die Daten auf mehrere Plattformen verteilt. Allerdings liegt hier der Fokus auf der parallelen Bearbeitung von Operationen und auf der Schnelligkeit. Auch liegen die Knoten meist in einem Computer oder zumindest an einem Standort vor, anstatt geographisch verteilt zu sein.¹⁴²

Man unterscheidet zwischen drei Architekturtypen:¹⁴³

- In der *Shared-Memory-Architektur* teilen sich mehrere Prozessoren einen Haupt- und Sekundärspeicher. Die Vorteile liegen in der einfachen Programmierung und optimalen Lastenverteilung. Die Nachteile liegen in der Synchronisation des Speicherzugriffs und der geringen Fehlerisolierung. Fehler treten dann auf, wenn Daten gleichzeitig in mehrere lokale Prozessor-Caches geladen und dort geändert werden: die Caches sind dann nicht mehr kohärent.
- Bei der *Shared-Disk-Architektur* besitzt jeder Prozessor einen Hauptspeicher und teilt sich mit anderen den Sekundärspeicher. Auf jedem Knoten liegt eine Kopie der Datenbank vor, die mit anderen synchronisiert werden muss. Die Vorteile liegen hier in der einfachen Skalierbarkeit und der Möglichkeit, ganze Funktionen an andere Prozessoren weiterzugeben.
- In der *Shared-Nothing-Architektur* hat jeder Prozessor seinen eigenen Speicher, so dass diese Architektur den verteilten Datenbanken sehr ähnlich ist. Optimal sind hier die Lastenverteilung, leichte Skalierbarkeit und gute Fehlerisolierung. Nachteilig sind die höheren Hardwarekosten.

¹⁴¹ Vgl. Ghandeharizadeh, S. et al. (2003), S. 194.

¹⁴² Vgl. Begg, C. et al. (1999), S. 656.

¹⁴³ Vgl. Vossen, G. (2000), S. 57.

Durch das steigende Datenvolumen und die ständige Forderung nach mehr Performance werden die parallelen Datenbanken als Entwicklungstrend immer wichtiger. Daher geht die BSI-Studie von einer weiten Verbreitung bis 2007 aus.¹⁴⁴

4. ZUSAMMENFASSUNG

Die Welt der Datenbanksysteme entwickelt sich weiter. Welche Richtungen sie ansteuert, ist dabei nur mittelfristig zu beantworten. In den nächsten Jahren werden auf jeden Fall Datenbanksysteme wichtig sein, die nicht nur große Datenbestände verwalten können, sondern darüber hinausgehenden Ansprüchen gerecht werden: Sie müssen über ein Datenmodell verfügen, welches multimediale, geographische, semistrukturierte und andere komplexe Daten speichern kann. Ihre Daten sollen sie in verteilten Systemen ablegen können und darauf einen performanten Zugriff gewährleisten. Der Datenaustausch mit Applikationen und fremden Datenbanksystemen muss reibungslos funktionieren. Um den Benutzer zufrieden zu stellen, müssen sie neue Methoden zur Aufbereitung von Daten und zur Generierung von Wissen beherrschen.

Die Ansprüche werden nicht sinken, sondern wachsen: Komplexe Objekte, XML, Vernetzung, Verteilung, Anwendungsintegration, Standardisierung, Informationssysteme und Parallelität sind nur einige der Themen, welche auch in Zukunft die Datenbankhersteller herausfordern und den Bereich der Datenbanksysteme spannend gestalten.

¹⁴⁴ Vgl. Alkassar, A. et al. (2003), S. 220.

LITERATURVERZEICHNIS

Alkassar, Ammar et al. (2003):

Kommunikations- und Informationstechnik 2010+3: Neue Trends und Entwicklungen in Technologien, Anwendungen und Sicherheit, Ingelheim 2003.

Begg, Carolyn et al. (1999):

Datenbanksysteme: Eine praktische Anleitung zu Design, Implementierung und Management, München 1999.

Blażewicz, Jacek / Morzy, Tadeusz (2003):

Management of Data: State-of-the-Art and Emerging Trends, in: Blażewicz, Jacek et al.: Handbook on Data Management in Information Systems, Berlin et al. 2003, S. 1-17.

Brinkhoff, Thomas / Weitkämper, Jürgen (2001):

Eine Architektur zur XML-basierten Repräsentation von bewegten Geo-Objekten, in: Heuer, Andreas et al. (Hrsg.): Datenbanksysteme in Büro, Technik und Wissenschaft, Berlin et al. 2001, S.127-143.

Disterer, Georg et al. (2003):

Taschenbuch der Wirtschaftsinformatik, 2. Auflage, München, Wien 2003.

Dorndorf, Ulrich / Pesch, Erwin (2003):

Data Warehouses, in: Blażewicz, Jacek et al.: Handbook on Data Management in Information Systems, Berlin et al. 2003, S. 387-430.

Elmasri, Ramez / Navathe, Shamkant B. (2002):

Grundlagen von Datenbanksystemen, 3., überarbeitete Auflage, München 2002.

Erbs, Heinz-Erich et al. (2003):

Datenbanken: Datenmodelle, Objekte, WWW, XML, Berlin, Offenbach 2003.

Ester, Martin / Sander, Jörg (2000):

Knowledge discovery in databases: Techniken und Anwendungen, Berlin et al. 2000.

Ghandeharizadeh, Shahram et al. (2003):

High Performance Parallel Database Management Systems, in: Blażewicz, Jacek et al.: Handbook on Data Management in Information Systems, Berlin et al. 2003, S. 194-220.

Hadeler, Thorsten / Arentzen, Ute (2000):

Gabler Wirtschaftslexikon, 15. Auflage, Wiesbaden 2001.

Halpin, Terry (2001):

Information modeling and relational databases: from conceptual analysis to logical design, London 2001.

Hansen, Hans Robert / Neumann, Gustaf (2001):

Wirtschaftsinformatik I, 8., völlig neubearbeitete und erweiterte Auflage, Stuttgart 2001.

Härder, Theo / Rahm, Erhard (1999):

Datenbanksysteme: Konzepte und Techniken der Implementierung, Berlin et al. 1999.

Helmer, Sven et al. (2003):

XML-Datenbanksysteme und ihre Anwendung, in: it – Information Technology, 45. Jg., Heft 3/2003, S. 137-142.

Horx, Matthias / Wippermann, Peter (1996):

Was ist Trendforschung?, Düsseldorf 1996.

IT-Director (2002):

Transformation von Daten in Wissen, in: IT-Director, 7. Jg., Heft 11/2002, S. 57.

Kazakos, Wassilios et al. (2002):

Datenbanken und XML, Berlin et al. 2002.

Kemper, Alfons / Eickler, André (2001):

Datenbanksysteme: eine Einführung, 4., überarbeitete und erweiterte Auflage, München et al. 2001.

Kemper, Alfons / Moerkotte, Guido (2003):

Object-Oriented Database Systems, in: Blažewicz, Jacek et al.: Handbook on Data Management in Information Systems, Berlin et al. 2003, S. 78-193.

Kleiner, Carsten / Lipeck, Udo W. (2001):

Web-Enabling Geographic Data with Object-relational Databases, in: Heuer, Andreas et al. (Hrsg.): Datenbanksysteme in Büro, Technik und Wissenschaft, Berlin et al. 2001, S.127-143.

Kramer, Christian (2001):

Analyse der Eignung von XML für die Verwendung von Datenmanagement in Datenbanken, Diplomarbeit, Berlin 2001.

Meier, Andreas / Wüst, Thomas (2000):

Objektorientierte und objektrelationale Datenbanken: ein Kompass für die Praxis, 2., überarbeitete und aktualisierte Auflage, Heidelberg 2000.

Michels, Jan-Eike et al. (2003):

The SQL Standard, in: it – Information Technology, 45. Jg., Heft 1/2003, S. 30-38.

Mittermeier, Ludwig (2003):

XML und Datenbanken: Naiv nativ, in: iX, Heft 8/2003, S. 42-52.

Pernul, Günther / Unland, Rainer (2003):

Datenbanken im Unternehmen: Analyse, Modellbildung und Einsatz, 2., korrigierte Auflage, München, Wien 2003.

Rolland, Fred (2003):

Datenbanksysteme im Klartext, München 2003.

Schulz, Hajo / Siering, Peter (2003):

Datendiener: Freie Datenbankserver im Vergleich, in: c't, Heft 5/2003, S.142-147.

Silberschatz, Abraham et al. (1999):

Database system concepts, 3. Auflage, Boston et al. 1999.

Stickel, Eberhard (Hrsg.) (1997):

Gabler – Wirtschaftsinformatik – Lexikon, Wiesbaden 1997.

Vossen, Gottfried (2000):

Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme, 4., korrigierte und ergänzte Auflage, München et al. 2000.

Zhang, Weiping / Ritter, Norbert (2001):

Leistungsuntersuchung von ORDB-gestützten objektorientierten Anwendungssystemen, in: Heuer, Andreas et al. (Hrsg.): Datenbanksysteme in Büro, Technik und Wissenschaft, Berlin et al. 2001, S. 227-243.

INTERNETVERZEICHNIS

Bager, Jo (1999):

Datenbanken: Wie man Informationen am besten organisiert, <http://www.heise.de/ct/99/04/130/>, 02.06.2000, Abruf am 01.12.2003.

Bry, François / Seipel, Dietmar (1996):

Deduktive Datenbanken, <http://www.gi-ev.de/informatik/lexikon/inf-lex-deduktive-db.shtml>, 1996, Abruf am 02.12.2003.

Carr, Eric (2001):

6 trends in business intelligence, <http://asia.cnet.com/newstech/applications/printfriendly.htm?AT=38027571-39001103t-39000011c>, 24.10.2001, Abruf am 01.12.2003.

Deutsche Informatik Akademie (2003):

9. Datenbank-Tutorientage 2003, http://www.dia-bonn.de/downloads/dbtt_2003.pdf, 2003, Abruf am 02.12.2003.

Devarakonda, Ramakanth S. (2002):

Object-Relational Database Systems - The Road Ahead, <http://www.acm.org/crossroads/xrds7-3/ordbms.html>, 01.03.2002, Abruf am 01.12.2003.

Fredegar, Florian (2003):

Was für unterschiedliche Arten an Programmiersprachen gibt es?, http://www.selflinux.org/selflinux-devel/html/programmieren_allgemein03.html, 6.10.2003, Abruf am 02.12.2003.

IDG Interactive (2003):

Studie: Deutsche länger online, <http://www.tecchannel.de/news/internet/13696/>, 24.11.2003, Abruf am 28.11.2003.

Jenz, Dieter E. (2002):

ebXML kommt - und keiner merkt's?, <http://www.aboutit.de/view.php?ziel=/02/28/08.html>, 17.08.2002, Abruf am 02.12.2003.

Klischewski, Ralf (2003):

Interoperabilität und Kooperation in der internetbasierten Verwaltung, http://swt-www.informatik.uni-hamburg.de/publications/files/klischewski_emisa03.pdf, 2003, Abruf am 02.12.2003.

Meehan, Michael (2001):

EDI, ebXML groups agree to cooperate, <http://www.computerworld.com/printthis/2001/0,4814,61840,00.html>, 02.07.2001, Abruf am 02.12.2003.

Münz, Stefan (2001):

Entstehung des Internet, <http://selfhtml.teamone.de/intro/internet/entstehung.htm>, Abruf am 28.11.2003.

North, Ken (1998):

Complex Processing, Analytical Processing, and Transactions, http://ourworld.compuserve.com/homepages/Ken_North/complex.htm, Juni 1998, Abruf am 02.12.2003.

Oebbeke, Alfons (2003):

EDI / EDIFACT, http://www.glossar.de/glossar/z_edi.htm, Abruf am 02.12.2003.

Sattler, Kai-Uwe (2003):

Anfragetechniken für heterogene Datenbanksysteme, Habilitation, <http://www.witi.cs.uni-magdeburg.de/~sattler/papers/Habilschrift.pdf>, 16.05.2003, Abruf am 02.12.2003.

Schmitz, Ludger (2002):

Relationale Datenbanken und XML, <http://www.cowo.de/index.cfm?pageid=255&artid=42163>, 23.10.2002, Abruf am 01.12.2003.

Siemens (2003):

Computer Telefonie Integration, http://w3.siemens.de/solutionprovider/_online_lexikon/8/f000478.htm, Abruf am 02.12.2003.

Udell, John (2003):

The future of XML documents and relational databases, http://www.infoworld.com/article/03/07/25/29FEdocs_1.html, 25.07.2003, Abruf am 02.12.2003.

EHRENWÖRTLICHE ERKLÄRUNG

”Ich erkläre ehrenwörtlich:

1. dass ich meine Hausarbeit ohne fremde Hilfe angefertigt habe,
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe,
3. dass ich meine Hausarbeit bei keiner anderen Prüfung vorgelegt habe.
4. Alle Quellen, die dem World Wide Web entnommen oder in einer sonstigen digitalen Form verwendet wurden, sind der Arbeit beigefügt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.“