Grundlagen komponentenorientierter föderierter Datenbanksysteme

Andreas Christiansen

Institut für Technische Informationssysteme
Otto-von-Guericke-Universität Magdeburg
Postfach 4120, D-39016 Magdeburg, Germany

E-mail: `christia@iti.cs.uni-magdeburg.de`
Tel.: `++49/391/67-12220` Fax: `++49/391/67-12020`

Zusammenfassung

In diesem Papier sollen am Beispiel des von der OMG entwickelten komponentenorientierten CORBA-Ansatzes [OMG96] grundlegende Aspekte einer auf einem solchen Ansatz basierenden Datenbankföderation vorgestellt werden. Dazu werden zunächst relevante Teile des CORBA-Standards sowie Grundlagen aus dem Bereich föderierter Datenbanksysteme aufgearbeitet. Darauf aufbauend werden ausgewählte Möglichkeiten zur Ausnutzung der Vorteile komponentenorientierter Systeme bei der Gestaltung einer Datenbankföderation präsentiert.

1 Einführung

In zahlreichen Unternehmen und Institutionen existieren historisch entstandene Datenmengen, welche durch den parallelen Einsatz verschiedener Spezialsysteme, basierend auf unterschiedlichen Datenbanksystemen, ein hohes Maß an Heterogenität aufweisen. Insbesondere die durch die sich häufig überlappenden Datenmengen bedingte Redundanz induziert einen hohen Kostenbedarf beim Betrieb eines heterogenen Informationssystems. Wesentlich verstärkt wird dieses Problem durch die rasche Entwicklung und Ausbreitung des Network-Computing, wobei eine globale Sicht auf weltweit heterogen verteilte Datenbestände notwendig wird. Vor diesem Hintergrund werden im Rahmen des Projektes **SIGMA**$_{FDB}$[1] [SCC$^+$97] die Grundlagen föderierter Datenbanksysteme erarbeitet. Neben der Schematransformation und -integration, Untersuchungen zur Konsistenzsicherung und der Einbeziehung von Dateiverbünden in eine Datenbankföderation bildet die Ableitung einer geeigneten Architektur einen Schwerpunkt in diesem Projekt.

Aufgrund der Komplexität einer Datenbankföderation gilt es eine Basisarchitektur zu finden, welche sich vor allem durch Offenheit, Hardware- und Betriebssystemunabhängigkeit und Adaptierbarkeit in moderne Rechnerarchitekturen auszeichnet. In diesem Zusammenhang erscheint es sinnvoll, komponentenorientierte Ansätze wie CORBA [OMG96] oder OLE [Sen95] zu untersuchen, da diese bereits in ihrer Konzeption Anforderungen aus dem Bereich der föderierten Datenbanksysteme berücksichtigen.

2 Grundlagen

In diesem Abschnitt werden wichtige Begriffe und Grundlagen föderierter Datenbanksysteme sowie ausgewählte Teile der *Common Object Request Broker Architecture* (CORBA) vorgestellt.

2.1 Föderierte Datenbanksysteme

Ein *föderiertes Datenbanksystem* (FDBS) besteht aus dem *föderierten Datenbankmanagementsystem* (FDBMS) und aus einem oder mehreren *Komponentendatenbanksystemen*

[1]**SIGMA**$_{FDB}$ (**S***chema* **I***ntegration and* **G***lobal integrity* **M***aintenance* **A***pproach for* **F***ederated* **D***ata* **B***ases*), gefördert durch das Land Sachsen-Anhalt unter FKZ 1987 A / 0025

(CDBS) [SL90]. Ziel des FDBS-Einsatzes ist es, logisch und möglicherweise physisch verteilte heterogene Datenbanksysteme zu integrieren, wobei die Autonomie bestehender lokaler Applikationen gewahrt wird. Die Heterogenität zwischen den zu föderierenden CDBS spiegelt sich in unterschiedlichen Laufzeitumgebungen, verschiedenen Datenmodellen und Datenbankschemata wider.

Die zentrale Aufgabe eines FDBMS besteht in der Bereitstellung einer globalen Schnittstelle für den einheitlichen und transparenten Zugriff auf alle föderierten Daten. Darüber hinaus soll das FDBMS die Funktionalität eines Datenbankmanagementsystems offerieren. Insbesondere die Konsistenzsicherung der heterogen verteilten Daten, ein verteiltes Transaktionsmanagement sowie die Bereitstellung einer Anfragesprache werden gefordert.

Abb. 1 zeigt das Architekturmodell eines FDBS. Die CDBS werden durch eine zentrale Ebene gekoppelt, welche die Dienste des FDBMS bereitstellt. Die lokalen Applikationen

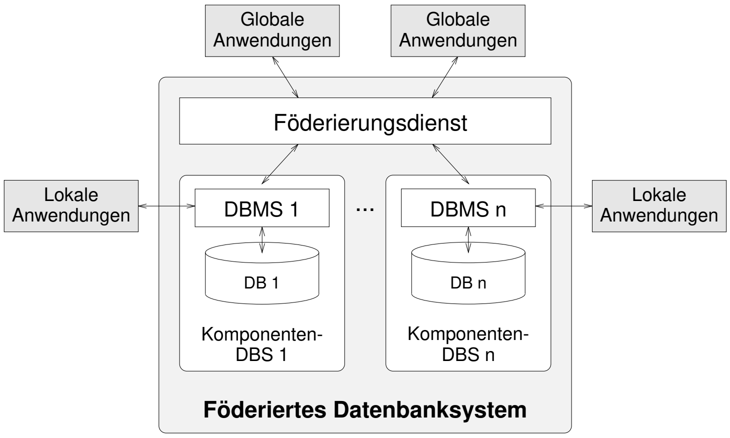

Abbildung 1: Komponenten eines föderierten Datenbanksystems

operieren weiterhin auf den Schnittstellen der lokalen Datenbanksysteme. Die Umsetzung einer globalen Integritätssicherungskomponente kann zwischen einem lokalen DBMS und einer lokalen Anwendung eine zusätzliche Ebene erfordern.

Die Schematransformation und -integration bildet einen wesentlichen Schwerpunkt beim Aufbau einer Datenbankföderation. Grundlegend für den gesamten Prozeß der Schematransformation und -integration ist die in [SL90] vorgeschlage 5-Ebenen Schemaarchitektur, welche ein geeignetes kanonisches Datenmodell voraussetzt. Durch Überführen der lokalen Schemata unter Benutzung des kanonischen Datenmodells in Komponentenschemata wird die Heterogenität auf Datenmodellebene überwunden. Aus den Komponentenschemata werden die zu föderierenden Schemateile exportiert und zu einem föderierten Schema integriert.

2.2 OMA/CORBA

Mit der *Object Management Architecture* (OMA) [OMG96] entwickelte die *Object Management Group* (OMG) ein Modell, welches die Interaktion von Objekten in heterogenen Netzwerkumgebungen beschreibt. Kern und zentrale Kommunikationskomponente dieses Modells bildet ein *Object Request Broker* (ORB). Zur Unterstützung der Entwicklung von Anwendungsobjekten wurden im Rahmen des OMA-Vorschlages zahlreiche, allgemein nutzbare Objektdienste entwickelt und in der *Common Object Service Specification* (COSS) [OMG94] spezifiziert. Ferner greift die OMA auf ein integriertes Facilitiy-Konzept zurück, welches es erlaubt, Objekte hinsichtlich ihrer Dienste entsprechend eines speziellen Anwendungsbereiches (vertikal) oder mehrere Anwendungsbereiche betreffend (horizontal) zu klassifizieren. Anwendungsobjekte, COSS-Objekte und Facilities kommunizieren via ORB miteinander, dessen Architektur durch die *Common Object Request*

Broker Architecture (CORBA) beschrieben wird. Fundamental in der in Abb. 2 darge-

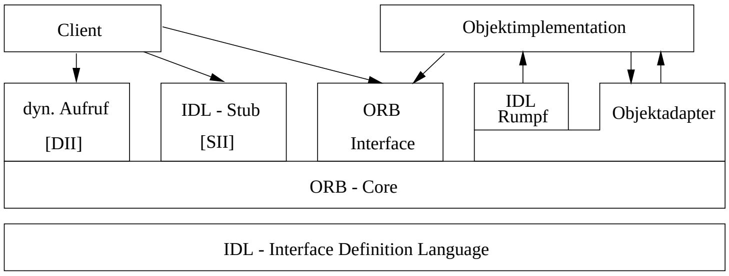

Abbildung 2: Common Object Request Broker Architecture

stellten Architektur ist die konsequente Trennung der Objektimplementierungen von deren Schnittstellen, welche in einer *Interface Definition Language* (IDL) [OMG96] beschrieben werden. Aus den IDL-Beschreibungen werden durch einen IDL-Compiler (nicht in Abb. 2 dargestellt) sowohl die Stubs zur Client-Adaption als auch die Codeskelette für die Implementierung der Serverobjekte erzeugt. Schnittstellen und Objektimplementierungen werden beim Kernel dieser Architektur (ORB-Core) registriert. Aufgrund dieser Informationen ist der ORB in der Lage, Dienstanforderungen zwischen Client- und Serverobjekten zu koordinieren. Eine Dienstanforderung erfolgt entweder durch Aufrufen der erzeugten Stub-Methoden *Static Invocation Interface* (SII) oder durch dynamisches Konstruieren der Dienstanforderung *Dynamic Invocation Interface* (DII) aus den Informationen des Interface-Repositories, welches vom ORB verwaltet wird.

3 CORBA als Grundlage einer FDBS-Architektur

Grundsätzlich kann festgestellt werden, daß durch den hohen Standardisierungsgrad von CORBA ein genereller Komplexitätsabbau beim Entwurf von verteilten Informationssystemen zu erkennen ist. Insbesondere ermöglicht ein ORB die Interaktion von Objekten in heterogenen Hardware- und Betriebssystemumgebungen, wobei es das *Internet Inter ORB Protocol* (IIOP) [OMG96] erlaubt, diese Feststellung auf Weitverkehrsnetze auszudehnen. Unter der Annahme, daß in einem FDBS-Szenario Heterogenität auf Hardware- bzw. Betriebssystemebene zu überwinden ist, stellt CORBA eine geeignete Kommunikationstechnologie in einer FDBS-Umgebung bereit. Ein Vorschlag für eine FDBS-Architektur, welche den CORBA-Standard primär zur Kommunikation nutzt, wird in [DDK$^+$95] vorgestellt.

Aufgrund der Mächtigkeit des CORBA-Standards erscheint es sinnvoll, diesen auf Einsatzmöglichkeiten sowohl hinsichtlich einer Eignung für den Entwurf und Betrieb eines FDBS als auch zur Datenrepräsentation auf föderierter Ebene zu untersuchen. Dabei werden das CORBA-Objektmodell [OMG96] als globales Datenmodell und die Existenz eines föderierten Datenbankschemas [SS96] vorausgesetzt.

Ein wichtiges Merkmal eines FDBS ist dessen Offenheit gegenüber den zu föderierenden CDBS. In diesem Zusammenhang muß beim Entwurf einer FDBS-Architektur die Heterogenität verschiedener Datenbankschnittstellen berücksichtigt werden. Eine wichtige Voraussetzung für eine globale Sicht auf alle föderierten Daten bildet eine abstrakte Schnittstelle zu den CDBS, welche die herstellerspezifischen Details verbirgt und Protokollabbildungen hinsichtlich Transaktionsmanagement und Anfrageverarbeitung realisiert. Zur Spezifikation dieser Schnittstelle kann der abstrakte Charakter einer IDL-Beschreibung ausgenutzt werden. Basiert ein CDBS-Adapter auf einer IDL-Beschreibung, ergeben sich verschiedene Vorteile gegenüber Alternativmethoden[2]. Die

[2]Alternativ könnte die Schnittstellenbeschreibung auf der abstrakten Klasse einer objektorientierten Programmiersprache basieren

Schnittstellenbeschreibung ist programmiersprachunabhängig. Darüber hinaus bedingt eine IDL-Beschreibung die Umsetzung eines Datenbankadapters als verteiltes Objekt. Somit ist es möglich, die lokalen Datenbankschnittstellen auf dem CDBS-Knoten durch ein CORBA-Objekt zu kapseln und die Kommunikation des FDBS mit den verschiedenen CDBS zu homogenisieren. Ferner können Aufgaben des FDBS direkt dem ORB übertragen werden, was einen Komplexitätsabbau zur Folge hat. Beispielhaft sei hier das Registrieren und Verwalten lokaler Datenbankadapter erwähnt.

Neben der Neuspezifikation von FDBS-Schnittstellen auf IDL-Basis bestehen zahlreiche Möglichkeiten, bereits entworfene und im Rahmen der COSS-Spezifikationen veröffentlichte Schnittstellenbeschreibungen zur Modellierung von FDBS-Komponenten zu verwenden.

1. *Object Naming Service* (ONS) [OMG93b]
   Der ONS unterstützt die Identifikation verteilter Objekte auf der Basis logischer Namen, welche an eine physische Objektreferenz gebunden werden. Aufgrund der Unterstützung hierarchischer Namensräume ist die Identifikation eines Objektes durch einen zugeordneten Namen abhängig vom aktuellen Namenskontext. Das Prinzip hierarchischer Namensräume kann als Basisgerüst zur Behandlung der Objektidentität in einem FDBS herangezogen werden, da in einem FDBS die Problematik besteht, globale Objekte eindeutig zu identifizieren und gleichzeitig den Zugriff auf die zugeordneten lokalen Objekte zu sichern [SS95].

2. *Object Query Service Specification* (OQSS) [OMG95]
   Mit der OQSS wird ein Objektdienst spezifiziert, welcher eine Anfragebearbeitung über heterogen verteilten Objekten beschreibt. Ein wesentliches Merkmal dieses Objektdienstes ist die Delegation von Anfragen und die Föderation systemgebundener Anfragesysteme. Dabei unterstützt dieser Objektdienst sowohl das Weiterleiten einer Anfrage an einen weiteren dieser Spezifikation genügenden Anfrageserver als auch das Weiterleiten an die Anfrageschnittstelle eines konkreten DBMS. In Bezug auf die Anforderungen an die Anfrageschnittstelle eines FDBMS kann offensichtlich das Prinzip der Anfragedelegation als Grundlage zur Verteilung von Anfragen auf föderierter Ebene benutzt werden. Dazu ist ein zentraler Anfrageserver vorstellbar, welcher die Anfragen entprechend dem föderierten Schema aufspaltet und diese an die lokalen Datenbankadapter delegiert, welche ihrerseits die Anfragen dem CDBS propagieren. Diese Methodik ermöglicht die Integration lokaler Anfrageoptimierer in die Bearbeitung der FDBS-Anfragen. Ein auf dem OQSS basierender Anfragemanager besitzt zusätzlich das Potential, Anfragen an Datenhaltungssysteme zu unterstützen, welche nicht die Funktionalität eines DBMS bereitstellen.

4 Zusammenfassung und Ausblick

In den vorangestellten Abschnitten wurden die Grundlagen föderierter Datenbanksysteme sowie die des CORBA-Standards grob umrissen. Daran anschließend erfolgte eine Diskussion verschiedener Vorschläge, die Dienste eines FDBS als CORBA-Komponenten abzubilden, welche jedoch im Einzelnen einer genaueren Untersuchung bedürfen.

Zusätzlich zu den in Abschnitt 3 diskutierten Möglichkeiten, COSS-Spezifikationen auf den FDBS-Entwurfsprozeß anzuwenden, sind Untersuchungen geplant, die Transaktionsverwaltung in einer FDBS-Umgebung auf der Basis des *Object Transaction Service* (OTS) [OMG97] zu realisieren. Auch in der Umsetzung einer aktiven Integritätssicherungskomponente [TC96], welche als Basisgerüst den *Event Notification Service* (ENS) [OMG93a] verwendet, wird eine Möglichkeit zur Anwendung des CORBA-Paradigmas auf das Design einer FDBS-Architektur gesehen.

Zukünftig gilt es ferner zu untersuchen, ob neben dem physischen Entwurf und Betrieb eines FDBS der CORBA-Standard auch auf das konzeptionelle FDBS-Design angewendet werden kann. In diesem Zusammenhang sind Untersuchungen einer IDL-basierten Schematransformation und -integration geplant. Unter dem Gesichtspunkt, daß das FDBS vollständig oder teilweise in Form verteilter Objektstrukturen in mehrere Komponenten zerlegt wird, ergeben sich neue Anforderungen hinsichtlich der Verfügbarkeit des föderierten Schemas.

Literatur

[DDK+95] A. Dogac, C. Dengi, E. Kilic, F. Ozhan, S. Nural, C. Evrendilek, U. Halici, B. Arpinar, P. Koksal, N. Kesim und S. Mancuhan. METU Interoperable Database System. *ACM SIGMOD Record*, 24(3):56–61, September 1995.

[OMG93a] OMG (Object Management Group). *Event Notification Service, OMG Document 93.07.03*, Juli 1993.

[OMG93b] OMG (Object Management Group). *Object Naming Service, OMG Document 93.05.02*, Mai 1993.

[OMG94] OMG (Object Management Group). *Common Object Services Specification, Volume 1, OMG Document Number 94.3.1*, März 1994.

[OMG95] OMG (Object Management Group). *Object Query Service Specification, OMG TC Document 95.1.1*, Januar 1995.

[OMG96] OMG (Object Management Group). *The Common Object Request Broker: 2.0/IIOP Specification, OMG Document Number 96.08.04*, August 1996.

[OMG97] OMG (Object Management Group). *Object Transaction Service, OMG Document 97.02.15*, Februar 1997.

[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*, Informatik aktuell, S. 322–331, Berlin, 1997. Springer-Verlag.

[Sen95] U. Sendler. *CAD Office Integration–OLE für Design und Modellierung.* Springer-Verlag, Berlin, 1995.

[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.

[SS95] I. Schmitt und G. Saake. Managing Object Identity in Federated Database Systems. In M. Papazoglou, Herausgeber, *OOER'95: Object-Oriented and Entity-Relationship Modeling, Proc. of the 14th Int. Conf., Gold Coast, Australia, December 1995*, *Lecture Notes in Computer Science*, Band 1021, S. 400–411, Berlin, 1995. Springer-Verlag.

[SS96] I. Schmitt und G. Saake. Flexible Generation of Global Integrated Schemata using GIM. In S. Conrad, M. Höding, S. Janssen und G. Saake, Herausgeber, *Kurzfassungen des Workshops "Föderierte Datenbanken", Magdeburg, 22.04–23.04.96*, Informatik-Bericht, Nummer 96–1, S. 29–43. Institut für Technische Informationssysteme, Universität Magdeburg, April 1996.

[TC96] C. Türker und S. Conrad. Using Active Mechanisms for Global Integrity Maintenance in Federated Database Systems. In S. Conrad, M. Höding, S. Janssen und G. Saake, Herausgeber, *Kurzfassungen des Workshops "Föderierte Datenbanken", Magdeburg, 22.04–23.04.96*, Informatik-Bericht, Nummer 96–1, S. 51–65. Institut für Technische Informationssysteme, Universität Magdeburg, April 1996.