Integration und Einhaltung von Integritätsbedingungen in FDBSen*

Sven Janssen
Otto-von-Guericke-Universität Magdeburg
Institut für Technische Informationssysteme
Postfach 4120, D–39016 Magdeburg
E-mail: janssen@iti.cs.uni-magdeburg.de
Tel.: ++49-391-67-12220 Fax: ++49-391-67-12020

Zusammenfassung

In diesem Papier soll ein kurzer Überblick über die Integration und Einhaltung von Integritätsbedingungen föderierter Datenbanksysteme (FDBSe) gegeben werden. Die Schritte des FDBS-Entwurfsprozesses werden unter dem Aspekt der Integration von Integritätsbedingungen überblicksartig vorgestellt. Dabei werden Probleme in Abhängigkeit von Anforderungen an das FDBS aufgezeigt. Anschließend erfolgen Betrachtungen zur Wahrung der im Entwurfsprozeß auftretenden Integritätsbedingungen.

1 Einleitung

Ein Trend der derzeitigen Datenbankforschung besteht in der Integration existierender und oft heterogener Datenbanken. Dies ergibt sich aus der Forderung nach neuen datenbankübergreifenden Anwendungen oder Einhaltung von Abhängigkeiten zwischen verschiedenen Datenbanken. Zur Sicherung getätigter Investitionen in Unternehmen sollen die lokalen Datenbankapplikationen weiterhin lauffähig sein, was eine weitestgehende Erhaltung der Autonomie der an der Föderation beteiligten Systeme erfordert. Das an unserem Institut gestartete Forschungsprojekt *Föderierung heterogener Datenbanksysteme und lokaler Datenhaltungskomponenten zur systemübergreifenden Integritätssicherung: Kurztitel* ***SIGMA***$_{\mathrm{FDB}}$[1], beschäftigt sich auf diesem Gebiet u.a. mit der Methodik zur Integration und der Ableitung und Einhaltung von Integritätsbedingungen [HTJ+95]. Dabei geht es auch um die Einbeziehung von Datenhaltungssystemen ohne oder nur mit geringfügiger Datenbankfunktionalität, da solche Komponenten in vielen mittelständischen Unternehmen eine weite Verbreitung gefunden haben und innerhalb des Forschungsprojektes eine Rolle spielen. Im Projekt sollen FDBSe als Rahmensystem zur Integration der an der Fabrikplanung beteiligten Software-Werkzeuge (Entwurfs-, Planungs-, Simulations- und Animationswerkzeuge) einschließlich ihrer lokalen Datenbestände fungieren.

Dazu wird in Kapitel 2 ein kurzer Einblick in die Architektur föderierter Datenbanksysteme gegeben. Nach der Begriffsbestimmung und Klassifikation von Integritätsbedingungen wird auf deren Integration und Einhaltung eingegangen.

2 Architektur föderierter Datenbanksysteme

Ein *föderiertes Datenbanksystem (FDBS)* besteht aus einer Menge von kooperierenden, autonomen, möglicherweise heterogenen *Komponentendatenbanksystemen (KDBSen)* [SL90]. Die Kooperation erfolgt über eine globale Föderierungsschicht, das *Föderierte Datenbankmanagementsystem (FDBMS)*. Die wesentlichen Aufgaben dieser *erweiterbaren* Föderierungsschicht bestehen darin, eine globale, homogene Schnittstelle für den einheitlichen und transparenten Zugriff auf föderierte Daten bereitzustellen sowie datenbankübergreifende Integritätsbedingungen einzuhalten. Dabei sollen die lokalen Datenbankapplikationen

*Diese Arbeit wurde gefördert durch das Land Sachsen-Anhalt unter FKZ: 1987A/0025 "Föderierung heterogener Datenbanksysteme und lokaler Datenhaltungskomponenten zur systemübergreifenden Integritätssicherung"

[1] **SIGMA**$_{FDB}$ (**S***chema* **I***ntegration and* **G***lobal integrity* **M***aintenance* **A***pproach for* **F***ederated* **D***ata* **B***ases*)

so wenig wie möglich beeinträchtigt werden. Bei der Entwicklung eines solchen FDBSs muß die Heterogenität der KDBSe bezüglich Hardware, Betriebssystem, Datenmodell, Schema, Zugriff und der Daten selbst, überwunden werden. Bei der Föderierung heterogener Datenbanksysteme werden prinzipiell zwei Ansätze unterschieden: der *lose gekoppelte* und der *eng gekoppelte* Ansatz, wobei wir den eng gekoppelten Ansatz im genannten Forschungsprojekt wegen der Forderung nach Erhalt datenbankübergreifender Integriätsbedingungen favorisieren.

3 Integration und Einhaltung von Integritätsbedingungen

Ein wesentlicher Aspekt bei der Entwicklung eines FDBS ist die Wahrung der Integrität der Daten. Nach einer Begriffsbestimmung und Klassisfikation wird in 3.2 auf die Integration und in 3.3 auf die Erhaltung von Integritätsbedingungen eingegangen.

3.1 Begriffsbestimmung und Klassifikation

Unter *Integrität* wird hier die Korrektheit der Daten und deren Verwendung innerhalb eines Datenbankmanagementsystems (DBMSs) verstanden. Dies schließt laut [GR94] den *Datenschutz*, die *Datenkonsistenz* und die *Datensicherheit* auf Seiten des DBMSs ein. Unter Datensicherung wird die Bewahrung der Daten vor Verfälschung oder Vernichtung, unter Datenschutz die Sicherung vor mißbräuchlicher Verwendung verstanden. Die Konsistenz der Daten bezieht sich auf die semantische Integrität, also die Korrektheit der Daten im Sinne des modellierten Weltausschnittes.

Ein wichtiger Schritt im Hinblick auf die Integration von Datenbanken ist die Ermittlung und Klassifikation der Integritätsbedingungen, die im folgenden als *lokale Integritätsbedingungen (LIBen)* bezeichnet werden. Informationsquellen dazu sind Datenbankschemata, Wissen der Datenbankadministratoren, die in den Anwendungen enthaltenen Informationen sowie die Anwender der Datenbankapplikationen selbst.

Es gibt verschiedene Arten der Klassifikation von Integritätsbedingungen, auf die kurz eingegangen wird. Aufgrund des **Zeitlichen Kontexts** lassen sich *statische* und *dynamische* Integritätsbedingungen unterscheiden [HS95]. Statische Integritätsbedingungen beziehen sich auf Datenbankzustände, dynamische auf Zustandsübergänge, wobei sich diese noch in *transitionale* (ein Zustandsübergang) und *temporale* (Folge von Zustandsübergängen) Integritätsbedingungen unterscheiden lassen. Nach der **Datenmodellzugehörigkeit** wird in *modellinhärente* (impliziert aus der Strukturbeschreibung des Datenmodells) und *explizit formulierte* Integritätsbedingungen unterschieden. Innerhalb des Kontextes **Granularität** wird hier zwischen IB bezogen auf einzelne (A_1) oder mehrere (A_n) Eigenschaften (Attribute, Referenzen) innerhalb einer (T_1) oder mehrerer (T_n) Datenmodell-Objekttypen unterschieden.

Von Interesse ist dabei, wie und welche Integritätsbedingungen während des FDBS-Entwurfs zu behandeln sind, wie sich ihre Klassenzugehörigkeit ändert und welche Regeln in Abhängigkeit von den Anforderungen an das FDBS abgeleitet werden können. Grundlegende Arbeiten dazu befinden sich in [SL90, BLN86, RPG95, CHJ$^+$96].

3.2 Integration von Integritätsbedingungen

Der föderierte Datenbank-Entwurfsprozeß umfaßt nach [SL90, BLN86] bezüglich der Schemata die Schritte Schematransformation, -filterung, -integration und Ableitung externer Schemata. Analog zu diesen Schritten werden die Schritte bezüglich Integration der Integritätsbedingungen in Abbildung 1 zugeordnet. Diese Schritte sind in jedem Fall abhängig voneinander zu betrachten. Der FDBS-Entwurf hängt dabei stark von Anforderungen an das FDBS ab. Anforderungen sind beispielsweise höchstmögliche Wahrung der Autonomie der beteiligten Datenbanksysteme, flexible An- und Abkopplung von Systemen, Zugriffsanforderungen auf der globalen Schnittstelle, Definition und Wahrung systemübergreifender Integritätsbedingungen, schnelle Ausführung der globalen Datenbankoperationen, schnelle Reaktion im Fall einer Integritätsverletzung, etc., die teilweise im Widerspruch zueinander stehen. Dies hat natürlich auch Auswirkungen auf die Integration von IBen.

Der Erste Schritt umfaßt die *Ermittlung der lokalen Integritätsbedingungen.* Im Zusammenhang mit der Ermittlung der lokalen Schemata werden die LIBen dem Schema zugeordnet und innerhalb des Schemas durchnumeriert. Auch die aus den Anwendungen oder dem Wissen der Datenbankadministratoren gewonnen IBen sollten als LIBen mit einfließen. An dieser Stelle findet eine Klassifikation nach Datenmodellzugehörigkeit, zeitlichem Kontext, Typ- und Attributanzahl statt. Als Beispiel:

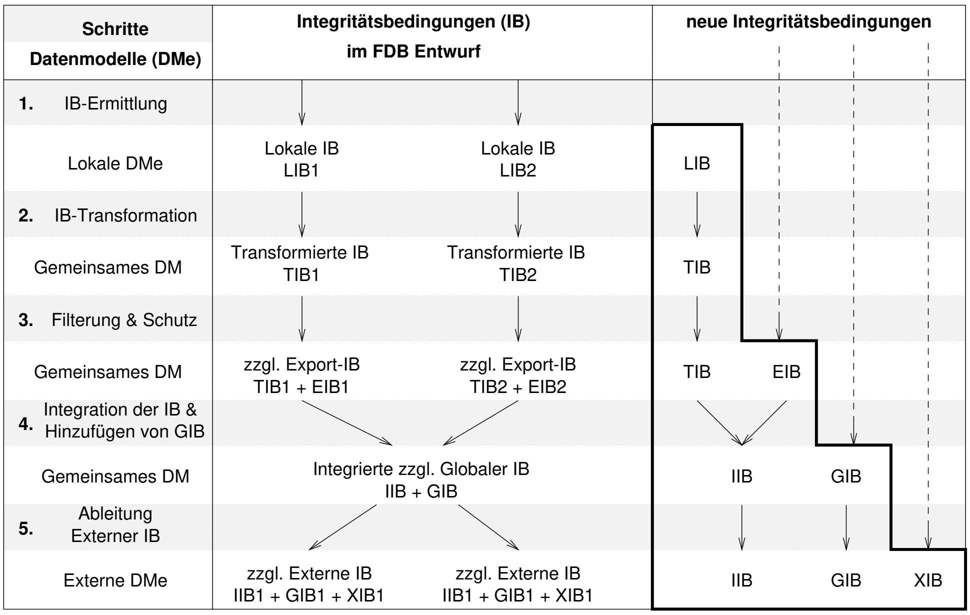

Abbildung 1: Integrationsprozeß von IBen.

```
LIS1.1: PERSON.Alter between 18 and 65 (explizit,statisch,T1,A1),
LIS2.1: Person.age >= 20 (explizit,statisch,T1,A1),
LIS2.2: Person.age <= 60 (explizit,statisch,T1,A1).
```

Anschließend findet eine *Transformation der LIBen* in sogenannte *Transformierte IBen (TIBen)* statt, da die Datenmodelltransformation auch eine entsprechende IB-Transformation bedingt. Dazu werden die IBen den Datenbank-Objekttypen und Attributen des Gemeinsamen Datenmodells (GDMs) zugeordnet. Bei dieser Transformation können IBen ihre Klassenzugehörigkeit ändern, indem sie z. B. von explizit formulierten in modellinhärente IBen des GDMs übergehen.

Der dritte Schritt beinhaltet die *Filterung der Schemata und IBen* und die *Vergabe von Schutzbedingungen*. Dabei werden nur die für die anschließende Integration relevanten IBen bezüglich des Exportschemas gefiltert und um *Export-IBen (EIBen)* erweitert. Diese EIBen können die Vergabe von Zugriffsrechten auf Daten des jeweiligen KDBSs beinhalten.

Der Schritt der *Integration von IBen* erfordert die Überführung der auf die einzelnen Exportschemata bezogenen IBen (in Abbildung 1: TIB1, EIB1, TIB2, EIB2) in *Integrierte IBen (IIBen)* des integrierten Schemas. Dies erfordert eine Auflösung von IB-Konflikten, die abhängig von den Anforderungen an das FDBS sind. Dabei können neue *Globale IBen (GIBen)* hinzukommen, die sich auf Interdatenbankabhängigkeiten beziehen. Als Beispiel seien hier folgende IIBen genannt:

```
IIS1: person.alter >= 20 (explizit,statisch,T1,A1),
IIS2: person.alter <= 60 (explizit,statisch,T1,A1).
```

Bei der *Ableitung externer Schemata und IBen* geht es um die Überführung der IIBen und GIBen in die IBen des Datenmodells des jeweiligen Exportschemas (Filterung). Das Hinzufügen von *Externen IBen (XIBen)* betrifft, ähnlich dem Schritt der Ableitung von Exportschemata und darin enthaltene IBen, die Filterung und Vergabe von Zugriffsrechten für den Nutzer.

3.3 Einhaltung von Integritätsbedingungen

Nach der Integration von IBen und dem Hinzufügen globaler IBen stellt sich die Frage nach der Einhaltung solcher IBen. Bei der Einhaltung ist zu klären, an welcher Stelle des FDBSs, wann die entsprechende IB überprüft werden soll und welche Reaktionen im Verletzungsfall ablaufen sollen.

Innerhalb eines FDBSs gibt es verschiedene Schichten, an denen solche IBen eingehalten werden können. Diese Schichten und die dort auftretenden IBen werden in Abbildung 2 deutlich.

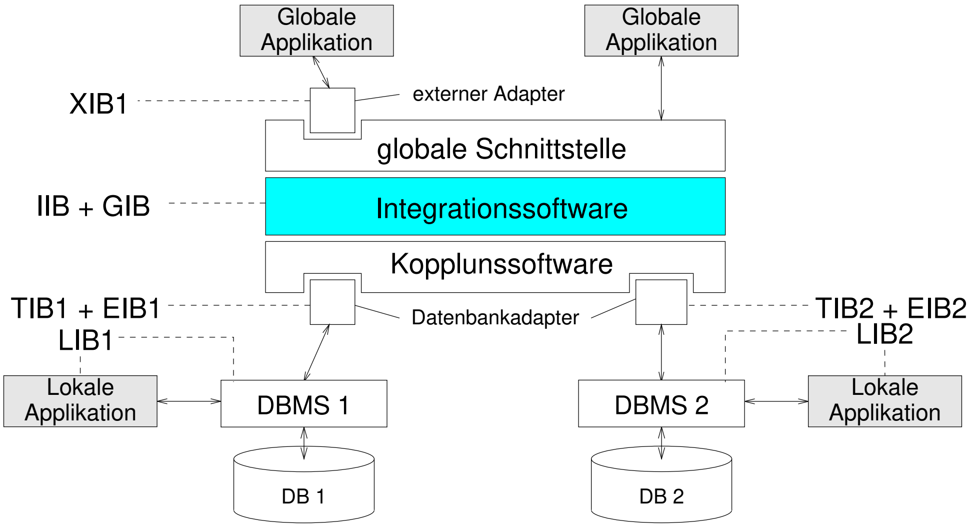

Abbildung 2: Überprüfbarkeit von IBen nach Auftreten in Schichten des FDBSs.

Lokale Integritätsbedingungen (LIBen), können durch das jeweilige KDBS überwacht werden, sofern diese Überwachungsfunktion im KDBS gegeben ist. Auf die Verletzung der LIBen durch eine globale Anwendung reagiert die entsprechende lokale Komponente. Besteht jedoch die Anforderung einer schnellen Reaktion im Fehlerfall, so sollte diese LIB in Form der zugehörigen TIB oder IIB durch den Datenbankadapter oder die Integrationssoftware überprüft werden. Ebenfalls sollten die in den Anwendungen überwachten LIBen in den Integrationsprozeß mit einbezogen werden. Diese sind dann in den darüberliegenden Schichten des FDBSs zu überwachen, da ansonsten lokale Inkonsistenzen entstehen können.

Zusätzliche EIBen können im Datenbankadapter oder innerhalb der Integrationssoftware überprüft werden. Können neue Applikationen direkt auf den Datenbankadapter zugreifen, so sollten die EIBen innerhalb des jeweiligen Datenbankadapters überwacht werden.

Die aus der Integration resultierenden IIBen können innerhalb der Integrationssoftware überwacht werden. Kommen innerhalb der globalen Schicht neue IBen hinzu, die nur eines der beteiligten KDBSe betreffen, so kann man versuchen, diese auf LIBen abzubilden und durch das KDBS überwachen zu lassen. Dies schränkt natürlich die Autonomie der KDBSe ein. Bestehen systemübergreifende IBen (dies können IIBen und GIBen sein), die durch lokale Datenbankoperationen verletzt werden können, so sind auch die lokalen Systeme geeignet in eine solche Überwachung einzubeziehen. Ein solcher Ansatz zur Wahrung globaler Integrittsbedingungen wird in [TC96] verfolgt. Ausgehend von einem globalen aktiven Regelmanagement können globale Integritätsregeln (IB zzgl. Aspekte der Überprüfung und Reaktion bei Verletzung) formuliert und durch Kooperation mit dem jeweiligen lokalen Regelmanagement überwacht werden.

Neben den Fragen der Zuordnung der Überprüfbarkeit zu den Schichten des FDBS bestehen weiterhin die Fragen nach der Reaktion im Fehlerfall, nach einem geeigneten Transaktionsmodell, nach der Realisierung der Kommunikation, etc.

4 Diskussion und Ausblick

Die hier aufgezeigten Schritte der Integration von Integritätsbedingungen sind ein erster Ansatzpunkt. Ziel ist es, eine Methodik zu entwickeln, anhand derer man den Integrationsprozeß bestehender Datenbank- und Datenhaltungssysteme in Abhängigkeit von Anforderungen an das FDBS vornehmen kann. Besonderes Augenmerk soll dabei auf die Erhaltung von datenbankübergreifenden Integritätsbedingungen gelegt werden, da sich die Notwendigkeit einer solchen Integration im genannten Forschungsprojekt $SIGMA_{FDB}$ gerade aus der semantischen Überlappung der Daten der am Fabrikplanungsprozeß beteiligten Systeme ergibt. Ein wichtiger Punkt ist auch die Einbeziehung von Datenhaltungsssystemen ohne oder mit geringfügiger Datenbankfunktionalität (Transaktionen, Mehrfachzugriff, Integritätssicherung, etc.). Die in dem Papier aufgeworfenen Fragen und Anforderungen müssen konkretisiert werden, um eine Methodik zur Integration entwickeln zu können. Ist eine entsprechende Methodik gefunden, ist eine Unterstützung der Schritte des FDB-Entwurfs durch geeignete Softwarewerkzeuge denkbar.

Literatur

[BLN86] C. Batini, M. Lenzerini und S. B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys*, 18(4):323–364, Dezember 1986.

[CHJ+96] S. Conrad, M. Höding, S. Janssen, G. Saake, I. Schmitt und C. Türker. Integrity Constraints in Federated Database Design. Preprint Nr. 2, Fakultät für Informatik, Universität Magdeburg, April 1996.

[GR94] R. Gabriel und H.-P. Röhrs. *Datenbanksysteme – Konzeptionelle Datenmodellierung und Datenbankarchitekturen*. Springer-Verlag, 1994.

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

[HTJ+95] M. Höding, C. Türker, S. Janssen, K.-U. Sattler, S. Conrad, G. Saake und I. Schmitt. Föderierte Datenbanksysteme — Grundlagen und Ziele des Projektes $\mathbf{SIGMA}_{FDB}$. Preprint Nr. 12, Fakultät für Informatik, Universität Magdeburg, Dezember 1995.

[RPG95] M. P. Reddy, B. E. Prasad und A. Gupta. Formulating Global Integrity Constraints during Derivation of Global Schema. *Data & Knowledge Engineering*, 16(3):241–268, Juli 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.

[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*, Nummer Bericht 96–01, S. 51–65. Institut für Technische Informationssysteme, Universität Magdeburg, April 1996.