Unternehmensweite Datenkonsistenz durch Integration bestehender Informationssysteme*

Stefan Conrad Can Türker

Otto-von-Guericke-Universität Magdeburg
Institut für Technische Informationssysteme
Postfach 4120, D–39016 Magdeburg, Deutschland
E-mail: {conrad|tuerker}@iti.cs.uni-magdeburg.de

Zusammenfassung

In diesem Papier diskutieren wir die Problematik der globalen, unternehmensweiten Datenkonsistenz und präsentieren einen Ansatz zur systemübergreifenden Konsistenzsicherung in föderierten heterogenen Informationssystemen.

Unser Vorschlag basiert darauf, aktive Mechanismen sowohl auf Föderierungsebene wie auch auf lokaler Komponentensystemebene für das Erkennen globaler Integritätsverletzungen einzusetzen. Aktive Mechanismen in Form von ECA-Regeln bieten die Möglichkeit, flexibel auf konsistenzverletzende Ereignisse zu reagieren.

Die systemübergreifende Konsistenz der Daten ist eine wesentliche Voraussetzung für die Realisierung und den Betrieb von Workflow-Management-Systemen, die mit verschiedenen Altsystemen kooperieren sollen.

1 Einleitung

Die gegenwärtige Situation in Unternehmen bzw. allgemein in Organisationen ist gekennzeichnet dadurch, daß im Laufe der Zeit eine Reihe von Informationssystemen unabhängig voneinander für zumeist unterschiedliche, aber manchmal auch überlappende Aufgaben entstanden sind (z.B. für Personalverwaltung, Auftragsabwicklung, Lagerhaltung, Konstruktion und Entwicklung, etc.).

Die Integration dieser verschiedenen Informationssysteme innerhalb eines Unternehmens kann aber zu einem wettbewerbsentscheidenden Faktor werden, denn Qualität und Verfügbarkeit von Daten innerhalb eines Unternehmens (aber auch zwischen kooperierenden Unternehmen) haben einen wesentlichen Einfluß auf den Erfolg. Daher ist die Integration eine notwendige Voraussetzung, um z.B. durch die Einführung von unternehmensweiten Workflow-Managementsystemen [Jab95] Abläufe in dem Unternehmen zu optimieren. Nur so kann langfristig ein Unternehmen effizient auf die sich ständig ändernden Anforderungen des Marktes reagieren.

Ein besonderer Aspekt föderierter Informationssysteme, den wir in diesem Beitrag näher betrachten wollen, ist die systemübergreifende Konsistenz der Daten. Während

*Diese Arbeit wurde teilweise gefördert durch das Land Sachsen-Anhalt under FKZ: 1987A/0025 "Föderierung heterogener Datenbanksysteme und lokaler Datenhaltungskomponenten zur systemübergreifenden Integritätssicherung" und durch die ESPRIT Basic Research Working Group No. 8319 ModelAge (A Common Formal **Model** of Cooperating Intelligent **Age**nts)

bisher die Daten separat in verschiedenen Systemen gehalten wurden, war eine systemgestützte Gewährleistung der Datenkonsistenz über Systemgrenzen hinweg unmöglich. Durch die Föderierung entstehen hier neue Möglichkeiten, aber auch neue Anforderungen. Durch die Integration der Systeme werden nun Anfragen und Anwendungen über die Grenzen der einzelnen Systeme hinaus unterstützt. Da aber in der Regel die Daten, die in den verschiedenen Systemen gespeichert werden, nicht völlig unabhängig voneinander sind, verlangen solche Anfragen und Anwendungen nun auch eine systemübergreifende Integrität der Daten, um sinnvolle und weiterverwertbare Ergebnisse liefern zu können.

Der von uns verfolgte Ansatz zur systemübergreifenden Konsistenzsicherung in föderierten Datenbanken basiert im wesentlichen auf Konzepten aktiver Datenbanksysteme [BZBW95, WC96]. Diese nutzen wir, um potentielle Integritätsverletzungen von den beteiligten Datenbanksystemen an die übergeordnete Föderierungsschicht mitzuteilen. In der Föderierungsschicht erlauben uns diese Konzepte, in flexibler Weise auf mögliche Integritätsverletzungen zu reagieren, indem wir z.B. kompensierende Maßnahmen vorgeben können.

Im folgenden werden wir zunächst kurz die wichtigsten Konzepte und Begriffe föderierter Datenbanken erläutern und zwei typische Anwendungsszenarien präsentieren (Kapitel 2). Danach stellen wir in Kapitel 3 unseren Ansatz zur Bewahrung unternehmensweiter Datenkonsistenz auf der Basis föderierter Datenbanksysteme unter Verwendung von Mechanismen aktiver Datenbanksysteme dar. Abschließend fassen wir die wichtigsten Aspekte zusammen und geben einen kurzen Ausblick auf laufende und zukünftige Arbeiten.

2 Föderierte Informationssysteme

Unter einem *föderierten Informationssystem* (FIS) verstehen wir ein *föderiertes Datenbanksystem* (FDBS) und Anwendungen zur Verknüpfung und Auswertung von Informationen [SL90]. Ein *föderiertes Datenbanksystem* besteht aus einer Menge von *autonomen heterogenen Komponentendatenbanksystemen*, die über eine flexible Föderierungsschicht kooperieren. Die wesentliche Aufgabe dieser *erweiterbaren* Föderierungsschicht besteht darin, eine globale, homogene Schnittstelle für den einheitlichen und transparenten Zugriff auf föderierte Daten bereitzustellen und dabei die lokalen Datenbankapplikationen so wenig wie möglich zu beeinträchtigen. Jedes beteiligte Informationssystem stellt der Föderation ein vom jeweiligen Datenbankadministrator erstelltes *Exportschema* zur Verfügung.

Im folgenden wollen wir kurz zwei unterschiedliche Anwendungsszenarien skizzieren, in denen föderierte Informationssysteme sinnvoll und gewinnbringend eingesetzt werden können.

Anwendungsszenario 1: Ein Dienstleistungsunternehmen, etwa ein Reisebüro, benötigt zur Erstellung und Planung von Reiseangeboten einen Zugriff auf die lokale Datenbank Reise DB und auf Informationen verschiedener unabhängiger Datenbanken der Bereiche Hotel, Bahn und Fluggesellschaft 1 (siehe Abb. 2). Ebenso benötigt das Informationssystem des Flughafen Zugriff auf Datenbanken verschiedener Fluggesellschaften (1, 2 und n), die ihre Dienste auf diesem Flughafen anbieten.

Zur Ermöglichung der Teilnahme der lokalen Datenbanksysteme an einer Föderation werden vom jeweiligen Datenbankadministrator Exportschemata bereitgestellt. Durch die

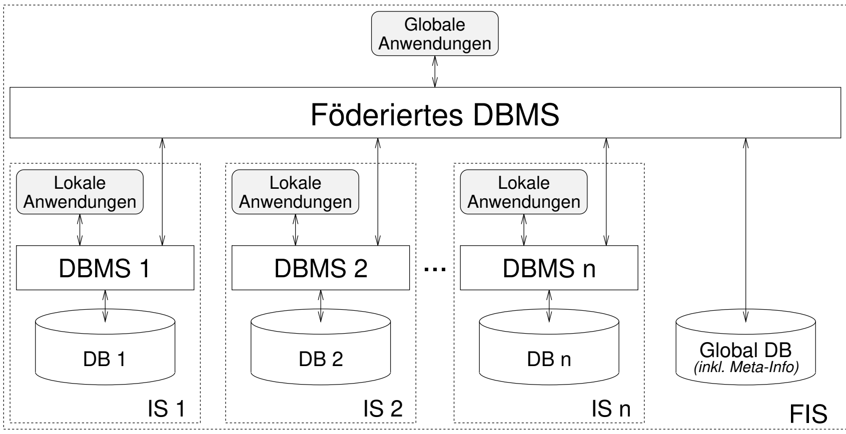

Abbildung 1: Architektur föderierter Informationssysteme.

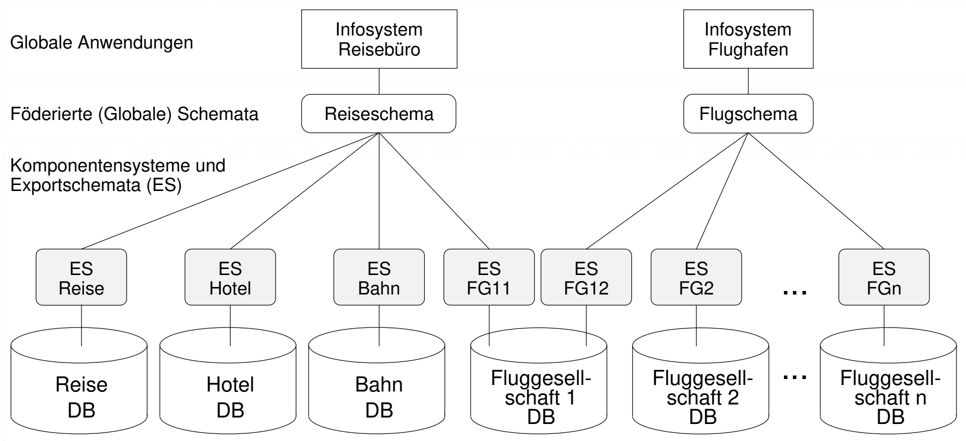

Abbildung 2: Unternehmensübergreifende Föderation.

Teilnahme der Datenbank der Fluggesellschaft 1 an zwei Föderationen werden dementsprechend zwei Schemata Exportschema FG 11 und 12 abgeleitet. Vom Anwendungsentwickler des föderierten Informationssystems werden diese Exportschemata je nach Anwendung zu einem föderierten Schema (Reiseschema bzw. Flugschema) zusammengefaßt. Im Beispiel des Informationssystems Reisebüro sind im Reiseschema alle Informationen der Exportschemata zu Hotels (Anzahl freier Zimmer/Betten, Preise, Lage, etc.), Bahnlinien (Fahrpläne, Preise), Fluggesellschaft 1 (Flugpläne, Preise, Service) zugänglich. Diese Informationen sind bei der Reise- und/oder Urlaubsplanung des Kunden durch das Reisebüro relevant. Im föderierten Schema Flugschema des Informationssystem des Flughafens sind alle für dieses System relevanten Informationen zugänglich (z. B. Flugpläne der einzelnen Gesellschaften, Serviceangebote, Preisvergleiche). Die föderierten Schemata beinhalten also nur die Informationen der einzelnen Exportschemata, die sie benötigen.

Anwendungsszenario 2: Eine andere Einsatzform ist bei unternehmensinternen Föderationen gegeben. Ein produzierendes Unternehmen (siehe Abb. 3) will zur Optimierung seiner Geschäftsprozesse Unternehmensdaten aus verschiedenen Abteilungen unter einer einheitlichen globalen Schicht integrieren und dabei die vorhandenen lokalen Applikationen sowenig wie möglich verändern. Die Daten befinden sich in den Datenbanken Konstruktion, Planung/Produktion und Lager und überlappen sich teilweise, d.h. Daten

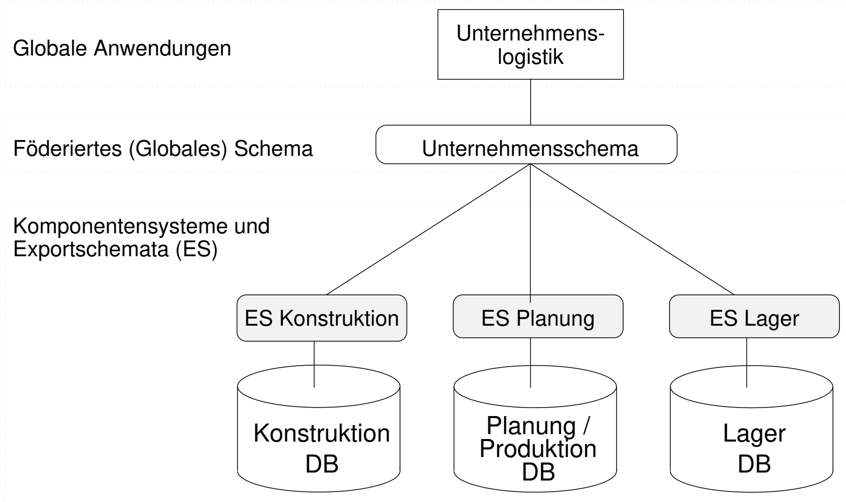

Abbildung 3: Unternehmensinterne Föderation.

werden redundant gehalten. Der Bereich Konstruktion verwaltet Konstruktionsdaten der erzeugten Produkte. Die Planung und Produktion verwaltet Maschinenaufstellpläne, Arbeitspläne, Ressourcendaten, etc. Die Lagerhaltung greift auf spezifische Lagerdaten wie etwa Produktnummern, Mengen und Lagerorte zu. Anforderungen an ein solches föderiertes Informationssystem sind die unternehmensweit einheitliche Beschreibung und Verwendung von Daten (einheitliche Namensgebung), die Auswertung von Informationen aller Bereiche für das Management (Kosten für Konstruktion/Planung/Fertigung, Prozeßoptimierung, statistische Erhebungen), die Sicherung globaler Integritätsbedingungen (z.B. eindeutige Produktnummern oder aggregrierte Bedingungen, etwa daß die Summe der Konstruktions- und Planungskosten kleiner sein muß als die Produktionskosten), etc.

Ein wichtiger Aspekt hierbei ist die Definition und Überwachung globaler Integritätsbedingungen, auf den wir im folgenden eingehen werden.

3 Aktive Föderationsplattform zur Integritätssicherung unternehmensweiter Daten

In diesem Kapitel präsentieren wir eine Rahmenarchitektur für aktive föderierte Datenbanken, die als Integrationsplattform zur Konsistenzsicherung unternehmensweiter Daten eingesetzt werden kann. Darüber hinaus skizzieren wir exemplarisch, wie Integritätsbedingungen in einer solchen Umgebung spezifiziert und durchgesetzt werden können.

3.1 Aktive Regeln zur Spezifikation und Durchsetzung von unternehmensweiten Integritätsbedingungen

Für die Integritäts- und Konsistenzsicherung sind verschiedene Ansätze bekannt (z.B. [Lip89, CW91]). In der Regel werden integritätsverletzende Operationen durch die Transaktionsverwaltung verhindert bzw. zurückgewiesen, d.h. die zugehörige Transaktion abgebrochen. Eine flexiblere Behandlung von Inkonsistenzen bieten Regelmechanismen, die dazu genutzt werden, (beliebige) konsistenzwiederherstellende Operationen im Konfliktfall einzuleiten.

In den letzten Jahren hat sich das Konzept der *Event-Condition-Action-Regeln* (kurz *ECA-Regeln*) im Bereich aktiver Datenbank zur Beschreibung aktiver Fähigkeiten durchgesetzt [DBB+88]. Mit ECA-Regeln können Situationen im Ablauf eines Programms spezifiziert werden, auf die das System (aktiv) mit der Ausführung entsprechender Operationen reagieren soll. Wir möchten den ECA-Regel-Mechanismus dazu verwenden, Ereignisse, die potentiell die unternehmensweite Datenintegrität verletzen können, zu erkennen und im Falle einer Integritätsverletzung mit der Ausführung integritätswiederherstellender Aktionen zu reagieren:

define rule *Unternehmensweite_Integritäts_Regel*
on *potentiell konsistenzverletzendes Ereignis*
if *unternehmensweite Integritätsbedingung ist verletzt*
do *konsistenzwiederherstellende Aktion(en)*

Wir betonen ausdrücklich, daß es nicht das Ziel ist, globale Integritätsregeln in Form von ECA-Regeln zu spezifizieren. Vielmehr soll der ECA-Regel-Mechanismus zur Realisierung deskriptiv formulierter Integritätsbedingungen eingesetzt werden. Hierzu sind mit Hilfe des Datenkatalogs der Föderation, der bei der Integration der Komponentensysteme angelegt wird, ausführbare ECA-Regeln zu generieren. Dies kann bestenfalls halbautomatisch geschehen. Oft ist jedoch das Eingreifen des Regelentwerfers (in der Regel der Datenbankadministrator der Föderierung) notwendig. Zum Beispiel muß im Falle einer globalen Schlüsselbedingung anhand der Metadaten festgestellt werden, welche lokalen Änderungsoperationen auf den entsprechenden Attributen bzw. welche lokalen Einfüge- oder Objekterzeugungsoperation der korrespondierenden Relationen bzw. Klassen davon betroffen sind. Für jede dieser Operationen muß dann eine ECA-Regel derart generiert werden, daß die Operationsanforderung bzw. das Ende der Operationsausführung der zuständigen Integritätssicherungskomponente angezeigt werden. Da diese Regeln den Komponentensystemen zugeordnet sind, um lokale Ereignisse zu erkennen, sprechen wir hier von *lokalen ECA-Regeln*. Zur Koordination dieser lokalen ECA-Regeln wird eine ECA-Regel auf der Föderierungebene benötigt, die die Signale der lokalen Regeln auswertet und im Falle einer Integritätsverletzung geeignete Korrekturmaßnahmen einleitet. Solche ECA-Regeln bezeichnen wir als *globale* ECA-Regeln. Man beachte, daß ECA-Regeln somit als Basis-Kommunikations- bzw. Koordinationsmechanismus zwischen der Föderierungsschicht und den Komponentensystemen eingesetzt werden.

3.2 Rahmenarchitektur für Aktive Föderierte Datenbanken

Zur Realisierung von lokalen und globalen ECA-Regeln benötigen wir aktive Mechanismen sowohl auf der lokalen wie auch auf der globalen Ebene eines föderierten Systems. In Abb. 4 wird eine um aktive Fähigkeiten erweiterte föderierte Datenbankarchitektur [TC95, CT95, HTJ+95] vorgestellt. Auf der Föderierungsebene wird eine Föderierungsschicht bereitgestellt, welche auch integrierte aktive Mechanismen à la REACH [BZBW95] anbietet. Diese können zur Realisierung von globalen ECA-Regeln herangezogen werden.

Die Realisierung der lokalen ECA-Regeln ist etwas schwieriger, da hierfür aktive Mechanismen auf Komponentensystemebene benötigt werden. Hier gibt es prinzipiell zwei Möglichkeiten. Falls bereits aktive Mechanismen, z.B. Trigger, auf lokaler Ebene existieren, werden diese zur Erkennung und Weiterleitung von lokalen Ereignissen eingesetzt. Wenn lokal jedoch keinerlei aktive Funktionalität zur Verfügung steht bzw. die bereitgestellte Funktionalität nicht zur Weiterleitung von Ereignissen außerhalb des Komponen-

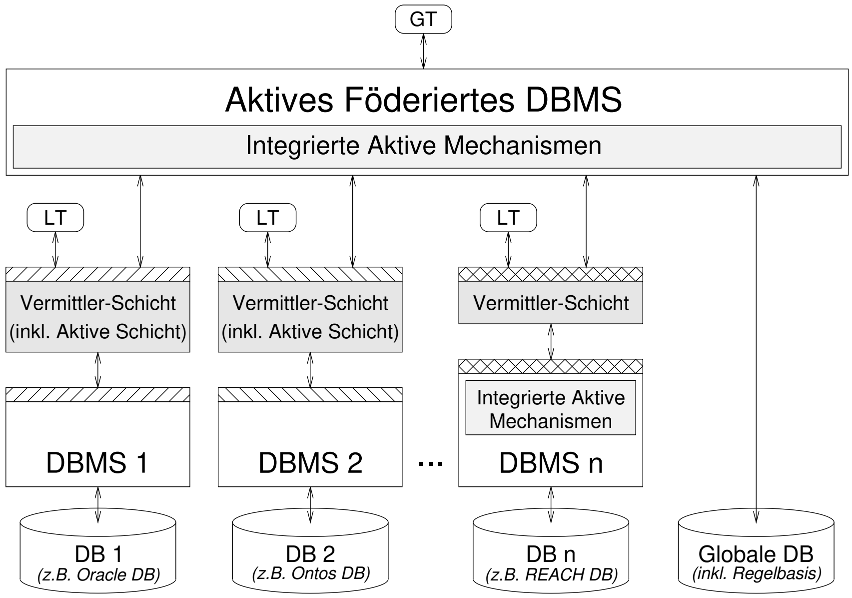

Abbildung 4: Aktive föderierte Datenbank-Rahmenarchitektur.

tensystems möglich ist, wird das entsprechende Komponentensystem mit einer zusätzlichen Vermittler-Schicht versehen. Alle Anforderungen lokaler Transaktionen (LT) an das Komponentensystem werden über diese Schicht „umgeleitet". In dieser Schicht können dann lokale Ereignisse vor bzw. nach der Ausführung einer Operation erkannt und an die Föderierungsschicht weitergemeldet werden. Damit existierende lokale Anwendungen weiterhin unverändert laufen können, bietet die Vermittler-Schicht die gleiche Schnittstelle wie das originale Komponentensystem.

Das Zusammenspiel der globalen und lokalen ECA-Regeln läuft wie folgt ab. Das Erkennen von Ereignissen, die sich ausschließlich auf die globale Datenbank beziehen, erfolgt bereits auf der Föderierungsebene, während für das Erkennen der übrigen (*primitiven*) Ereignisse die Kooperation/Kommunikation mit den Komponentensystemen erforderlich ist. In diesen Fällen müssen die durch die lokalen ECA-Regeln spezifizierten Ereignisse von den einzelnen Systemen (durch *lokale* Regelmanager in den Vermittler-Schichten) erkannt und an den globalen Regelmanager der Föderierungschicht weitergeleitet werden. Dieser Regelmanager ist dann für die Komposition von *zusammengesetzten* Ereignissen sowie für die Verarbeitung der entsprechenden globalen ECA-Regeln verantwortlich. Hierzu muß der globale Regelmanager zunächst Maßnahmen zur Auswertung der spezifizierten Bedingungen einleiten und im Falle einer erfolgreichen Bedingungsauswertung die notwendigen Aktionen starten, indem er entweder Operationen (Funktionen bzw. Methoden) auf der globalen Datenbank ausführt oder entsprechende Operationen der Komponentensysteme aufruft. Somit wird garantiert, daß jederzeit globale wie auch lokale Transaktionen zurückgewiesen werden können, wenn sie die Integrität der föderierten Datenbank verletzen.

Auch wenn wir hier nur globale Schlüsselbedingungen betrachtet haben, erlaubt unser Ansatz auch die Überprüfung und Gewährleistung anderer wichtiger Arten von Integritätsbedingungen auf globaler Ebene. Insbesondere Bedingungen, die referentielle Integrität zwischen Objekten in verschiedenen Datenbanken ausdrücken, spielen eine große Rolle in föderierten Systemen.

3.3 Beispiel für die Aktive Integritätssicherung

In diesem Unterabschnitt skizzieren wir exemplarisch, wie unternehmensweite Integritätsbedingungen in unserer aktiven föderierten Datenbankumgebung spezifiziert und durchgesetzt werden können. Zur Illustration verwenden wir ein typisches Anwendungsszenario, das häufig in Unternehmen mit mehreren Abteilungen anzutreffen ist (siehe zweites Beispiel in Kapitel 2). In jeder dieser Abteilungen (Konstruktion, Planung/Produktion und Lager) kommen verschiedene, heterogene Datenbanksysteme zum Einsatz, die entsprechend den Bedürfnissen der jeweiligen Abteilungen ausgewählt wurden. Das Unternehmen ist bestrebt, die im Laufe der Jahre unabhängig entstandenen Datenbestände (virtuell) zu integrieren, um dadurch schneller auf unerwünschte oder inkorrekte Zustände reagieren zu können. Wir nehmen an, daß in den Abteilungen unseres Beispielunternehmens folgende Komponentensysteme eingesetzt werden:

- In der Planung/Produktion wird ein relationales Datenbanksystem verwendet, das einfache Trigger-Mechanismen anbietet, zur Verwaltung der allgemeinen Daten über geplante, laufende und abgeschlossene Projekte, Produktionsprozesse, Produkte, etc.
- In der Konstruktionsabteilung werden (technische) Daten über Produkte in einem objektorientierten Datenbanksystem verwaltet, das keinerlei aktive Mechanismen unterstützt.
- Im Lager kommt ein aktives objektorientiertes Datenbanksystem, das ECA-Regeln anbietet, zur Erfassung der Produktionsbestände zum Einsatz.

Nehmen wir weiterhin an, daß die bestehenden Datenbankschemata der lokalen Systeme durch einen föderierten Datenbankentwurfsprozeß adäquat in ein objektorientiertes föderiertes Schema integriert wurden (vgl. [SS96]). Im folgenden wollen wir kurz skizzieren, wie eine unternehmenweite Schlüsselbedingung spezifiziert und anhand von ECA-Regeln durchgesetzt werden kann.

Obwohl in objektorientierten Datenbanksystemen Objekte durch eindeutige, interne Identifikatoren unterschieden werden, wird im allgemeinen der Schlüsselmechanismus unterstützt, um dem Benutzer die Möglichkeit zu bieten, Objekte nach eigenen Kriterien unterscheiden zu können. Das Schlüsselkonzept spielt auch in föderierten Datenbanken eine wichtige Rolle. Jedes Objekt muß nicht nur im Kontext des Komponentensystems eindeutig identifizierbar sein, sondern auch im Kontext der föderierten Datenbank. Ein Beispiel dafür ist, daß alle Produkte in dem Unternehmen eine eindeutige Produktnummer haben müssen. Solche komponentensystemübergreifenden Bedingungen würde man im föderierten Schema deklarativ wie folgt definieren (gemäß ODMG-93-Notation):

```
interface Produkt
( extent Produkte
  key     produktnummer )
{
  attribute string<6> produktnummer;
  ...
};
```

Abstrakt könnte diese globale Schlüsselbedingung folgendermaßen als ECA-Regel spezifiziert werden:

define rule *Unternehmensweite_Schlüsselbedingungs_Regel*
on *Ändern der Produktnummer oder Erzeugen eines Produkt-Objekts*
if *Unternehmensweite Schlüsselbedingung ist verletzt*
do *Abbruch der triggernden Transaktion*

Die Realisierung dieser globalen Schlüsselbedingung erfolgt in der vorgeschlagenen Architektur wie folgt. Da Produkte in allen drei Komponentendatenbanken verwaltet werden, müssen alle entsprechenden Änderungsoperationen auf den lokalen Produkt-Objekten überwacht werden, d.h. mit lokalen ECA-Regeln versehen werden. Für die Konstruktionsdatenbank muß z.B. eine lokale ECA-Regel für den Konstruktoraufruf der Konstruktionsklasse abgeleitet werden.

define rule *Lokale-Regel-1*
on *Konstruktion::Konstruktion(Parameter)*
do *GDB::notify(Globale-Regel, DB1::Konstruktion::Konstruktion(Parameter))*

Das Eintreten des Konstruktoraufrufs wird durch diese lokale Regel an die globale Koordinationsregel weitergeleitet, die dann die notwendigen Maßnahmen zur Überprüfung der Integritätsbedingung einleitet. In diesem Fall muß die globale Regel in allen anderen Komponentendatenbanken nachschauen, ob bereits ein Produkt mit dieser Produktnummer existiert. Falls dies der Fall ist, muß die triggernde Änderungsoperation zurückgesetzt werden.

Man beachte, daß durch die existierenden ECA-Regelsprachen nahezu beliebige Integritätsbedingungen ausgedrückt werden können. Insbesondere können temporale Bedingungen zur Spezifikation langfristiger Entwicklungen der föderierten Datenbank angegeben werden.

3.4 Andere Ansätze

Die von uns vorgestellte Verwendung aktiver Mechanismen zur globalen Interitätssicherung kann als Ergänzung zu einer Vielzahl von aktuellen Vorschlägen für die Transaktionsverwaltung in Multidatenbanksystemen genutzt werden. Daher schlagen wir auch kein eigenes Transaktionskonzept für föderierte Informationssysteme vor. Im folgenden gehen wir daher kurz auf einige bereits bestehende Transaktionsmodelle ein.

In [SRK92] werden sogenannte Polytransaktionen eingeführt, mit denen Interdatenbankabhängigkeiten und ihre Auswirkungen bei der Datenmanipulation behandelt werden. In diesem Ansatz können verschiedene Arten von Integritätsbedingungen bereits explizit beschrieben werden und Wiederherstellungsprozeduren definiert werden, die im Falle der Integritätsverletztung durch die Transaktionsverwaltung angewendet werden.

In [BÖH+92] wird ein Transaktionmodell für aktive verteilte Objektsysteme vorgeschlagen: das DOM Transaktionsmodell. Dieses Modell umfaßt verschiedene Arten von Transaktionen, z.B. geschachtelte, vitale und nicht-vitale, Kompensations- und Ausweich-Transaktionen. Diese Vielfalt von Transaktionsarten erlaubt es, die Integritätssicherung in dem föderierten System auf die speziellen Bedürfnisse der Anwendungen geeignet zuzuschneidern.

Auch in anderen Arbeiten (z.B. in [SSW95]) werden vergleichbare Transaktionsmodelle vorgeschlagen, die im wesentlichen auf offen geschachtelten Transaktionen basieren und so ein frühes Beenden von Subtransaktionen ermöglichen.

Für alle diese Transaktionsmodelle ist die von uns vorgeschlagene Architektur mit aktiven Mechanismen geeignet. Allerdings sei hier deutlich darauf hingewiesen, daß unser Ansatz, der speziell auf die Aspekte der systemübergreifenden Integritätssicherung in föderierten Informationssystemen zugeschnitten ist, im allgemeinen nicht die Probleme der Transaktionsverwaltung in föderierten Systemen lösen kann (für einen Überblick siehe z.B. [BGS95]).

4 Resümee und Ausblick

In diesem Beitrag haben wir einen Ansatz zur Gewährleistung unternehmensweiter Datenkonsistenz auf der Basis föderierter Datenbanksysteme unter Verwendung aktiver Regeln vorgestellt. Aktive Regeln erlauben eine flexible Umsetzung systemübergreifender Integritätsbedingungen. Gerade in heterogenen Systemumgebungen benötigen wir solche übergeordneten Mechanismen, um potentielle Integritätsverletzungen über Systemgrenzen hinweg mitteilen zu können. Darüber hinaus kann in einheitlicher Weise die Reaktion auf Integritätsverletzungen festgelegt werden.

Einige technische Detailfragen sind noch zu lösen, z.B. die Integration unseres Ansatzes in bestehende Transaktionsmodelle für föderierte Datenbanksysteme (siehe auch [CT95]). Es ist aber schon jetzt festzuhalten, daß die Bedeutung unternehmensweiter Datenkonsistenz nicht hoch genug eingeschätzt werden kann. Die fehlende Integration der in einem Unternehmen vorhandenen Datenbestände führt zwangsläufig zu Inkonsistenzen in dem Sinne, daß verschiedene Abteilungen mit unterschiedlichen, zum Teil nicht zueinander passenden Daten arbeiten. Dies kann im Extremfall sogar zu unternehmerischen Fehlentscheidungen aufgrund unzuverlässiger Daten führen. In der alltäglichen Praxis führt eine solche Situation zu einer Vielzahl von zusätzlichen zeitaufwendigen und damit letztendlichen teuren Arbeitsvorgängen, um z.B. Daten zwischen Abteilungen aufeinander abzugleichen.

Aus diesen und weiteren Gründen wird die unternehmensweite Integration von bestehenden und neu zu entwickelnden Informationssystemen zu einem immer wichtigeren Wettbewerbsfaktor. Eine solche Integration ist die Voraussetzung für die Optimierung von Geschäftsabläufen und für die schnelle Anpassung an sich ändernde Wettbewerbsbedingungen. Dabei darf die Autonomie der bestehenden Systeme nicht unmäßig eingeschränkt werden, um auch bestehende Anwendungen weiterhin ablaufen lassen zu können. Hier bieten die im Bereich föderierter Datenbanken entwickelten Konzepte adäquate Umsetzungsmöglichkeiten. Der von uns in diesem Beitrag vorgeschlagene Ansatz zur systemübergreifenden Integritätssicherung in föderierten Systemen ergänzt diese Konzepte derart, daß die Realisierung einer unternehmensweiten Datenintegrität in greifbare Nähe rückt.

In diesem Beitrag haben wir Ergebnisse aus einem laufenden Projekt [HTJ+95] präsentiert. Gegenwärtig arbeiten wir an der Integration unseres Ansatzes mit einem konkreten Transaktionsmodell. Begleitend dazu entwickeln wir eine Methodik für die Föderierung bestehender heterogener Datenbanken unter besonderer Berücksichtigung von Integritätsbedingungen [CHJ+96]. Nachdem in einem solchen Schematransformations- und Integrationsprozeß globale Integritätsbedingungen gefunden und deklarativ formuliert worden sind, müssen sie auf globale und lokale ECA-Regeln abgebildet werden. Da dies nur bedingt automatisiert werden kann, werden wir hierfür eine methodische Unterstützung für den globalen Datenbankadministrator entwickeln.

Literatur

[BGS95] Y. Breitbart, H. Garcia-Molina und A. Silberschatz. Transaction Management in Multidatabase Systems. In W. Kim, Hrsg., *Modern Database Systems*, S. 573–591. ACM Press, 1995.

[BÖH+92] A. Buchmann, M. T. Özsu, M. Hornick, D. Georgakopoulos und F. Manola. A Transaction Model for Active Distributed Object Systems. In A. K. Elmagarmid, Hrsg., *Database Transaction Models for Advanced Applications*, S. 123–151. Morgan Kaufmann Publishers, 1992.

[BZBW95] A. P. Buchmann, J. Zimmermann, J. A. Blakeley und D. L. Wells. Building an Integrated Active OODBMS: Requirements, Architecture, and Design Decisions. In P. S. Yu und A. L. P. Chen, Hrsg., *Proc. 11th IEEE Int. Conf. on Data Engineering (ICDE'95)*, S. 117–128. IEEE Computer Society Press, 1995.

[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, 1996.

[CT95] S. Conrad und C. Türker. Active Integrity Maintenance in Federated Database Systems. Preprint Nr. 9, Fakultät für Informatik, Universität Magdeburg, 1995.

[CW91] S. Ceri und J. Widom. Deriving Production Rules for Incremental View Maintenance. In S. Baker, R. Agrawal und D. Bell, Hrsg., *Proc. 17th Int. Conf. on Very Large Data Bases (VLDB'91)*, S. 577–589. Morgan Kaufmann Publishers, 1991.

[DBB+88] U. Dayal, B. Blaustein, A. Buchmann, S. Chakravarthy, D. Goldhirsch, M. Hsu, R. Ladin, D. McCarthy und A. Rosenthal. The HiPAC Project: Combining Active Databases and Timing Contraints. *ACM SIGMOD RECORD*, 17(1):51–70, 1988.

[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, 1995.

[Jab95] S. Jablonski. Workflow-Management-Systeme: Motivation, Modellierung, Architektur. *GI Informatik Spektrum*, 18(1):13–24, 1995.

[Lip89] U. W. Lipeck. *Dynamische Integrität von Datenbanken.* Informatik Fachberichte 209, Springer-Verlag, 1989.

[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.

[SRK92] A. Sheth, M. Rusinkiewicz und G Karabatis. Using Polytransaction to Manage Interdependent Data. In A. K. Elmagarmid, Hrsg., *Database Transaction Models for Advanced Applications*, S. 555–581. Morgan Kaufmann Publishers, 1992.

[SS96] I. Schmitt und G. Saake. Integration of Inheritance Trees as Part of View Generation for Database Federations. In *Proc. 15th Int. Conf. on Conceptual Modelling (ER'96).* LNCS, Springer-Verlag, 1996. *To appear.*

[SSW95] W. Schaad, H.-J. Schek und G. Weikum. Implementation and Performance of Multilevel Transaction Management in a Multidatabase Environment. In O. Bukhres, T. Özsu und M.-C. Shan, Hrsg., *Proc. 5th Int. Workshop on Research Issues on Data Engineering: Distributed Object Management (RIDE-DOM'95)*, S. 108–115. IEEE Computer Society Press, 1995.

[TC95] C. Türker und S. Conrad. Aktive Mechanismen zur Konsistenzsicherung in föderierten Datenbanksystemen. In C. Eckert, H.-J. Klein und T. Polle, Hrsg., *7. GI-Workshop "Grundlagen von Datenbanken"*, S. 148–152. Hildesheimer Informatik Berichte, 1995.

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