Erschienen in: Eckert, C. and Klein, H.-J. and Polle, T., (Hrsg.), 7. GI-Workshop "Grundlagen von Datenbanken", Hildesheimer Informatik Berichte, Seiten 148–152, Juni 1995.

Aktive Mechanismen zur Konsistenzsicherung in föderierten Datenbanksystemen*

Can Türker   Stefan Conrad

Otto-von-Guericke-Universität Magdeburg
Institut für Technische Informationssysteme
Postfach 4120, D–39016 Magdeburg
E-mail: {tuerker|conrad}@iti.cs.uni-magdeburg.de
Tel.: ++49-391-67-{12994|18066}   Fax: ++49-391-67-12020

Zusammenfassung

In diesem Papier präsentieren wir die Idee eines beabsichtigten Projektes, das sich mit der Problematik der systemübergreifenden Konsistenzsicherung in föderierten heterogenen Datenbankumgebungen beschäftigt. Dazu sollen auf einer übergeordneten Ebene aktive Mechanismen realisiert werden, welche die unterschiedlichen Möglichkeiten der Integritätssicherung der darunterliegenden Einzelsysteme ausnutzen und verbinden.

Schlüsselwörter: Konsistenzsicherung, ECA-Regeln, aktive Datenbanken, föderierte Datenbanksysteme.

1 Einleitung

Der aktuelle Trend im Bereich moderner Informationssysteme geht dahin, bestehende (heterogene) Systeme zu einem umfassenden System zu föderieren, wobei die Einzelsysteme ihre Autonomie nicht vollständig aufgeben. Dies hat den Vorteil, daß bestehende Anwendungen unverändert weiterlaufen können, während neuen Anwendungen die Möglichkeit geboten wird, föderierte Daten verwenden zu können [SL90].

Ein wesentlicher Aspekt beim Zusammenschluß heterogener Systeme betrifft die Konsistenz der föderierten Daten. Obwohl die Föderierung von autonomen heterogenen (Datenbank-) Systemen Thema vieler Veröffentlichungen war, ist die systemübergreifende Konsistenzsicherung in solchen Systemen bislang kaum behandelt worden.

Aktive Mechanismen zur Konsistenzsicherung in aktiven Datenbanken (z.B. REACH [BZBW95], Sentinel [CKTB95], etc.) stellen ein aktuelles und vielversprechendes Prinzip zur Konsistenzsicherung in einzelnen, aber auch in föderierten Datenbanksystemen dar. Sie beruhen auf regelbasierter Konsistenzüberwachung und -erzwingung. Für potentiell konsistenzverletzende Aktionen werden Regeln definiert, die die für die jeweilige Situation geeigneten Korrekturmaßnahmen festlegen. Tritt eine solche Aktion (d.h. ein Ereignis)

*Diese Arbeit wurde teilweise gefördert durch die ESPRIT Basic Research Working Group No. 8319 ModelAge (A Common Formal **Model** of Cooperating Intelligent **Age**nts).

ein, wird die entsprechende Regel aktiviert, die dann die konsistenzwahrenden Maßnahmen initiiert (falls erforderlich). So ist es möglich, eine regelorientierte Wissensbasis zur Konsistenzsicherung des gesamten Datenbestandes anzulegen.

Die bislang erforschten Prinzipien aktiver Datenbanken beziehen sich ausschließlich auf einzelne Datenbanksysteme oder aber zumindest auf homogene Datenbanksysteme. Unser Ziel ist es, diese Prinzipien im Kontext föderierter, heterogener Datenbanksysteme zu analysieren und eventuelle Erweiterungen aktiver Mechanismen zu betrachten. Da die kommerziellen Datenbanksysteme in der Regel keine aktiven Mechanismen zur Konsistenzsicherung anbieten, sind die aktiven Mechanismen auf der Ebene des föderierten Systems, also datenbankübergreifend anzusiedeln. Jedoch bieten fast alle kommerziellen Datenbanksysteme eigene Integritätsüberwachungsmechanismen an, die aber nur einen Bruchteil der Möglichkeiten aktiver, regelbasierter Mechanismen leisten. Aus Effizienzgründen sollten diese mitgelieferten Mechanismen (wenn möglich) miteinbezogen werden. Angestrebt wird dabei eine Aufteilung derart, daß die aktiven, regelbasierten Mechanismen die globale Kontrolle bei der Konsistenzsicherung übernehmen und die lokal zu behandelnden Teile an die einzelnen Datenbanksysteme weitergeben.

Zur Realisierung eines solchen Konzeptes sind zunächst geeignete (erweiterte) Konzepte zur Spezifikation und Verarbeitung von datenbankübergreifenden (Integritäts-) Regeln für föderierte Datenbankumgebungen abzuleiten bzw. weiterzuentwickeln.

2 Architektur föderierter Datenbanksysteme

Ein *föderiertes Datenbanksystem* (FDBS) besteht aus einer Menge von *autonomen heterogenen Komponentendatenbanksystemen* (CDBSen), die über eine flexible Föderierungsschicht kooperieren (vgl. [SL90]). Die wesentliche Aufgabe dieser *erweiterbaren* Föderierungsschicht besteht darin, eine globale, homogene Schnittstelle für den einheitlichen und transparenten Zugriff auf föderierte Daten bereitzustellen.

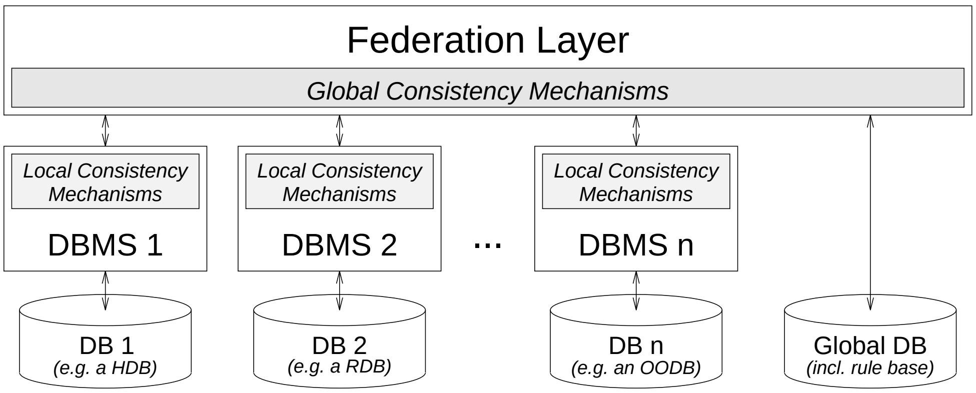

Abbildung 1: Architektur föderierter Datenbanksysteme.

Wir gehen davon aus, daß auf der Föderierungsebene ein globales, gemeinsames Datenmodell existiert, auf das die einzelnen (Komponenten-) Datenmodelle abgebildet werden können. Aufgrund der Anforderungen, daß ein föderiertes Datenbanksystem komplexe Objekte unterstützen und zudem leicht erweiterbar (d.h. flexibel) bleiben soll, ist für das *globale* Datenmodell der Föderierungsschicht ein objektorientiertes Datenmodell zu wählen. Wir nehmen hier an, daß dieses Modell dem ODMG93-Standard entspricht (vgl. [RS94]).

Ferner soll die Architektur des föderierten Systems so ausgelegt sein, daß die lokalen Schnittstellen der einzelnen, (teilweise) autonomen CDBSe erhalten bleiben, so daß existierende Applikationen unverändert weiterlaufen können. Auf der Ebene der Föderierungsschicht sollen Mechanismen zur Konsistenzsicherung bereitgestellt werden (siehe Abb. 1), die über die in den CDBSen angebotenen Integritätsmechanismen hinausgehen.

3 Integration aktiver Mechanismen

Aufbauend auf der in Kap. 2 skizzierten Architektur, verfolgen wir in diesem Abschnitt den Ansatz, aktive Mechanismen zur systemübergreifenden Konsistenzsicherung von föderierten Daten auf der Föderierungsebene zu integrieren. Ziel ist es, die von den Einzelsystemen bereitgestellten Integritätmechanismen zur *globalen* Konsistenzsicherung heranzuziehen.

Als Beschreibungsmittel von aktiven Eigenschaften haben sich die *ECA-Regeln* [DBB+88] als de facto Standard durchgesetzt. Wir möchten ECA-Regeln dazu verwenden, auf der Föderierungsebene potentiell konsistenzverletzende Ereignisse, auf die das System mit entsprechenden konsistenzwahrenden Aktionen reagieren soll, zu spezifizieren. Aus Effizienzgründen sollen ECA-Regeln aber nur in den Fällen verwendet werden, in denen die jeweiligen Konsistenzbedingungen nicht geeignet durch die vorhandenen, lokalen Integritätsmechanismen realisiert werden können.

Die Integration des ECA-Regelkonzeptes in die in Abb. 1 skizzierte föderierte Umgebung kann wie folgt erfolgen. Auf der Föderierungsebene werden globale Integritätsmechanismen in Form von *globalen* ECA-Regeln bereitgestellt. Diese ECA-Regeln werden in der globalen Datenbank (in der sich die Daten der Föderierungsschicht befinden) abgelegt.

Die Detektion von Ereignissen, die sich ausschließlich auf die globale Datenbank beziehen, kann bereits auf der Föderierungsebene erfolgen, während für die Detektion der übrigen (*primitiven*) Ereignisse [GD94] die Kooperation/Kommunikation mit den Komponentensystemen erforderlich ist. In diesen Fällen müssen die Ereignisse lokal von den einzelnen Systemen (*durch lokale Regelmanager*) detektiert und an die entsprechende Komponente der Föderierungschicht (*dem globalen Regelmanager*) weitergeleitet werden. Dieser Regelmanager ist dann für die Komposition von *zusammengesetzten* Ereignissen [CM91] sowie für die Verarbeitung der entsprechenden 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 ausgeführt oder entsprechende Operationen der Komponentensysteme aufruft.

4 Diskussion und Ausblick

ECA-Regeln scheinen aufgrund ihrer Mächtigkeit und Flexibilität ein geeigneter Mechanismus zur Konsistenzsicherung auch für föderierte Datenbankumgebungen zu sein. Die allgemein zu beantwortende Frage ist, wie mächtig darf bzw. soll die zu verwendende ECA-Regelsprache sein, um als geeignetes Mittel zur Spezifikation von Integritätsregeln in solchen Umgebungen dienen zu können?

Ereignisteil: Gemäß [GD94] muß ein aktives System die Spezifikation und Detektion von diversen Ereignistypen, wie z.B. Zeitereignisse, Methodenaufrufe (Operationsaufrufe) oder zusammengesetzte Ereignisse [CM91] unterstützen. Für föderierte Umgebungen ergeben sich nun die folgenden Fragestellungen, die zu lösen sind:

- Welche der vorgeschlagenen Ereignistypen sollten bzw. müssen von einem föderierten System für Konsistenzsicherungszwecke unterstützt werden? Sind Erweiterungen nötig?

- Wie können Ereignisse in CDBSen detektiert werden? In aktiven objektorientierten DBSen wird hierzu das Method Wrapping benutzt (vgl. [BZBW95]). Doch wie können Ereignisse in RDBMSen erkannt und signalisiert werden? Eine naheliegende Lösung ist, Trigger (als lokale Regelmanager) dazu zu verwenden, das Eintreten eines Ereignisses dem globalen Regelmanager anzuzeigen. Falls aber keine Trigger unterstützt werden, muß ebenfalls eine Art "Wrapper" in das System integriert werden, so daß Ereignisse vor bzw. nach dem Aufruf einer SQL-Operation signalisiert werden können. Wenn bestehende, nicht mit aktiven Fähigkeiten (z.B. mit Triggermechanismen) versehene CDBSe föderiert werden, entsteht das Problem, daß weder die CDBSe selbst noch ihre Anwendungen Ereignisse detektieren und signalisieren können. In diesen Fällen müssen die Anwendungen "gewrapped", d.h. neu übersetzt werden. Dies jedoch steht im starken Widerspruch zur Forderung nach lokaler Autonomie.

- Wie können Ereignisse, die in den CDBSen detektiert werden, der Föderierungsschicht bekannt gemacht werden? Hier müssen *lokale* ECA-Regeln, die die Detektion der lokalen Ereignisse an den entsprechenden globalen Regelmanager weiterleiten, von der globalen ECA-Regeldefinition abgeleitet werden. Für die Erzeugung von lokalen ECA-Regeln sind entsprechende Kooperationsmechanismen zwischen lokalen und globalen Regelmanagern erforderlich.

Bedingungsteil: Welche Arten von Bedingungen können bzw. dürfen formuliert werden? Wünschenswert sind hier Bedingungen, die über die Zustände der diversen CDBSe spezifiziert werden können.

Aktionsteil: Welche Aktionen sind zulässig? Beliebige Methoden- bzw. Operationsaufrufe sollen als Aktionen möglich sein, d.h. sowohl benutzerdefinierte Methoden als auch Systemoperationen, wie z.B. SQL-Anweisungen (insert, update, delete, etc.), sollen ausführbar sein. Weiterhin sollen vordefinierte Transaktionen/Applikationen der jeweiligen CDBSe als zulässige Sequenzen von Aktionen genutzt werden können.

Es muß ebenfalls möglich sein, systemübergreifende Aktionen aus einer ECA-Regel heraus zu starten, d.h. es muß die Möglichkeit bestehen, mehrere Methoden (bzw. Operationen) auf verschiedenen CDBSen aufrufen zu können. Hierfür müssen jedoch geeignete Synchronisationsmechanismen vom föderierten System bereitgestellt werden.

Kopplungsmodi: Ein wesentlicher Flexibilitätsfaktor für aktive Systeme sind die bereitgestellten Kopplungsmodi. Entsprechend [BBKZ94] muß ein aktives System diverse Kopplungsmodi unterstützen. Die Kopplungsmodi bestimmen, wann und in welchen Transaktionsgrenzen die Aktionen einer ECA-Regel ausgeführt werden. Hier ist die Frage zu klären, welche der vorgeschlagenen Kopplungsmodi im Rahmen einer föderierten Umgebung für die Konsistenzsicherung notwendig bzw. geeignet sind? Als Grundlage für die Realisierung diverser Kopplungsmodi wird allerdings die Existenz eines föderierten Transaktionsmechanismus vorausgesetzt. Dieser Mechanismus sollte zumindestens geschachtelte Transaktionen unterstützen, damit systemübergreifende Aktionen in Subtransaktionen innerhalb einer globalen Transaktion ablaufen können.

Weitere Probleme betreffen in erster Linie die Definition von globalen ECA-Regeln. Während der transparente Zugriff auf föderierte Daten als wesentliches Kriterium für föderierte Systeme mit *nur einem globalen* Föderierungsschema gilt, ist die *(Orts-) Transparenz* bei der Regeldefinition nicht erwünscht. In letzteren Fall ist man eher daran interessiert, auf ein *bestimmtes* Ereignis in einem *bestimmten* CDBS reagieren zu können. Daher muß der Regeldefinierer genau wissen, wo sich die entsprechenden Daten und Operationen befinden.

Als abschließende Bemerkung möchten wir festhalten, daß die Wahrung der lokalen Autonomie der CDBSe und die globale Konsistenzsicherung von föderierten Daten zwei gegensätzliche Anforderungen darstellen. Eine globale Konsistenzsicherung ist ohne Einschränkungen der lokalen Autonomie nicht geeignet realisierbar.

Die in diesem Papier vorgestellten und weitere Fragestellungen sollen in dem Projekt "Virtuelle Fabrik" im Rahmen des Förderschwerpunkts "Neue Produktionsverfahren/Experimentelle Fabrik" des Landes Sachsen-Anhalt behandelt werden. Als Basis für ein zukunftweisendes Fabrikplanungssystem soll ein föderiertes Datenbanksystem als integrierendes Rahmensystem entstehen, das verschiedenen Entwurfs– , Planungs– und Simulationswerkzeugen als gemeinsame Basis dienen soll. Die verschiedenen, zum Teil bereits bestehenden Systemen bringen eine heterogene Datenhaltung mit sich, für die eine übergreifende Konsistenzsicherung erforderlich ist.

Danksagung: An dieser Stelle möchten wir uns bei Gunter Saake, Ingo Schmitt, Jan Kusch und Thorsten Hartmann für die hilfreichen Diskussionen bedanken.

Literatur

[BBKZ94] H. Branding, A. Buchmann, T. Kudrass und J. Zimmermann. Rules in an Open System: The REACH Rule System. In N.W. Paton und M.H. Williams, Herausgeber, *Proc. of the 1st Int. Workshop on Rules in Database Systems, Edinburgh, August 1993*, S. 111–126. Springer-Verlag, Workshops in Computing, 1994.

[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, Herausgeber, *Proc. of the 11th Int. Conf. on Data Engineering (ICDE'95), Taipei, Taiwan*, S. 117–128. IEEE Computer Society, März 1995.

[CKTB95] S. Chakravarthy, V. Krishnaprasad, Z. Tamizuddin und R. H. Badani. ECA Rule Integration into an OODBMS: Architecture and Implementation. In P. S. Yu und A. L. P. Chen, Herausgeber, *Proc. of the 11th Int. Conf. on Data Engineering (ICDE'95), Taipei, Taiwan*, S. 341–348. IEEE Computer Society Press, März 1995.

[CM91] S. Chakravarthy und D. Mishra. An Event Specification Language (SNOOP) For Active Databases and its Detection. Technical Report, University of Florida, September 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, März 1988.

[GD94] S. Gatziu und K. R. Dittrich. Events in an Active Object-Oriented Database System. In N.W. Paton und M.H. Williams, Herausgeber, *Proc. of the 1st Int. Workshop on Rules in Database Systems, Edinburgh, August 1993*, S. 23–39. Springer-Verlag, Workshops in Computing, 1994.

[RS94] E. Radeke und M. H. Scholl. Federation and Stepwise Reduction of Database Systems. In W. Litwin und T. Risch, Herausgeber, *Proc. of the 1st Int. Conf. on Applications of Databases, Sweden*, S. 381–399. Springer-Verlag, LNCS 819, Juni 1994.

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