

Ereignisgetriebene CORBA-Dienste für heterogene, verteilte Informationssysteme

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

der Fakultät für Informatik

der Universität Karlsruhe (Technische Hochschule)

genehmigte

D i s s e r t a t i o n

von

Arne Koschel

aus Gehrden

Tag der mündlichen Prüfung:

Erster Gutachter:

Zweiter Gutachter:

1. Juli 1999

Prof. Dr. Peter C. Lockemann

Prof. Dr. Sebastian Abeck

Inhaltsverzeichnis

1	Einleitung	1
1.1	Gesamtziel – Ereignisgetriebene Dienste für heterogene, verteilte Systeme	1
1.2	Ausgangslage: Basistechnologien und deren Schwachpunkte	3
1.3	Anwendungshintergrund: UIS als heterogene, verteilte IS	7
1.4	Problemstellung	10
1.5	Lösungsansatz, Arbeitsthesen und Ziele der Arbeit	11
1.6	Gliederung der Arbeit	13
2	Grundbegriffe und Anwendungsszenario	-15
2.1	Aktive Kernfunktionalität	15
2.1.1	Ereignisse	16
2.1.2	Ereigniserkennung und Ereignis-Propagierung	17
2.1.3	ECA-Regeln in aktiven DBMS	19
2.2	Grundlagen: Dienstorientierung	20
2.2.1	Komponenten und Konnektoren	21
2.2.2	Der CORBA-Standard	21
2.3	Architekturkonzept für heterogene, verteilte Informationssysteme	25
2.3.1	Merkmale heterogener, verteilter Informationssysteme	25
2.3.2	Gesamtintegration durch Föderationsarchitektur	28
2.3.3	Der allgemeine Kapsel-Begriff	30
2.4	Konfigurierbarkeit	33
2.5	Ein Beispiel-Szenario für den Einsatz ereignisgetriebener Dienste	34
2.6	Resümee	36
3	Ziele und Aufgaben	-37
3.1	Transfer der aktiven Kernfunktionalität aus aktiven DBMS	40
3.2	Dienstorientierung unter Einsatz offener, standardisierter, objektorientierter Vermittlungsschichten	41
3.2.1	Entflechtung monolithischer aktiver DBMS in Dienste	42
3.2.2	Einsatz offener, standardisierter, objektorientierter Vermittlungsschichten	42
3.2.3	Einsatz von CORBA und der OMA für dienstorientierte Umgebungen	43
3.3	Heterogene, verteilte Informationsquellen bzw. Systemkomponenten – Integration und Ereigniserkennung	44
3.3.1	Rahmenaufgaben: Kapseln; erweiterte ECA-Regeln; Transaktionen	45
3.3.2	Integration heterogener Informationsquellen: Ereigniserkennung	47
3.3.3	Integration heterogener Informationsquellen: Bedingungen und Aktionen	48
3.3.4	Verteilung von Systemkomponenten	49
3.4	Aufgabenbereich: Konfigurierbarkeit	50
3.5	Resümee	51
4	Literaturübersicht	-53
4.1	Vorauswahl relevanter Literatur	54
4.2	Analyse relevanter Literatur	55
4.2.1	ECA-Regelverarbeitung in aktiven DBMS	55
4.2.2	Verteilte ADBMS-artige ECA-Regelverarbeitung	58
4.2.3	CORBA-basierte Systeme zur ADBMS-artigen ECA-Regelverarbeitung	59
4.3	Zusammenfassende Einordnung und Bewertung	62
4.4	Resümee und weiteres Vorgehen	64

5 Entflechtung aktiver DBMS - - - - -	65
5.1 DBMS-Entflechtung: Basisverfahren, präzisiertes Verfahren für ADBMS	66
5.1.1 Basisverfahren: Entflechtung monolithischer DBMS	67
5.1.2 Diskussion und Präzisierung: Entflechtung für monolithische <i>aktive</i> DBMS	68
5.1.3 Ziele bei der Entflechtung aktiver DBMS	69
5.1.4 Instanziierung: Verfahren zur Entflechtung <i>aktiver</i> DBMS	71
5.2 Domänenanalyse – Aktive Funktionalität in aktiven DBMS	72
5.2.1 ECA-Regelmodell: Definition von ECA-Regeln	73
5.2.2 ECA-Regelverwaltung (Rule Management) aktiver DBMS	75
5.2.3 ECA-Regelausführungsmodell: Ausführung von ECA-Regeln	75
5.3 Architekturanalyse – Architekturen durch Entflechtung	79
5.3.1 Schritt 0: Ausgangspunkt – Monolithisches aktives DBMS	80
5.3.2 Entflechtungsschritt 1: Aktivitätsdienst – Abtrennen aktiver Funktionalität	83
5.3.3 Entflechtungsschritt 2: Ein Ereignisdienst und ein Regeldienst	85
5.3.4 Entflechtungsschritt 3: Integration heterogener Informationsquellen	91
5.4 Resümee	95
6 Dienstkonzeption – ADBMS-artige Ereigniserkennung für heterogene, verteilbare Quellen	97
6.1 Konzeption des modularen Monitor-Dienstes	99
6.1.1 Terminologie	99
6.1.2 Konzept der Monitor-Kapseln	102
6.1.3 Wesentliche Schnittstellen des Monitor-Dienstes	106
6.2 Monitor-Unterstützung heterogener Ereignisquellen	108
6.2.1 Kategorisierung: Monitor-Unterstützung heterogener Ereignisquellen	109
6.2.2 Quellenkategorien: Auswahl repräsentativer existierender Ereignisquellen	112
6.3 ADBMS-artiges, erweitertes Ereignismodell für heterogene Quellen	114
6.3.1 Rahmen zur Ereignismodellierung	115
6.3.2 Ereignistypen und Ereignis-Semantikparameter	116
6.3.3 Ereignisinstanzen	119
6.3.4 Modelltypen zur Ereignismodellierung	121
6.3.5 Ereignismodell: IDL-Modellierung von Ereignistypen und -instanzen	123
6.4 Resümee	127
7 Verfahren – ADBMS-artige Ereigniserkennung für heterogene, verteilbare Quellen - - -	129
7.1 Quellenkategorie-spezifische Monitor-Verfahren	131
7.1.1 Monitor-Verfahren und ECA-Semantikparameter	131
7.1.2 Aktive Quellen mit Rückruf-Mechanismus	134
7.1.3 Aktive Quellen mit internen Aktionen – Spiegelrelationen	138
7.1.4 Quellen mit Delta-Aktivität	143
7.1.5 Pure Ereignisquellen	145
7.1.6 Passive Quellen mit Anfrageunterstützung	146
7.1.7 Protokollierte, passive Quellen	150
7.1.8 Blockquellen	153
7.1.9 Generische Ereigniserkennung auf Kapsel-Ebene	154
7.1.10 Zusammenfassung und Einordnung der Monitor-Verfahren	157
7.1.11 Resümee	159
7.2 Dynamische Ereignistypdefinition (DETD)	160
7.2.1 DETD: Basisvorgehen, Grundkonzept	160
7.2.2 Quellenspezifische Funktionalität am Beispiel des RDBMS Oracle	163
7.2.3 DETD am Beispiel „Aktive Quelle mit Triggern und Rückruf“	166
7.2.4 DETD: Fazit	169
7.3 Semi-automatische Monitor-Kapsel-Generierung	169
7.4 Resümee	171
8 Prototyp C²offein – Architektur und Realisierung unter CORBA - - - - -	173
8.1 ECA-Regelmodell und ECA-Ausführungsmodell für C ² offein	174

8.1.1	Parameter des ECA-Ausführungsmodells von C^2 offein	174
8.1.2	ECA-Regeldefinitionssprache „ C^2 offein-RDL“	177
8.2	Konzept der Konfigurierbarkeit in C^2 offein	178
8.2.1	Gesamt-Konfigurationskonzept von C^2 offein	179
8.2.2	Spezifikation einer Konfiguration	180
8.3	Architektur und ausgewählte Realisierungsaspekte von C^2 offein	182
8.3.1	Systemarchitektur von C^2 offein	182
8.3.2	Ereignis-Monitor-Kapseln zur Ereigniserkennung	184
8.3.3	Ereignisdienst: Ereignisverwaltung mit Ereignishistorie	184
8.3.4	Ereignisdienst: Detektor für komplexe Ereignisse	185
8.3.5	Regeldienst: Regelausführung	187
8.3.6	Regeldienst: ECA-Regelverwaltung	189
8.3.7	Regeldienst: Informationsquellenzugriffe – Bedingungsvaluierung, Aktionsausführung	190
8.3.8	Konfigurationskomponente	191
8.4	Entwicklungswerkzeuge, Status und Erfahrungen	193
8.4.1	Entwicklungswerkzeuge	193
8.4.2	Status und Erfahrungen bei der Implementierung	194
8.5	Resümee	196
9	Bewertung der erzielten Ergebnisse	197
9.1	Aufgabenerfüllung	198
9.1.1	Transfer der ADBMS-Kernfunktionalität	198
9.1.2	Architekturen für ereignisgetriebene Dienste	199
9.1.3	ADBMS-artige Ereigniserkennung für autonome, heterogene, verteilbare Quellen	200
9.1.4	Fazit	201
9.2	Leistungsverhalten	201
9.2.1	Phase 1: Leistungsmessungen am Beispiel	201
9.2.2	Phase 2: Simulationsmodell zur Leistungsanalyse	204
9.3	Einsetzbarkeit in Anwendungen: Anwendungsbeispiele	206
9.3.1	Anwendungsbeispiel: WWW-Ozon-Ticker	206
9.3.2	Anwendungsbeispiel: CORBA-Monitor	207
9.3.3	Anwendungsbeispiel: Metadatenfortschreibung	208
9.3.4	Anwendungsbeispiel: ECA-Regeln kombiniert mit „Push“-Technologie	209
9.4	Klassifikationsschema: Anwendungen ereignisgetriebener Dienste	211
9.4.1	E-Klasse: Reine Ereigniserkennung und Ereignisweitergabe	211
9.4.2	EA-Klasse: Ereigniserkennung und Aktionsausführung	212
9.4.3	CA-Klasse: Reine Produktionsregel-Verarbeitung	212
9.4.4	ECA-Klasse: Vollständige ECA-Verarbeitung, potentiell mit Rückgriffen	213
9.4.5	Tabellarische Zusammenfassung der Anwendungsklassifikation	214
9.5	Resümee	214
10	Zusammenfassung und Ausblick	217
10.1	Zusammenfassung	217
10.2	Ausblick	220

1 Einleitung

„Der Weg ist das Ziel“
- Asiatische Weisheit

1.1 Gesamtziel – Ereignisgetriebene Dienste für heterogene, verteilte Systeme

Herkömmliche Informationssysteme basieren vorwiegend auf einer einzigen Art von Informationsquelle, wie z.B. auf einem Datenbanksystem. Heutige Informationssysteme jedoch müssen, oft dienstorientiert¹, eine Vielzahl verschiedenartiger, autonom arbeitender Informationsquellen integrieren. Beispiele solcher Quellen sind Datenbank-Management-Systeme (DBMS) oder Dateien, aber insbesondere auch Nicht-Datenquellen, wie Berechnungsprogramme. Die Informationsquellen sind folglich stark heterogen und darüberhinaus ggf. in Netzwerken verteilt.

Der Einsatz moderner Integrationstechnologie in Form objekt-orientierter Vermittlungsschichten („Middleware²“) wie CORBA erlaubt es gegenwärtig, zumindest die Heterogenität von Systemplattformen (Betriebssysteme, Netzwerkprotokolle, Programmiersprachen) und die Verteilung von Systemkomponenten zu verdecken. Die Mittel einer solchen Schicht ermöglichen es ferner, Informationsquellen bzw. Systemkomponenten zu kapseln und in dieser gekapselten Form für Entwickler bereitzustellen. Der Zugriff auf die gekapselten Quellen erfolgt somit benutzergetrieben über entsprechende Anwendungen. Er ist damit aus Sicht der dienstgebenden Informationsquellen *passiv*.

Um die effektive Kontrolle, Koordination oder Kooperation der solcherart integrierten Bestandteile heterogener, verteilter Informationssysteme zu ermöglichen und der nutzerseitig resultierenden Informationsüberflutung Herr werden zu können, sind jedoch ergänzend *aktive* Mechanismen

-
1. Der *Dienstbegriff* ist in der Literatur ein durchaus schillernder. So sind Dienste im ISO/ODP-Sinne z.B. funktionsorientiert spezifiziert [ISO95]. In der vorliegenden Arbeit wird der Begriff des Dienstes auf abstrakterer Ebene genutzt, und zwar stellt ein Dienst über eine Menge von Schnittstellen eine zu erbringende Funktionalität bereit. Mit dieser Verwendung lehnt sich der Dienstebegriff an den diesbezüglich weitergefaßten Begriff des CORBA-Standards an, um dem Kontext der Arbeit gerecht zu werden.
 2. *Terminologie*: In dieser Arbeit werden auf konzeptioneller Ebene möglichst weitgehend deutsche Informatikbegriffe verwendet, also z.B. Vermittlungsschicht anstelle von „Middleware“, Kapsel anstelle von „Wrapper“ oder Faden anstelle von „Thread“. Muß auf dieser Ebene auf englische Begriffe zurückgegriffen werden, z.B. zur Verdeutlichung bei deren erstmaliger Verwendung, so werden sie in Anführungszeichen gesetzt. Ansonsten werden nur punktuell englische Begriffe und Namen eingesetzt, und zwar auf implementierungsnaher Ebene und teilweise in Form von Zitaten, also z.B. CORBA-Client bzw. -Server, DBMS- bzw. DB-INSERT oder Oracle-Pipe. Dies gilt auch für einige spezielle Begriffe aus dem ADBMS-Kontext, da in der Literatur i.w. nur englische Begriffe verwendet werden, so daß deutsche Übersetzungen hierfür nicht zweckmäßig scheinen.
-

in Form ereignisgetriebener Dienste erforderlich. Hierfür erlauben derartige ereignisgetriebene Dienste bspw. die gezielte Spezifikation relevanter Änderungsereignisse, über die der Benutzer bei Eintritt des Ereignisses aktiv, also automatisch, informiert wird.

Die *Basisthese* der vorliegenden Arbeit lautet, daß sich solche ereignisgetriebenen Dienste durch die Weiterentwicklung der bekannten Mechanismen aktiver Datenbank-Management-Systeme (ADBMS) kombiniert mit den Mitteln der genannten Vermittlungsschichten bereitstellen lassen. ADBMS-Mechanismen bieten aktives Verhalten über sog. ECA-Regeln (ECA: „Event Condition Action“) an, die auf einem klar definierten ECA-Regel- und ECA-Ausführungsmodell basieren.

ADBMS-artige aktive Funktionalität direkt für diensteorientierte Umgebungen mit heterogenen, verteilten Informations- bzw. Ereignisquellen einzusetzen, schlägt jedoch wegen folgender Merkmale bestehender aktiver DBMS fehl:

- Aktive DBMS sind gegenwärtig typischerweise proprietär und monolithisch implementiert. Ihre aktive Funktionalität ist damit nicht in Form eigenständiger Dienste verfügbar.
- Selbst wenn diese aktive Funktionalität eigenständig herausgelöst zur Verfügung stünde, wäre sie nicht oder nur sehr eingeschränkt in der Lage, mit der Heterogenität verschiedenartigster Informations- bzw. Ereignisquellen umzugehen oder mit der Verteilbarkeit von Systemkomponenten fertig zu werden.
Vermittlungsschichten wie CORBA sind zwar sehr gut geeignet die Basisheterogenität von Systemplattformen zu verdecken, versagen aber noch bei der Bewältigung derartiger Quellenheterogenität. Sie beinhalten z.B. praktisch keine Mechanismen zur Ereignisverarbeitung für die genannten stark heterogenen Quellen.

Zur Bewältigung dieser Probleme soll die vorliegende Arbeit einen Beitrag leisten. Der Arbeit *übergeordnetes Gesamtziel* ist es damit, die Rahmenkonzeption sowie Kernelemente eines konfigurierbaren Baukastens für ereignisgetriebene Dienste in heterogenen, verteilten Informationssystemen bereitzustellen und prototypisch zu erproben. Die Arbeit geht also, gemäß der obigen Ausführungen, von *zwei Voraussetzungen* aus:

- Bei den Mechanismen für ereignisgetriebenes Verhalten wird auf die Technologie aktiver Datenbank-Management-Systeme (ADBMS) zurückgegriffen, d.h. auf ADBMS-artige aktive Funktionalität mit ihrer klar definierten Semantik.
- Zur Verteilung von Systemkomponenten und zur Überwindung der technischen Basisheterogenität von Systemplattformen wird der Industriestandard CORBA, eine objektorientierte Vermittlungsschicht, eingesetzt.

Innerhalb des übergeordneten Gesamtziels lassen sich eine Reihe von Zielen erkennen, von denen die eigene Arbeit zwei behandeln wird. Die nächsten Abschnitte dienen dazu, die oben angedeuteten Probleme etwas genauer darzustellen, um daraus abschließend die beiden Ziele der vorliegenden Arbeit abzuleiten. Hierzu werden zunächst die verwendeten Basistechnologien und deren Schwachpunkte skizziert und sodann wird der Bereich der Umweltinformationssysteme (UIS) als typische Ausprägung heterogener, verteilter Informationssysteme vorgestellt, dies zusammen mit motivierenden Anwendungsbeispielen aus UIS.

1.2 Ausgangslage: Basistechnologien und deren Schwachpunkte

Die Grundlage für die Entwicklung ereignisgetriebener Dienste ergibt sich aus einer geeigneten Kombination der oben genannten Basistechnologien bzw. Faktoren, wobei deren Schwachpunkte durch passende Modifikationen bzw. Erweiterungen zu bewältigen sind. Die Einflußfaktoren, illustriert in Abbildung 1.1, sind die aktive Funktionalität aus aktiven DBMS, Dienstorientierung und Verteilung mittels CORBA und Merkmale heterogener, verteilter Informationssysteme, wie z.B. die Ereignisquellenvielfalt.

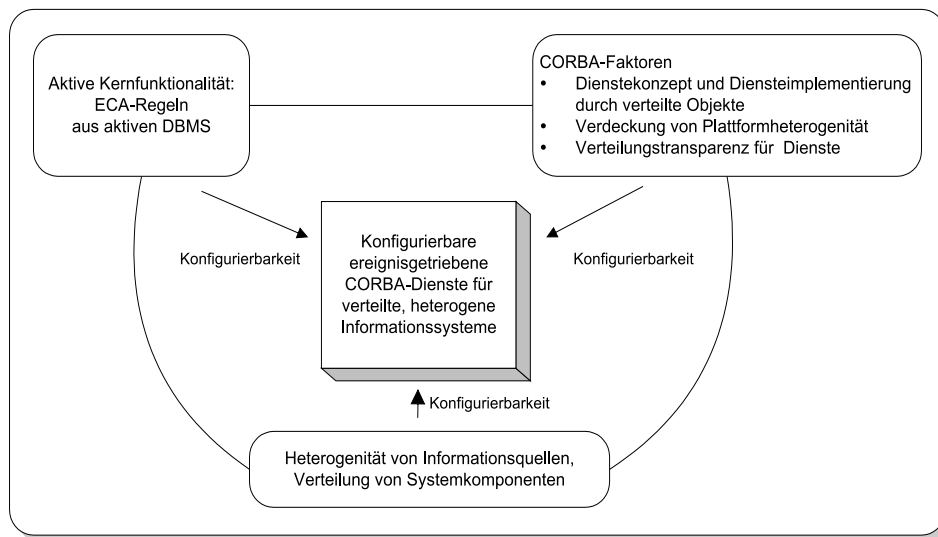


Abbildung 1.1 Einflußfaktoren für konfigurierbare, ereignisgetriebene Dienste

Technologie aktiver Datenbank-Management-Systeme

DBMS sind allgemein akzeptiert als essentielle Bestandteile der Infrastruktur heutiger Informationssysteme aller Art. Sie stellen zuverlässige Basistechnologie bereit, sofern akzeptiert wird, daß es sich um recht zentralistisch orientierte, monolithische Systeme handelt, die eine weitreichende Menge von Diensten unter einheitlicher Schnittstelle bereitstellen. Beispiele solcher Dienste sind Datenverwaltung und Anfragebearbeitung, Transaktionen oder sog. „Backup/Recovery“.

Solche Dienste werden in zwei Formen bereitgestellt: *Passive DBMS* reagieren nur auf explizite Dienstanforderungen, wie z.B. eine Datenbankanfrage. *Aktive DBMS (ADBMS)* verhalten sich von ihrer Philosophie her aktiv, d.h. getrieben durch Ereignisse. Sie nutzen hierzu das ECA-Paradigma mit der Bedeutung: Wenn ein überwachtes Ereignis E eintritt, das eine zu überprüfende Bedingung C erfüllt, dann wird eine Aktion A ausgelöst. Die Universalität des „Event Condition Action“-Paradigmas wird durch die Physiologie mit ihrem Reiz-Reaktion-Schema („Stimulus Response“) wissenschaftsübergreifend bestätigt. Organismen, die Reizen („Events“) ausgesetzt werden, haben, bei entsprechender Disponierung („Condition“), mit bestimmten Reaktionen („Action“) zu antworten [RW93]. Wie einleitend angedeutet, ist dieses Paradigma in aktiven

DBMS mittels sog. ECA-Regeln implementiert, bekannt auch in eingeschränkter Form als DBMS-Trigger.

Die Kernbestandteile ADBMS-artiger aktiver Funktionalität sind drei Elemente: Ein ECA-Regelmodell, das die Definition von Ereignissen, Bedingungen und Aktionen gestattet, ein ECA-Ausführungsmodell, das die Semantik von Regelausführungen definiert, und eine ECA-Regelverwaltung, mit der Regeln erstellt und modifiziert werden können. Die Definition dieser Kern-Funktionalität aktiver DBMS ist inzwischen konsolidiert. In letzter Zeit entstanden eine Reihe von Veröffentlichungen [DG96, FT95, Pat99, The96, WC96], in denen diese Funktionalität mit klar definierter Ausführungssemantik festgelegt ist.

Technologietransfer in neue Umgebungen – Baukästen aus Diensten

Für eine ganze Reihe von Informationssystemen wird die Bedeutung von DBMS unverändert Bestand haben. Gegenwärtig entflechten jedoch Industrie und Verwaltung ihre monolithischen Organisationen in kleine, relativ autonome Einheiten. Daher wird es immer mehr Anwendungsbereiche geben, in denen zentral organisierte, monolithische Datenbanksysteme keine adäquaten Lösungen mehr darstellen. Informationssysteme werden aus einer Vielzahl in Netzwerken verteilter, technisch und semantisch heterogener Informationsquellen bestehen. Dies können Datenquellen sein, z.B. Daten- oder Wissensbanken, aber auch Funktionen, z.B. Berechnungsbibliotheken, oder gar komplette, bereits bestehende Anwendungssysteme. Der Einsatzbereich kann von der Integration mehrerer Komponenten komplexer Anwendungssysteme, z.B. Engineering-Umgebungen [Bül95a, Bül95b, SvdB93], über unternehmensweite Systeme, z.B. sog. „Data Warehouses“ [Wid95] für Führungsinformationssysteme, bis hin zu unternehmensübergreifenden Systemen reichen. Die Integration und Interoperabilität solcher heterogener Informationsquellen ist eine der großen aktuellen und künftigen Aufgaben [Bro93, OHE95] der Informatik. Zur effizienten Kontrolle, Koordination und Kooperation werden Dienste für aktives Verhalten in derartigen Informationssystemen eine wesentliche Rolle spielen.

Die derzeit noch monolithischen DBMS zu entflechten und ihre bewährte Funktionalität in Form separat nutzbarer Dienste bereitzustellen, ist deshalb eine der aktuellen Herausforderungen der Datenbankforschung [Bla96, GD97, DBWG96, Vas95]. Um aktives Verhalten in Form ereignisgetriebener Dienste mit ADBMS-artiger Semantik leistungsfähig unterstützen zu können, ist also konkret die Herauslösung der konsolidierten aktiven Mechanismen aus aktiven DBMS [GKvBF98, KGFvB98] von zentraler Bedeutung.

Konsequenterweise werden Dienste damit nicht mehr nur zentral angeboten, sondern verteilbar in Netzwerken, also mit nur noch lose gekoppelter Kommunikation untereinander. Die jüngeren Entwicklungen im Rechnerkommunikationsbereich und das Zusammenwachsen von Datenbank- und Telekommunikationstechnologie zeigen genau in diese Richtung. Durch Organisationen in Industrie und Forschung wird hier versucht, Baukästen aus Diensten bereitzustellen, in denen Dienste aus Datenbanken wichtige Kernbestandteile sind. Angeboten werden diese Dienste meist als Bestandteile von Vermittlungsschichten („Middleware“) auf mittlerer Architekturebene, die Anwendungsprogramme und Informationsquellenzugriffe über klare Schnittstellen voneinander

isolieren. Bereits in früheren Ansätzen (siehe z.B. [Bat86, CD87, Com82]) wurde versucht, Baukästen für DBMS (sog. „Extensible Database Systems“) bereitzustellen. Aktive Funktionalität, Heterogenität und Verteilung waren dort jedoch keine Schwerpunkte. Hier soll die vorliegende Arbeit neue Ergebnisse erzielen.

Industriestandard CORBA – eine dienstbasierte, objektorientierte Vermittlungsschicht

Im Zuge der Entwicklungen für verteilte Systeme entstanden in den letzten Jahren eine Reihe von Ansätzen für Vermittlungsschichten. Diese Ansätze ermöglichen die Interoperabilität von Elementen verteilter Informationssysteme und somit den Aufbau föderierter, also lose gekoppelter Gesamtsysteme. Sie unterscheiden sich u.a. in der „Breite“ des Ansatzes bezüglich bereitgestellter oder vorgesehener Funktionalität, in der Offenheit (offener Standard oder proprietäres System), in der Verbreitung bzw. Verbreitungsaussicht und im Grad der Unterstützung heterogener Systemplattformen und des Dienste-Konzepts.

Ein besonders wichtiger Ansatz für die Entwicklung bzw. Integration offener großer heterogener Anwendungssysteme ist die sog. „Common Object Request Broker Architecture – CORBA“ [OMG90, OMG95e, Vin97] der „Object Management Group – OMG“. CORBA ist besonders wichtig, weil es sich hierbei um einen *Industriestandard* einer Organisation von inzwischen weit über 800 DV-Herstellern, Anwendern und Forschungseinrichtungen handelt, der sich durch diese breite Unterstützung bereits in einem umfangreichen Marktsegment etabliert hat [Adl95, Bro93, OH95a]. Hinzu kommt, daß die Normungsgremien der OMG vergleichsweise schnell arbeiten und viele Elemente von CORBA bereits in einer Reihe von Implementierungen vorliegen [KKT95, KKTvB96, OMG94a, OMG95c, OMGa, vBKK⁺95].

Die OMG definiert CORBA als plattform- und programmiersprachenunabhängige Vermittlungsschicht für verteilte Dienste, die durch CORBA-Objekte implementiert werden. Den Rahmen der OMG-Arbeiten bildet die sog. „Object Management Architecture (OMA)“. Die OMA verfolgt konsequent die Idee eines Baukastens aus Diensten. Sie besteht aus einem Kern, dem sog. „Object Request Broker (ORB)“, sowie einer Sprache zur Beschreibung von Objektschnittstellen, die sog. „CORBA Interface Definition Language – IDL“. Hinzu kommen allgemein nutzbare grundlegende Objektdienste, wie ein Namensdienst usw., genannt die CORBAServices, sowie darauf aufbauende anwendungsspezifische und anwendungsübergreifende Bibliotheken, die sogenannten CORBAfacilities. CORBAfacilities sind allerdings bisher im wesentlichen „in Arbeit“. Jedoch stellen bereits ORB und CORBAServices, einen sehr guten Dienstebaukasten dar, der entsprechend gut als Föderationskern für heterogene, verteilte Informationssysteme dienen kann.

Generell werden mit CORBA zwei Ziele verfolgt [MZ95, OH95a]. Das Endziel ist eine softwaretechnische Infrastruktur zur Entwicklung verteilter, objektorientierter Anwendungen verschiedenster Anwendungsbereiche. Dies ist ein fortlaufender Prozeß, der auch Neuentwicklungen berücksichtigt. CORBA dient außerdem zur Integration bestehender Systeme, wie Datenbanken oder Anwendungsprogramme, in CORBA-basierte Umgebungen. Hierfür zentrales Mittel ist das Konzept sogenannter IDL-Kapseln („IDL Wrapper“). Informell ausgedrückt wird durch eine Kapsel um ein bestehendes System eine Schale mit definierten Schnittstellen gelegt. Hiermit ist

der syntaktisch einheitliche Zugriff auf diese Systeme gegeben. Nicht Teil von CORBA – auch künftig in den CORBAfacilities nur für Spezialfälle – ist hingegen der Aufbau solcher Kapseln und die Semantik hinter ihren Schnittstellen. „The ambitious goal of CORBA is to turn everything into nails, and give everyone a hammer. The nails are the IDLized services, and the hammer is the IDL interface to these services“ [OH95a].

Beispiel 1.1 Ein Beispiel eines zu integrierenden Systems kann ein DBMS sein, für das eine Kapsel über eine Schnittstelle das Beantworten einer SQL-Anfrage mit einer passenden Tupelmenge ermöglicht. Eine andere Kapsel könnte Ereignisse infolge von Änderungen in der Datenbank erkennen. Kapseln bzw. IDL-Kapseln sind damit wichtige Bestandteile ereignisgetriebener Dienste für CORBA-basierte Informationssysteme.

Die beiden anderen „großen“ allgemeinen, offenen Vermittlungsschicht-Ansätze für Dienstebaukästen sind OSF DCE und Active (Open) Group (Microsoft) ActiveX/DCOM. Gegenüber diesen bietet CORBA das deutlich klarere objektorientierte Konzept und die wesentlich weiterreichende Unterstützung technisch heterogener Systemplattformen. Künftig möglicherweise vergleichbar in der Funktionalität, wenn auch mit deutlich schwächerem Architektur-Grundkonzept, ist DCOM. Als Industriestandard gilt CORBA gegenüber DCOM als der offenere Ansatz und besonders für große heterogene Informationssysteme als besser geeignet. DCOM hingegen ist besonders in stark Microsoft-orientierten Systemwelten bedeutsam. Künftig ebenfalls Bedeutung erlangen werden wohl, bedingt durch die starke Verbreitung der Programmiersprache Java, die sogenannten Enterprise JavaBeans, die jedoch noch in einer frühen Entwicklungsphase sind. Hier ist wiederum ein Zusammenwachsen mit CORBA zu beobachten, z.B. in den derzeit in der Spezifikationsphase befindlichen CORBAbeans (inzwischen auch CORBA Components genannt). Beschreibungen aller Ansätze finden sich in [Adl95, HM98, OH95b, OHE96, OMG98b, OMG97a, VD97].

CORBA ist durch seine Eigenschaften und seine breite Unterstützung einer der wichtigen, aktuellen Standards. Der Einsatz von CORBA als Infrastruktur empfiehlt sich, je mehr Heterogenität und Verteilung bei der Implementierung entsprechender Informationssysteme eine Rolle spielen. Ingenieurwissenschaftliche Arbeiten für und mit diesem Standard sind deshalb von einschlägiger Bedeutung.

Konfigurierbare ereignisgetriebene Dienste unter CORBA-Einsatz

Die OMA enthält in den CORBAservices bereits einige aus Datenbanksystemen bekannte Dienste, u.a. einfache Formen von Anfragebearbeitung, Persistenz und Transaktionen. Sehr unzureichend unterstützt sind bislang im Rahmen der OMA jedoch aktive Mechanismen. Der CORBA Event Service ist hier zu nennen, der jedoch nur einen Übertragungskanal für Ereignisse darstellt. Unwesentlich weiter gehen der unlängst verabschiedete CORBA Notification Service und der in der Konzeptionsphase befindliche CORBA Messaging Service. Verabschiedete CORBAservices finden sich jeweils aktuell unter [OMGa].

10 Zusammenfassung und Ausblick

*„Das Ende des Weges ist Ebenbild der Morgenröte,
ein neuer Anfang.“
- unbekannt*

10.1 Zusammenfassung

Das übergeordnete Ziel der vorliegenden Arbeit war es, die Rahmenkonzeption und Kernelemente eines konfigurierbaren Baukastens für ereignisgetriebene Dienste in heterogenen, verteilten Informationssystemen bereitzustellen und prototypisch zu erproben. Erst solche Dienste ermöglichen die effektive Kontrolle, Koordination und Kooperation der Bestandteile dieser heterogenen, verteilten Informationssysteme.

Ausgangspunkt der Arbeit waren die Mechanismen der mittlerweile gut erforschten aktiven DBMS. Diese bieten aktives, also ereignisgetriebenes Verhalten, und sie basieren dabei auf klar definierten, inzwischen in ihrer Kernfunktionalität konsolidierten ECA-(Event-Condition-Action)Regel- und Ausführungsmodellen. Die unmittelbare Übernahme ADBMS-artiger Funktionalität für dienstorientierte, heterogene, verteilte Umgebungen scheitert jedoch an den Charakteristika von ADBMS, denn:

- ADBMS sind typischerweise proprietär und monolithisch implementiert, d.h. ihre aktive Funktionalität ist nicht in Form von eigenständigen Diensten verfügbar.
- Diese Funktionalität wäre, selbst wenn sie herauslösbar wäre, nicht oder nur sehr eingeschränkt in der Lage, mit der Heterogenität von Informations- bzw. Ereignisquellen umzugehen oder die Verteilung von Systemkomponenten zu bewältigen.

Diese Schwächen ließen sich nur durch eine konsequente Fortentwicklung der ADBMS-Technologie und deren Kombination mit objektorientierten Verteilungskonzepten überwinden. Die wesentliche Fortentwicklung der ADBMS-Technologie umfaßte insbesondere Mittel, um mit der Heterogenität von Informations- bzw. besonders Ereignisquellen umgehen zu können.

Als technische Basis wurde hierzu der bedeutendste Industriestandard für objektorientierte Vermittlungsschichten, CORBA, eingesetzt. Hierdurch konnte die ADBMS-artige aktive Funktionalität

lität gut in Form ereignisgetriebener Dienste angeboten werden. Ferner verdeckt CORBA die technische Basisheterogenität von Systemplattformen sowie die potentielle Verteilung von Systemkomponenten.

Auf eine gute Integration der eigenen Ergebnisse in CORBA wurde besonderer Wert gelegt, um – zusätzlich zur bedeutsamen Fortentwicklung von ADBMS-Technologie – auch für diesen Standard einen ingenieurwissenschaftlichen Beitrag zu leisten. Die erzielten Ergebnisse sind zudem gut übertragbar, da die verwendeten CORBA-Mittel sich in ähnlicher Form auch in anderen verteilten Objektsystemen, wie bspw. DCOM wiederfinden.

Nach der Literaturanalyse in Kapitel 4 ließen sich aus zunächst *funktionaler Sicht* (d.h. also ohne bspw. Qualitätsaspekte wie Leistung näher zu behandeln) innerhalb des übergeordneten Gesamtziels mindestens zwei Zielbereiche erkennen, die in bisher bestehenden Lösungsansätzen nicht bzw. nur sehr unzureichend gelöst wurden:

- Erarbeitung von Architekturen CORBA-basierter, ereignisgetriebener Dienste für aktive Kernfunktionalität aus aktiven DBMS und
- ADBMS-artige Ereigniserkennung und -verarbeitung für heterogene, verteilte Informationsquellen.

Auf diese beiden Zielbereiche konzentrierte sich die weitere Arbeit. Folgende wesentliche Ergebnisse wurden hierfür erzielt:

Architekturen CORBA-basierter, ereignisgetriebener Dienste für aktive Kernfunktionalität aus aktiven DBMS.

In Kapitel 5 wurde, basierend auf einem Verfahren für passiver DBMS, ein Verfahren zur Entflechtung aktiver DBMS entwickelt und erstmals angewendet. Es gelang damit, eine modulare, dienstebasierte Rahmenarchitektur mit verteilten Komponenten zu schaffen und deren klar definierte, konsolidierte aktive Funktionalität alternativ auch in Teilen nutzbar zu machen. Entwickler werden hierdurch von dem Zwang befreit, auch dann ein vollständiges aktives DBMS einsetzen zu müssen, wenn sie nur eine partielle aktive Funktionalität, wie z.B. eine reine Ereigniserkennung, benötigen. Zudem eröffnete der Transfer ADBMS-artiger aktiver Funktionalität in heterogene, verteilte Systemumgebungen eine Fülle neuer interessanter Einsatzbereiche, die für derartige Funktionalität bisher kaum zugänglich waren.

Zunächst wurde dadurch eine umfassende Analyse von ECA-Regelmodell und ECA-Ausführungsmodell aktiver DBMS geliefert. Die anschließend erarbeiteten Entflechtungsarchitekturen beginnen bei einem „einfachen Aktivitätsdienst“ als Aufsatz für passive DBMS und reichen bis zu einer komplexen ADBMS-artigen ECA-Regelverarbeitung für heterogene, verteilte Informationssysteme.

ADBMS-artige Ereigniserkennung und -verarbeitung für heterogene, verteilte Informationsquellen.

Essentielle Grundlage ereignisverarbeitender Mechanismen ist die Erkennung von Ereignissen, insbesondere die Erkennung primitiver Ereignisse innerhalb eines breiten Spektrums heterogener Informations- bzw. Ereignisquellen. CORBA erlaubt es zwar, die Heterogenität von Systemplattformen zu verdecken und eine transparente Verteilung im Netz zu ermöglichen, hat aber den Bereich der Ereignisverarbeitung bisher nur unzureichend behandelt.

Um diese gravierenden Lücken im Bereich der Erkennung primitiver Ereignisse zu schließen, wurde in Kapitel 6 und in Kapitel 7 ein modularer Monitor-Dienst für heterogene Ereignisquellen konzipiert. Die Konzeption der Schnittstellenspezifikation des Monitor-Dienstes basiert auf CORBA-IDL und sieht den Einsatz von Monitor-Kapseln („Monitoring Wrapper“) vor. Ergänzt wird die Konzeption durch ein IDL-basiertes Ereignismodell. Mit diesen Ergebnissen ist erstmals der modelltechnische Rahmen der ADBMS-artigen Erkennung primitiver Ereignisse auch für stark heterogene Ereignisquellen umfassend erarbeitet.

Ein detailliertes Schema zur Kategorisierung der Monitor-Unterstützung von heterogenen Ereignisquellen erlaubt deren präzisen Vergleich hinsichtlich ADBMS-artiger Ereigniserkennung und insbesondere eine wesentlich bequemere Erstellung der Monitor-Kapseln hierfür. Bei Kenntnis der Ereignisquellenkategorie sind die Monitor-Kapseln schematisch entwickelbar und partiell können sogar Schablonen für sie rechnergestützt generiert werden. In diesem Zusammenhang wurden erste Ansätze von Verfahren zur dynamischen Ereignistypdefinition und zur Schablonengenerierung von Monitor-Kapseln als Entwicklerunterstützung erarbeitet.

Die bekannten Monitor-Verfahren für die einzelnen Ereignisquellenkategorien wurden umfassend analysiert und bewertet. Die detaillierte Feststellung, inwieweit die jeweiligen Verfahren die ADBMS-artige Ereignisverarbeitung realisieren können, war das bedeutendste Ergebnis dieser Untersuchungen. Verfahrensspezifisch wurde ausführlich ermittelt, in welchem Umfang die ADBMS-artige Semantik (ADBMS-ECA-Semantikparameter) der Ereignisverarbeitung unterstützbar ist.

Mit diesen Ergebnissen ist erstmals die Adaption und der Transfer ADBMS-artiger Semantik der Ereigniserkennung für stark heterogene Ereignisquellen erreicht. Insgesamt sind wesentliche Fortschritte bei der Entflechtung von aktiven DBMS, die jetzt passend für dienstorientierte, heterogene, verteilte Umgebungen zur Verfügung stehen, und bei der ADBMS-artigen Erkennung primitiver Ereignisse aus stark heterogenen Ereignisquellen und ihrer Weiterverarbeitung erzielt worden. Da die erzielten Ergebnisse stets gut in eine CORBA-Umgebung integriert wurden, steht außerdem jetzt auch für diesen Standard erstmals eine umfassende Lösung zur anwendungsindividuellen Ereigniserkennung für stark heterogene Quellen zur Verfügung.

C²offein-Prototyp

Die wesentlichen erarbeiteten Konzepte wurden anhand des sogenannten C²offein-Prototyps (Kapitel 8) realisiert. Das ECA-Ausführungsmodell von C²offein wurde zudem gegenüber ADBMS für heterogene, verteilte Systemumgebungen modifiziert und erweitert. In dieser Form stellt es eine deutliche Fortentwicklung entsprechender ADBMS-ECA-Ausführungsmodelle dar.

Die Prototyp-Architektur von C²offein erfüllt bereits eine Reihe von Aufgaben hinsichtlich der Konfigurierbarkeit von ereignisgetriebenen Diensten. Sie erlaubt die separate Nutzung von sinnvollen (ereignisgetriebenen) Diensten und sie erlaubt deren Re-Kombinieren zu größeren Systemen, bis hin zur gesamten ECA-Regelverarbeitung. Damit stellt C²offein bereits einen wesentlichen Schritt für ein Werkzeug dar, das spezifische ADBMS-artige aktive Funktionalität für individuelle Anwendungsbedürfnisse anbieten kann.

Die abschließende Evaluierung in Kapitel 9 zeigte, daß die oben aufgestellten Ziele umfassend erreicht wurden. Ergänzend wurden erste Leistungsmessungen zu C²offein vorgestellt. Vor allem an Beispielen aus Umweltinformationssystemen, als typische Vertreter heterogener, verteilter Informationssysteme, wurde die Einsetzbarkeit der erzielten Ergebnisse gezeigt. Ein erstes Schema zur Auswahl von passender aktiver Funktionalität nach spezifischen Anwendungsmerkmalen rundete die in der Arbeit erzielten Ergebnisse ab.

10.2 Ausblick

Die in dieser Arbeit erzielten grundlegenden Ergebnisse eröffnen den Raum für eine ganze Reihe weiterer denkbarer Forschungsaktivitäten in unterschiedlichen Bereichen. Nur einige der sich aufdrängenden Fragestellungen sind:

- *Transaktionsverarbeitung.* Die Transaktionsverarbeitung stellt eine wesentliche Eigenschaft (aktiver) DBMS dar. Hierzu wurden in dieser Arbeit im Rahmen der Ereigniserkennung Grundlagen gelegt. Sinnvoll wäre eine weitergehende Untersuchung von Kopplungsmodi oder gar der Einsetzbarkeit „moderner“ Transaktionskonzepte, zum Beispiel SAGA's [JK97], im Rahmen der gesamten ECA-Regelverarbeitung für heterogene, verteilte Umgebungen.
 - *Informationsquellenzugriffe.* Die Integration weitergehender Techniken für Informationsquellenzugriffe in Bedingungsprüfungen oder Aktionsausführungen, also die Entwicklung zugehöriger Mediationskonnektoren.
 - *Erweitertes ECA-Regel- und -Ausführungsmodell.* Eine „vollständige“ Erweiterung des C²offein-ECA-Regel- und -Ausführungsmodells für heterogene, verteilte Umgebungen, z.B. durch ergänzenden Einbezug von Qualitätsparametern. Hier könnten auch Untersuchungen zur formalen Semantik solcher Modelle sinnvoll sein.
- In diesem Zusammenhang wären auch Untersuchungen interessant, inwieweit die automa-

tische Orthogonalisierung von Regelmengen vorgenommen werden kann, um sie bspw. parallel durch mehrere Regelverarbeitungen verarbeiten zu können.

- *Qualitätsaspekte.* Generell wäre die umfassende Untersuchung und Optimierung von Qualitätsaspekten („Quality of Service“) für die erarbeiteten Ergebnisse von Bedeutung, insbesondere für die Weiterentwicklung eines Systems wie C²offein zu einer „Produktionsversion“. Dies beinhaltet Aspekte wie Leistung, Ausfallsicherheit usw. Hier ist bspw. die Integration entsprechender Techniken aus dem Bereich der verteilten Systeme bzw. der Telematik denkbar.
Ebenfalls interessant sind hier Aspekte wie Zugriffssicherheit bei verteilt verarbeiteten Regeln oder die Administrierbarkeit von Systemen wie C²offein bzw. der von ihnen verarbeiteten Regeln.
- *Entwicklungswerkzeuge.* Für den weitergehenden Einsatz speziell von C²offein wäre eine Erweiterung des Systems um die üblichen Entwicklungswerkzeuge sinnvoll, zum Beispiel Modellierungswerkzeuge in Verbindung mit Regelgeneratoren, ein graphischer Regeleditor sowie ein „Debugger“ in Verbindung mit dem in Kapitel 9 skizzierten CORBA-Monitor für C²offein.
- *Konfiguration ereignisgetriebener Dienste.* Der gesamte Bereich der Konfiguration ereignisgetriebener Dienste ist in dieser Arbeit nur angerissen worden. Die systematische Ermittlung weiterer Kriterien für die Anwendung von Konfigurationen und die Verfeinerung des in Kapitel 9 erarbeiteten Schemas könnten Grundlage einer zumindest partiell automatischen Generierung anwendungsindividueller Konfigurationen von Systemen wie C²offein sein. Natürlich ist auch die Weiterentwicklung der konfigurierbaren Systemmerkmale selbst eine interessante Aufgabe.
- *Weitergehender Einsatz, Regelverarbeitungs-föderationen.* Die Einsatzpotentiale von Systemen wie C²offein wurden in Kapitel 9 nur angerissen. Die umfassende Anwendung solcher Systeme bei großen Regelmengen in heterogenen, verteilten Informationssystemen verspricht interessante Ergebnisse. Dies könnte bis zu Fragestellungen gehen, inwieweit in solchen ggf. unternehmensweiten oder gar unternehmensübergreifenden Informationssystemen mit unterschiedlichen Regelverarbeitungen umgegangen wird, wodurch kooperierende Regelverarbeitungen in „Regelverarbeitungs-föderationen“ denkbar werden.

Mit den erzielten Ergebnissen wurde das Tor für eine Fülle weiterer Forschungs- und Anwendungsarbeiten aufgestoßen, denn die Forschung im Bereich der diensteorientierten, ADBMS-artigen ECA-Regelverarbeitung in heterogen, verteilten Systemumgebungen steht erst am Anfang.