

Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

Ralf Bruns · Jürgen Dunkel

Event-Driven Architecture

Softwarearchitektur für ereignisgesteuerte
Geschäftsprozesse

 Springer

Prof. Dr. Ralf Bruns
Fachhochschule Hannover
Abteilung Informatik
Ricklinger Stadtweg 120
30459 Hannover
Germany
ralf.bruns@fh-hannover.de

Prof. Dr. Jürgen Dunkel
Fachhochschule Hannover
Abteilung Informatik
Ricklinger Stadtweg 120
30459 Hannover
Germany
juergen.dunkel@fh-hannover.de

ISSN 1439-5428

ISBN 978-3-642-02438-2

e-ISBN 978-3-642-02439-9

DOI 10.1007/978-3-642-02439-9

Springer Heidelberg Dordrecht London New York

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

© Springer-Verlag Berlin Heidelberg 2010

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Einbandentwurf: KuenkelLopka GmbH

Gedruckt auf säurefreiem Papier

Springer ist Teil der Fachverlagsgruppe Springer Science+Business Media (www.springer.com)

*Für Maria und Fritz Bruns,
Keno, Lara und Sabine!*

Für Cristina!

Vorwort

Eine Vielzahl von internen und externen Ereignissen beeinflusst die Abläufe und Geschäftsprozesse in Unternehmen – eine Warenlieferung trifft verspätet ein, ein Kunde annulliert seine Bestellung, ein Temperatursensor misst einen erhöhten Wert oder eine Fertigungsmaschine fällt aufgrund eines Defekts aus. Die Unternehmen sind gezwungen, auf diese Ereignisse angemessen und möglichst zeitnah zu reagieren – die Unternehmen agieren demnach ereignisgesteuert. Ereignissteuerung in Unternehmen ist heutzutage eine selbstverständliche Praxis und Bestandteil der natürlichen Arbeitsweise von Unternehmen.

Gegenstand des Buches

Event-Driven Architecture (EDA) als Architekturstil und *Complex Event Processing (CEP)* als Softwaretechnologie rücken Ereignisse als zentrales Strukturierungskonzept in den Fokus der Softwarearchitektur. Die resultierenden ereignisgesteuerten Anwendungssysteme ermöglichen eine realitätsnahe Abbildung der ereignisgesteuerten Geschäftsprozesse eines Unternehmens. Ereignisse werden identifiziert, transformiert, korreliert und bewertet. Ereignisströme werden analysiert, komplexe Sequenzen von Ereignissen mit kausalen, temporalen und räumlichen Bedingungen erkannt sowie fachliche Aktionen abgeleitet, die aus diesen Ereignismustern resultieren. Obwohl die Ereignisse häufig in extrem großen Volumina auftreten können, steht mit Complex Event Processing eine leistungsfähige Technologie zur Verfügung, um diese in Echtzeit zu verarbeiten.

Event-Driven Architecture als Architekturstil für Unternehmensanwendungen verfolgt das Ziel, die Agilität, Reaktionsfähigkeit und Echtzeitfähigkeit der Geschäftsprozesse eines Unternehmens zu erhöhen. Die Ereignisorientierung stellt einen wichtigen Schritt dar, um die bestehende Diskrepanz zwischen den realen Geschäftsprozessen und deren informationstechnischer Unterstützung zu vermindern. Event-Driven Architecture in Verbindung mit Complex Event Processing ermöglicht eine neue Qualität von Unternehmensanwendungen.

Ziele des Buches

Dieses Buch widmet sich ausführlich dem Thema Event-Driven Architecture und seiner fortgeschrittensten Form, dem Complex Event Processing. Es behandelt das Thema aus drei unterschiedlichen Sichtweisen, die gemeinsam eine umfassende Gesamtsicht dieser noch sehr jungen, aber zukunftsweisenden Form von Softwarearchitektur vermitteln sollen:

1. *Fachliche Sicht*: Welchen Nutzen bringt EDA für ein Unternehmen?

Bei Event-Driven Architecture handelt es sich um einen neuen Ansatz mit enormem fachlichen Potenzial. Das Potenzial wollen wir herausarbeiten, die Grenzen konventioneller Ansätze aufzeigen und besonders aussichtsreiche Anwendungsgebiete für EDA identifizieren.

2. *Konzeptionelle Sicht*: Was verbirgt sich softwaretechnisch hinter EDA?

Die zentralen Ideen und Konzepte von EDA und CEP werden sowohl auf einer konzeptionellen als auch detailliert auf einer technischen Ebene erläutert. So ergibt sich ein vollständiges Gesamtbild der technologischen Grundlagen und es bildet sich ein tiefes Verständnis der konkreten Umsetzung.

3. *Praktische Sicht*: Wie können EDA-Anwendungen realisiert werden?

Noch existieren relativ wenige EDA-Anwendungen im produktiven Einsatz in der Wirtschaft und folglich liegt wenig Erfahrungswissen vor. Anhand eines durchgehenden Fallbeispiels lässt sich die praktische Anwendung der Ereignisverarbeitung schrittweise nachvollziehen. Auch wesentliche Methoden für den Entwurf und die Entwicklung von professionellen EDA-Anwendungen werden vorgestellt.

Insgesamt betrachtet, spannt dieses Buch den Bogen vom fachlichen Nutzen über die technischen Hintergründe bis zum praktischen Einsatz und liefert somit eine fundierte und umfassende Sicht auf das Thema Event-Driven Architecture.

Leserkreis

Das Buch wendet sich an einen breiten Leserkreis: Einerseits an Entscheidungsträger, Softwarearchitekten und Softwareentwickler aus der betrieblichen Praxis, die in ihren Unternehmen ereignisgesteuerte Geschäftsprozesse durch moderne und zukunftsfähige Softwaresysteme unterstützen müssen. Andererseits an Studierende und Hochschullehrer/innen der Informatik sowie der Wirtschaftsinformatik, die einen praxisbezogenen Einblick in die grundlegenden Konzepte von ereignisorientierten Softwarearchitekturen und deren praktischer Umsetzung erhalten möchten.

Aufbau des Buches

Teil I – Event-Driven Architecture – Einführung führt umfassend in die Thematik der Ereignisverarbeitung als dynamischer Architekturstil für Unternehmensanwendungen ein. Die große fachliche Bedeutung von Ereignissen für Unternehmensanwendungen wird aufgezeigt, auf die Unzulänglichkeiten von konventionellen Softwarearchitekturen in diesem Kontext eingegangen und das fachliche Potenzial der

Ereignisorientierung herausgestellt. Es werden die grundlegenden Ideen, Prinzipien und Konzepte von Event-Driven Architecture und Complex Event Processing vorgestellt, bewertet und aussichtsreiche Anwendungsgebiete identifiziert.

Dieser Teil des Buches wählt eine nichttechnische Betrachtungsweise, so dass die Leser, die nicht so sehr an der technischen Umsetzung interessiert sind, durch diesen Teil einen Gesamtüberblick über Event-Driven Architecture erhalten und das fachliche Potenzial einschätzen können.

Teil II – Complex Event Processing – Konzepte vertieft die konzeptionellen Inhalte aus Teil I systematisch, indem Complex Event Processing als *die* zentrale Technologie einer Event-Driven Architecture ausführlich und detailliert vorgestellt wird. Die Repräsentation von Ereignissen durch Ereignismodelle, die Verarbeitung von Ereignissen mittels Ereignismustern und -regeln, der Aufbau einer CEP-Anwendung als Netzwerk von interagierenden, ereignisverarbeitenden Komponenten sowie die Behandlung der Ereignisse in den nachgelagerten Anwendungssystemen werden Schritt für Schritt erläutert.

Die konkrete technische Umsetzung der präsentierten Konzepte wird einerseits anhand einer vereinfachten Regelsprache für die Ereignisverarbeitung, andererseits anhand der Open-Source-CEP-Engine *Esper* vorgestellt. Ein durchgehendes Fallbeispiel aus dem Bereich Verkehrskontrolle und -steuerung verdeutlicht exemplarisch, wie die vorgestellten Konzepte in einem größeren fachlichen Zusammenhang eingesetzt werden.

Teil III – Entwurfsmuster und Architekturen für EDA stellt wichtige Konzepte für die professionelle Entwicklung von EDA-Anwendungen in der Praxis vor. Softwareentwicklungsprojekte im industriellen Maßstab sind äußerst komplexe Vorhaben. Das gilt selbstverständlich auch für EDA-Projekte. Diese Komplexität besteht auf zwei Ebenen:

1. Systemkomplexität: Wie wird das System richtig gebaut?
2. Prozesskomplexität: Wie wird der Entwicklungsprozess richtig gestaltet?

Mit Entwurfsmustern und Referenzarchitekturen für EDA-Anwendungen werden in diesem Teil wesentliche Konzepte für den Entwurf und die Entwicklung von EDA-Anwendungen behandelt. Sowohl Entwurfsmuster als auch Referenzarchitekturen repräsentieren wichtiges Erfahrungswissen, um ein ereignisgesteuertes System richtig zu bauen, also um die Systemkomplexität zu beherrschen. Für die Gestaltung des Entwicklungsprozesses, also um die Prozesskomplexität zu beherrschen, werden Vorgehensschritte und ein Vorgehensmodell vorgestellt, die auf die spezifischen Anforderungen von ereignisgesteuerten Systemen zugeschnitten sind.

Teil IV – Stand, Zusammenfassung und Ausblick beschließt das Buch. Der erreichte Entwicklungsstand von Event-Driven Architecture wird beleuchtet, die wichtigsten Inhalte des Buches zusammengefasst und ein Ausblick auf die Einführung von EDA in die Praxis gegeben.

Webseite zum Buch

Natürlich gibt es auch eine Webseite zum Buch, auf der zusätzliche Informationen zum Buch sowie zu den Themen Event-Driven Architecture und Complex Event Processing bereitgestellt sind. Die URL der Seite lautet:

`http://event-driven-architecture.org`

Auf dieser Seite finden Sie unter anderem:

- Dateien mit den Source-Code-Beispielen aus dem Buch
- interessante Links zu den Themen EDA und CEP
- Links zu wichtigen kommerziellen Herstellern von EDA-Plattformen und Open-Source-Produkten für CEP

Für weitere Fragen, Hinweise auf Fehler, Meinungen und Anregungen stehen wir Ihnen sehr gerne zur Verfügung. Unsere Kontaktdaten finden Sie auf der Webseite.

Entstehung und Danksagung

Der Anstoß zu diesem Buch kam durch unsere Arbeiten im Bereich Softwarearchitektur in der Abteilung Informatik der Fachhochschule Hannover. Erste Berührungspunkte mit Event-Driven Architecture und Complex Event Processing ergaben sich über Abschlussarbeiten. Wir waren sofort von dem Thema überzeugt, so dass es zunächst Einzug in unsere Lehrveranstaltungen fand und schließlich zu Kooperationsprojekten mit industriellen Partnern führte.

Wir hoffen mit dem vorliegenden Buch die Aufmerksamkeit für die Ereignisverarbeitung als Architekturstil zu erhöhen und damit einen Beitrag für die weitere Entwicklung des Themengebiets zu leisten. Es würde uns sehr freuen, wenn Event-Driven Architecture in Zukunft einen signifikanten Platz im Kanon der Architekturkonzepte für die Entwicklung von zukunfts- und leistungsfähigen Anwendungsarchitekturen in der betrieblichen Praxis einnehmen würde.

Zum Schluss möchten wir uns ganz besonders bei unseren Familien bedanken, die uns in den vergangenen Monaten mit so viel Geduld unterstützt haben – danke euch allen!

Hannover,
Dezember 2009

*Ralf Bruns
Jürgen Dunkel*

Inhaltsverzeichnis

Teil I Event-Driven Architecture – Einführung

1	Einführung und Motivation	3
1.1	EDA und CEP – Einführung	3
1.2	Motivation	5
1.3	Historischer Kontext	9
2	Ereignisse in Unternehmensanwendungen	13
2.1	Bedeutung von Ereignissen	13
2.1.1	Ereignisgesteuerte Geschäftsprozesse	14
2.1.2	Herausforderungen im Umgang mit Ereignissen	15
2.1.3	Ereignismuster und -abstraktion	18
2.2	Grenzen von konventionellen Softwarearchitekturen	23
2.2.1	Geschäftssicht in konventionellen Architekturen	24
2.2.2	Technologiesicht in konventionellen Architekturen	26
2.3	Ereignisorientierung als Architekturstil	28
2.3.1	Geschäftssicht in ereignisorientierten Architekturen	28
2.3.2	Technologiesicht in ereignisorientierten Architekturen	30
2.3.3	Fachliche Gesamtsicht der Ereignisorientierung	31
2.4	EDA und SOA	34
2.4.1	Grundidee von SOA	34
2.4.2	Gegenüberstellung von SOA und EDA	36
2.4.3	SOA + EDA = Event-Driven SOA	38
2.5	Anwendungsgebiete für EDA	39
2.5.1	Monitoring	40
2.5.2	Sensornetzwerke	41
2.5.3	Analytik	44
2.5.4	Enterprise Application Integration	45

3 Event-Driven Architecture und Complex Event Processing im Überblick 47

3.1 EDA – Grundkonzepte 47

 3.1.1 Ereignisse – Ereignisgesteuert – EDA 48

 3.1.2 EDA mit CEP 56

 3.1.3 EDA-Architektur und -schichten 61

3.2 CEP – Grundkonzepte 65

 3.2.1 Ereignismodelle, -muster und -regeln 66

 3.2.2 Netzwerk von CEP-Komponenten 68

3.3 EDA – Bewertung 72

3.4 Fallstudie: Verkehrskontrolle und -steuerung 76

Teil II Complex Event Processing – Konzepte

4 Ereignismodelle 85

4.1 Ereignisse und Ereignisströme 85

4.2 Ereignisbeziehungen und -hierarchien 91

4.3 Modelle 95

4.4 Ereignis-Constraints 97

4.5 Fallstudie: Verkehrskontrolle und -steuerung 99

5 Ereignisverarbeitung 105

5.1 Regeln zur Ereignisverarbeitung 105

5.2 Konzepte zur Beschreibung von Ereignismustern 108

5.3 Aktionen in Ereignisregeln 115

5.4 Ereignisregeln mit Esper 125

 5.4.1 Ereignisse in Esper 127

 5.4.2 Aufbau von Ereignisregeln in Esper 128

 5.4.3 Ereignismuster in Esper 129

 5.4.4 Aktionen in Esper 136

5.5 Event Processing Agents und Event Processing Engines 139

5.6 Event Processing Networks 142

5.7 Realisierungsplattformen und Sprachtypen 145

5.8 Fallstudie: Verkehrskontrolle und -steuerung 149

 5.8.1 Event Processing Network des Verkehrskontrollsystems 149

 5.8.2 Ereignisregeln im Verkehrskontrollsystem 153

6 Ereignisbehandlung 157

6.1 Aufgaben der Ereignisbehandlung 157

6.2 Integration der CEP-Komponente 159

6.3 Ereignisgesteuerte Prozesse 164

6.4 Visualisierung von Ereignissen 167

6.5 Erzeugen von Geschäftsereignissen 170

6.6 Fallstudie: Verkehrskontrolle und -steuerung 172

Teil III Entwurfsmuster und Architekturen für EDA

- 7 Entwurfsmuster** 177
 - 7.1 Muster in Event-Driven Architecture 177
 - 7.2 Architekturmuster 179
 - 7.2.1 EDA-Schichten 179
 - 7.2.2 Ereignisagenten 181
 - 7.2.3 Verarbeitungs-Pipeline 183
 - 7.3 Designmuster 185
 - 7.3.1 Konsistenz von Ereignissen 186
 - 7.3.2 Reduktion der Ereignismenge 188
 - 7.3.3 Transformation von Ereignissen 191
 - 7.3.4 Synthese von Ereignissen 194
- 8 Referenzarchitekturen** 201
 - 8.1 Referenzarchitekturen für Event-Driven Architecture 201
 - 8.2 Allgemeine Referenzarchitektur 203
 - 8.3 EDA für Sensornetzwerke 208
 - 8.4 EDA für Analysesysteme 210
- 9 Vorgehen bei der Entwicklung von EDA-Anwendungen** 213
 - 9.1 Einführung 213
 - 9.2 Prozessschritte für EDA 214
 - 9.3 Vorgehensmodell für EDA 216

Teil IV Stand, Zusammenfassung und Ausblick

- 10 Stand, Zusammenfassung und Ausblick** 223
 - 10.1 Aktueller Entwicklungsstand 223
 - 10.2 Zusammenfassung 225
 - 10.3 Ausblick: Einführung von EDA in die Praxis 226
- Literaturverzeichnis** 231
- Sachverzeichnis** 237

Abkürzungen

API	Application Programming Interface
BAM	Business Activity Monitoring
BI	Business Intelligence
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling Notation
CEP	Complex Event Processing
CQL	Continuous Query Language
CRM	Customer Relationship Management
CSV	Comma-Separated Value
EAI	Enterprise Application Integration
ECA	Event-Condition-Action
EDA	Event-Driven Architecture
EPA	Event Processing Agent
EPL	Event Processing Language
EPN	Event Processing Network
EQL	Esper Query Language
ESB	Enterprise Service Bus
ESP	Event Stream Processing
FIFO	First-In-First-Out
GPS	Global Positioning System
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
JMS	Java Message Service
LUW	Logical Unit of Work
MOM	Message-oriented Middleware
OCL	Object Constraint Language
OWL	Web Ontology Language
POJO	Plain Old Java Object
RDF	Resource Description Framework
RDF-S	RDF Schema

RFID	Radio Frequency Identification
RIF	Rule Interchange Format
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SLA	Service Level Agreement
SOA	Service-oriented Architecture
SQL	Structured Query Language
WSDL	Web Services Description Language
XML	Extensible Markup Language

Teil I
Event-Driven Architecture – Einführung

Teil I – Event-Driven Architecture – Einführung führt umfassend in die Thematik der Ereignisverarbeitung als dynamischer Architekturstil für Unternehmensanwendungen ein. Die große fachliche Bedeutung von Ereignissen für Unternehmensanwendungen wird aufgezeigt, auf die Unzulänglichkeiten von konventionellen Softwarearchitekturen in diesem Kontext eingegangen und das fachliche Potenzial der Ereignisorientierung herausgestellt. Es werden die grundlegenden Ideen, Prinzipien und Konzepte von Event-Driven Architecture und Complex Event Processing vorgestellt, bewertet und aussichtsreiche Anwendungsgebiete identifiziert.

Dieser Teil des Buches wählt eine nichttechnische Betrachtungsweise, so dass die Leser, die nicht so sehr an der technischen Umsetzung interessiert sind, durch diesen Teil einen Gesamtüberblick über Event-Driven Architecture erhalten und das fachliche Potenzial einschätzen können.

Kapitel 1 – Einführung und Motivation leitet in die Thematik der komplexen Ereignisverarbeitung ein. Es wird aufgezeigt, dass die Verarbeitung von Ereignissen für die realen Prozesse in Unternehmen und die Kommunikationsvorgänge zwischen Unternehmen alltägliche und gängige Praxis sind und sich daraus die Motivation für einen neuen, ereignisorientierten Stil der Softwarearchitektur ableitet. Die historische Entwicklung ereignisverarbeitender Systeme in der Informatik wird betrachtet und die jeweiligen Einflüsse auf EDA und CEP identifiziert.

Kapitel 2 – Ereignisse in Unternehmensanwendungen beleuchtet das Thema Ereignisorientierung primär aus fachlicher Sicht. Die fachliche Bedeutung von Ereignissen für die Geschäftsprozesse eines Unternehmens ist Gegenstand dieses Kapitels. Insbesondere wird das fachliche Potenzial, das sich durch die Ereignisorientierung als Architekturstil in einer EDA ergibt, herausgestellt. Konventionelle Softwarearchitekturen und Service-orientierte Architekturen (SOA) werden mit EDA verglichen und potenziell vielversprechende Einsatzgebiete für EDA betrachtet.

Kapitel 3 – EDA und CEP im Überblick erläutert die wichtigsten Begriffe, Ideen und Entwurfsprinzipien, die sich hinter Event-Driven Architecture und Complex Event Processing verbergen. Architekturschichten und -elemente einer EDA werden vorgestellt, so dass sich ein vollständiges Gesamtbild des Architekturstils, seiner Kernelemente und Haupteigenschaften ergibt. Das Kapitel schließt mit einer Bewertung von EDA für Unternehmensanwendungen und der Vorstellung einer Fallstudie.

Kapitel 1

Einführung und Motivation

Zusammenfassung Agilität und Effizienz der betrieblichen Geschäftsprozesse sind entscheidende Faktoren für die Leistungs- und Wettbewerbsfähigkeit von modernen Unternehmen. Ereignisse spielen hierbei eine entscheidende Rolle, da alle Abläufe in der realen Welt durch eine Vielzahl und Vielfalt von unterschiedlichen Ereignissen beeinflusst werden. Event-Driven Architecture (EDA) und Complex Event Processing (CEP) repräsentieren einen neuen Stil von Unternehmensanwendungen, bei dem Ereignisse in das Zentrum der Softwarearchitektur rücken. Dieses Kapitel führt in die Thematik der Ereignisverarbeitung mit EDA und CEP ein und motiviert die Bedeutung von Ereignissen für die realen Prozesse in Unternehmen bzw. Unternehmensnetzwerken. In verschiedenen Bereichen der Informatik werden Ereignisse schon seit einiger Zeit betrachtet. Der letzte Abschnitt geht auf diese Bereiche ein und zeigt deren Einflüsse auf die moderne Ereignisverarbeitung.

1.1 EDA und CEP – Einführung

Unternehmen sind seit jeher einer permanenten Evolution ausgesetzt. Die Marktbedingungen und -teilnehmer ändern sich und es gilt, die sich wandelnden gesetzlichen Rahmenbedingungen einzuhalten sowie die Unternehmensziele neu zu definieren und umzusetzen. Deshalb streben Unternehmen nach *Agilität* und *Effizienz* ihrer Geschäftsprozesse, um auf diese Evolution angemessen reagieren und die sich daraus ergebenden Chancen aktiv gestalten zu können.

Die Informationstechnologie (IT) bildet seit vielen Jahren die Basis für die Umsetzung der geschäftlichen Aufgaben und hat somit entscheidenden Einfluss auf die Realisierung der unternehmerischen Ziele. Trotz großer Fortschritte, wie sie beispielsweise in den letzten Jahren durch Service-orientierte Architekturen (SOA) erzielt wurden, besitzen die betrieblichen IT-Systeme jedoch häufig nicht die erforderliche Flexibilität, um die fachliche Agilität entsprechend abzubilden.

Erschwerend kommt hinzu, dass die internen und externen Anforderungen an Agilität stetig wachsen, sei es durch erweiterte technische Möglichkeiten (z.B. durch

mobile Endgeräte, die permanent online sind, oder den Einsatz von Sensoren zur Messung von physikalischen Werten) oder verändertes Kundenverhalten (z.B. der gestiegenen Erwartungshaltung an die Geschwindigkeit der Reaktion eines Unternehmens auf eine Kundenanfrage).

Ereignissen kommt hierbei eine entscheidende Rolle zu, da alle Vorgänge und Abläufe in der realen Welt durch eine Fülle von unterschiedlichen Ereignissen beeinflusst werden.

Event-Driven Architecture (EDA) und Complex Event Processing (CEP) repräsentieren einen neuen Stil von Unternehmensanwendungen, bei dem *Ereignisse* in das Zentrum der *Softwarearchitektur* rücken – *Ereignisorientierung als Architekturstil*.

Während EDA einen ereignisorientierten *Entwurfstil* für Anwendungsarchitekturen (enterprise application architecture) beschreibt, handelt es sich bei CEP um eine Technologie zur *dynamischen Verarbeitung* von großen Volumina von *Ereignissen in Echtzeit*. EDA ist demnach ein generelleres Konzept als CEP, und CEP kann als Kernbestandteil einer EDA angesehen werden, wie in Abbildung 1.1 symbolisiert.

Aufgrund des noch jungen Fachgebiets der *Ereignisverarbeitung* (event processing) und der rasanten Entwicklung von EDA und CEP innerhalb der letzten Jahre haben insbesondere Business-Analysten und die Hersteller von Softwareprodukten diverse Schlagworte im Umfeld der Ereignisverarbeitung geprägt. Abbildung 1.1 listet eine Auswahl dieser Begriffe auf.

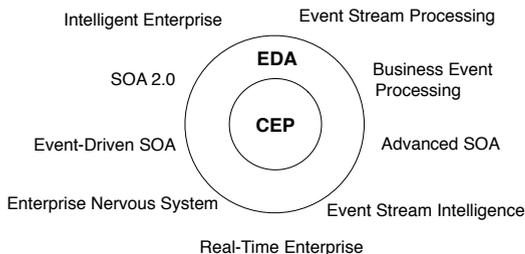


Abb. 1.1 Schlagworte im Umfeld von EDA und CEP

Die Grundkonzepte, die sich hinter den diversen unterschiedlichen Bezeichnungen verbergen, sind jedoch immer die gleichen: *Event-Driven Architecture* und *Complex Event Processing* – die Themen dieses Buches.¹

¹ Mit *Event-Driven Architecture* und *Complex Event Processing* verwenden wir in diesem Buch die etablierte englische Fachterminologie, entsprechend eindeutige deutsche Bezeichnungen existieren bislang nicht. Übersetzungen wie ereignisgesteuerte Architekturen oder komplexe Ereignisverarbeitung sind nicht etabliert.

Die **Ereignisorientierung** als Gestaltungsprinzip von Softwaresystemen bildet einen wichtigen Schritt auf dem Weg, die bestehende konzeptionelle Lücke zwischen den Geschäftsinteressen eines Unternehmens und den IT-Systemen zu verkleinern. Ereignisorientierung besitzt das Potenzial, dass die Architekturen von Unternehmensanwendungen *agiler, reaktionsschneller* und *echtzeitfähig* werden.

1.2 Motivation

Moderne Unternehmen agieren heutzutage in unternehmensübergreifenden Netzwerken von Partnerunternehmen, öffentlichen Institutionen, Dienstleistern, Informationsdiensten und diversen weiteren Institutionen. Die Qualität und Effizienz der Leistungserbringung eines Unternehmens hängt im wesentlichen Maße von der Qualität und dem Fluss der Informationen in diesem Netzwerk ab.

Ereignisse in Unternehmensnetzwerken

Ein verteiltes Unternehmensnetzwerk ist beispielhaft für ein Unternehmen aus der Logistikbranche in Abbildung 1.2 visualisiert. Das Logistikunternehmen muss im Rahmen der Erbringung einer logistischen Dienstleistung mit diversen anderen Institutionen bzw. mit deren Informationssystemen *kommunizieren*. Für ein Logistikunternehmen sind dies u.a. behördliche Institutionen wie Zoll oder Hafenbehörde, Fertigungsunternehmen, deren produzierte Waren transportiert werden sollen, Geschäftskunden mit vermittelten Transportaufträgen, Empfänger, zu denen die Waren gebracht werden müssen, Transportunternehmen, die als Subauftragnehmer Teilstrecken des Transportweges übernehmen, diverse weitere Partnerunternehmen und andere externe Institutionen wie Wetter- und Verkehrsinformationsdienste. Häufig bilden die Logistikprozesse wichtige Teilprozesse in einer komplexen Zulieferkette (supply chain) und müssen entlang der Prozesskette mit den involvierten Partnern *integriert* werden.

Die an der Leistungserbringung beteiligten Institutionen kommunizieren durch den Austausch von *Nachrichten* miteinander, die zwischen den Institutionen bidirektional hin und her fließen. Als Medium der Kommunikation können sehr unterschiedliche Formen zum Einsatz kommen, z.B. persönliche Gespräche, Internet, Telefon, E-Mail, EDI oder Fax.

Jede Nachricht enthält die Daten eines relevanten Vorkommnisses oder Sachverhalts, über das die interessierten Parteien im Netzwerk informiert werden sollen. Eine Nachricht lässt sich somit als Repräsentation eines eingetretenen *Ereignisses* interpretieren, über dessen Auftreten die Partner in Kenntnis gesetzt werden sollen. Die Kommunikation im Netzwerk besteht demnach aus dem Austausch von Nachrichten mit *Ereignisdaten*. Die Unternehmen reagieren auf die Ereignisse, die sie

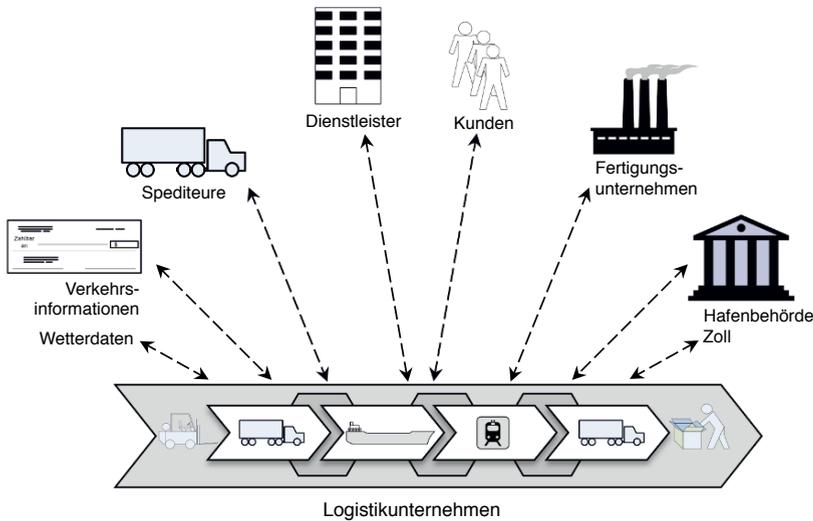


Abb. 1.2 Logistikunternehmen im Netzwerk mit verbundenen Institutionen

empfangen, und erzeugen neue Ereignisse, die sie an ihre Partner senden. Das ganze System ist *„ereignisgesteuert“*, da es auf dem *Erkennen*, *Verarbeiten* und *Reagieren* von bzw. auf Ereignissen beruht und die Ereignisse das *Mittel* der Kommunikation bilden. In einem realen Unternehmensnetzwerk können extrem große Volumina von Ereignissen ausgetauscht werden, tausende bis zu mehrere hunderttausend Ereignisse pro Sekunde.

Das Logistikunternehmen in Abbildung 1.2 steht beispielhaft für ein verteiltes unternehmensübergreifendes System. Generell lässt sich festhalten, dass die Geschäftsprozesse global agierender Unternehmen, gleich welcher Branche, auf dem Austausch von Ereignissen durch Nachrichten innerhalb des Unternehmens und unternehmensübergreifend mit Partnerinstitutionen über ein verteiltes Netzwerk beruhen.

Ereignisgesteuerte Unternehmen und Institutionen sind demnach in der Realität weit verbreitet. *Ereignisgesteuerte* Unternehmen sind dadurch gekennzeichnet, dass

- sie Ereignisse *erkennen*, *analysieren* und darauf *reagieren*,
- sie durch den Austausch von (Ereignis-)Nachrichten miteinander *kommunizieren*,
- sie neue Ereignisse *erzeugen* und diese an ihre Partner *senden* und
- ihre *Aktivitäten* durch Ereignisse *ausgelöst* werden.

Abbildung 1.3 zeigt einen typischen Prozessablauf für ein Logistikunternehmen, in dem der Transport einer Ware mit unterschiedlichen Verkehrsträgern, ausgehend vom Versand der Ware bis zur Ablieferung bei einem Empfänger, dargestellt ist.

Im Zuge der Prozessdurchführung können eine Fülle von höchst unterschiedlichen Ereignissen auftreten, die Einfluss auf die Leistungserbringung – sowohl po-

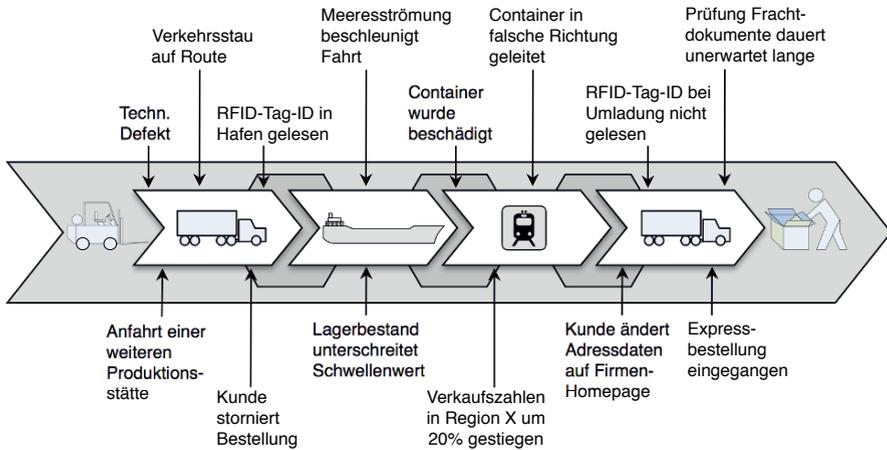


Abb. 1.3 Logistischer Prozess mit Beispielen für Ereignisse

sitiv als auch negativ – haben können. In der Abbildung sind einige Ereignisse beispielhaft aufgeführt. Die Ereignisse können *unvorhergesehen* (z.B. technischer Defekt am LKW), *erwartet* (z.B. RFID-Transponder bei Umladung im Hafen gelesen) oder *erwartet, aber nicht eingetreten* sein (z.B. RFID-Transponder bei Umladung auf LKW *nicht* gelesen) und sowohl aus *internen* (z.B. Lagerbestand unterschreitet vorgegebenen Schwellenwert) als auch *externen* Quellen (z.B. Meldung über Verkehrsstau auf geplanter Route) stammen.

Typische Fragestellungen

Einzelne Ereignisse sind isoliert betrachtet häufig wenig aussagekräftig, z.B. das Lesen eines RFID-Transponders an einem bestimmten Standort. Entscheidend sind Fragestellungen auf der Ebene der betrieblichen Geschäftsprozesse, deren Antworten direkten Einfluss auf operative Entscheidungen haben. Um diese beantworten zu können, müssen jedoch oft mehrere Ereignisse im Zusammenhang betrachtet werden. Typische Fragen mit hohem fachlichem Informationswert sind beispielsweise die folgenden:

- “Warum erfolgte die Lieferung einer Ware verspätet?“
Die Antwort könnte sowohl mehrere interne Transportereignisse als auch externe Ereignisse, z.B. Verkehrsstaus oder Wetterbedingungen, umfassen.
- “Erreicht meine Ware heute noch den Bestimmungsort?“
Die Antwort könnte sich aus dem Zusammenhang von RFID-Ereignissen, die anzeigen, wo sich die Ware zurzeit befindet, und externen Ereignissen über gemeldete Verkehrsprobleme und Wetterprognosen ergeben.
- “Wo wurde der Container fehlgeleitet?“
Die Antwort könnte aus den Beziehungen zwischen RFID-Ereignissen, die den

letzten Standort bestimmen, und Transportereignissen, wie z.B. LKW am Standort X zur Zeit Y entladen, abgeleitet werden.

- “Halten unsere Abläufe die gesetzlichen Vorgaben ein?“
Die Antwort könnte aus Ereignissen resultieren, die Probleme mit behördlichen Institutionen signalisieren, z.B. Verzögerungen beim Zoll oder bei der Prüfung der Frachtdokumente.
- “Besteht eine Chance, die gleichzeitig ablaufenden Prozesse zu optimieren?“
Beispielsweise lassen sich Transporte durch die Zusammenfassung von Lieferungen aufgrund von räumlichen Korrelationen, den Austausch von Gütern zwischen gleichzeitigen Lieferprozessen durch die Nutzung freier Kapazitäten oder die kurzfristige Umleitung von Waren aufgrund von Expressbestellungen effizienter durchführen.
- “Laufen unsere derzeitigen Prozesse störungsfrei?“
Beispielsweise deutet das *Nicht-Lesen* eines erwarteten RFID-Transponders an einer Umladungsstation auf eine Fehlleitung des entsprechenden Gegenstands hin. Geeignete Maßnahmen zur Wiederauffindung des Gegenstands können zeitnah eingeleitet werden.

In einem Unternehmen existieren natürlich diverse weitere Fragestellungen der obigen Art. Diese Fragen reichen von der operativen Steuerung der Unternehmensabläufe, dem Erkennen von Marktchancen und Optimierungspotenzial, dem Erkennen von Störungen bzw. unerwarteten Situationen bis zur Sicherstellung der Konformität zu gesetzlichen Richtlinien.

Die Beantwortung dieser Art von Fragen setzt ein *Verstehen* der Ereignisse und ihrer Zusammenhänge entlang der Prozesskette voraus. Verstehen bedeutet in diesem Kontext die Bewertung des Einflusses der Ereignisse auf die geschäftlichen Abläufe und strategischen Ziele des Unternehmens. Durch die unmittelbare Relevanz der Fragen für die operativen Prozesse ist es essenziell, die Antworten in *Echtzeit* zu erhalten, um zeitnah reagieren zu können.

Konsequenzen für die Unternehmensanwendungen

Konventionellen IT-Anwendungsarchitekturen liegt eine eher ablauforientierte Sicht auf die Unternehmensanwendungen zugrunde. Diese Systeme sind in der Regel zu starr und unflexibel, um das Verständnis von Geschäftsereignissen und die zeitnahe Ableitung von fachlichen Reaktionen zu leisten. Gesucht ist eine neue Art von Anwendungsarchitektur, die es ermöglicht, die Auswirkungen von Ereignissen auf die Unternehmensziele, -richtlinien und Geschäftsprozesse abzubilden.

Da heutige Unternehmensarchitekturen in der Regel nicht explizit auf Ereignisse im Unternehmenskontext eingehen, besteht häufig eine Diskrepanz zwischen den realen Geschäftsprozessen und deren informationstechnischer Unterstützung. Roy Schulte formuliert dies wie folgt:

“The real world is mostly event driven, and event-driven situations are best addressed by event-driven business applications (to eliminate the impedance mismatch).“

(Quelle: R. Schulte, Gartner Inc., Vortrag “Event Processing in Business Applications“, 15. März 2006)

Um diesen Herausforderungen gerecht werden zu können, ist für Unternehmen das *Erkennen*, *Analysieren* und *Reagieren* von bzw. auf Ereignisse mindestens genauso wichtig wie das Ausführen von vordefinierten Prozessen.

Mit *Event-Driven Architecture* als Architekturstil und *Complex Event Processing* als leistungsfähiger Technologie zur Ereignisverarbeitung stehen zwei ausdrucksstarke Ansätze zur Entwicklung ereignisgesteuerter Geschäftsanwendungen zur Verfügung. Mit EDA und CEP lassen sich Geschäftsprozesse *wirklichkeitsnah* und *problemadäquat* abbilden und somit die Geschäftsziele und IT-Systeme in Einklang bringen.

Unternehmen, die den *ereignisorientierten Architekturstil* in ihre Unternehmensanwendungen integrieren, können *schneller* und *flexibler* auf sich verändernde Geschäftsbedingungen *reagieren* und erreichen insgesamt eine höhere *Agilität* und *Effizienz* ihrer Prozessabläufe. Wohingegen konventionelle Softwarearchitekturen für die Verarbeitung von Ereignissen oft komplexes und kostspieliges Software Engineering erfordern.

1.3 Historischer Kontext

Event-Driven Architecture und Complex Event Processing haben sich in den letzten Jahren als eigenständiges Fachgebiet der *Ereignisverarbeitung* (event processing) etabliert. In der Informatik wird jedoch die Verarbeitung von Ereignissen bereits seit vielen Jahren untersucht.

Ereignisbasierte Konzepte wurden unabhängig voneinander in unterschiedlichen Bereichen entwickelt. Eine ausführlichere Diskussion der Entwicklung der Ereignisverarbeitung findet sich in [66] und [67].

- Die *diskrete Ereignissimulation* (discrete event simulation) verfolgt das Ziel, das natürliche Verhalten von Systemen rechnergestützt zu simulieren. Ausgehend von Eingabedaten erzeugt die Simulation Ereignisse, die die Interaktion zwischen Teilkomponenten des Systems nachbilden. Die sich während der Simulation ergebende Folge von Ereignissen repräsentiert somit das Verhalten des Modells. Die Analyse der Ereignisfolge berücksichtigt *kausale Abhängigkeiten* zwischen Ereignissen und leitet Systemkenngrößen aus den simulierten Ereignissen bzw. deren *Beziehungsmustern* ab [66].
- In *Computer-Netzwerken* stellt das Senden oder Empfangen von Datenpaketen jeweils ein signifikantes Ereignis dar. Zur Strukturierung der Kommunikationsaufgaben werden in Netzwerken die Kommunikationsereignisse in aufeinander aufbauende *Schichten* (layer) gegliedert, beispielsweise im ISO/OSI-Referenzmodell oder im TCP/IP-Referenzmodell. Diese Modelle basieren auf der *Abstraktion* von Ereignissen – ausgehend von einfachen Netzwerkdetails (unterste Ebene) bis zu aussagekräftigen Anwendungsereignissen (höchste Ebene) nimmt der *Abstraktionsgrad* von Schicht zu Schicht zu [66]. Die in diesem Kontext entstandenen Netzwerkmanagement-Werkzeuge ermöglichen das Verfolgen und die Kontrolle von Ereignissen im Netzwerk. Diese Werk-

zeuge können als frühe Vorgänger des Business Activity Monitoring (BAM) angesehen werden und enthalten bereits wesentliche grundlegende Konzepte von Echtzeitkontrollsystemen.

- *Aktive Datenbanken* (active databases) [85] erweitern traditionelle Datenbankmodelle um Techniken zur unmittelbaren Reaktion auf Datenänderungen, und zwar mit dem Ziel einer *Echtzeitverarbeitung*. Mit Hilfe sogenannter *ECA-Regeln* (event-condition-action rules) können einzelne *Ereignisse* sowie einfache *Muster* von Ereignissen definiert werden, deren Auftreten im operativen Betrieb dann *Aktionen auslösen* kann, wie z.B. den Aufruf einer Anwendungsfunktion. Spezifische Sprachen zur Formulierung von ECA-Regeln beinhalten Konzepte wie *zusammengesetzte Ereignisse* (composite events) und *temporale Constraints* [66].
- In der *Softwarearchitektur* hat sich der Austausch von Ereignissen in Form von Nachrichten als *Kommunikationsstil* zwischen verteilten Systemkomponenten etabliert. Insbesondere im Bereich der Anwendungsintegration (Enterprise Application Integration, EAI) kommt dieser Kommunikationsstil zum Einsatz. Ereignisse sind Bestandteil von Unternehmensarchitekturen (enterprise architecture) [40, 51] und Entwurfsmustern (design pattern) wie beispielsweise dem Observer-Pattern [43].

Die Kommunikationsinfrastruktur basiert auf *ereignisgesteuerter Kommunikation* auf einer niedrigen, technischen Ebene und unterstützt eine *ereignisgesteuerte Anwendungsarchitektur*, also eine Event-Driven Architecture, auf der Ebene der Unternehmensanwendungen [66].

Weitere, wichtige Konzepte mit starkem Einfluss auf EDA und CEP stammen aus dem Gebiet der *Künstlichen Intelligenz* (artificial intelligence). Hier sind insbesondere die Arbeiten zur Repräsentation und zum Schließen über *zeitabhängige Informationen* (*temporales Schließen*, temporal reasoning) [4] sowie die Konzepte zu *regelbasierten Systemen* (production systems) [78] und *wissensbasierten Systemen* (knowledge-based systems) zu nennen.

Im Jahr 2002 veröffentlichte David Luckham sein grundlegendes Buch “*The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*“ [63], in dem er die diversen Vorarbeiten im Kontext der Verarbeitung von Ereignissen zum Konzept des *Complex Event Processing (CEP)* zusammenführte. Das Buch stellt einen wesentlichen Meilenstein für die Etablierung der Ereignisorientierung als Fachdisziplin dar.

Inzwischen haben etliche Softwarehersteller den Markt für Ereignisverarbeitung erkannt und bieten diverse professionelle Softwareplattformen für Complex Event Processing bzw. Event Stream Processing an. Neben kommerziellen Werkzeugen von Herstellern, u.a. Progress, Aleri, IBM, Oracle, Tibco, Coral8, Stream-Base Systems, sind auch leistungsfähige Open-Source-Produkte, u.a. Esper [34] oder Drools [54], verfügbar.

Mit der *Event Processing Technical Society (EPTS)*² wurde im Jahr 2008 erstmalig eine internationale Fachorganisation gegründet, die explizit auf das Fachgebiet der Ereignisverarbeitung fokussiert ist. Die EPTS verfolgt die Ziele, das Verständnis und die Entwicklung von Prinzipien und Technologien zur Ereignisverarbeitung zu fördern, die Entwicklung von Standards zu unterstützen und ein kooperatives Umfeld für Kommunikation und Lernen in diesem Fachgebiet zu bieten.

² Weitere Informationen zur Event Processing Technical Society (EPTS) finden sich unter der URL <http://www.ep-ts.com>.