



Xpert.press



Ralf Bruns · Jürgen Dunkel

Event-Driven Architecture

Softwarearchitektur für
ereignisgesteuerte Geschäftsprozesse

 Springer

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