



Universität
Rostock



Traditio et Innovatio

Event-Driven Architectures (Seminar)

Organisatorisches

Univ.-Prof. Dr.-Ing. habil. Gero Mühl

Architektur von Anwendungssystemen (AVA)
Fakultät für Informatik und Elektrotechnik (IEF)
Universität Rostock

Dozent

Name	Univ.-Prof. Dr.-Ing. habil. Gero Mühl
Adresse	Architektur von Anwendungssystemen Fakultät für Informatik und Elektrotechnik Universität Rostock Albert-Einstein-Str. 22
Büro	Raum 268
Telefon	0381 / 498 7621 (Sekretariat)
E-Mail	gero.muehl@uni-rostock.de
WWW	http://www.wava.informatik.uni-rostock.de
Sprechstunde	Dienstag, 13:00 – 14:00 Uhr und nach Vereinbarung, fällt bis auf Weiteres aus



What is an event?

“a software message indicating that something has happened” (computing)

“a set of outcomes to which a probability is assigned” (probability theory)

“a notable occurrence at a particular point in time” (UML)

“an object in time”
(philosophy)

“a point in space at an instant in time, i.e., a location in space-time” (relativity theory)

“a type of synchronization mechanism”
(synchronization primitive)

“a happening of interest”
(publish/subscribe)

“a significant change in state” (EDA)

“results of a fundamental interaction between subatomic particles” (particle physics)

Examples for Events

- > A method of an object is called or returns
- > An e-mail is received
- > A new stock quote is available
- > A key is pressed on a keyboard
- > A computer user has moved the mouse
- > A customer has issued a new order
- > The warehouse is low on a certain item
- > An USB stick is connected to a port
- > A new employee is hired
- > The data of a hard disk read is now available
- > The temperature has raised above a limit
- > A machine in a factory has failed
- > A hard disk is about to fail
- > ...

Events occur on various levels of abstraction ranging from hardware interrupts to state changes of business processes

Properties of Events

- > Observable happening
 - > Leads to **change in state**
 - > Different state before and after the event
- > Instant in time
 - > **No duration** (practically very brief)
 - > Compare event vs. process
- > Event vs. **notifications**
 - > Events just occur, they do not travel
 - > Notifications are messages that describe event occurrences
 - > Event/notification often used interchangeably

Why Bothering with Events?

AP The Associated Press 
@AP

 Following

Breaking: Two Explosions in the White House and Barack Obama is injured

 Reply  Retweet  Favorite  More

876
RETWEETS

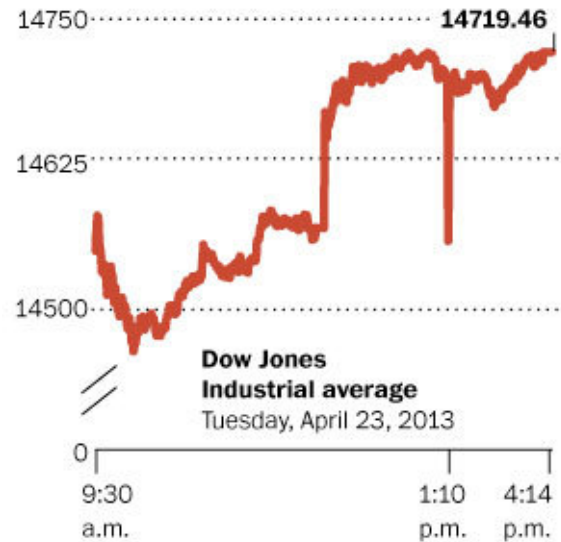
32
FAVORITES



1:07 PM - 23 Apr 13

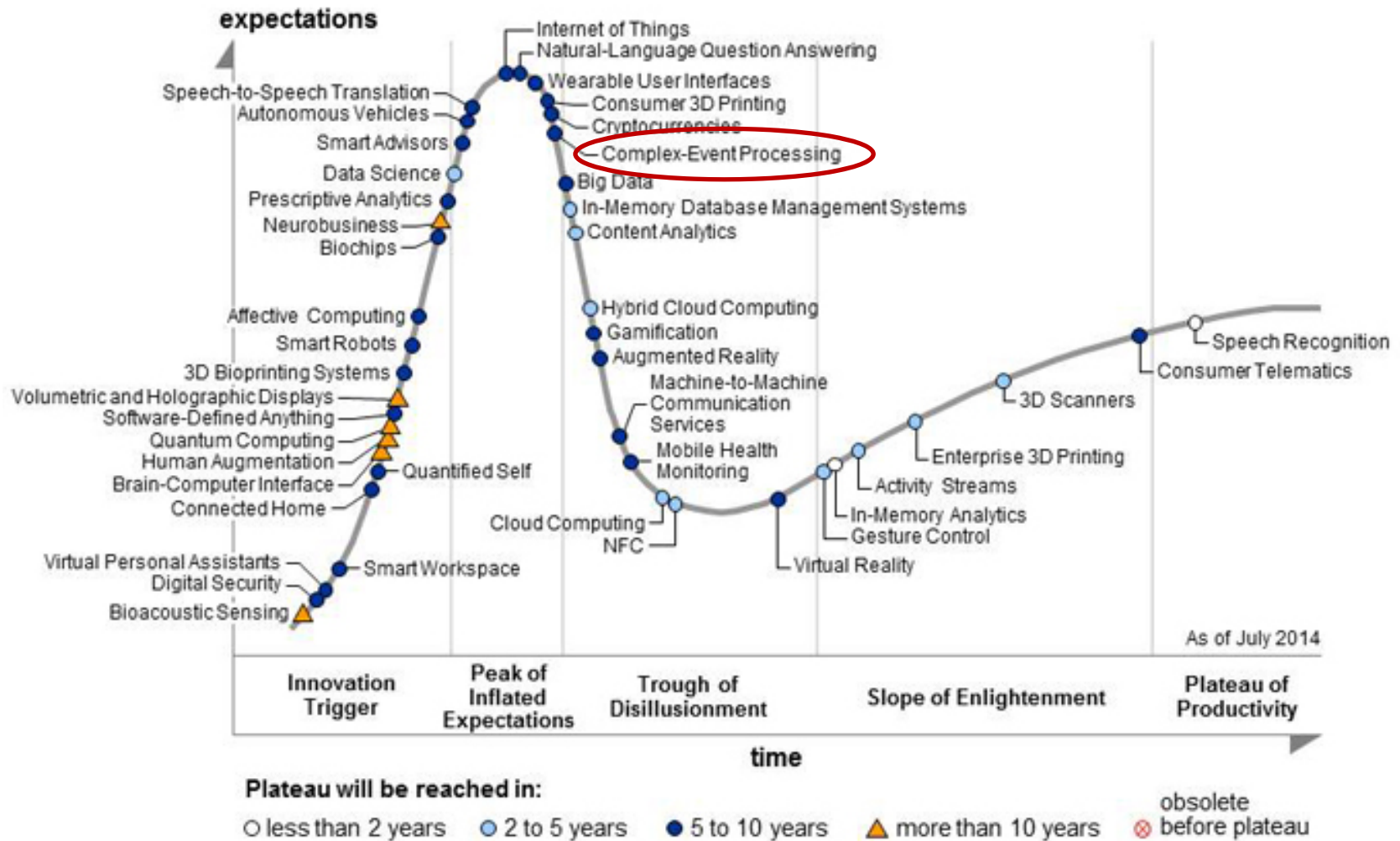
Reaction to fake tweet

The Dow Jones dropped when a fake Associated Press tweet said that there were explosions at the White House.



Source: Bloomberg

Still Asking „Why Bother?“



Three styles of interaction

> Time-driven interaction

- > Initiator: the advent of time within the system
- > Participants: primarily affects the system itself

> Request-driven interaction

- > Initiator: the client issuing a request
- > Participants: client and server

> Event-driven interaction

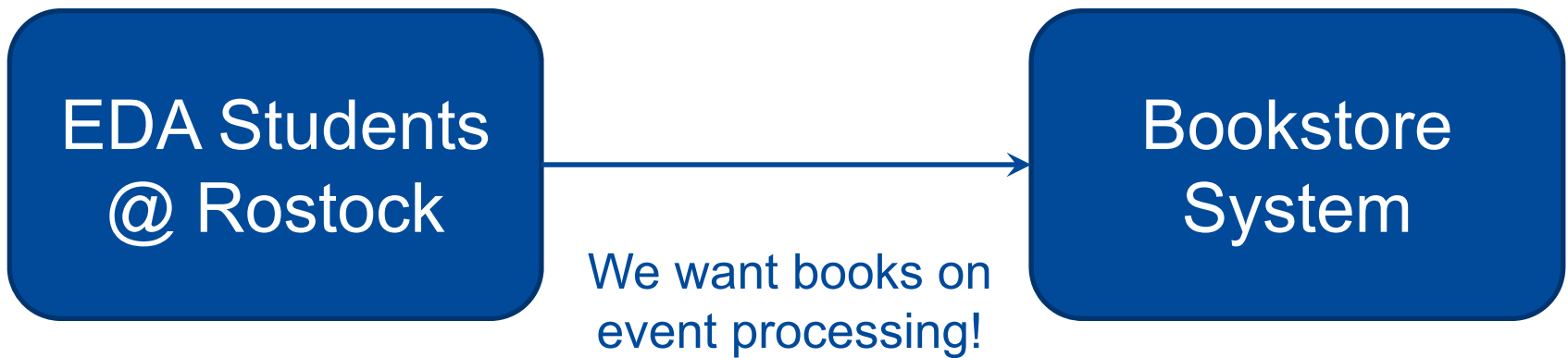
- > Initiator: the (detector of the) event
- > Participants: whoever is interested
(theoretically open-ended)

Time-driven systems

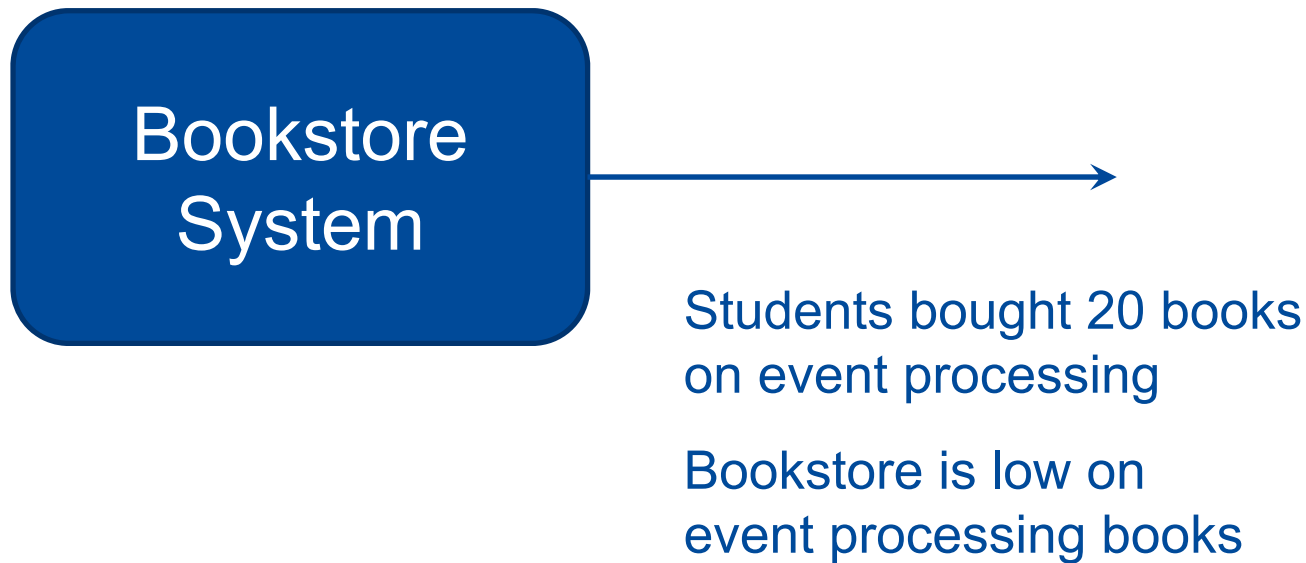
Bookstore System

Determine the ten best
selling books every week
at Sunday 21:00

Request-driven systems



Event-driven systems

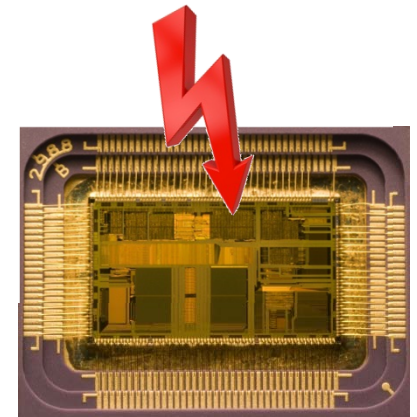


Event-driven systems used in computing?

- > Asynchronous systems and operations
 - > To inform the system about an (external) incident
 - > Notify about success or failure of an operation executed by a different thread or in a different context
- > Examples
 - > Interrupts
 - > Graphical user interfaces (GUIs)
 - > Active databases
 - > Publish/subscribe systems

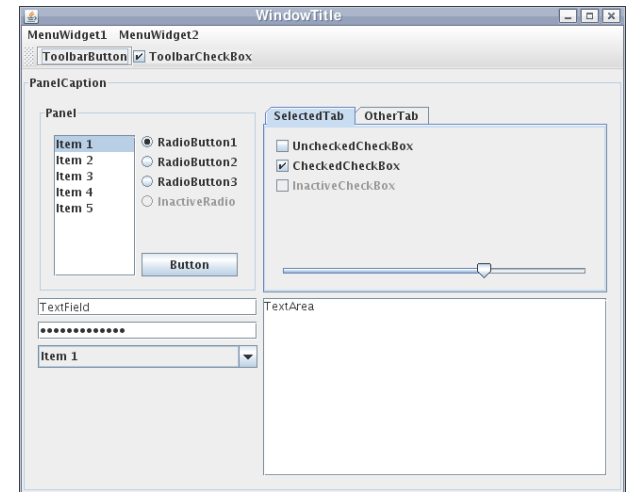
Interrupts

- > Interrupt
 - > Signal to the processor emitted by hardware or software (e.g., IRQ, trap, exception)
 - > Indicates a situation that needs immediate attention (e.g., device in need of service, division by zero)
- > Handling of interrupts
 - > Current thread of execution is halted
 - > State of suspended thread is saved
 - > **Interrupt service routine (ISR)** is executed
 - > Thread state is restored after finishing the ISR
 - > Execution of original thread is resumed



Graphical User Interfaces (GUIs)

- > GUI toolkits / libraries
 - > Build user interfaces from prefabricated visual controls (e.g., input fields, buttons)
 - > Component-oriented development
- > Handling of user input
 - > Input data is provided asynchronously (e.g., key strokes, mouse clicks, finger taps)
 - > Toolkit determines control to which the input is applied
 - > Toolkit calls event handler that was previously registered for control and type of input (i.e., event type)
 - > Event handler either implements application logic or further delegates the call
 - > Inversion of control



Active Databases

- > Active Databases
 - > supplement regular DBMS with reactive behavior
 - > reactive behavior often specified with ECA rules
 - > feature is also known as **database triggers**
- > **Event-Condition-Action (ECA) rules**
 - > **Event** identifies happening of interest
(e.g., insertion/deletion of a tuple, abort of a transaction)
 - > **Condition** evaluates the context of the event's occurrence
 - > predicates over event attributes and database queries
 - > **Action** are executed when both event and conditions match
 - > May include **stored procedures** or even external processes
 - > May trigger new events to be fired and other ECA rules to be processed

Publish/Subscribe System

- > Publish/Subscribe
 - > Group communication paradigm for distributed components
 - > Publish (and send) notifications about occurred events
 - > Subscribe to (and receive) notifications of interest
- > Notification Service
 - > Matches published notifications with active subscriptions
 - matching algorithms
 - > Forwards matching notifications to their subscribers
 - routing algorithms
 - > Often implemented by a set of cooperation brokers
 - > Notifications (and subscriptions) are stepwise forwarded in the broker network
 - > Well suited to disseminate information to many clients

Advantages of the Event-driven Style

- > **Reduces latency** → increases responsiveness and agility
 - > Inform about and process events immediately when they occur
 - > Push on event vs. periodic poll
- > **Example: Client wants to receive stock quotes from a server**
 - > Max. message delay between client and server: 100 ms
 - > Client request size / server reply size: 1 KB / 100 KB
 - > Max tolerable latency for updates: 1s
 - > Stock quotes are updated every 10s on average
- > **What bandwidth is needed when**
 - > a) client polls the readings periodically
 - > b) server pushes the readings when updated

Architecture

“A specific model of a micro-chip or CPU.” (Hardware)

“The art or science of building.” (Merriam Webster)

“Architecture is invention.”
(Oscar Niemeyer)

“Architecture begins where engineering ends.” (Walter Gropius)

“From Ancient Greek ἀρχιτέκτων (arkhitéktōn, ‘architect’) meaning chief builder” (etymology)

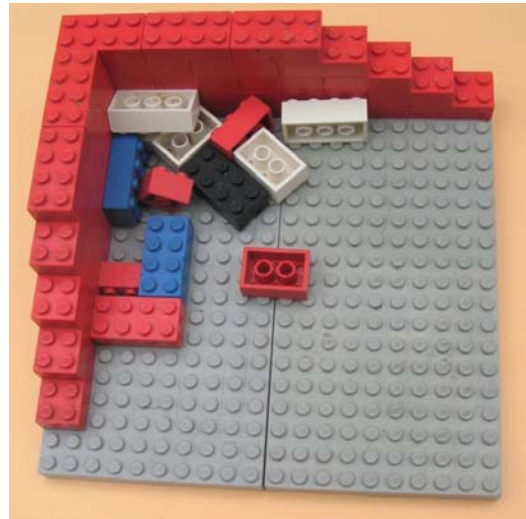
“Architecture starts when you carefully put two bricks together. There it begins.”
(Ludwig Mies van der Rohe)

“A method or style of building.” (Merriam Webster)

(System) Architecture

- > “An architecture is a set of rules defining how elements can be composed to create a new system”
 - > Basic building blocks → elements
 - > Elements can be composed → new system
 - > Systems are elements, too → hierarchical composition
 - > Composition rules define possible compositions

- > Example
 - > Lego™



Event-Driven Architectures (EDA)

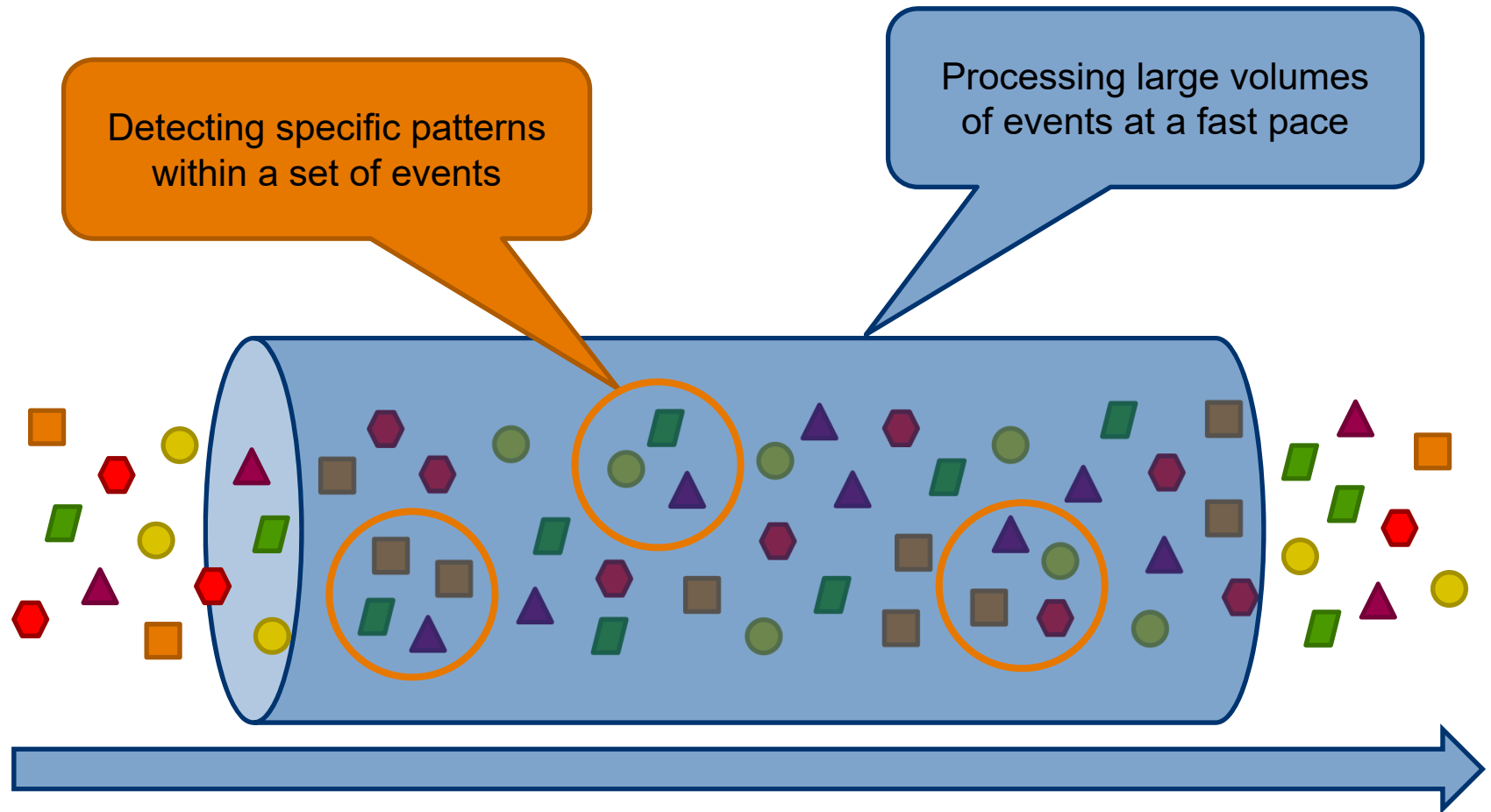
- > Software architecture pattern facilitating
 - > the specification and modelling,
 - > the production and publication,
 - > the subscription and propagation,
 - > the detection and consumption, and
 - > the processing and reactionof / to events

- > Events are the central concept regarding the
 - > specification
 - > design and
 - > interaction styleof the architecture.

Event-Driven Architecture (EDA)

- > In an EDA, the components focus on their own original business functionality → **self-focused components**
- > The system's functionality emerges (mainly) from the **implicit interaction** of its components
- > If a component changes its state, this is detected and information about this event is saved → **event detection**
- > To interpret events, usually additional information about their occurrence is needed → **context information**
- > Other components are informed about events that are of interest to them and react accordingly to them
- > Informing components of events can be done in different ways (e.g., by calling a method of the receiving component)

Complex Event Processing



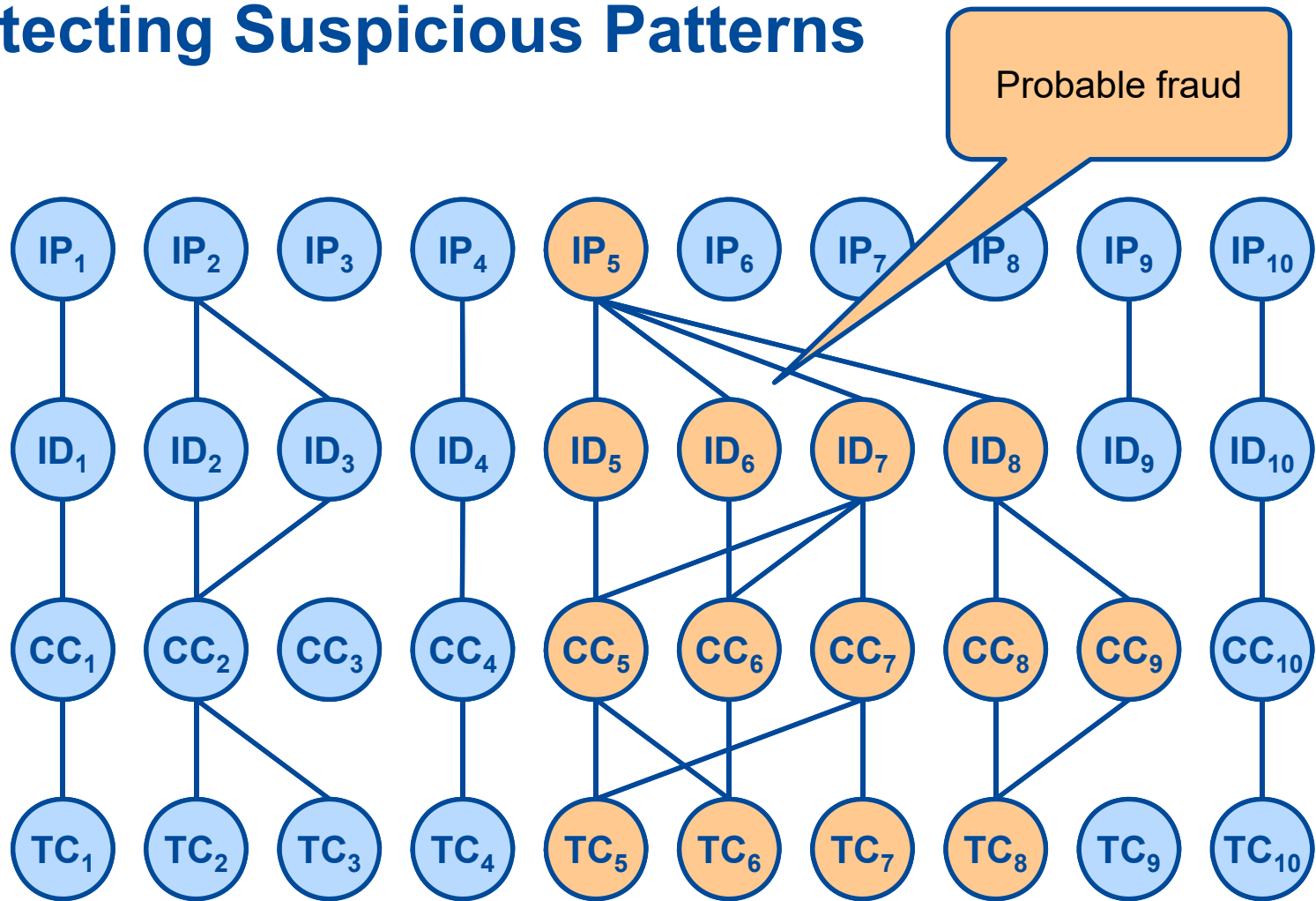
Example: e-Commerce Fraud

- > Scenario
 - > Online payment in e-commerce
- > Payment transactions with
 - > User ID
 - > IP address
 - > Geo location
 - > Tracking cookie
 - > Credit card number
- > Payment gateways/providers process payment transactions for many web shops



How to detect probable fraud attempts?

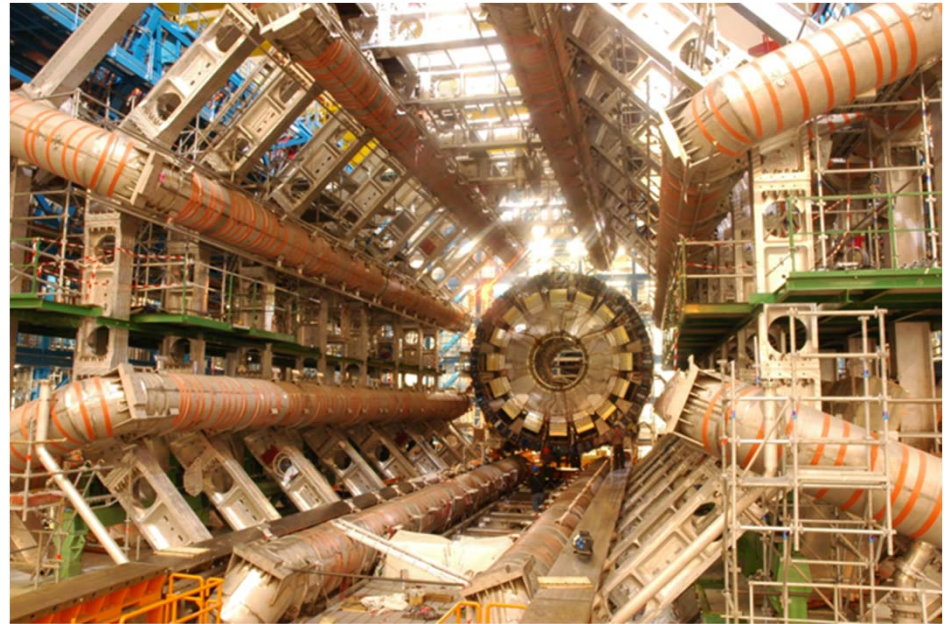
Detecting Suspicious Patterns



IP address (IP), user ID (ID), credit card number (CC), tracking cookie (TC)

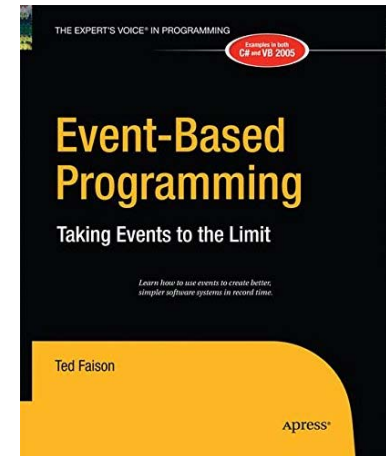
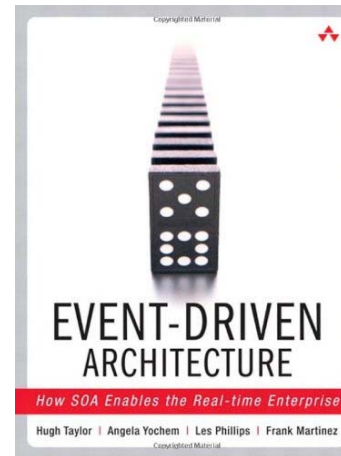
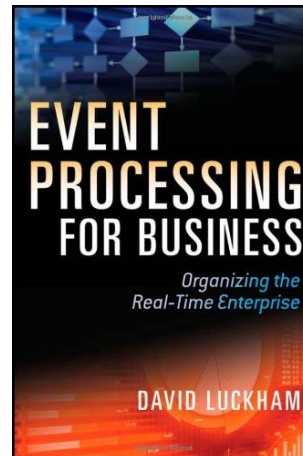
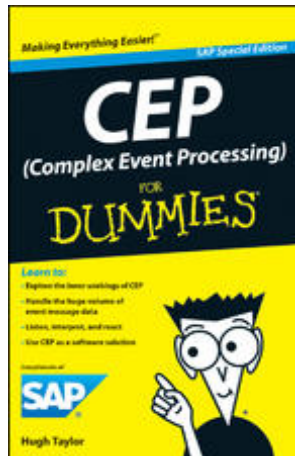
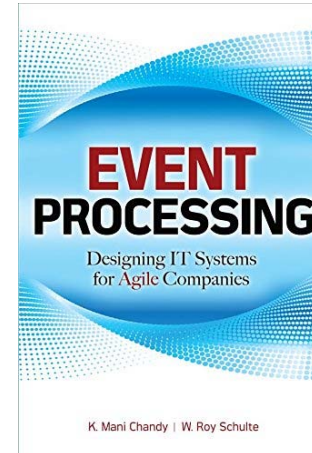
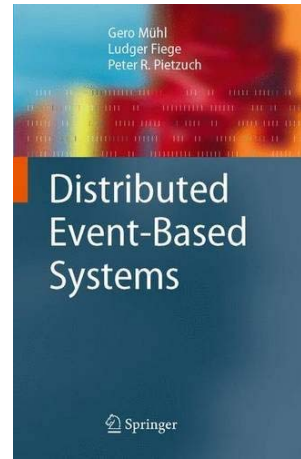
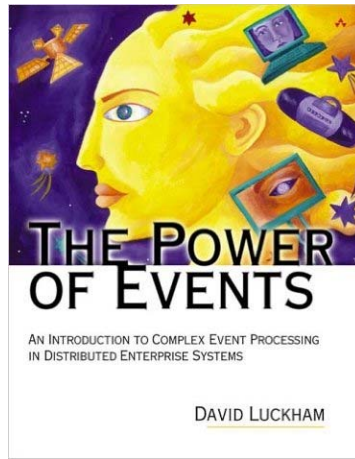
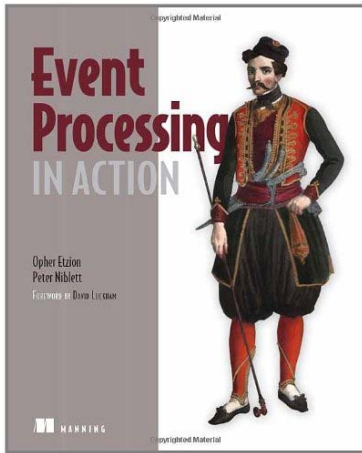
Example: LHC Particle Accelerator

- > Scenario
 - > Particle detection
- > Atlas detector
 - > One of seven particle detectors at the Large Hadron Collider (LHC)
 - > 46m long, diameter of 25m, weight of 7000 t, 3000km cable
- > Generated data
 - > Detecting collision events at 40 MHz each having a resolution of 70 „megapixels“



How to process these large amounts of data?

Literature



Lehrziele des Seminars

> Allgemeine Lehrziele

- > Selbstständige Einarbeitung in das gewählte Themengebiet inklusive Literaturrecherche
- > Aneignen der Terminologie des Themengebiets
- > Anfertigung einer Ausarbeitung unter Anwendung wissenschaftlicher Literatur und Methoden
- > Erarbeiten eines strukturierten und informativen Vortrags
- > Vortragen und zur Diskussion stellen der eigenen Ergebnisse

> Themenspezifische Lehrziele

- > Kennenlernen der Eigenschaften und Anwendungsmöglichkeiten von ereignisgetriebenen Architekturen

Durchführung der Veranstaltung

- > Vergabe der Vortragsthemen per Losverfahren
 - > Los legt die Reihenfolge fest, in der die Studierenden aus den noch nicht vergebenen Themen ihres auswählen können
 - > Dozent führt Losverfahren durch
- > Dozentenvorträge als Video über die Gestaltung von Ausarbeitungen und Vorträgen
- > Abgabe der vorläufigen Versionen der Ausarbeitung und der Vortragsfolien am 20.05., 12:00 Uhr MEZ
 - > Kommentierung erfolgt zeitnah durch Dozenten
- > Abgabe der nachgebesserten Vortragsfolien am 13.06., 12:00 Uhr MEZ
- > Vorträge per Videokonferenz nach dem 13.06.
- > Endgültige Ausarbeitung fällig am 07.08., 12:00 Uhr MEZ

Durchführung der Veranstaltung

- > Alle Abgaben per E-Mail, Dokumente als PDF-Attachment
- > Verlängerung von Terminen nur *vor* Fristablauf und bei vorliegen *wichtiger* Gründe möglich!
- > Selbstständige Einarbeitung in das Thema
 - > Offene Fragen möglichst durch eigene Recherche klären
 - > Bei zwingender Notwendigkeit an den Dozenten mit der Bitte um Hilfestellung wenden
- > Plagieren von Teilen der Ausarbeitung oder des Vortrags führt ohne Vorwarnung direkt zum Nichtbestehen des Moduls

Ausarbeitung

- > Eigenständig verfasster Text, durchgehend in Deutsch oder Englisch, kein Mischmasch
- > Wissenschaftliche Bearbeitung des Themas
 - > Strukturierter Text, korrekte Zitate, [Herausarbeiten der interessanten Aspekte](#), ordentliches Literaturverzeichnis etc.
- > Umfang
 - > 6-7 Seiten im ACM SIG Proceedings Style
 - > Vorlagen für LaTeX und Word
 - <http://www.acm.org/sigs/publications/proceedings-templates>
- > Empfohlen wird die Verwendung von LaTeX, es darf aber auch ein anderes Textverarbeitungsprogramm genutzt werden
- > Abgabe der Ausarbeitung als ein PDF
- > Endgültige Version soll eventuelle Ergebnisse der Vortragsdiskussion mit beinhalten

Vortrag

- > Selbst erstellte Folien in Deutsch oder Englisch, kein Mischmasch beider Sprachen
- > Sprache des Vortrags *muss* mit der auf den Folien verwendeten Sprache übereinstimmen
- > Stilrichtlinien bei der Vortragserstellung beachten
- > Abgabe der Folien als PDF

- > Dauer des Vortrags etwa 20-25 Minuten plus Diskussion
- > Vortrag vorher üben und Zeit einhalten
- > Beantworten der in der Diskussion gestellten Fragen wichtig

- > Präsentation per Videokonferenz

Modulprüfung

- > Voraussetzungen für das Bestehen der Modulprüfung sind
 1. Anhören aller Vorträge und Teilnahme an den zugehörigen Diskussionen
 - > bei Krankheit Attest vorlegen
 - > bei anderen wichtigen Gründen *vorher* absprechen
 2. Termingerechte Abgabe der Ausarbeitung *und* des Vortrags
 3. Halten des Vortrags *am vereinbarten Termin*
 4. Termingerechte Nachbesserung der Ausarbeitung soweit als Konsequenz aus der Diskussion nach dem Vortrag erforderlich

- > Modulnote ergibt sich je zu 50% aus der Seminarausarbeitung und dem Seminarvortrag

Seminarthemen

1. Einführung EDA und Pub/Sub
2. Content-Based Routing
3. Matching Algorithmen für Pub/Sub
4. Request/Reply vs. Event-based
5. Pub/Sub-Standards: DDS
6. Pub/Sub-Standards: JMS
7. Pub/Sub-Standards: MQTT
8. P2P-Based Pub/Sub am Beispiel von Hermes

Seminarthemen

9. Einführung CEP

10. CEP mit Esper

11. CEP mit Apache Flink

12. Event Algebren am Beispiel von Snoop

13. Active Databases

14. Streaming Databases am Beispiel von Stream

15. SQL Match Recognize

16. Event-Driven Programming

Spezielle Literatur

1. K. M. Chandy, *Event-Driven Applications: Costs, Benefits and Design Approaches*, California Institute of Technology, 2006
2. G. Hohpe, *Programmieren ohne Stack: ereignis-getriebene Architekturen*, OBJEKTspektrum, 02/2006, 18-24
3. G. Mühl, L. Fiege, and A. P. Buchmann. *Evaluation of cooperation models for electronic business*. In Information Systems for E-Commerce, Conference of German Society for Computer Science / EMISA, 81-94, 2000.
4. R. Bruns, J. Dunkel, *Event-Driven Architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse*, Springer Verlag, 2010.
5. N. W. Paton, O. Díaz, *Active database systems*. ACM Computing Surveys 31(1), 63-103, 1999.

Spezielle Literatur

6. S. Chakravarthy, V. Krishnaprasad, E. Anwar, S.-K. Kim, *Composite Events for Active Databases: Semantics, Contexts and Detection*, Proceedings of the 20th International Conference on Very Large Data Bases, 606-617, 1994.
7. T. Faison, *Event-Based Programming: Taking Events to the Limit*, Apress, 1st edition, 2006
8. D. Luckham, *The Power of Events*, Addison-Wesley Professional, 2002
9. K. Chandy, W. Schulte, *Event Processing: Designing IT Systems for Agile Companies*, Mcgraw-Hill Professional, 2009
10. H. Taylor, A. Yochem, L. Phillips, F. Martinez, *Event-Driven Architecture: How SOA Enables the Real-Time Enterprise*, Addison-Wesley Professional, 2009

Spezielle Literatur

11. O. Etzion, P. Niblett, *Event Processing in Action*, Manning Pubn, 2010
12. R. W. Schulte, *A Real-Time Enterprise Is Event-Driven*, Gartner Group, 2003
13. R. W. Schulte, *The Growing Role of Events in Enterprise Applications*, Gartner Group, 2003
14. R. W. Schulte, *The Case for Event-Driven Design*, Gartner Group, 2003
15. R. W. Schulte, *Event-Driven Applications: Definition and Taxonomy*, Gartner Group, 2003

Spezielle Literatur

16. R. W. Schulte, Y. Natis, *Event-Driven Architecture Complements SOA*, Gartner Group, 2003
17. R. W. Schulte, *The Business Value of Event-Driven Processes*, Gartner Group, 2003
18. S. Acharya, M. Franklin, and S. Zdonik. *Balancing push and pull for data broadcast*. In SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data, pages 183–194, 1997. ACM Press.
19. G. Mühl, L. Fiege, P. Pietzuch. 2006. *Distributed Event-based Systems*. Springer.
20. M.-T. Schmidt, B. Hutchison, P. Lambros, R. Phippen, *The Enterprise Service Bus: Making service-oriented architecture real*, IBM System Journal, 44(4), 2005

Spezielle Literatur

21. K. M. Chandy, M. Charpentier, A. Capponi, *Towards a theory of events*.
In: DEBS '07: Proceedings of the 2007 inaugural international conference on Distributed event-based systems, ACM, 180-187, 2007
22. K. M. Chandy, W. R. Schulte, *What is Event Driven Architecture (EDA) and why does it Matter?*
23. D. Luckham, *A Short History of Complex Event Processing*,
Part 1: Beginnings
24. D. Luckham, *A Short History of Complex Event Processing*,
Part 2: the rise of CEP

Vielen Dank für Ihre Aufmerksamkeit!

Univ.-Prof. Dr.-Ing. Gero Mühl

`gero.muehl@uni-rostock.de`

`http://wwwava.informatik.uni-rostock.de`