Event Processing Concepts and Technologies (in IS): History and Current Trends

IARIA ComputationWorld 2013, Valencia, Spain
Keynote

image: http://www.wikimedia.org

Prof. Dr. Arne Koschel

Hochschule Hannover
University of Applied Sciences and Arts
Hannover, Germany

What you will hear here

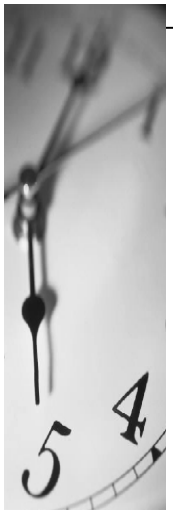
- Get some take on the roots and history of

  current (Complex) Event Processing

  and a little idea about some near future

- Disclaimer: We won't go "deep in details", but still some concepts, architectures, code, and application examples ahead. ;-)

Agenda



1. **Motivation for Event Processing (EP)**
2. EP: (Some) Basic Definitions and Concepts
3. History of Event Processing
4. Present / near Future Event Processing – An Outlook
5. Conclusion and Summary

1. Motivation

"Statements" about Event Processing

- "Companies Must Adopt Modern Event Processing Techniques to improve Operations"

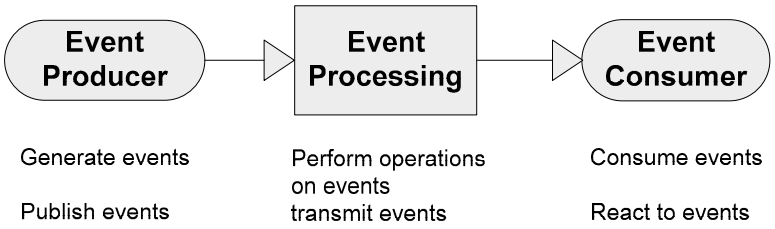
  and

  "If EDA is the first big idea on which event processing is based, then event-driven complex-event processing (CEP) is the second"
  - (Gartner, Oct 2009, http://www.gartner.com/it/page.jsp?id=1212113)
- Events are today (2013) everywhere in almost everyone's (IT) life
  - Email, Facebook, Google (Glass), Stock Markets, Twitter, the Weather ;-), What's Up, . . .

but there are other voices as well

- On the Maturity of Complex event Processing (CEP) (T. Bass, 2008)
  - CEP is mature? CEP is really not ESP? CEP is really event-driven SOA? CEP is really real-time BI? CEP is really low latency, high throughput, white-box COTs algorithmic trading? CEP is really not a type of BPM? CEP is not really for detecting complex events? Complex does not really mean complex? (Bass, 2008, http://www.unix.com/complex-event-processing-rss-news/67495-maturity-cep.html

Event Processing – The Idea



Example: Event Condition Action (ECA) Rules

General Syntax

on <event>
[**if** <condition>] -- optional part
do <action>

Example ECA rule, here with a temporal event

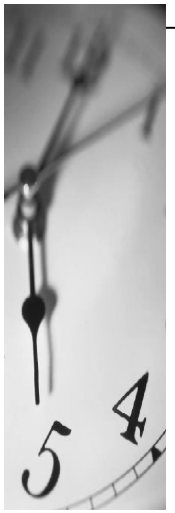
- Start keynote processing at 30.05.2013, 09:15

define rule KeynoteProcessing **is**
on T(30.05.2013,09:15)
do KeynoteSpeach.Start (IARIA_ComputationWorld2013)
end rule;

1. Motivation
Other Examples

- CEP – Automated Stock Trading
- Airport Processes for just 1 flight – “Events/sensors Everywhere”
- Distributed Situation Detection - “Ozone Alarm”

Agenda



1. Motivation for Event Processing (EP)
2. **Event Processing: (Some) Basic Definitions and Concepts**
3. History of Event Processing
4. Present / near Future Event Processing – An Outlook
5. Conclusion and Summary

2. Definitions and Concepts

Event

- Event (IT perspective)
  - *A happening of interest at a certain point of time (at a certain location)*
  - *An object that represents, encodes, or records an event, generally for the purpose of computer processing*
    . . .
  - **Examples**
    - The time/location event "09:21" in "Valencia, Spain"
    - The hardware simulation event: "Component C1 created me at time T1 with data values A and passed me on to components C2 and C3"
    - An email message confirming an airline reservation
    - Stock tick message that reports a stock trade
    - A message that reports a temperature sensor reading
  - **Notes**
    1. Events are processed by computer systems by processing their representations as event objects.
       The same activity may be represented by more than one event object;
       Each event object might record different attributes of the activity.
    2. Overloading: Event objects contain data. The word "event" is overloaded so that it can be used as a synonym for event object. In discussing event processing, the word "event " is used to denote **both** the everyday meaning (anything that happens) and the computer science meaning (an event object or message).

Similar in [3]

2. Definitions and Concepts

Event Type; Event Attributes

- Event Type
  - *A class of event objects.*
  - **Examples**
    - The type of all price quotations;
      The type of all sensor readings for any kind of sensor
  - **Notes**
    - All events must be instances of an event type. An *event* has the structure defined by its type.
    - The structure is represented as a collection of event attributes.
- Event Attributes
  - *A component of the structure of an event.*
  - **Examples**
    - A unique event identifier used to reference the event
    - The "time stamp" and "source" of the event's creation
    - The "price attribute" in an stock price quotations;
    - The "temperature attribute" in a sensor reading

Similar in [3]

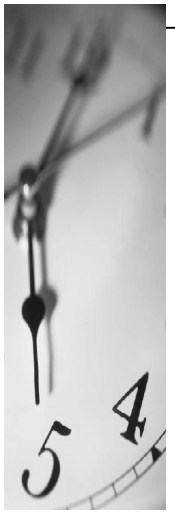
2. Definitions and Concepts

Complex Event; Composite Event

- **Complex event**
  - *An event that is an abstraction of other events called its members.*
- **Examples**
  - The 1929 stock market crash – an abstraction denoting many thousands of member events
  - The 2004 Indonesian Tsunami – an abstraction of many natural events
  - A CPU instruction – an abstraction of register transfer level (RTL) events
  - A completed purchase – an abstraction of the events in a business transaction to purchase something
- **Composite event**
  - *A derived, complex event that is created by combining base events using a specific set of event constructors such as disjunction, conjunction, sequence, etc. A composite event always includes the base (member) events from which it is derived.*
  - **Examples**
    - (E1 AND E2) OR (E3)
    - "door bell rings" FOLLOWED-BY "dog barks" FOLLOWED-BY "post man has been bitten"
  - **Notes**
    - In the Active Database terminology often "complex event" is used instead of composite event.

Similar in [3]

Agenda



1. Motivation for Event Processing (EP)
2. EP: (Some) Basic Definitions and Concepts
3. **History of Event Processing**
4. Event Processing – An Outlook
5. Conclusion and Summary

3. History of EP

Acknowledgement

- In 2007 Prof. David Luckham created an overview about the history of (Complex) Event Processing.
  - [1] A Short History of Complex Event Processing, Part 1: D. Luckham, 2007, http://www.complexevents.com
  - [2] A Short History of Complex Event Processing, Part 2: D. Luckham, 2007, http://www.complexevents.com
- A slightly extended/modified form will guide us from here.

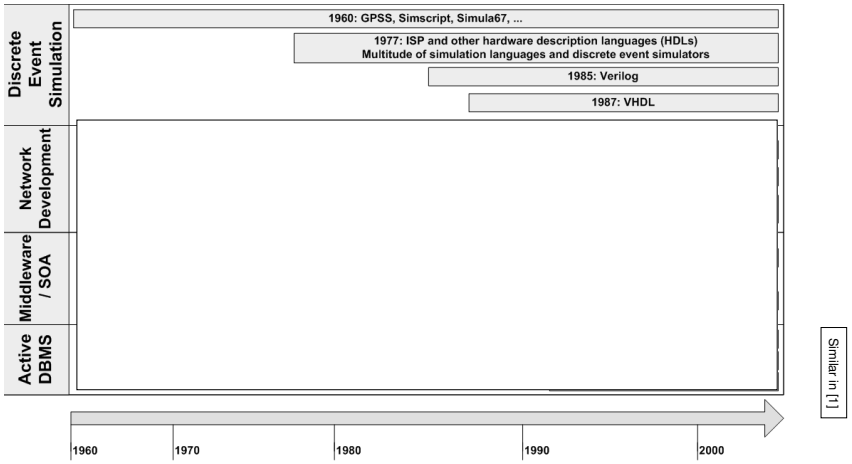
3. History of EP

Historical ingredients of Event Processing

- From [1] we see 4 major historical EP ingredients, which influenced the “EP soup”
  - Discrete event simulation
  - Computer networks
  - Active Database Management Systems (ADBMS)
  - Middleware.

3. History of EP

Discrete Event Simulation – starting in “the ancient past”



3. History of EP - Discrete Event Simulation

Discrete Event Simulation - Overview

- „Event Processing started with discrete event simulation in the 1950’s.
- The basic idea was that the behavior of a system – be it a hardware design, control system, avionics, factory production line or natural phenomenon like weather – could be modeled by a computer program written in a “simulation language”. Given input data, the program would create events that mimicked the interactions between components of the system. Each event happened at a time recorded by a clock. Of course, some events could happen at the same time. But the clock would eventually increase its reading by discrete “ticks” representing the progress of real time. Such models were called discrete event simulations. “ (from [1])

3. History of EP - Discrete Event Simulation

Discrete Event Simulation – Overview cont'd

- „The events had the form of messages like "component C1 created me at time T1 with data values A and B and sent me to components C2 and C3".

  The simulator had to schedule the flow of the events between components in the model, the execution of the components, and the ticking of the clock. " (from [1])

- Some more widely known simulation languages were Simula, VHL, Verilog

3. History of EP - Discrete Event Simulation

Example - Simula

- OO simulation programming language from Norway Computing Centre (Kristen Nygaard, from the 1960s)

- Example code description:

  "Sam, Sally, and Andy are shopping for clothes. They share one fitting room. Each one of them is browsing the store for about 12 minutes and then uses the fitting room exclusively for about three minutes, each following a normal distribution."

```
Simulation Begin
   Class FittingRoom; Begin
      Ref (Head) door;
      Boolean inUse;
      Procedure request; Begin
         If inUse Then Begin
            Wait (door);
            door.First.Out;
         End;
         inUse:= True;
      End;
      Procedure leave; Begin
         inUse:= False;
         Activate door.First;
      End;
      door:- New Head;
   End;

   Procedure report (message); Text message; Begin
      OutFix (Time, 2, 0); OutText (": " & message); OutImage;
   End;

   Process Class Person (pname); Text pname; Begin
      While True Do Begin
         Hold (Normal (12, 4, u));
         report  (pname & " is requesting the fitting room");
         fittingroom1.request;
         report (pname & " has entered the fitting room");
         Hold (Normal (3, 1, u));
         fittingroom1.leave;
         report (pname & " has left the fitting room");
      End;
   End;

   Integer u;
   Ref (FittingRoom) fittingRoom1;

   fittingRoom1:- New FittingRoom;
   Activate New Person ("Sam");
   Activate New Person ("Sally");
   Activate New Person ("Andy");
   Hold (100);
End;
```

W Source: http://en.wikipedia.org/wiki/Simula#Simulation

3. History of EP - Discrete Event Simulation

Example – Verilog / SystemVerilog

- Hardware simulation language
  - originating from research (Moorby, Goel).
  - From C syntax inspired.
  - Initially 1983/84.

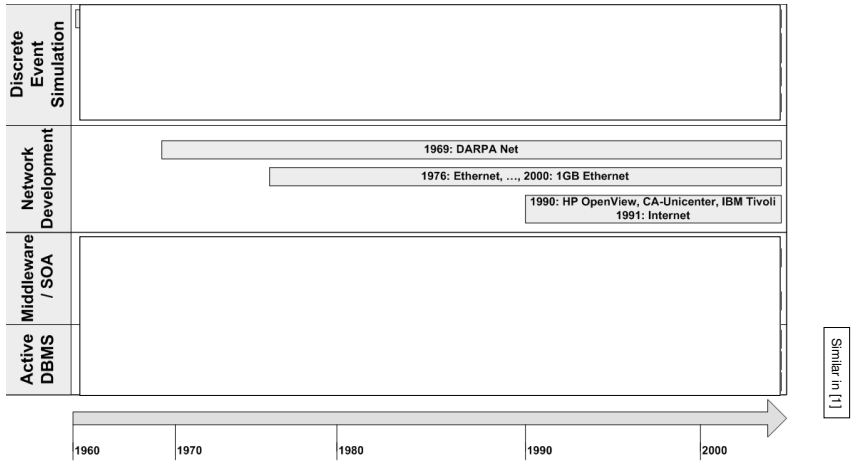
- Code sample: Flip Flop

```
module toplevel(clock,reset);
  input clock;
  input reset;

  reg flop1;
  reg flop2;

  always @ (posedge reset or posedge clock)
    if (reset)
      begin
        flop1 <= 0;
        flop2 <= 1;
      end
    else
      begin
        flop1 <= flop2;
        flop2 <= flop1;
      end
endmodule
```

W Source: http://en.wikipedia.org/wiki/Verilog#Example

3. History of EP

Network Development – continuing in "the Middle Ages"



3. History of EP - Networks

It started with the ARPA net

- „Another kind of event processing was involved in the development of computer networks, starting in the late 1960's with the ARPA net.
- The focus was on making reliable communication between computers across networks by means of events containing sequences of binary data – so called packets. Transmitting or receiving a packet was an event. The basic work involved developing protocols for communicating sequences of packets reliably when the network itself might be unreliable and subject to errors. "

3. History of EP - Networks

Networks and Event Models

- TCP/IP
- ISO 7-layer Communication Model
- Both use „event layers" and „event hierarchies"

3. History of EP - Networks

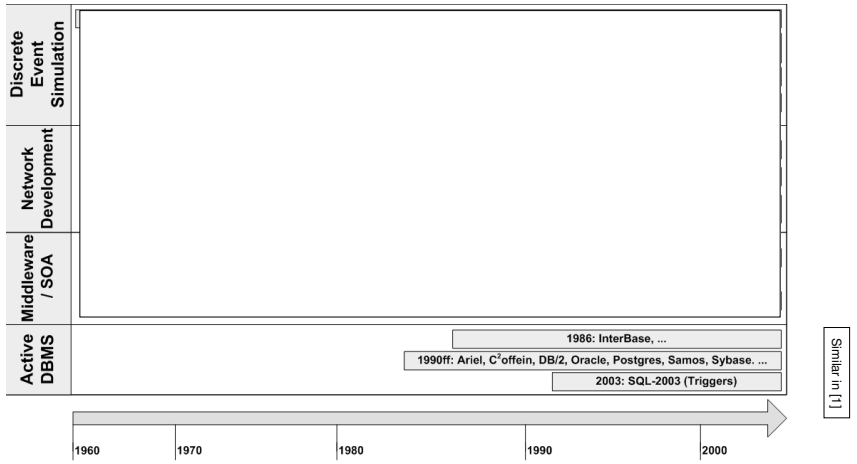
Network Monitoring

- Network Monitoring tools
  - Track and trace events in the network such as
    - Node up / down
    - Node's response time
    - Resource usage at nodes (Disk / CPU / Memory, ….)
- Typical tool examples
  - Nagios, CA-Unicenter, HP Openview, IBM Tivoli, …
- The „early adopters" kind of today's Business Activity Monitoring (BAM)

3. History of EP

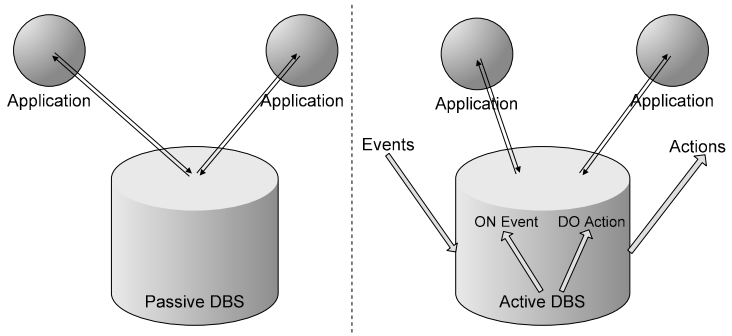
Active Database Management Systems –

"just" 20-10 years ago, major EP language influencers



3. History of EP – Active DBMS

What is an Active DBMS (vs. Passive DBMS) ?



A DBMS ist *active,* if it reacts to (external or internal) *Events* by means of (external or internal) *Actions*

3. History of EP – Active DBMS

Active DBMS style Event Condition Action (ECA) Rules

General Syntax

on <event>
[**if** <condition>]
do <action>

Example ECA rules; here ECA rules with temporal events

- Start production at 23.01.2013, 09:25:
  define rule Production **is**
  on T(23.01.2013,09:25)
  do ProductionSystem.Start (ProductionOrder)
  end rule;

- If a machine is not ready again from 5 minutes after breakdown, start repair process
  define rule CheckMachines (Machine m) **is**
  on T(m.SetState (OutOfOrder) + 00:05)
  if m.state = OutOfOrder
  do Repair (m)
  end rule;

3. History of EP – Active DBMS

Known example: Active (R)DBMS EP with "Triggers"

```
AFTER UPDATE
    OF parts_on_hand
    ON inventory
```
← **E**vent

```
WHEN (
  :new.parts_on_hand < :new.reorder_point
)
```
← **C**ondition

Action ↓

```
FOR EACH ROW

DECLARE
    x NUMBER;

BEGIN
    SELECT COUNT(*) INTO x
        FROM pending_orders
        WHERE part_no = :new.part_no;

    . . . More statements to order new parts. . .
END;
```


3. History of EP – Active DBMS

Example: ECA Rule Execution Model (from Active DBMS Manifesto)

roles of E, C, and A part	**roles** ⊆ {*optional*, mandatory, none}
coupling modes	**coupling mode** ⊆ {immediate coupled, deferred coupled, *immediate decoupled, deferred decoupled*}
signalling point	**point of time** ⊆ {*pre, post, instead*}
signalling granularity	**granularity** ⊆ {*instance oriented, set oriented*}
net-effect	**net-effect** ⊆ {*true, false*}
conflict resolution	**strategy** ⊆ {*parallel, arbitrary, priority*, static, dynamic}
event consumption policy	**policy** ⊆ {*recent, chronicle, continuous, cumulative*}
scope of event consumption	**scope** ⊆ {*local, set local,* global, set global}
event lifespan	**lifespan** ⊆ {*explicit, implicit*}

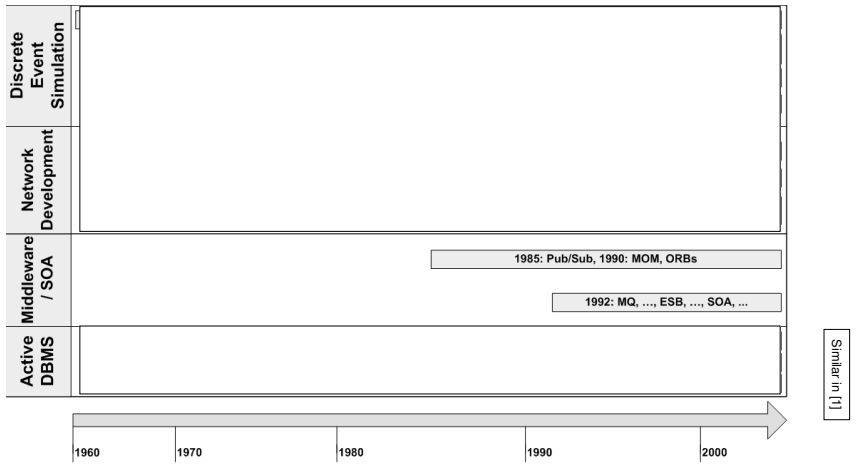
3. History of EP – Active DBMS

Technology Examples

- Several research Active DBMS with ECA rules or triggers …
  - Ariel, Samos, Sentinel, Postgres, …
- … and several commercial ones
  (typically in commercial RDBMS: SQL triggers
  but also event proc. in O-RDBMS and OO-DBMS)
  - DB/2, Ingres, Oracle, SQL Server, Sybase, Versant, …

3. History of EP

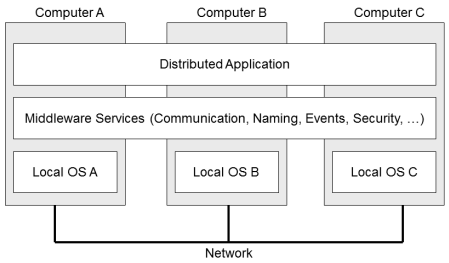
Middleware / SOA – ”also just” 20 years ago, well “alive and kicking”



3. History of EP – Middleware/SOA

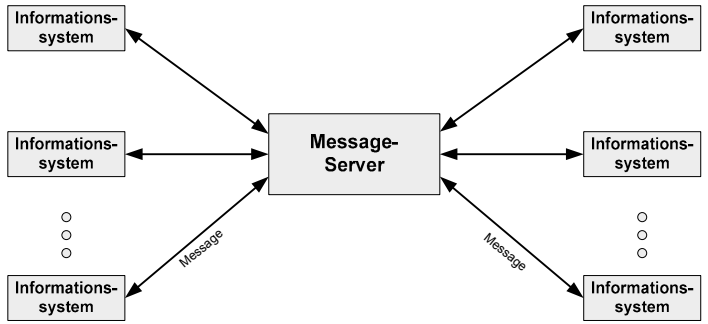
Middleware

- Middleware
  - Infrastructure software to enable communication between software components (across the network)
  - Abstracts from network, OS, (may be) programming language etc.
- Placement of middleware



3. History of EP – Middleware/SOA

Example: Message Oriented Middleware (MoM)



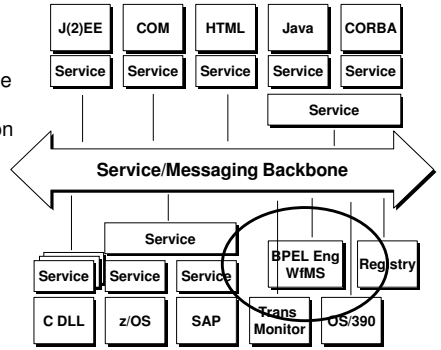
Asynchronous (event) messages within the server
(if necessary – in several MoM implementations – with transaction control)

Product/technology examples: JMS, Joram, WS MQ, CORBA Notification, …

3. History of EP – Middleware/SOA

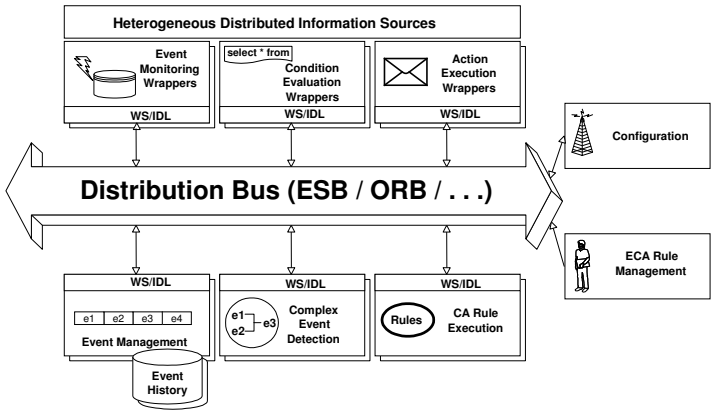
Middleware Concept: SOA

- Service / Messaging Backbone
  - Often: Logical / Technical "Enterprise Service Bus (ESB)"
- Services with formally described interface
- Integration of components often based on productive systems
- Different communication paradigms (Synchronous / asynchronous)
- Implementation possible with different technologies / platforms, e.g.
  - CORBA, RMI, Web Services; Java EE, .Net, . . .
  - Now: Often Web Services used combined with e.g. Java EE
- Event / task flow Processing, e.g. using
  - BPEL / WfMS Engines



3. Event Processing History - Example Middleware-based EP Architecture

Combining ADBMS+ESB/ORB: Distributed ECA Rule Processing (C²offein)



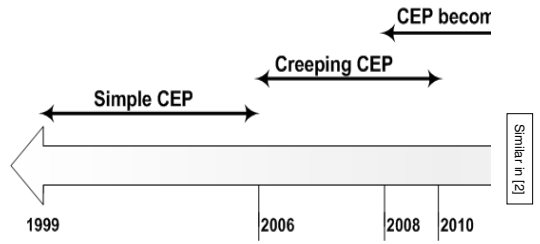
Source [6]

"Present History" EP – Complex Event Processing (CEP)

3. Present EP

Present Event Processing (IS focus)

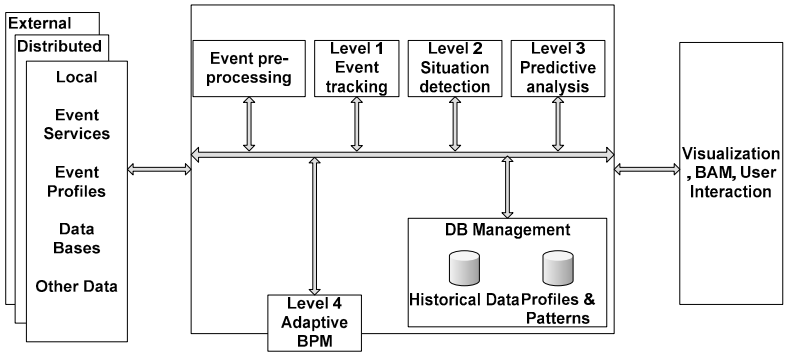
- Complex Event Processing (CEP) & Event Stream Processing
- Event Driven Architectures (EDA)
- Busines Process Management (BPM / Workflows) including Busines Activity Monitoring (BAM)



Similar in [2]

3. Present CEP - Example Architecture

Sample: CEP: Functional Reference Architecture



Adopted from [5]

3. Present CEP - Sample Event Processing Languages

Sample: Declarative Event Stream Queries "SQL style"

- Data stream is an unbounded sequence of time-stamped tuples.
- Time-stamps are only time points.
- Idea: use SQL like syntax do query such streams including queries to other data as well
- Sample in Continuous Query Language (CQL):
  "Order Tracking", "Large Orders ", "Order Payments"

```
SELECT Istream O.customer, S.trackingId
FROM O[Range 2 Hours], S[Range 2 Hours]
WHERE O.id = S.orderId
```

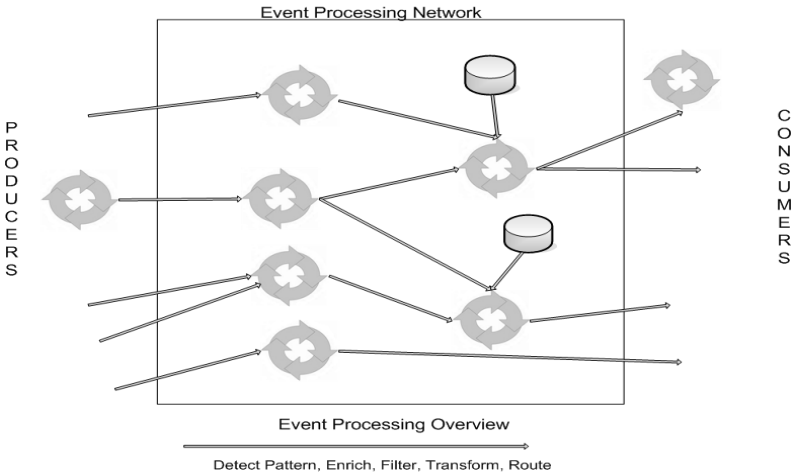
```
SELECT Istream( count(id) )
FROM O[Range 24 Hours]
WHERE O.total > 1000
```

```
SELECT Istream( payment, count(id) )
FROM O[Range 24 Hours]
GROUP BY O.payment
```

Sample source: Diss., Eckert, 2008

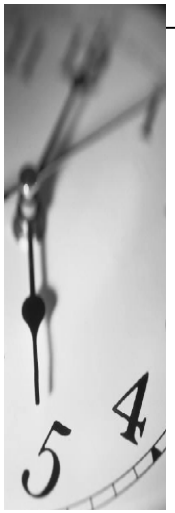
3. Present CEP - Sample Event Processing Agents

Typical (C)EP Concept: "Processing Agents" – Big Picture



Adopted from [4]

Agenda

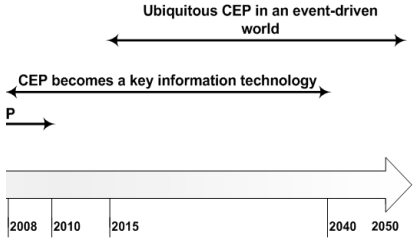


1. Motivation for Event Processing (EP)
2. EP: (Some) Basic Definitions and Concepts
3. History of Event Processing
4. **Event Processing**
   – Present and near Future
5. Conclusion and Summary

4. Present / near future EP

Present and (Near) Future Event Processing

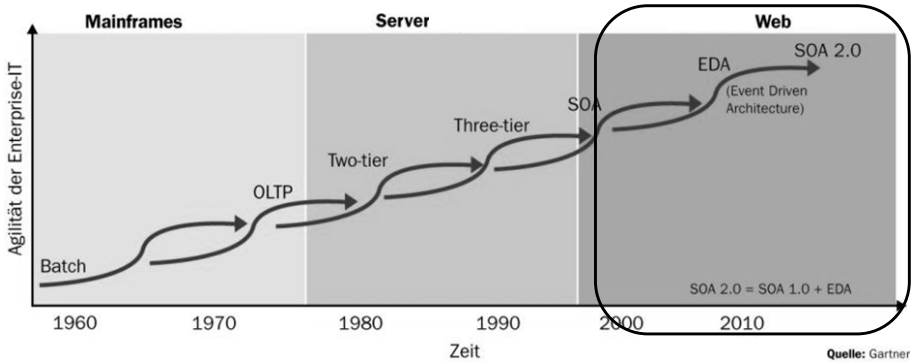
- Event Driven SOA (ED-SOA)
- Distributed Complex Event Processing (CEP)
- A myriad of events everywhere
- Event Processing across global clouds ?



Similar in [2]

4. Present / near future EP

ED-SOA / SOA 2.0 – EDA & SOA combined



4. Present / near future EP - Event Processing Example Architecture

Present Sample: ED-SOA

- SOA to integrate synchronous, strongly typed services
- Event messaging for mass data processing for data with simpler structure

4. Present / near future EP

About the near Future?

- As we know

"Prediction is difficult, especially about the future"

Yogi Berra, baseball catcher

4. Present / near future EP

Some EP future topics from literature

- Some future topics from [12] and EPTS (www.ep-ts.com)
  - From narrow to wide (application areas)
  - From monolithic to diversified (specialized EP)
  - From proprietary to standards based (event structure, event languages, …)
  - Event processing in virtual platforms / "the cloud"
  - Pattern-based intelligent EP
  - . . .

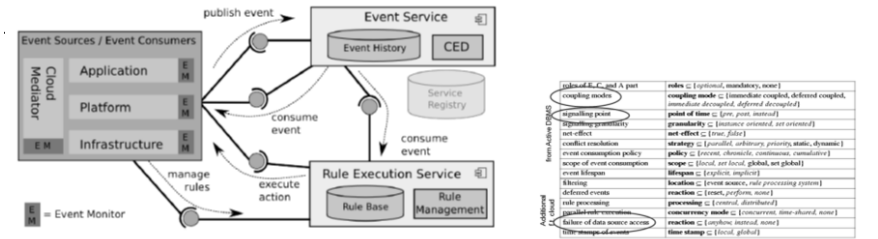
4. Present / near future EP

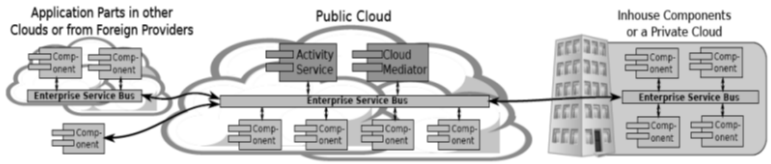
Events from the Cloud

- Events from everywhere
  - Sensors (RFID, temperature, location, …)
  - Google GLASS
  - Web / internet events
  - IT Applications
  - News
  - . . .

4. Future EP

Event Processing (OM4SPACE) in the hybrid cloud





4. Future EP

Globally Federated EP across the Cloud "rules the World"

Some References

Want to know more?

1. A Short History of Complex Event Processing, Part 1: D. Luckham, 2007, http://www.complexevents.com
2. A Short History of Complex Event Processing, Part 2: D. Luckham, 2007, http://www.complexevents.com
3. Event Processing Glossary – Version 1.1: D. Luckham, R. Schulte (Edt.), 8/2008, http://www.complexevents.com
4. Event Processing – a semantic approach: O. Etzion, IBM, DEBS tutorial, 2007
5. Mythbusters: Event Stream Processing v. Complex Event Processing: T. Bass, ACM DEBS 2007, Toronto, June 20, 2007
6. Ereignisgetriebene CORBA-Dienste für heterogene, verteilte Informationssysteme: A. Koschel, Dissertation, Universität Karlsruhe, July 1999. http://www.ubka.uni-karlsruhe.de/cgi-bin/psview?document=1999/informatik/5
7. http://www.complexevents.com: D. Luckhams Web-Site dedicated to Event Processing
8. DEBS: Distributed Event-Based Systems, ACM conf. series, www.debs.org
9. The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems: D. Luckham, Addison-Wesley, 2002
10. Event Processing: Designing IT Systems for Agile Companies: K. Chandy, W. Schulte, McGraw-Hill, (Gartner), 2009.
11. Active Rules for Databases: N. W. Paton (Edt.), Springer, 1999.
12. Event-Processing in Action; Opher Etzion, Peter Niblett: Manning, 2011.

5. Conclusion

Conclusion

- Event Processing has a long tradition with roots in
  - Discrete event simulation
  - Computer networks
  - Active Database Management Systems (ADBMS)
  - Middleware.
- Despite the tradition EP is *very alive* and will become and is already a key factor in IT applications around the globe

That‘s it for now! – Questions ?

Arne Koschel
University of Applied Sciences and Arts,
Hannover, Germany
akoschel /at/ acm.org