1. Motivation

“Statements” about Event Processing

- “Companies Must Adopt Modern Event Processing Techniques to improve Operations”
- “If EDA is the first big idea on which event processing is based, then event-driven complex-event processing (CEP) is the second”
- Events are today (2014) everywhere in almost everyone’s (IT) life
  - Email, Facebook, Google (Glass), Stock Markets, Twitter, the Weather ;-
- “Prediction is difficult, especially about the future”
  - Yogi Berra, baseball catcher

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

Event Processing – The Idea

Event Producer ➔ Event Processing ➔ Event Consumer

- Generate events
- Perform operations
- Consume events
- Publish events
- Transmit events
- React to events

Example: Event Condition Action (ECA) Rules

General Syntax

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

Example ECA rule, here reaction on a temporal event

- Start talk at 4. Feb. 2014, 14:00
- Define rule TalkProcessing is
- on T(04.02.2014,14:00)
do Talk.Start (OOP_2014)
end rule:

Example: CEP – Automated Stock Trading

- Trading Rule
  - IF
  - MSFT price moves outside 2% of MSFT:15-min-vol-open
  - AND
  - VIX price moves up by 5%
  - OR
  - VIX price move down by 2%
  - THEN
  - CALL, SELL

- Heterogeneous, distributed sources
- Full source spanning
- ECA rules required

Example: 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 "14:11" in "Munich, Germany"
  - An email message confirming an airline reservation
  - A message that reports a stock trade
  - A message that reports a temperature sensor reading

- **Notes**
  - 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.

- **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).

Event Type; Event Attributes

- **Event Type**
  - A class of event objects.
  - **Example**: The type of all price quotations;

- **Event Attributes**
  - A component of the class (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 "temperature attribute" in a sensor reading

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 regular 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.
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.

Acknowledgement

- In 2007 Prof. David Luckham created an overview about the history of (Complex) Event Processing.

- An extended / modified form will guide us from here.

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])

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, Verilog, and VHL

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.”

Example – Verilog / SystemVerilog

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

- Code sample: Flip Flop
3. History of EP - Networks

Network Development
– continuing in “the Middle Ages”

- 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

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)

- If a machine is not ready again from 5 minutes after breakdown, start repair process
  - define rule CheckMachines (Machine m)
  - on T(m,SatStatus (OutOfOrder) + 0:05)
  - if m.state = OutOfOrder
  - do Repair (m)

---

3. History of EP – Active DBMS

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

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

---

3. History of EP – Active DBMS

Well known example: Active (R)DBMS EP with “Triggers”

**General Syntax**

- AFTER UPDATE
  - on parts_on_hand
  - OF parts_on_hand < new.reorder_point
  - WHEN (new.parts_on_hand)

**Example**

- FOR EACH ROW
  - DECLARE x NUMBER;
  - BEGIN
  - SELECT COUNT(*) INTO x
  - FROM pending_orders WHERE part_no = new.part_no;
  - x = x + 1;
  - Endif x > new.reorder_point
  - do More statements to order new parts...
3. History of EP

**Middleware / SOA** — "also just" 20 years ago, well "alive and kicking"

Middleware

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

Placement of middleware

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, etc.

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 / WMS Engines
3. Event Processing History - Example Architecture "in action"

Example: "Automated Metadata Update"
Active DBMS and Middleware combined in C2offein (vgl. [6])

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

3. Present EP
Present Event Processing (focus: information systems)

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

3. Present CEP - Example Architecture
Example: CEP – Functional Reference Architecture
3. Present CEP - Sample Event Processing Languages

Example: 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 to query such streams including queries to other data as well
- Sample in Continuous Query Language (CQL): “Order Tracking”, “Large Orders”, “Order Payments”

```
SELECT t.stream_name, t.customer, t.trackingId
FROM t[Range 3 hours]
WHERE t.id = t.id
```

3. Present CEP - Sample Event Processing Languages

Example: Imperative Event Processing Languages (EPLs)

- Imperative EPL using APAMA
  Excerpt from a “Stock Order System”

```
package com.apama.labs.positions;

// Position service
public class PositionService {
    private Position position;
    private final String position;

    public PositionService(PositionService service) {
        this.position = service.getPosition();
    }

    public void setPosition(Position position) {
        this.position = position;
    }

    public Position getPosition() {
        return position;
    }
}
```

3. Present CEP - Sample Event Processing Agents

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

- Filtering
  - Stateless Agent that filters events. It is the most common agent, and a filtering agent has been required by many applications
- Routing
  - Itinerary-based routing e.g. mail delivery
  - Subscription-based routing e.g. news feed Intelligent (criteria based) routing
- Pattern Detection
  - Time series pattern detector
  - “Set at a time” pattern detector
  - Basic pattern detector
  - Fixed set of most common patterns

Business Activity Monitoring (BAM)

Business Activity Monitoring (BAM)

- "BAM is the real-time reporting, analysis and alerting of significant business events, accomplishing by gathering data, key performance indicators and business events from multiple applications." [Gartner 2007]

---


CC_ITM's (HsH) BAM – Architecture (2010ff)

---

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?

---

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

ED-SOA / SOA 2.0 – EDA & SOA combined

From literature

Some EP future topics

  - 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

Events from the Cloud

- Sensors (RFID, temperature, location, ...)
- Google GLASS
- Web / internet events
- IT Applications
- News
  - ...
4. Future EP

Event Processing (HSH:OMSPACE) in the hybrid cloud

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

Some References

Want to know more?


That’s it for now! – Questions?

Arne Koschel
Hochschule Hannover
akoschel /at/ acm.org