Ringvorlesung Industrie 4.0

Event-driven Architecture und Complex Event Processing: Basistechnologien für Industrie 4.0

Prof. Dr. Christian Janiesch
Juniorprofessor für Information Management
Josef-Stangl-Platz 2
97070 Würzburg
http://www.bwl.uni-wuerzburg.de/lehrstuehle/bwljp1/
Christian Janiesch

UNI WÜ

Karlruhe Institute of Technology

Westfälische Wilhelms-Universität Münster

Julius-Maximilians-Universität Würzburg

SAP

EDA und CEP – Prof. Dr. Christian Janiesch
IoT in Sports

- **Data collection (configurable)**
  - 300+ sensors throughout the boat
  - server in the hull
  - 3,000 variables at ten times a second
  - several video feeds and numerous still images of the sail wing every second
  - ~1 gigabyte of raw data and ~200 gigabytes of video per day
- **Real-time analytics**
  - chase boat is the analytical hub
  - analysis may require combination of 10, 20, or 30 variables fitted through a time-based algorithms
  - predictions on what will happen in the next few seconds, or minutes, or even hours in terms of weather analysis
- **Performance sailing technology**
  - each crew member wore a ruggedised PDA on his wrist and received a real-time, customized feed
  - wireless network a bottleneck
A Logistics Example

- Traffic jam on route
- Ocean current speeds up journey
- Container was routed the wrong way
- Freight forwarding documents delayed
- Technical defect
- RFID tag read in harbor
- Container was damaged
- RFID tag was not read
- Delivery truck routed to an additional factory
- Low stock alert
- Customer cancels order
- Sales in region x increased by 20%
- Customer changes address on self service portal
- New express order received

Bruns, Dunkel (2010)
IoT Supply Chain

http://www.dhl.com/
Typical BI Stack

EDA und CEP – Prof. Dr. Christian Janiesch

Cf. Chaudhuri et al. (2011)
Reporting vs. Analytics

What just happened?

Business Activity Monitoring

What happened?

Where exactly?

What actions are needed?

Why is this happening?

What actions are needed?

What's the best that can happen?

Predictive modeling
Forecasting/extrapolation
Statistical analysis
Alerts
Query/drill down
Ad-hoc reports
Standard reports

Degree of intelligence

Competitive advantage

Analytics

Access and reporting

Report on CEP – Prof. Dr. Christian Janiesch

Davenport, Harris (2007)
"Business activity monitoring" (BAM) is Gartner's term defining how we can provide real-time access to critical business performance indicators to improve the speed and effectiveness of business operations.

Unlike traditional real-time monitoring, BAM draws its information from:
- multiple application systems and
- other internal and external (inter-enterprise) sources,
  enabling a broader and richer view of business activities.

BAM is the next big thing that application integration vendors want and need.
- Understand **influence of internal as well as external events** on business
- Data on **different layers of abstraction** including technical and non-technical events
- **Real-time decision** support
- Immediate **notifications on exceptions** of business policies
- **Statistics** for relevant metrics
Information Latency

Business Value

Value lost through latency

- Business-relevant event occurs
- Event data stored
- Analysis information delivered
- Action initiated
- Action completed

Time

- Data latency
- Analysis latency
- Decision latency
- Implementation latency

Infrastructure latency

Reaction time

zur Mühlen, Shapiro (2010)
When was your last dental check-up?

Time-, Request-, Event-driven Interactions

Time for a check-up!

Time
Request
Event

Chandy, Schulte (2009)
Time- & Request-driven vs. Event-driven

Records
- Added
- Removed

Database
- Query
- Result

Reports

Events
- Queued

Event-driven Engine
- Notified

Visualizations

EDA und CEP – Prof. Dr. Christian Janiesch

SAP AG (2008)
Without Events We Are At Risk!
An Event…

- **Form**
  - …is an **object**
  - …has **attributes** and **data components**

- **Significance**
  - …**signifies** an **activity**

- **Relativity**
  - …is **related** to other activities by **time**, **causality**, and **aggregation**
  - …have the **same** relationships to one another as the activities they signify

Luckham (2002)
Relativity of Events

- **Time**
  - Order of events
  - Multiple Timestamps
  - Multiple Clocks

- **Cause**
  - In-/dependence of events

- **Aggregation**
  - An event may signify *multiple* other events
  - Abstraction → complex event

Luckham (2002)
Kinds of Events

- **Normal** event type
  - Delivery of a package on time
  - This event type is already handled by current IT apps

- **Anticipated “abnormal”** event type
  - Penalty likely because of delayed shipment
  - We don’t expect shipments to be delayed, but we are prepared for that eventuality.

- **Unanticipated** event type
  - Your network was attacked with a new type of attack
Typical BI Stack

- **Data sources**
  - External Data Sources
  - Operational Databases

- **Data movement, streaming engines**
  - Extract Transform Load (ETL)
  - Complex Event Processing Engine

- **Data warehouse servers**
  - Relational DBMS
  - MapReduce engine

- **Mid-tier servers**
  - OLAP Server
  - Enterprise search engine
  - Data mining, text analytic engines
  - Reporting Server

- **Front-end applications**
  - Search
  - Spreadsheet
  - Dashboard
  - Ad hoc query

Chaudhuri et al. (2011)
Event-driven Architecture (EDA)

- Event Processing...
  - ...is **computing** that **performs operations** on events. Common event processing operations include reading, creating, transforming, and deleting events.

- Event-driven architecture...
  - ...is a recent addition to the list of architecture paradigms for system landscapes emphasizing the **orchestration of applications and processes** through **events** which can originate anywhere from **external** sensors to **internal** or external business IT systems.

- Often based on Complex Event Processing (CEP)
EDA Applications

- **Predictive processing**: Mitigate or eliminate predicted events
- **Observation**: Make quick observations into exceptional business behavior and notify appropriate people
- **Active diagnostics**: Diagnose problems based on symptoms and resolve them
- **Information dissemination**: Get correct information in the right granularity to the right person at the right time
- **Dynamic operational behavior**: React to events as part of business transactions, achieving low latency decisions and quick reaction to threats and opportunities

Etzion, Niblett (2010)
EDA Application Characteristics

- **Sense and Respond**
  - *Timely* response when reality deviates from expectation

- **Asynchrony**
  - Timing of events are *not controlled* by the enterprise, communication is usually unidirectional

- **Global situational awareness**
  - Awareness by correlating *multiple sources* of data from outside the enterprise with enterprise data

- **Errors**
  - *External data* is more *noisy*

Chandy (2006)
Event-driven Architecture Challenges

- Technical Complexity
  - Processes are a network of tasks in disparate systems
  - Processes span across functional and company boundaries

- Technological Complexity
  - Organically evolved and thus heterogeneous IT architectures
  - Hardware, OS, programming languages, formats, interfaces… differ

- Communication with Physical World
  - Sensor to event data
  - Standardized back channel

- Amount of Data
  - Ever growing amount of smart items
  - Capturing data is easy, deciding was is important not so much

Luckham (2002)
“Complex Event Processing (CEP) is a defined set of tools and techniques for analyzing and controlling the complex series of interrelated events that drive modern distributed information systems.”

- Roles
  - Producer (source), processor, consumer (sink)

- Events Sources
  - Technical, system, business

- Event Processing
  - Sense, process/analyze, respond

Luckham (2002)
From Singular Events to Complex Events

abstract

complex events

event stream

concrete

Bruns, Dunkel (2010)
Real-time Event-driven Architecture

Event Producer
- ERP
- Sensors
- Virtualized Infrastructure

Event Channel

Event Processor
- Complex Event Processing
- Contextual Data

Event Channel

Event Consumers
- Dashboards & Notifications
- ERP
- Virtualized Infrastructure
- **(Any) Execution Environment**
  - ...produces events

- **Logging Component**
  - Track **state** of single process instances
  - Store audit data for **legal** purposes
  - **Trade-off**: Logging granularity vs. performance

- **Event Publisher**
  - (Pre-)**Filter** events
  - **Format** events
  - **Publish** events
There is, at present, no single standard for the way that event instances or event types are represented!

Reasons include the following:
EventID="1df94d91-ffe2-1e50-374e-000c2958989c"
Timestamp="2011-06-08T17:03:30.991+2:00"
ServerID="CE1_00_server0"
ProcessDefinitionID="e469530c095b1908f4711dfa0100022fa206747"
ProcessInstanceID="10cc902094d911df95f5000c29589898c"
ProcessName="Order">

<bpa:EventDetails CurrentState="Closed.Completed.Success"
PreviousState="Open.Running.InProgress" />

<bpa:DataElements>
  <bpa:DataElement name="eventContext" value="process" />
  <bpa:DataElement name="administrators" value="Administrator" />
  <bpa:DataElement name="initiator" value="Administrator" />
  <bpa:DataElement name="priority" value="3" />
  <bpa:DataElement name="startTime"
    value="2011-06-08T17:03:05.623" />
  <bpa:DataElement name="completionTime"
    value="2011-06-08T17:03:31.010+2:00" />
</bpa:DataElements>
</bpaf: Event>
- **Input Adapter**
  - **transforms** events into an internal format and puts events into input event stream

- **Event Processing Network (EPN)**
  - is composed of **Event Processing Agents (EPA)**
  - EPA **monitor** events streams to **detect** and **act** on events
  - EPA **filter**, **match**, and **derive** (translate, aggregate, split,…)

- **Output Adapter**
  - translates events into **metrics, messages** or **function calls**
Event Processing Agent Operations

Event Processing Agent

Filter
Transformation
Pattern detect

Translate
Aggregate
Split
Compose

Enrich
Project

Etzion, Niblett (2010)
Pattern Detect EPA Examples

Etzion, Niblett (2010)
Event Context: Temporal Examples

Etzion, Niblett (2011)
Event Context: Spatial

- **Fixed location**
  - Within the house

- **Entity distance location**
  - Within 2 km from the motel

- **Event distance location**
  - Within 10 km from the accident

Etzion, Niblett (2011)
Pattern Detect EPA Examples

Etzion, Niblett (2010)
- `select * from EVENTTYPE`

- Every time an event of the type `EVENTTYPE` occurs, it is picked up

- select * from EVENTTYPE.win:length(5)

- Every time an event of the type EVENTTYPE occurs, it is picked up; also the last four events are picked up as well

Filters and Where Clauses

- `select * from EVENTTYPE(amount>=200) .win:length(5)`

- Every time an event of the type `EVENTTYPE` occurs, which amount is 200 or more, it is picked up; also the last four events are picked up as well.

select * from EVENTTYPE.\texttt{win:time(4 sec)}

- Every time an event of the type EVENTTYPE occurs, the events of the last four seconds are picked up as well

• select * from ET.\texttt{win:time\_batch}(4 \ sec)

• In a fixed window of four seconds all events of the type \texttt{EVENTTYPE} are picked up and reported

Event Consumer

- **Input Adapter**
  - Receives events and transforms them into internal format

- *(Any)* Execution Environment
  - Visualizes or acts upon an event

- **Typical Consumers**
  - Other **BAM/ CEP** environments (→ complex events)
  - Real-time monitoring **dashboards**
  - **Messaging** infrastructure (SMS, e-mail, IM, …)
  - **Process execution engines** (start, suspend, cancel,…)

EDA und CEP – Prof. Dr. Christian Janiesch
Example Consumer Screenshots
Events in BPM encompasses serve two purposes:

- **designing, simulating, monitoring**, and **optimizing** systems in a deliberate and systematic way that is conscious of end-to-end business processes.
- Use of BPM software, such as **orchestration** engines and workflow engines, at run time to direct the sequence of execution of software components and human activity steps in a process.

Both concepts of BPM involve events, but they don't always involve event objects or the discipline of event processing.
Timing Data from Process Log Events

- Turnaround Time
- Idle Time
- Change-over Time
- Processing Time
- Suspend Time

Activity instance becomes ready for execution, work item is placed on user’s work list

User selects Work Item
User starts Work Item
User suspends activity
User resumes activity
User completes activity instance
Event-driven BPM Roles

- BPM as a producer of
  - state changes
  - lack of state changes
  - payload

- BPM as a consumer for
  - reactions to a situation
    - Instantiate/ reserve/ release resources
    - Start new process
    - Change process (data at gate)
    - Suspend process
    - Abort process
    - Skip process
    - Terminate process
Implementation Proof of Concept

```xml
<EventPublisher>
  <AggregateStream>
    <FilterStream id="IncorrectQuantityFilter" store="memory" istream="Claims">
      <FilterExpression>like(Claims.claim_type, 'incorrect_quantity')</FilterExpression>
    </FilterStream>
    <FilterStream id="IncorrectProductFilter" store="memory" istream="Claims">
      <FilterExpression>like(Claims.claim_type, 'incorrect_product')</FilterExpression>
    </FilterStream>
    <AggregateStream id="IncorrectProductQuantityCount" store="memory" istream="IncorrectProductQuantityUnion">
      <ColumnExpression key="false" name="Counter">cast(double, count(IncorrectProductQuantityUnion.order_id))</ColumnExpression>
      <ColumnExpression key="true" name="NameOut">'Ident'</ColumnExpression>
    </AggregateStream>
    <FlexStream id="OrderByNullDo1" store="memory" istream="OrderByEntryAccuracyJoin">
      <Column datatype="string" key="true" name="NameOut"/>
      <Column datatype="double" key="false" name="valueOut"/>
      <Method name="OrderByEntryAccuracyJoin" stream="OrderByEntryAccuracyJoin">
        [ string nameOut, double valueOut ] rec;
        double entry;
        entry := OrderEntryAccuracyJoin.valueOut;
        if (!isNull( entry )) entry := 1;
        rec := setOpcode([ nameOut = 'OrderByEntryAccuracy', valueOut = entry ], upsert);
        output rec;
      </Method>
    </FlexStream>
    <AggregateStream id="ReleasedByAgentCount" store="memory" istream="ReleasedByAgent">
      <ColumnExpression key="false" name="Counter">cast(double, count(ReleasedByAgent.ProcessInstanceId ))</ColumnExpression>
      <ColumnExpression key="true" name="NameOut">'Ident'</ColumnExpression>
    </AggregateStream>
  </AggregateStream>
</EventPublisher>
```
Scenario: Heavy Machinery Maintenance
Modeling CEP and BPM Interaction

Input Adapter 1

OrderEvent

OrderEvent 2

Split

CustomerDB

Get Customer data:

- customer.limit
- customer.creditrating

CustomerEvent

CustomerDB

EnrCustomerEvent

EnrPaidOrder Event

OrderIndicator Event

Private IT Resource

CustomerEvent

OrderEvent 2

Sum

OrderSumEvent

Aggregate

Translate

AllOrderLimit = AllOrderValue - customer.limit

Translated

Output Adapter 2

Output Adapter 1

NotCreditWorthyEvent

Janiesch, Diebold (2016)
Latency Revisited: Prediction

Schwegmann et al. (2013)
Latency Revisited: Autonomy

- What if the agent...
  - 1. offered **no** assistance
  - 2. offered a complete set of **alternatives**
  - 3. narrowed a **selection** to a few alternatives
  - 4. suggested **one** alternative
  - 5. asked for **approval**
  - 6. allowed a time for the human to **veto** the machine decision
  - 7. executed automatically, and then **informed** the human
  - 8. informed the human only **if asked**
  - 9. informed the human only **if it wants to**
  - 10. **decided everything**

Parasuraman et al. (2000)
Q and A

Ventana Research (2012)
Juniorprofessur für Information Management

Josef-Stangl-Platz 2
97070 Würzburg

http://www.bwl.uni-wuerzburg.de/lehrstuehle/bwljp1/