# Event-Driven Business Process Management and its Practical Application Taking the Example of DHL

Rainer von Ammon[1] , Christoph Emmersberger[2], Florian Springer[2,] Christian Wolff[3]

[1] CITT GmbH, Konrad-Adenauerallee 30, D-93051 Regensburg, Germany
[2] Senacor Technologies AG, Vordere Cramergasse 11, D-90478 Nürnberg, Germany
[3] University of Regensburg, Media Computing, Universitätsstrasse 31, D-93040 Regensburg

Rainer.Ammon@citt-online.com, Christoph.Emmersberger@senacor.com, Florian.Springer@senacor.com,
christian.wolff@sprachlit.uni-regensburg.de

**Abstract.** The recently coined term «Event-Driven Business Process Management» (EDBPM) is a combination of actually two different disciplines: Business Process Management (BPM) and Complex Event Processing (CEP). The common understanding behind BPM is that each company's unique way of doing business is captured in its business processes. For this reason, business processes are today seen as the most valuable corporate asset. In the context of this article, BPM means a software platform which provides companies the ability to model, manage, and optimize these processes for significant gain. As an independent system, Complex Event Processing (CEP) is a parallel running platform that analyses and processes events. The BPM- and the CEP-platform correspond via events which are produced by the BPM-workflow engine and by the IT services which are associated with the business process steps. In this paper we present a general framework for EDBPM as well as first use cases in the context of logistics and financial services.

**Keywords:** Complex Event Processing, CEP, Business Process Management, Event-driven BPM, Event Pattern, Software as a Service, SaaS, SOA, EDA

## 1. Event-Driven Business Process Management

Business process modeling became popular in the context of enterprise reorganization and modernization in the early 1990's. A business process is a structured set of activities designed to produce a specified measurable result for a particular customer or market. The common understanding behind Business Process Management (BPM) is that each company’s unique way of doing business is captured in its business processes. In the last years, BPM has become a common tool in larger IT projects and workflow engines able to execute business processes became quite popular. Business processes are today seen as the most valuable corporate asset.

The term “Event-Driven Business Process Management” is a combination of actually two different disciplines: Business Process Management (BPM) and Complex Event Processing (CEP). In this context BPM means a software platform which provides companies the ability to model, manage, and optimize these processes for significant gain. As an independent system, Complex Event Processing (CEP) is a parallel running platform that analyses and processes events. The BPM- and the CEP-platform correspond via events which are produced by the BPM-workflow engine and by the IT services which are associated with the business process steps. Also events coming from different event sources in different forms can trigger a business process or influence the execution of the process or a service, which can result in another event. Even more, the correlation of these events in a particular context can be treated as a complex, business level event, relevant for the execution of other business processes or services.

In this article we describe the development of event-driven BPM (section 2) as a combination of Complex Event Processing and Business Process Management. We present the challenges in realizing event-driven BPM and describe a conceptual reference model for it (section 3). Section 4 discusses a practical use case of EDBPM in the context of the large logistics company DHL and we demonstrate the feasibility of event-driven BPM showing that it is a powerful tool with a clear business value which has its role in actively or even proactively ensuring better business process insight and situation-aware dynamic business process execution (section 4). The conclusion outline future steps in current running and imminent projects (section 5).

## 2. Definitions and Historical Background

BPM platforms address the operative and tactical level of supporting business including all activities that ensure efficient business process execution such as change management, incident management, problem/disruption management, and all activities which address business process control, delivery and planning such as as business activity and quality management, continuity management. This apparently amounts for situation awareness by reactive measures and detection of critical events, e.g. serious incidents which lead to process disruptions.

Complex Event Processing (CEP, [4]) is a dicipline that deals with event-driven behavior. It is an emerging technology for obtaining relevant situational knowledge from distributed systems in realtime or almost realtime by selection, aggregation, and event abstraction for generating higher level complex events of interest.

We define Event-Driven Business Process Management as event-driven combination of BPM and CEP. The solutions are executed by a BPM platform in combination with a separated CEP integration system, where both are decoupled with each other via interchanging events.

The term “Event Driven Business Process Management” was first used in June 2003 in a white paper of Bruce Silver Associates in connection with the FileNet P8-BPM platform [3]. The term was understood as a synthesis of workflow and Enterprise Application Integration (EAI). Actually, a concept of event processing and real-time BAM was described in that white paper, but only as single event processing without knowing anything about CEP which had not been introduced at the time. Although the book “The power of events” of David Luckham [4] has already been published in 2002, it wasn’t well known until 2004 and there was no working CEP platform yet. In November 2007, the term was explicitly used as a combination of BPM and CEP in an interview with Ruma Sanyal from BEA Systems [5].

## 3. Reference Model for Event-Driven BPM

In the following, we roughly outline the basic components needed for operational EDBPM systems. As shown in Fig. 1, basic elements can be taken from BPM platforms as well as CEP applications. EDBPM comprises two modeling layers for businesses processes as well as for events (to be described below). Basically this connects the pre-operational abstraction of process modeling with the runtime observations of business process-related events occurring at execution time.

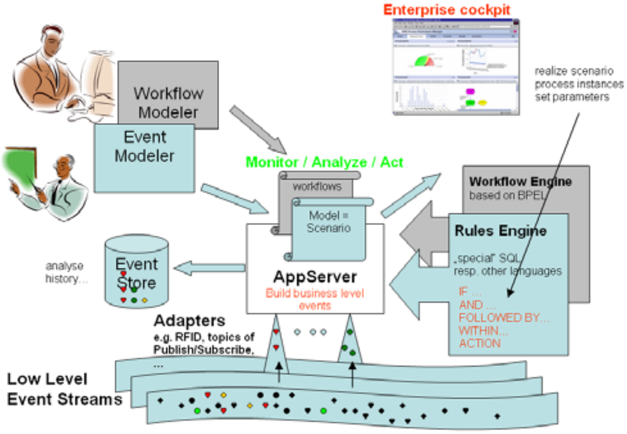

**Figure 1 A reference model of the adoption of a CEP/BAM/BPM platform**

Figure 1 shows the principle of how a BPM- and BAM/CEP platform work together on the basis of events. Grey components show workflows with respect to BPM, while blue components deal with real-time BAM/CEP phenomena. With this architecture in mind and given the recent availability of commercial CEP platforms, there will be two different kinds of EDBPM specialists in the future: Workflow modelers or business analysts and event modelers. Their task profile is roughly outlined in the following sections

### 3.1. The Business Process Modeler

The workflow modeler identifies business processes, starting from the value chain of an enterprise. He analyses and reengineers or optimizes the business processes by using a Business Process Analyzing- (BPA) tool based on a standard notation like the Business Process Modeling Notation (BPMN), invented 2002 by IBM and now standardized by the Object Management Group (OMG) since 2005, published February 2006 as version 1.0 ([8], for an overview of BPM languages see [9]). A business process model is a graphically designed network based on symbols e.g. for flow objects like activities, gateways and also events. Depending on their position in the network, BPMN categorizes the events as start events, intermediate events and stop events. Each event category has different event types like timer-, message-, exception events etc. In the present release of BPMN 1.1, there are around 25 event types. A new release BPMN (V. 2.0) is announced for Q3/2008. The modeling of business processes is a highly skilled task for a workflow modeler respectively a business analyst, but completely different from the task of an event modeler for real-time BAM/CEP as the perspective is completely different: The event modeler acts with different kinds of events which are produced by the business process instances themselves or by other event sources, like SNMP traps, RFID tags, log file entries etc.

The business process models can be transformed into an executable notation of a workflow engine or a BPM platform respectively. Executable notations are currently being standardized by OASIS as Business Process Execution Language (BPEL, [10]) or as a different standard XPDL by the Workflow Management Coalition (WfMC, [11]). To make a business process model really executable, it has to be remodeled more fine grained directly in the BPM platform. From the point of view of the IT department, e.g. so-called compensations and exceptions have to be modeled, if an IT-service cannot be executed or fails. This task cannot be done by the business analyst who does not know anything about the internals of IT services associated with the process steps. So, in the future as well as with upcoming BPM platforms and standards a new procedure for modeling and implementing business processes is needed. Business analysts from the operating department have to work closely with IT specialists, modeling the process and its associated IT services together, visualized as BPMN, but directly in an integrated modeling tool of the BPM platform. This may be enhanced by a user interface for defining technical information, like porttypes, ports, partnerlinks, peoplelinks etc. which are needed for the process execution by a BPEL-engine. The upcoming BPM platforms are also able to produce events automatically when a service is called or cannot be executed and fails and so on. These are the event types the event modeler has to know for realizing BAM dashboards and enterprise cockpits.

### 3.2. The Event Modeler

By cooperating with the process owners of the operating departments or even with the C-level management of an enterprise, the event modeler has to define which BAM view has to be monitored in a dashboard, which alerts are to send to which roles in the organization and which actions shall be started automatically if a certain event pattern occurs. Derived from such BAM views, the event modeler looks for the needed event types and their instances flowing through the event streams of an enterprise or which are saved in an event store. It is a highly skilled task to define the right event patterns for a real-time BAM view. The event modeler has to know the different event sources like messages types in the Java Message Service (JMS) which are realized on the basis of a publish/subscribe model. He has to install the corresponding event adapters delivered by the CEP platform as “out of the box” pre-built features or the event modeler has to care about the development of not yet existing adapters.

### 3.3. Event Processing Languages

The event modeler defines event patterns for a BAM view on the basis of an Event Processing Language (EPL). At present, there is no standard for an EPL although different commercial solutions for event processing languages exist (see [12] and [16] for an example). The CEP community, founded as a discipline at the first CEP symposium in Hawthorne/New York in March 2006 is discussing the right standard for an EPL very controversially so far. It seems that there will be different EPL-approaches for different domains like algorithmic trading, fraud detection, BAM, etc. At present, the CEP community is gathering use cases and is classifying them according to their corresponding domains [13]. The CEP platforms come with an SQL-like language (e.g. Coral8, Esper, Oracle, StreamBase, cfl. [14]) or provide a rule-based EPL approach (e.g. AMiT from IBM or Reaction RuleML from RuleML, [15]) or some have an abstract user interface which hides any language and generates code like Java (e.g. Tibco, AptSoft). The models of business processes and event scenarios are deployed into a middleware platform, e.g. into an application server, which is responsible for aspects like high availability, scalability, failover, transparency of heterogeneous infrastructures. First EDBPM solutions are just available yet, and we report on a first use case for coupling BPM with CEP technology in the following chapter.

## 4. A USE CASE of the logistics domain

We are currently working on several use cases for the reference model sketched above: Prototype EDBPM implementations are under way in different business environments (logistics, financial services, automotive, see [16], [17] for detailed descriptions of these use cases). The logistics use case of DHL will be described below in more detail as it has reached the highest level of maturity in comparison with our other EDBPM use cases, so far.

Our use case investigates the technical feasibility of combining actual tools for Complex Event Processing (CEP) and Business Activity Monitoring with the latest Enterprise Service Bus (ESB) infrastructure technology which is used to efficiently deploy and manage business and IT services as a distributed service oriented architecture. In particular, we evaluate if it is possible to technically integrate the Oracle components for CEP, BPEL and BAM into the existing SOA Framework of Sopera/Deutsche Post and to test the functionality and quality by means of a typical business process of the Deutsche Post. This use case serves as a proof-of-concept implementation for a solution combining the goals of a SOA with the advantages of CEP. At the same time it is a prototypical implementation of the reference model described in section 3 above.

For evaluating the capabilities and the interplay of the components, a typical business process of DHL, addressing “shipment”, “monitoring/investigation” and “claim” of packages has been modeled and implemented as test bed for the evaluation of the advantages of combining CEP and SOA. The Sopera services implementing the process functionalities are capable of sending process information and service status information as event streams directly to the BAM and CEP components. The CEP engine queries the event streams according to predefined CEP patterns for detecting relevant complex events. These complex events are displayed within the BAM component in dashboard views and are used to create alerts for predictive business.

The following Oracle components have been integrated into the Sopera ESB in a proof-of-concept prototype implementation (for details, see [18]):

- Oracle BPEL (Business Process Management),
- Oracle BAM (Business Activity Monitoring), and
- Oracle CEP (Complex Event Processing).

### 4.1. Process Overview

Based on the business process implemented the following excerpt explains the goals of predictive business analysis for the example of a late delivery of a package. The detection of possible errors is based on an event pattern that deals with events having influence on shipment.

A package is delivered by using several types of transport (e.g. plane, truck, van, etc.) which have to pass several areas / regions on a pre-defined way. In every area miscellaneous events may influence shipment and may eventually inhibit timely delivery of the package, e.g. a traffic congestion hinders the van from delivering the package on time to the customer or a storm prevents a plane from starting. Fig. 2 shows an excerpt of the business process that deals with shipment. The package is allocated to a specific type of transportation which is responsible for shipping the package one part of its way. These steps are processed until the package is delivered to the customer. These events influence on shipment as described.

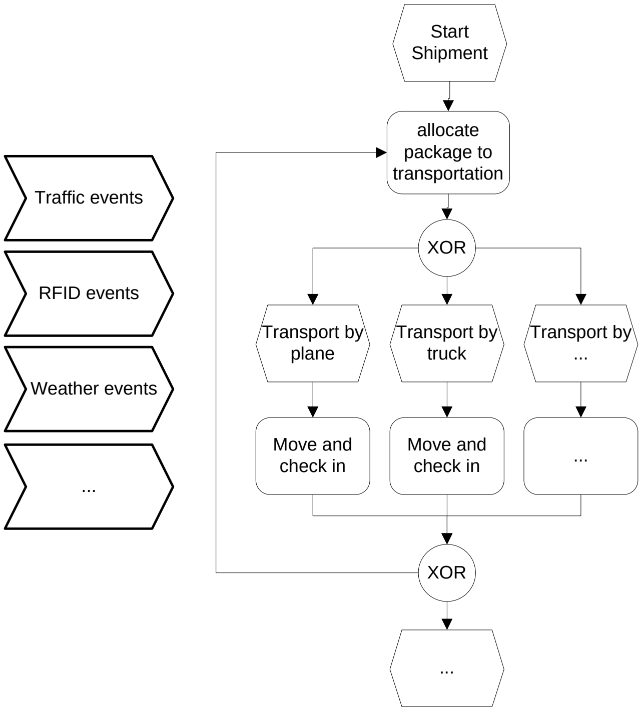

**Figure 2: Excerpt of the Shipment Process**

### 4.2. The External Process Influence Pattern

The following paragraph describes the event correlation in a patterned structured way. The event patterns are schemas based on research on past business process activities. Patterns are found when events occur regularly and always lead to the same failure. After a pattern has occurred, CEP is able to create a complex event which may trigger proactive reactions. One of the patterns that can be found within the shipment process is the “External Process Influence Pattern”. It describes the influences of external events that affect the shipment delivery, e.g. when traffic congestion is encountered within a certain area, and a truck enters the zone, it is most likely that the shipment will not be on time. This event pattern will be described in the following paragraphs.

Overview

The external process influence pattern describes how to detect external influences on a logistics-related delivery process. The processes need to reveal information about their state as well as other events needed to be captured to provide the information about external process influences. External process influences occur continuously during process execution. External events are aggregated with process events so that forecasts are enabled, if the process execution succeeds or if any failures occur. The forecast is based on key performance indicators (KPIs) which have been defined in advance (cf.

[19]). The external process influence pattern allows forecasting process success as well as reacting to process failing predictions in real-time.

Example

A logistics related delivery process is a complex process containing many steps until its completion. For example, a package has to be taken over several routes via several means of transportation till it arrives at the destination. Within these routes, there are many influences which cannot be considered at the process design but have influences on the success of the delivery process. Such influences are traffic congestion, accidents, or bad weather conditions.

Context

Delivery processes are executed and traced by sending out events at certain processing points. Additional information enters the system via several event streams. During process execution there are several reasons why process execution does not complete.

Problem

A business analyst will strive to guarantee the successful execution of a business process. The main questions in case of a failed process are: At which step does the process fail? and What is the reason for the business process failure?

Solution

The CEP platform collects events fired by the business process (process events). It also gathers events coming from external sources (external events). So there are basically two events needed before being able to predict a business process failure.

| Process Event | External Event |
|---|---|
| Key identifying an order | Key identifying an order |
| Affected item identifier | Physical area where the event happens |
| Location of the business process | Degree of failure probability |
| Physical area where the event happens | |

**Table 1 Events and their attributes**

The forecast of a loss is based on the area (areaCode) match of a process event, on the next processing step, with the area of an external event as well as a matching key that identifies the order (timeStamp).

Structure

Two types of events are necessary to aggregate the needed information. In fig. 3 these events are ordered according to their abstraction level:

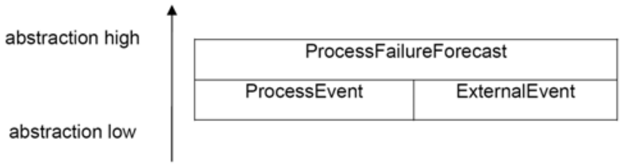

**Figure 3 Event abstraction layer**

On the lowest abstraction level, there are the event types *ProcessEvent* and *ExternalEvent*. The moment taken into consideration is always the moment when the event arrives at the system. This means that information about the location within the process is relative to the time as well as the physical location of an item. Events of the type *ProcessEvent* need to contain the following information:

- *timeStamp*: key identifying the order of the processed events,
- *itemId*: key that identifies a processed item,
- *processLoc*: location identifying the position in the business process and
- *areaCode*: identifier locating the item is physically in the process.
- The type *ExternalEvents* must provide the following fields:
    - *timeStamp*: key identifying the order of the processed events,
    - *areaCode*: identifier locating the area of the external information and
    - *probability*: factor that allows forecasting the process failure.

The type *ProcessFailureForecast* is the event which is generated by the correlation of the *ProcessEvent* and the *ExternalEvent*. It is further processed to arrive in a BAM view or an alerting system. To retrieve the information needed, it is necessary to process the following fields:

- *timeStamp*: key identifying the exact time when the event is correlated,
- *itemId*: key that identifies the processed item,
- *processLoc*: location identifying the position in the business process now,
- *areaCode*: identifying where the item will be physically located next and
- *probability*: factor if the item passes in the next step through the area

*Dynamics*

Each process step generates several *ProcessEvents* on the one side while in the meantime *ExternalEvents* are continuously coming in the system. Once a *ProcessEvent* enters the system, the system matches the process information with the data of the *ExternalEvents* for generating a *ProcessFailureForecast* event.

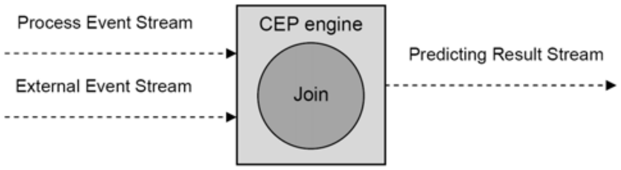

**Figure 4 Dynamics of the external process influence pattern**

The *ProcessFailureForecast* event contains the information about how likely the process will fail in the next step of the execution. This forecast has to be done by using metrics derived from relevant KPIs.

*Implementation*

In order to implement the pattern described above, the following steps are necessary:

1. *Collecting events of the type ProcessEvent*: The *ProcessEvents* can be fired either directly out of the process or hard coded out of the services.
2. *Collecting events of the type ExternalEvent*: The continuous collection of *ExternalEvents* occurs independently of the business process instance. An example for an external event flow is TMC (traffic message channel) which radio stations send out as a service in addition to their traffic reports.
3. *Correlating the events in a continuous query*: The correlation of the *ProcessEvent* and the *ExternalEvent* enriches the process information with valuable forecasting information.
4. *Generation of the complex event type ProcessFailureForecast*: The result of the correlation is the *ProcessFailureForecast* as a complex event type.

The "External Process Influence Pattern" was implemented in the course of the technology integration study at DHL/Deutsche Post AG. Basically, the capturing of both, process data as well as external data could be realized. Thus, the business is provided with a real-time overview of the actual state as well as with predictions on the status of future business activities.

*Variations*

Possible variations of the "External Process Influence Pattern" depend on the incoming event streams. Among them are the analysis of external weather information and its impact on the delivery process as well as human factor information which is related to the driver's performance.

*Consequences*

The "External Process Influence pattern" provides benefits in the following areas: *It is possible to predict the status of future process activities* by correlating the actual process data with the information of the external process influences.

Thus, this use case shows how *businesses may be enabled to take appropriate measurements in advance, avoiding* losses resulting from failed business processes.

The implementation of a EDBPM use case for the package delivery process at Deutsche Post AG – certainly a core process in logistics – basically shows that the combination of business process modeling and (real-time) complex event processing is feasible (proof-of-concept). This is a first step in the direction of real-time optimization and prediction of business process execution ([20]).

## 5. Conclusion

In this paper, we have established a first link between the flourishing area of modern business process management and BPM platforms on the one hand and complex event processing and CEP media, on the other hand. We have related the main concepts of both technologies in a general reference model for even-driven BPM and identified new specialized roles in the engineering process. By means of a use case of DHL the integration aspects, the added functionality as well as the business value of even-driven BPM are evaluated and demonstrated.

In summary, currently event-driven BPM is able to detect possible errors within a business process by using the CEP technology and alert responsible persons by using e.g. BAM. This leads to an added business value as compared to pure BPM solutions. However, in future business processes also have to react automatically to such errors, e.g. based on a conditional decision and reaction logic. Further key applications will take place in all domains and first projects on the basis of «Event Driven BPM»-platforms just start 2008:

- Logistics applications, e.g. at DHL/Deutsche Post [16]
- Finance applications, e.g. Deutsche Bank, TeamBank [17]
- Telco applications, e.g. Deutsche Telekom, T-Mobile

Current/future research projects, e.g. “Domain specific reference models for event patterns for a faster set-up of BPM/BAM applications” are currently being prepared [13]. Obviously, there are a number of further steps to be taken, among them:

- The practical evaluation of relevant use cases in other domains like financial services or eCommerce.
- The development of adequate description formats for complex events and event patterns.
- Theoretical models for a realtime business process modeling – observation / monitoring – optimization cycle.

## 6. References

[1] R. v. Ammon, H.-M. Brandl, H.-W. Düster et al., “Business Activity Monitoring of norisbank - Taking the Example of the Application easyCredit and the Future Adoption of Complex Event Processing (CEP)”, Proceedings of PPPJ’06, Mannheim, 2006, http://www.citt-online.com/downloads/Ammon-Greiner-ACM-PPPJ06-vs5-engl-final.doc, downloaded 2007-08-19

[2] R. v. Ammon, H.-M. Brandl, T. Greiner, D. Guschakowski, “Complex Event Processing in the Context of Business Ac-tivity Monitoring: An evaluation of

different approaches and tools taking the example of the Next Generation easy-Credit.", Preworkshop DEBS'07, H.-A. Jacobsen, Toronto, 2007, http://www.debs.msrg.utoronto.ca/collaborate/Workshop_Agenda

[3] Bruce Silver Associates, FileNet P8 – Event-Driven Business Process Management. Industry Trend Reports June 2003, http://www.ecm-unverzichtbar.de/Repository/48/files/69_EvBPM.pdf, http://www.complexevents.com/

[4] Luckham, D. The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. 2002. Addison-Wesley Professional

[5] Danielsson, K., Trotta, G. Key Requirements for Event-Driven BPM and SOA. http://www.ebizq.net/topics/bpm/features/8700.html

[6] Dheap V., Ward, P. Event-driven response architecture for event-based computing. Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research table of contents, Toranto, Ontario, Canada, 70 – 82 [IBM Centre for Advanced Studies Conference archive,].

[7] Ibrahim, M. and Etzion, O. 2006. Workshop on event driven architecture. In Companion To the 21st ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications (Portland, Oregon, USA, October 22 - 26, 2006). OOPSLA '06.

[8] OMG/BPMI, Business Process Modeling Notation (BPMN) Information, http://www.bpmn.org/, provides links to the current specification and relevant literature

[9] List, B.; Korherr, B. "An Evaluation of Conceptual Business Process Modelling Languages" Proc. 21st Annual ACM Symposium on Applied Computing SAC '2006, Dijon, France, April 2006, 1532-1539.

[10] van der Aalst, W. M. and Bisgaard Lassen, K. 2008. Translating unstructured workflow processes to readable BPEL: Theory and implementation. Inf. Softw. Technol. 50, 3 (Feb. 2008), 131-159.

[11] Workflow Management Coalition, 2005. Workflow Standard Process Definition Interface. -- XML Process Definition Language. Document Number WFMC-TC-1025. Version 2.00, October 2005, http://www.wfmc.org/standards/docs/TC-1025_xpdl_2_2005-10-03.pdf [accessed 03 / 2008].

[12] Albek, E., Bax, E., Billock, G., Chandy, K. M., and Swett, I. 2005. An Event Processing Language (EPL) for Building Sense and Respond Applications. In Proceedings of the 19th IEEE international Parallel and Distributed Processing Symposium (Ipdps'05) - Workshop 2 - Volume 03 (April 04 - 08, 2005). IPDPS. IEEE Computer Society, Washington.

[13] Ammon, R. v., Silberbauer, C., Wolff, C. Domain Specific Reference Models for Event Patterns – for Faster Developing of Business Activity Monitoring Applications. VIPSI 2007 Lake Bled, Slovenia, 8-11 October 2007

[14] Kobielus, J., “Complex event processing: still on the launch pad”, Network World , 08/16/2007, http://www.networkworld.com/columnists/2007/082707-kobielus.html [accessed 03 / 2008].

[15] Paschke, A.: The Reaction RuleML Classification of the Event / Action / State Processing and Reasoning Space, White Paper, October, 2006, arXiv:cs/0611047v1

[16] Brandl, H.-M., Guschakowski, D. Complex Event Processing in the context of Business Activity Monitoring. An evaluation of different approaches and tools taking the example of the Next Generation easyCredit. 2007. Diploma thesis http://www.citt-online.de/downloads/Diplomarbeit_BaGu_Final.pdf

[17] Emmersberger, C., Springer, F. Event Driven Business Process Management. An evaluation of the approach of Oracle and the Sopera Open Source Framework. 2008. Diploma thesis http://www.citt-online.de/downloads/EmmSpr_Diplomarbeit_Final.pdf

[18] McReynolds, St., 2007, Complex Event Processing in the Real World. White Paper, September 2007. Redwood Shores, CA, Oracle Corporation, http://www.oracle.com/technologies/soa/docs/oracle-complex-event-processing.pdf [accessed 03 / 2008].

[19] Peng, W., Sun, T., Rose, P., and Li, T. 2007. A semi-automatic system with an iterative learning method for discovering the leading indicators in business processes. In Proceedings of the 2007 international Workshop on Domain Driven Data Mining (San Jose, California, August 12 - 12, 2007). DDDM '07.

[20] Hayes-Roth, F.: Value Information at the Right Time (VIRT): Why less volume is more value in hastily formed networks. 2007 www.nps.edu/cebrowski/Docs/VIRTforHFNs.pdf

[21] Ranadive, V. The Power to Predict: How Real Time Businesses Anticipate Customer Needs, Create Opportunities and Beat the Competition. 2006. Mcgraw-Hill Professional.

[22] Widder, A., Ammon, R. v., Schaeffer, P., and Wolff, C. 2007. Identification of suspicious, unknown event patterns in an event cloud. In Proceedings of the 2007 inaugural international Conference on Distributed Event-Based Systems (Toronto, Ontario, Canada, June 20 - 22, 2007). DEBS '07, vol. 233. ACM, New York, NY, 164-170.