Mapping Clouds of SOA- and Business-related Events for an Enterprise Cockpit in a Java-based Environment

Daniel Jobst
TietoEnator Deutschland GmbH
Digital Innovations
80335 München, Germany
Daniel.Jobst@tietoenator.com

Gerald Preissler
Deutsche Post AG
SOPSOLUTIONS®
53250 Bonn, Germany
G.Preissler@deutschepost.de

ABSTRACT
This paper is about business process management (BPM) and business activity monitoring (BAM) using event processing. We will show why the management of business processes is important for all further steps towards an event-driven and real time enterprise. That is process automation using workflow engines and standards like the Web Service Business Process Execution Language (BPEL). As an underlying middleware platform we use the service oriented platform SOPware of Deutsche Post AG.

Events are emitted from all layers, the middleware platform layer and the business process layer, and figuratively build "event clouds". Event processing functionalities will correlate both "event clouds" and feed business activity monitoring. There, enterprise performance cockpits and dashboards depict the performance of the enterprise and of its processes.

Categories and Subject Descriptors
D.2.11 [**Software Architectures**]: patterns – *service oriented architecture, event processing.* D.2.8 [**Metrics**]: Process metrics – *business activity monitoring, enterprise cockpit.* D.3.2 [**Language Classifications**]: Object-oriented languages – *Java, J2EE.*

General Terms
Management, Measurement, Performance, Design, Standardization.

Keywords
Service oriented architecture, event processing, business activity monitoring.

1. TECHNOLOGIES AND ARCHITECTURE
In the following chapters we will concentrate on how to do monitoring better rather than the question what it is for. We will take into account recent developments in IT architecture and event processing technology.

1.1 Business process management (BPM)
According to the latest Gartner research paper on process management, today BPM is more than a collection of software tools. BPM is a management discipline with modeled business processes as its fulcrum. It aims to improve enterprise agility and the operational performance. Decisive BPM technologies are process modeling, process execution, a BPM suite, and accessing underlying resources using service orientated technologies. [1]

As modeling tool and modeling environment we are using the ARIS toolset in our use case. Processes in their finest granularity will be denoted in the Event-driven Process Chain (EPC) .

PPPJ 2006, August 30 – September 1, 2006, Mannheim, Germany.

According to [2] and the monitoring and event processing approach we are discussing in this paper, focusing on processes and explicit process models are vital for further process measurement and monitoring.

1.2 Process automation and service orientation
According to the approach we are introducing in this paper, process automation needs some preconditions.

First of all, it needs process models in the right granularity for SOA. The modeled process steps have to match the appropriate service calls or sub processes in the later automated processes.

Then it needs a service oriented platform with concepts providing for services like authorization, authentication, user management, data management, logging, exception handling, monitoring, and security. SOPware is a Java-based infrastructure that allows the easy integration of J2EE and J2SE-based applications in an SOA environment.

Thirdly, process automation needs a standardized process language which can be executed by a workflow engine much like a programming language. The Business Process Execution Language is such a language which is designed to orchestrate web services. For the current version 1.1 (BPEL4WS) and version 2.0 (WSBPEL) see [4].

1.3 Business activity monitoring (BAM)
BAM as a technology gives access to key business metrics. It is used to monitor business objectives, to evaluate operational risk, and to reduce the action time between an event happening and the actions taken [5].

According to [6], BAM incorporates the different technologies like BPM, BPEL, event processing, and SOA. It links events, services, and processes with rules, notifications, and human interaction.

Results of BAM are needed for controlling an enterprise and are important for other methodologies like the balanced scorecard (BSC) and Six Sigma or for compliance with regulations like the Sarbanes Oxley Act (SOX) or Basel II.

1.4 Event processing (EP)
Each enterprise has to deal with a huge number of events. These events can be low level network events or high level business events. All of them have some kind of representations in the IT system. In order to deal in real time with a huge number of events, EP provides different technologies and methodologies. One is to classify events in different hierarchical layers, to look for patterns of events within a layer, and propagate new complex events to

higher layers as introduced by [7]. Another way is to arrange events into streams and monitor, analyze, and act upon events as they appear in the stream [8].

EP is not a new technology. But with increasing hardware and network capacities and evolving standards and tools it gets a new momentum [9].

2. USE CASE

An abridged business process from the banking domain will be the consistent process example in this paper. It is a simple version of a credit application process. Once a credit application is received, a data validation service will be called. Then, depending on the amount of the application, either the scoring and approval will be done automatically or manually by an employee. After that the customer will be sent an email.

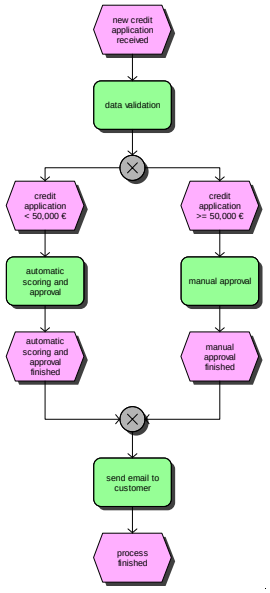

Figure 1. Process in EPC notation

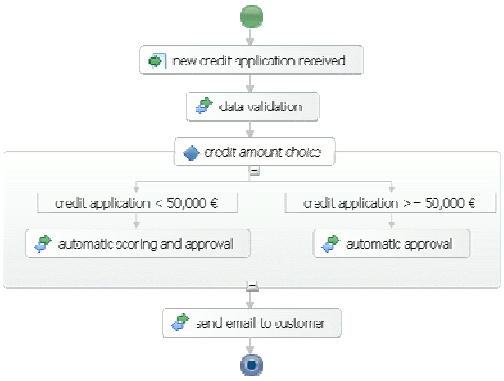

Figure 2. BPEL Process

Figure 1 shows the process in the EPC notation, Figure 2 is the process translated into BPEL. For our services “data validation”, “automatic scoring and approval”, “automatic approval”, and “send email to customer” in the example BPEL process in Figure 2 there is a service implementation within SOPware which will be called by the BPEL workflow engine.

The BPEL process will be published as a service itself. In terms of multi channeling this is important because independent of a channel (e.g. portal, call center, self service devices) always the same process is used. And it is important for the monitoring. The different channels will be a dimension which will be monitored and analyzed in the BAM environment.

The basic concepts of a service-oriented architecture as well as the SOA implementation chosen for our studies are described in chapter 3.

The overall architecture of the approach we introduce in this paper is shown in Figure 3.

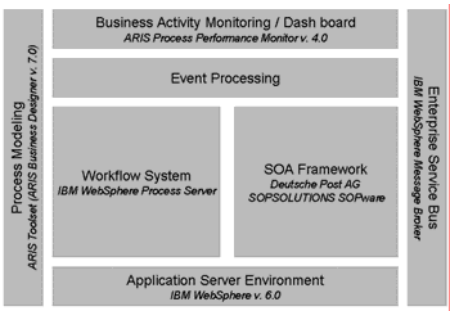

Figure 3. Our architecture

The workflow system, the SOA framework, and the enterprise service bus are based on a J2EE application server environment. The workflow system runs the BPEL processes and publishes business and process events. The SOA framework is Java based too and publishes the lower level SOA events. The enterprise service bus (ESB) is responsible for event transportation and format transformations through the different layers.

Here we have to understand the problem: BAM tools monitor and analyze events from an high level “event cloud” whereas technical monitoring processes events form an “event cloud” of lower level events. Figure 4 shows the two different monitoring views with their layers and typical vendors and tools. With EP both views can be mapped in order to reach a consistency in terms of monitoring completely end-to-end without replacing existing installations by mapping instances.

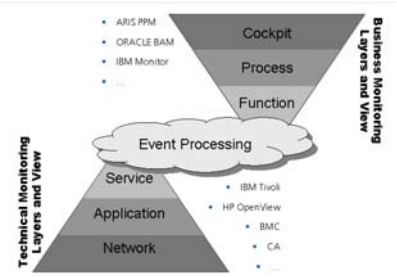

Figure 4. Technical and business monitoring layers and views

The event processing layer will correlate the actual business and process events (from the business instance) with the events from

the technical SOA events (from the implementation layer). This we will describe in chapter 5.

As BAM tool we use the ARIS Process Performance Manager (PPM) to build personalized views of a performance dashboard to visualize key performance indicators (KPI) such as "average process runtime". The event processing layer will deliver the events and metrics needed for the dashboard and to be able to drill down to aggregated process views or to single process instances.

3. SOP SERVICE ORIENTED ARCHITECTURE MODEL

To discuss the events generated in a service oriented architecture (SOA), an understanding of the components of a SOA and their interactions is necessary. As of the time of writing, there are no standard or established definitions for this. The following paragraphs will try to define the characteristics of a SOA that are relevant for the topic of this paper. This is not intended to be a discussion of SOAs in general – which would be beyond the scope of the document – but to foster a common understanding of the problem domain under discussion.

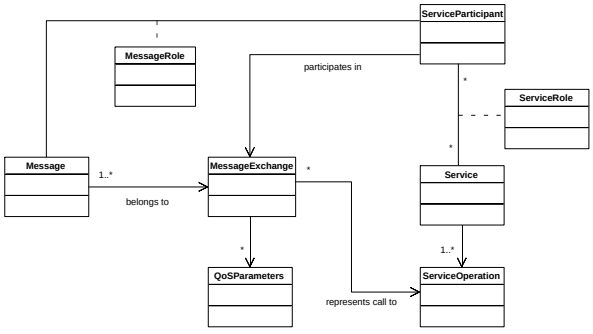

Figure 5. Elements of a SOA

Figure 5 shows an overview of the entities that are relevant for management purposes.

A service is the encapsulation of a defined unit of business logic. From a management point of view, a service is a set of service operations. Further functional properties such as the exact syntax of individual operations or their semantics are not considered here.

An entity acting within a SOA is called service participant. Each service participant can act as service provider or service consumer for one or more services.

The basic mechanism for service usage is the exchange of XML-encoded service messages between a service consumer and a service provider.

A message exchange groups one or more transmissions of individual messages that together complete one invocation of a service operation. The pattern of messages exchanged between consumer and provider during an exchange is a property of the operation. A number of such exchange patterns are defined in [17].

For each message exchange, a number of quality of service properties (QoS) can be defined. A common example for a QoS property is the maximum response time for an In-Out type operation. These properties can be defined in multiple ways, e.g. for all calls to a service operation or by individual negotiation between service provider and consumer. At runtime, a concrete set of QoS properties applies to an individual message exchange. The adherence to the parameters defined for each exchange should be monitored.

For studying the actual integration of a SOA implementation into a business activity monitoring solution, the service-oriented platform SOPware, provided by the `SOPSOLUTIONS` department of Deutsche Post AG, has been chosen. This platform provides a distributed component called SBB library that each service participant uses to consume or provide services. This component provides management and monitoring capabilities based on the Java Management Extensions. Part of these facilities is the generation of SOA-releted events as described below. In addition to the SOPware internal processing and reporting capabilities, this information can be propagated to existing monitoring or event-processing environments.

This paper focuses on a small detail of SOPware, for further information about the platform as a whole, please see [18].

4. EVENTS IN A SERVICE ORIENTED AND PROCESS ENVIRONMENT

4.1 Events

"An event is an object that is a record of an activity in a system."[7] This can be both a more technical event like "service response is five seconds overdue" and an event of a higher business level, e.g. "new credit application received".

What are common to all events are three aspects: form, significance, and relativity to other events (by time, causality, and aggregation). Form means that it is an object with significance meaning that this object is a representation of an activity that happened. In order to specify the activity, the object does consist of attributes describing the activity. Events relate to other events, this can be by time (event A happened after event B), causality (events referring to the same customer ID) or aggregation (event A was caused by the occurrence of events B and C). [7]

A first step towards standardization of events is the common base event (CBE) format. With the standardization of Web Services Distributed Management (WSDM) [10] this standard does specify a WSDM event format. The WSDM event format was implemented within the common event infrastructure (CBI)[1] and is known as the CBE [11]. The CBE does at least require a unique identifier (`globalInstanceId`), a creation timestamp (`creationTime`), an ID of a component which caused the event

[1] "The Common Event Infrastructure (CEI) is IBM's implementation of a consistent, unified set of APIs and infrastructure for the creation, transmission, persistence and distribution of a wide range of business, system and network Common Base Event formatted events." [11]

(`sourceComponentId`), and data structure identifying the situation that happened (`situation`) [12].

Now with a basic definition of events and the minimal requirements, the following two paragraphs define the events in the respective "event clouds".

4.2 SOA events

Based on the model described in chapter 3, a number of events can be defined that can be useful in monitoring the operation of a SOA.

The first class of events describes the lifecycle of service participants at runtime. Whenever a service participant is started or stopped, a corresponding message can be generated. Further events can be generated when a participant registers itself as provider or consumer for a particular service or service operation.

The second class of events provides insight into the usage of service operations. Each service participant can generate events that provide information about calls to service operations and their results. The level of detail provided by these events can range from aggregated data about a large number of service calls to fine-grained tracking data that allows the monitoring of individual message exchanges as they propagate through the messaging infrastructure. The following levels have been proven to provide practical information:

- An aggregation of statistical data (number of calls, successful completions, and response times) about calls to a particular service operation made or served by a service participant during a defined period of time.
- Information about the result of each individual message exchange.
- Tracking information that details each step of processing of an individual message exchange.

While the first level of detail is mainly useful for accounting purposes and capacity planning, the more fine-grained information provided by the other levels can be correlated with information about business events as detailed below.

The third class of events is concerned with quality of service parameters. Whenever a violation of a QoS parameter defined for a message exchange is noticed by the integration infrastructure, a corresponding event is generated.

Instances of the events discussed above are automatically generated by the SBB library and collected at a central location. Therefore, integration into a corresponding business activity monitoring becomes feasibly without any side-effects for the implementation of business applications using the service infrastructure.

The information contained in the events provides information about the event type, event source (service participant or message exchange), and data specific to the event type. This allows correlation of events within the context of the integration infrastructure. To use this information in the context of business activity monitoring, a correlation between SOA-based identifiers (namely message exchange identifiers) and business activity related identifiers (e.g. a business process identifier) needs to be established. The integration infrastructure provided by SOPware supports this in two ways:

An application using the SBB library to access services can specify an arbitrary identifier as an optional parameter to the API call to invoke a service operation. This identifier is observable in all tracking events that are generated for the message exchange that realizes the invocation.

If this is not possible, e.g. if legacy systems are integrated into the execution of a business process, a correlation message can be generated that connects the message exchange identifier with any business identifier that is part of the message payload and that can be specified by XPath [19] statements.

It is our goal to set the SOA-level events described here in relation to high-level events generated by the business process to get a better understanding of the operation of the whole infrastructure.

4.3 Business and process events

The workflow environment of the use case described in chapter 2 does provide the following CBE events for each invoke or receive in a BPEL process[2]: Entry, Exit, Expired, Failed, Completion forced, Retry forced, Skipped, Stopped, and Terminated.

The events above are possible activities during the BPEL process execution and are CBE. They will be called process events in this paper.

Business events are specified by the messages or events either a business service receives or produces such as "new credit application received".

Because the processes are modeled in BPEL and because they resemble the business process, the process events can indicate business events. In order to denote an event for "new credit application received" the CBE process event "Entry" of the receive process node can be used.

Another way to raise business events is with sensors in BPEL processes as explicit service calls. This was introduced in [16].

5. MONITORING WITH EVENTS

5.1 Use case scenarios

The essential idea behind existing monitoring tools (both business and technical) is to alert when single events happen and then to act upon them [13]. But the humblest low level event can have impact on high level processes and the overall performance of the enterprise as well as high level events can have future impact on IT assets. In this chapter we want to show on simplified examples bases that both "event clouds" can be mapped in order to extract useful information and monitoring results.

[2] IBM WebSphere Integration Developer (WID) 6.0.1 and IBM WebSphere Process Server 6.0

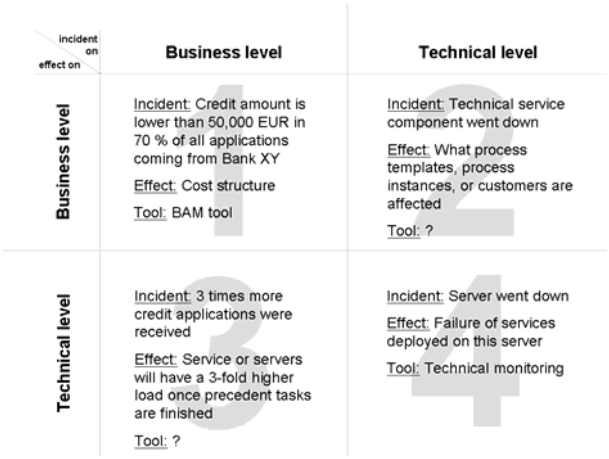

Figure 6. Examples of business and technical incidents and their effects

Figure 6 shows a matrix of four scenarios with business and technical incidents (horizontal) and business and technical effects (vertical). Incidents with an effect on the same level are easy to handle and are state-of-the-art (scenarios 1 and 4).

Scenarios 2 and 3 exceed the scope of the respective monitoring environment and the connections between them are hard to trace.

5.2 Event correlations and patterns

With event correlations and event patterns we will concentrate on scenarios two and three.

Scenario 2: Here we have to detect different activities. 1. That a service component went down or that it is not responding as agreed upon in a service level agreement (SLA). 2. What process instances were using the service component? 3. Which process template are affected, who are the customers, and other details?

For the purpose of illustrating how to write event patterns we will use the straw man event pattern language STRAW-EPL as introduced by [7].

Element	Declarations
Variables	MessageExchange mex, Time timeAt, Operation op, Time timeAgreed, Time actualTime, Correlation c, MessageId mid, CorrelationId cid, BusinessId bid, String processInstanceId, String processTemplateId, Customer cust
Event types	RESPONSETIME_EXCEEDED (mex, op, timeAgreed, actualTime) CORRELATION(c, mid, cid, bid) new_credit_application.Entry(processInstanceId, processTemplateId, cust)
Rel. operators	**and**
Pattern	RESPONSETIME_EXCEEDED **and** CORRELATION **and** new credit application.Entry
Context test	mex.getId=mid **and** bid=processInstanceId
Action	**create** BamScenario2Event("Scenario 2", …)

Figure 7. Event pattern for scenario 2 in STRAW-EPL

Here, two SOA events are correlated with a business event. Once the response time of a component is exceeded we take the correlation event which holds both the link to the service component and the business process instance data and correlate them. If the pattern matches a new event will be created.

We suggest the `BamScenario2Event` and `BamScenario3Event` to be a CBE and to be sent via the ESB to the BAM tool. The ESB can then transform it to proprietary event formats (e.g. "ARIS PPM Event Format") or pass it on to other CBE consumers. With this approach BAM tools could be loosely coupled just like services in a SOA.

Scenario 3: In scenario 3 we receive more credit applications than normal. If the credit amount is above 50,000 Euros, the process does start a human task and waits for the task to be finished before proceeding (which according to the BPEL is a service call). Before completing the manual tasks, no preceding service call is made, meaning that the SOA framework has no way of knowing about the imminent increase in service calls. Therefore, no pro-active measures, like the deployment of additional service providers, can be taken.

In order to measure the number of credit applications with events over a period of time we have to extend the STRAW-EPL with a sequence of events (identifier "sequence") with an index *i* and an identifier for a time window ("within").

Element	Declarations
Variables	String ProcessInstanceId(i), String ProcessTemplateId(i), Customer cust(i)
Event types	new_credit_application.Entry (processInstanceId(i), processTemplateId(i), cust(i))
Rel. operators	**and**
Pattern	**sequence** (new credit application.Entry(i)) **within** last 8 h
Context test	processTemplateId(i)=processTemplateId(i+1)
Action	**create** BamScenario3Event(…)

Figure 8. STRAW-EPL for scenario 3

With the creation of `Scenario3Event` and the business process knowledge the proceeding service calls (and thus the IT assets affected) are automatically identified.

Again, the event patterns above do not claim to be real world patterns. They do only demonstrate that with event patterns and EP a technical event cloud can be matched with a business event cloud.

5.3 BAM

Although the ARIS PPM tool does only meet the first two features of the BAM event processing checklist in [14] (i.e. real time event data computation and single event triggering), it is widely used and does use the EPC process notation. This does mean that a business process lifecycle can be reached from process modeling and process automation to monitoring and process re-design and so forth with only one business process notation.

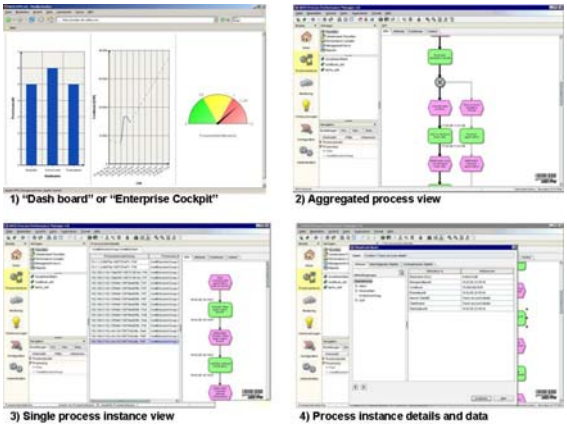

Figure 9. Different ARIS PPM BAM views

The functionality of the remaining three features of the BAM event processing checklist (event streams processing, event pattern triggering, and event pattern abstraction) would be met by implementing scenarios 2 and 3. What is still missing is a monitoring user interface to visualize the results of our EP scenarios.

The possibility we suggest is to use an existing BAM tool such as ARIS PPM. Since those tools are at least capable of single event processing and the visualization of single events such as `BamScenario2Event` or `BamScenario3Event`.

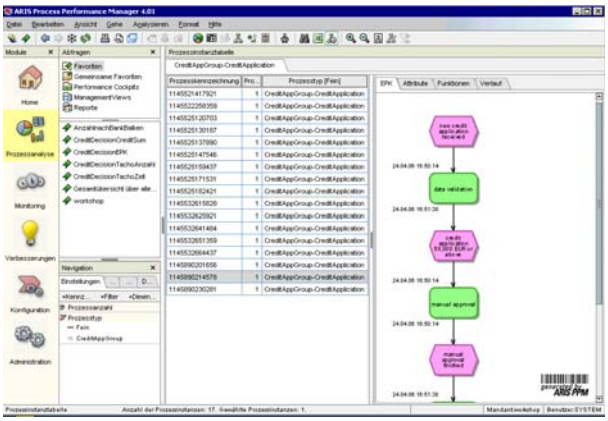

Figure 10. Scenario 2 visualization

Figure 10 shows a list of process instances where the pattern (see Figure 7) matched. Every process instance can be shown in the appropriate EPC process model exactly as it was executed. The single process nodes contain the business data monitored. This is where further information about the process, customer, or other details can be retrieved as shown in Figure 11.

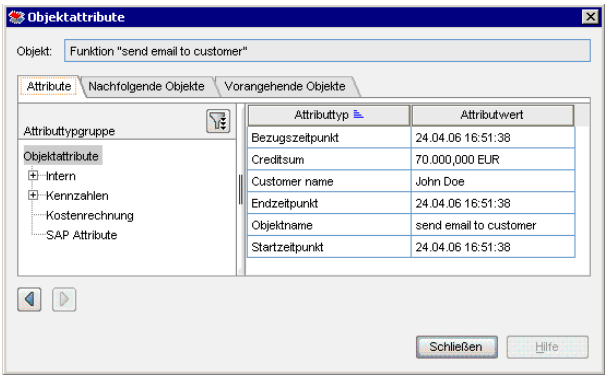

Figure 11. Details of a process node in ARIS PPM

The visualization of scenario 3 shows a speedometer like diagram with the number of process instances within a time window. Figure 12 shows that the number of running processes has reached a critical area. Due to the coupling of the BAM EPC view with the EPC process models all following activities and service calls of a process template or instance can be identified.

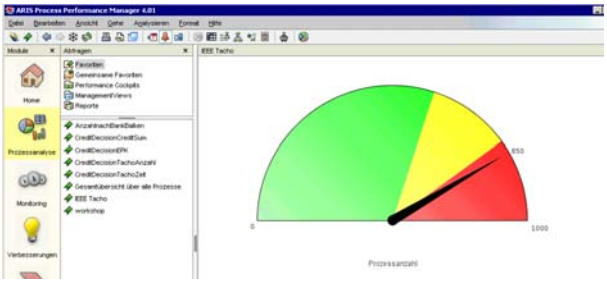

Figure 12. Scenario 3 visualization

6. CONCLUSION

EP as a very young discipline has not yet reached a level of agreed standards, wide spread tools, or good market penetration. It is our belief that it holds the potential to leverage synergies between the areas of BPM (management, modeling, and automation of processes), BAM (monitoring of processes) and SOA monitoring and management, among others.

Our study shows one possible way of utilizing the capabilities present in the technologies mentioned above to achieve this goal. We showed links between events (both business-events and events in an SOA), information processing, and processes in order to leverage modern application and monitoring platforms such as SOPware and ARIS PPM. As theoretically introduced in [20] our study shows furthermore that monitoring using EP in combination with BPEL processes provides a comprehensive and agile foundation for business process optimization that ensures efficient and flexible business processes.

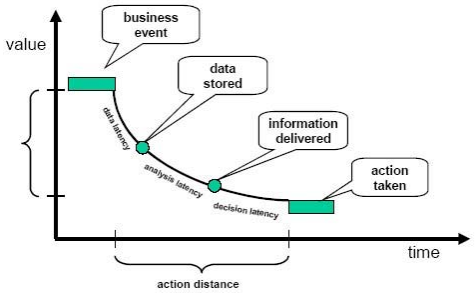

Figure 13. "Business Value of Data Freshness" [15]

To illustrate a business value behind our study we want to refer to the time-value curve of [15] where the author shows the connection between "data freshness" and the "business value". This "data freshness"[3] we can provide by applying EP in order to map the clouds of events and to complete functionalities today's BAM environments are not yet able to provide.

7. REFERENCES

[1] J.B. Hill, J. Sinur, D. Flint, M.J. Melenovsky, Gartner's Position on Business Process Management, 2006, Gartner Research, 16 February 2006, pp. 5-8.

[2] P. Küng, C. Hagen, M. Rodel, S. Seifert, Business Process Monitoring & Measurement in a Large Bank: Challenges and selected Approaches, Proceedings of the 16th International Workshop on Database and Expert Systems Applications, IEEE Publications, 2005.

[3] R. Davis, Business Process Modelling with ARIS. A Practical Guide, Springer, London, 2001.

[4] OASIS Web Services Business Process Execution Language (WSBPEL) TC, Organization for the Advancement of Structured Information Standards (OASIS), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel, accessed on 17 April 2006.

[5] B. Gassman, Who's Who in Business Activity Monitoring, 4Q05, Gartner Research, 12 April 2006, p. 3.

[6] J. Deeb, F. Knifsend, Using BAM to Empower the Organization: Unifying Events and Processes, http://www.BIJOnline.com/index.cfm?section=article&aid=130, 2005.

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

[8] Event Stream Processing: A New Computing Model, http://www.eventstreamprocessing.com, accessed on 15 April 2006.

[9] R.W. Schulte, The Growing Role of Events in Enterprise Applications, Gartner Research, 9 July 2003, pp. 2-3.

[10] Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1, OASIS, 9 March 2005, http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf, accessed on 19 April 2006.

[11] Common Event Infrastructure, Intelligent automation that saves time and improves resource utilization, IBM, http://www-306.ibm.com/software/tivoli/features/cei, accessed on 19 April 2006.

[12] Autonomic Computing Toolkit, Developer's Guide, IBM, August 2004, http://www-128.ibm.com/developerworks/autonomic/books/fpy0mst.htm#HDRAPPA, accessed on 19 April 2006.

[13] D.C. Luckham, Why Business Activity Monitoring Must Use CEP, 21 July 2004, http://complexevents.com/?p=23, accessed on 20 April 2006.

[14] D.C. Luckham, Can We Bring Some Order to the Business Activity Monitoring Space?, 16 June 2005, http://complexevents.com/?p=5, accessed on 20 April 2006.

[15] R. Hackathorn, Current Practices in Active Data Warehousing, November 2002, http://www.dmreview.com/whitepaper/WID489.pdf, pp. 23-24, accessed on 21 April 2006.

[16] D. Jobst, T. Greiner, Modern Process Management With SOA, BAM und CEP, From static process models to executable workflows and monitoring on business level, 1st Event Processing Symposium, Hawthorn NY, March 2006, http://complexevents.com/wordpress/slides/CITT_CEPSymposium_Jobst_Greiner.pdf, accessed on 21 April 2006.

[17] World Wide Web Consortium, Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts, March 2006, http://www.w3.org/TR/2006/CR-wsdl20-adjuncts-20060327/, accessed on 24 April 2006.

[18] J. Belger, Die service-orientierte Platform der Deutschen Post, JavaSPEKTRUM 1/06, p. 22.

[19] XML Path Language (XPath) Version 1.0, W3C Recommendation, 16 November 1999, http://www.w3.org/TR/xpath, accessed on 24 April 2006.

[20] J. Deeb, What Does Complex Event Processing Do for BPM, 13 June 2005, http://www.bpm.com/FeaturePrint.asp?FeatureId=171, accessed on 24 April 2006.

[21] B. Gassman, Gartner Study Reveals Business Activity Monitoring's Growing Value, Gartner Research, 18 April 2006

[3] According to the latest Gartner research today's latency requirements between business events and notification is less than 15 minutes. This is likely to decrease to less than 60 seconds over the next 6 years [21].