Business Activity Monitoring of norisbank Taking the Example of the Application easyCredit and the Future Adoption of Complex Event Processing (CEP)

Torsten Greiner, Willy Düster, Francis Pouatcha
norisbank AG
Rathenauplatz 12-18
D-90489 Nürnberg
torsten.greiner@norisbank.de

Rainer von Ammon, Hans-Martin Brandl, David Guschakowski
CITT GmbH / TietoEnator GmbH
Konrad-Adenauer-Allee 30
D-93051 Regensburg
rainer.ammon@citt-online.com

ABSTRACT

The kernel business process of easyCredit of the norisbank as a German Online Credit System is described with all its detailed process steps. Business Activity Monitoring is a basic condition for the successful operation of the system. The As Is-solution is faced with the future concept based on CEP/ESP. For that reason the process has to be redesigned in the sense of SOA and flexibly implemented by means of a BPEL-based Workflow Engine. In this connection the events of a BPEL-engine are differentiated from the events, which are being processed in a so called "event cloud" by a CEP system for a real time BAM. The possibilities and advantages of a real time able BAM are being shown taking the example of the easyCredit.

Categories and Subject Descriptors

D.2.11 [**Software Architectures**]: l*anguages*

General Terms

Performance, Design, Standardization, Languages, Theory

Keywords

Business Activity Monitoring, Complex Event Processing, Event Stream Processing, Business Process Management, Service Oriented Architecture, online instant credit system

1. INTRODUCTION

Business processes as a whole realize the business strategy of an enterprise in a save, correct and economic way. Thereby IT systems support and optimize the performance of the business processes. In the example of the browser based easyCredit application (rated credit with online instant confirmation on the Internet) of the norisbank AG is described a potential business process monitoring from an economic view based on key figures. This economical monitoring of the business processes on the base of key figures is called Business Activity Monitoring (BAM) [10][11].

The norisbank has realized BAM by a pipeline model. This model assumes that each credit application runs through several processing steps within its entire life cycle. The single steps can be imagined as a production line or a pipeline. In this pipeline model we can trace how many contracts respectively applications are located on which position of the pipeline. The actually traced values of a pipeline section can be compared with threshold ranges, so that in the case of significant deviations indicating a technical fault in the process an alarm can be raised.

In the following the essential basics for the realization of the business process monitoring of the rated credit application easyCredit with BAM is introduced. After that the real system is described, how it has successfully been implemented in the norisbank AG. At last it is shown, how new paradigms of Complex Event Processing (CEP) [9] will be used in the next generation of the easyCredit for realizing an intelligent, real time able BAM platform.

2. SOME REMARKS ABOUT THE BACKGROUND OF EASYCREDIT

The norisbank AG - a 100% affiliate of the DZ-Bank AG since October 1st, 2003 – runs about 100 branches in Germany. Their kernel business is the allocation of consumer credits. In April 2000 norisbank was the first bank in Germany to offer a rated credit application on the Internet with online instant confirmation. This application was certified by the TÜV (Technical Controlling Association, MoT) in 2003 as the world wide first bank product.

Due to the change of the ownerships at the norisbank AG it became necessary to develop a new fully automated and mass business capable rated credit application (easyCredit[1]). A technical relaunch, as well as an economical further development of the old application was accomplished.

In 2002 the norisbank introduced a new technical platform as a basis for a modern Internet branch bank [7]. The technical architecture of this platform was realized by the operation system

PPPJ'06, July 30th, 2006, Mannheim, Germany.

[1] http://www.easycredit.de

Solaris of SUN. This platform provides a scalable architecture, based on the standard of Java Enterprise Edition (Java EE). The Java EE model defines the standard for the implementation, configuration and for operating distributed applications. Furthermore Java EE is web-based, i.e. it is assumed that the clients (webbrowser capable systems) are interacting with the application via the protocol http or https. Thus redevelopment of the easyCredit application was made by the Java EE technology as well. This redevelopment, too, was certified in 2004 by MoT.

Meanwhile about 900 partner banks, with about 12.000 distribution agencies and about 32.000 branch users are using this application, but also the same amount of external customers from the Internet.

From the view of the end users (partner bank, Internet) it's not only of interest, whether the application and the process are available, but the cycle time of a contract within the norisbank is essential, too. The cycle time is here defined by the total processing time from the reception of the credit application until the payment onto the account of the customer. For this reason it is especially important to know, which contract has which actual status.

It is the aim of business process monitoring within a pipeline model to define traffic light status for each pipeline section which has to be controlled:

Green = amount of contracts in a pipeline section are in the defined interval

Yellow = amount of contracts in a pipeline section are approaching the defined interval limits

Red = amount of contracts in a pipeline section are outside the defined interval

Only in the case of an alert in time the throughput of the contracts can be constantly maintained with added effort of personal resources.

3. BUSINESS ACTIVITY MONITORING

For the realization of Business Activity Monitoring the understanding of the business processes are indispensable. In this connection the kernel business process of the easyCredit is to be understood in the way that the process is being started by the credit application of an external or internal customer and only ends with the delivery of an agreed result to the customer (end user). In the following only the business process steps, which are relevant for the monitoring of the application way "Internet guest", are shown and after that the realization of the BAM on the basis of the pipeline model are explained.

In this process a customer uses the Internet to get a credit offer. At first the customer gets to the "mini calculator" page. There he can choose diverse credit parameters like the amount of credit, duration, and optionally an insurance product. After that he will automatically be given the adequate interest rate as well as the monthly amount of interests. Has the customer decided to continue with these conditions, he has to enter his household data in the next step. These data include the monthly income and expenses as well as credit contracts, which may have to be repaid. In the following the customer has to enter his personal data inclusive his address. After that the employment data form has to be processed. As a next step the customer has to enter in the account data page, whereto the money shall be transferred and from which account the monthly rates shall be collected. All the entered data are checked by appropriate procedures for pausibility which will not be explained here in detail.

Does the customer continue with the application process, the credit application with all its data entered so far will be stored in the system. From this point the control of this application within the monitoring system is possible (see section B.1 in fig. 1). Only now an inquiry at the credit agencies Infoscore and Schufa will be made. According to the results the decision on the credit (CD) will be met. The result of the CD will be arranged as follows:

CD red: The customer will be briefly indicated that he cannot get a credit offer (e.g. because of insufficient creditworthiness). For the customer it is the end of the offer process. In the background however a further process automatically produces a letter of denial in the form of a PDF-document. Another process provides the generated PDF-document as a print job. All print jobs are being gathered, printed and sent off in a cyclic way in another process.

CD grey: The customer gets a non-binding offer on the screen assuming a averaged good creditworthiness, as there are no informations of the Schufa in this case. The customer will be informed that he can expect a written offer within short time. Parallel to this the resubmission "reviewing" is set in the background. That means that the further processing of the credit application will be continued by an employee of the norisbank.

CD green: In this case it is differentiated, whether the customer has entered the credit contracts, which will have to be repaid to norisbank. In this case an offer is shown to the customer on the screen and he will be informed that he will get the offer documents by post. In addition the resubmission "reviewing" is set.

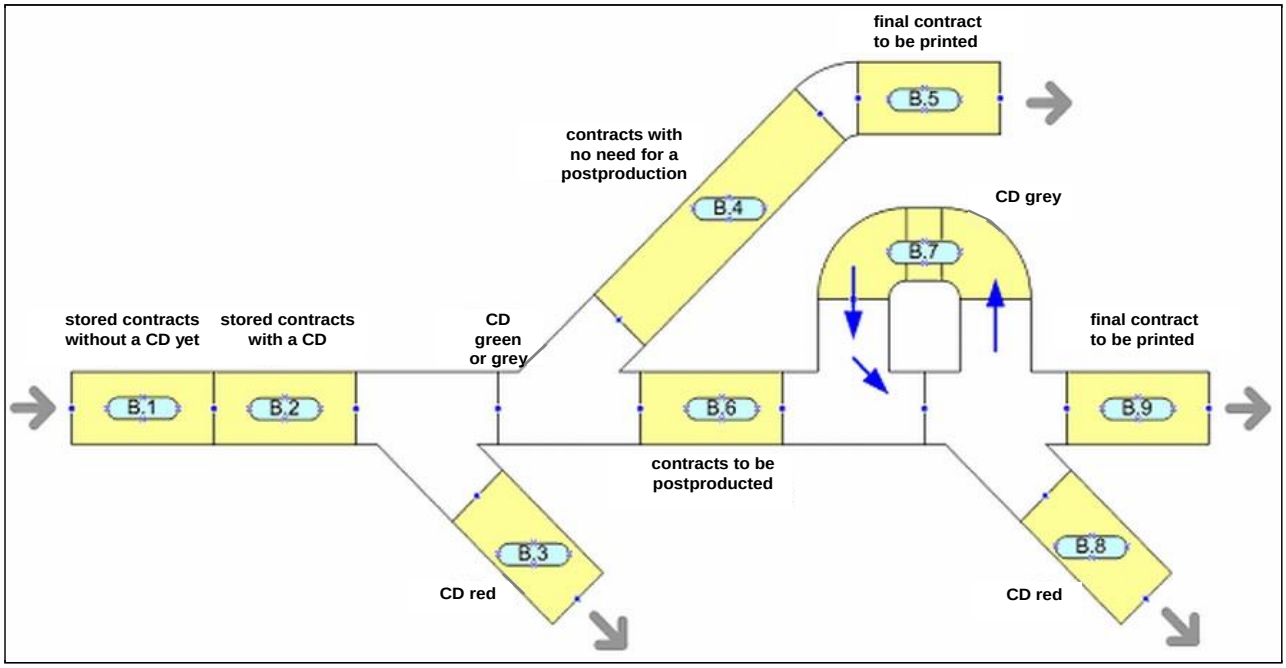
The result of this description is the simplified shown life cycle model of the application way "Internet guest" (see fig. 1).

4. REALIZATION OF THE PIPELINE MODEL

A pipeline model is used for the technical monitoring of the business processes by norisbank. This model assumes that every credit application within its total lifecycle runs through several processing steps, like in a production line or in a pipeline. In this pipeline model we can trace how many contracts respectively applications are located on which position of the pipeline.

Precondition for this is that each change of a status of an application is recorded with a corresponding time stamp in a database. The actually traced values of a pipeline section are then compared with the defined threshold ranges, so that in case of significant deviations indicating a technical fault in the process an alarm can be raised.

In order to check any faults in the business processes the following monitoring principles within the pipeline model will be used:



Legend:
B.1: stored applications, for which no credit decision (CD) was made yet.
B.2: stored applications, for which a credit decision was made already.
B.3: applications, for which the last credit decision was red.
B.4: applications, with CD green, which don't have to go into reviewing.
B.5: applications, with CD green, which don't have to go into reviewing, the contract of which is ready for printing.
B.6: applications, with CD green, which are being reviewed, for which however there hasn't been done any CD in the reviewing yet.
B.7: applications with CD grey in the reviewing.
B.8: applications with CD red in the reviewing.
B.9: agreed applications, the contract for which is ready for printing.

Figure 1: Simplified shown life cycle model of the application way "Internet guest"

- Finding the amounts in the single pipeline sections,
- Monitoring, whether the applications are moving forward in the pipeline,
- Checking, whether all applications/contracts within a certain time frame have reached a final status.

For this the following monitoring variants are used in detail:

Pipeline-snapshot: Establishing the absolute amount of applications within the single pipeline sections at defined times (e.g. each hour).

Pipeline-progress: Establishing, which application objects have newly reached the corresponding pipeline section within a defined time interval (e.g. each 15 minutes). The absolute amount of applications is established.

Application specific finalising: Here it is checked, whether all applications within a certain time interval have reached the defined final status (e.g. one working day per each application). Defined final states are the sections B.3, B.5, B.8, B.9.

Wait time-monitoring: Establishing, how much time it takes for an application from one pipeline section to the next respectively whether an application stays for too long in a certain pipeline section. Critical pipeline sections are: B.2, B.5, B.6, B.7, B.9.

The monitoring of the single application states is done by cyclical database requests. Hereby the time stamp of the corresponding status entry in the database serves as a reference. E.g. if the Schufa credit agency is not available, it is easily possible to recognize through evaluating the pipeline progress that the amount of applications with status CD grey have been raised significantly.

5. FURTHER DEVELOPMENT OF THE MONITORING PROCEDURE WITH CEP/ESP

The so far described approach assumes that each credit application with its total life cycle runs through several processing steps, like in a production line or within a pipeline. Precondition for a corresponding technical monitoring is however that each

status change of an application is recorded with a corresponding time stamp in a database. Only in this way a simple database request can establish, how many contracts are located in which place within the pipeline. The advantage of this procedure is the simple realization.

The disadvantage of this approach however is that only a frozen state of an existing constellation at time point t of the data base request can be seen. A real time monitoring of the economical states of the application is only possible, if status changes of applications are requested in the data base permanently. This means: if the answer for a request is wanted each second, the request has to be entered each second as well. This solution cannot be realized for hundreds of different requests. Therefore the database is a bottleneck in the case of complex status changes, which shall be monitored in real time. Also the triggers, which were integrated later in traditional databases, don't solve the problem, as these don't scale.

Event Stream Processing (ESP) or Complex Event Processing (CEP) are paradigms, which are helpful to react in real time to changes of states by corresponding informations [9]. With CEP/ESP messages, informations or data are correlated, aggregated, analyzed and evaluated in real time. These newly generated informations then provide the base for further decisions. Thus a CEP/ESP-platform becomes an intelligent BAM-tool, which also offers the possibility of dynamic visualization. In a next step this pipeline model shall be realized with methods of CEP/ESP.

5.1 Redesign as SOA and Use of a BPEL-based Workflow Engine

For the new approach of easyCredit it is necessary to redesign the architecture of the system in the sense of a SOA [3][16]. As a principal difference to EAI [12] a SOA is based on the business processes (see fig. 2). At each process step another enterprise internal or external process or a service respectively a software component can be called and eventually a change in a database or in a legacy system, e.g. in an ERP-system like R/3 or in a CRM-system like Siebel, can be caused. Fig. 2 shows the example of an online-credit-system that this architecture could arbitrarily cascade [1]. All software-components respectively services are defined e.g. by the Web-Service-Definition-Language (WSDL) and are bound as web-service to a process step.

However hereby it has to be guaranteed that for the aim of a real time BAM no performance problems will be caused by still relatively heavy and slow XML-based protocols like SOAP [15] because of longer latency times.

This also goes for the use of a BPEL-based workflow engine [13]. Though the business process can be standardized and flexibly implemented in this way and even be modified at run time of the system on a high level, i.e. by the means of workflow design tools by non-IT-experts, if applicable even directly by a business department, for the realization of new business and marketing strategies. On the other hand numerous, concurrent business process instances could cause performance and scalability problems of the BPEL-engine. This has to be considered in time towards the required real time performance of the BAM at the dimensioning of the system and the resource planning (sizing-project).

5.2 Events for the Workflow Engine Versus Events for CEP/ESP

The business process is controlled e.g. in the sense of an eEPC-notation [5] of events, like "credit application received", "credit application checked for completeness", "Schufa-information

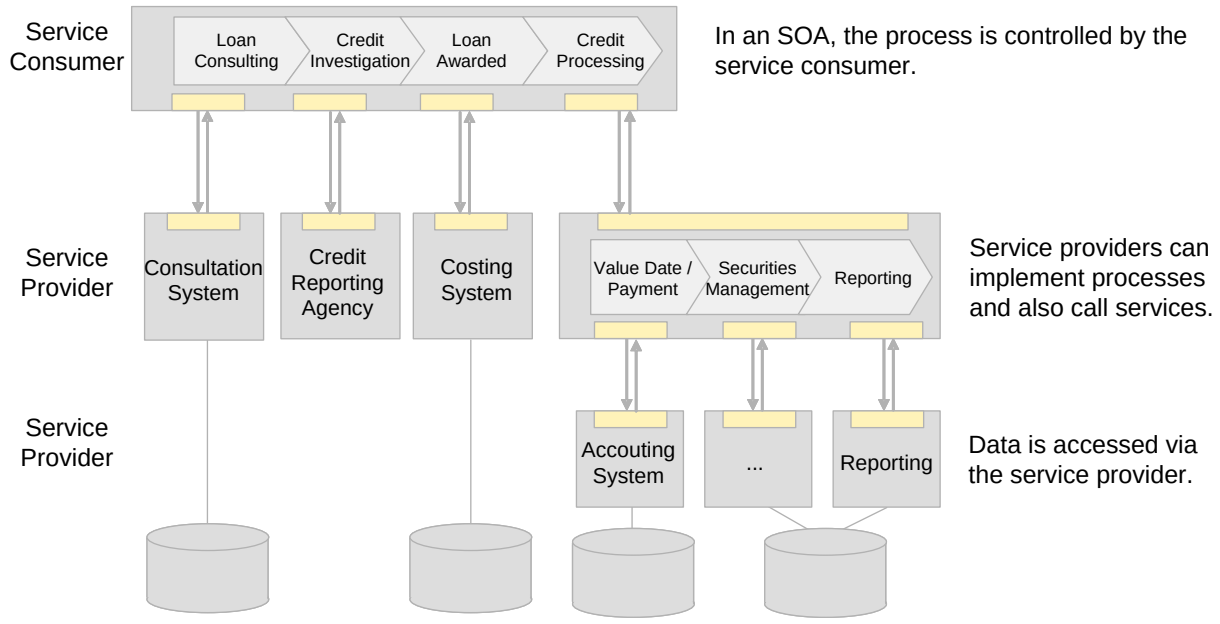


Figure 2: The SOA Challenge - design horizontal and vertical coupling of services

received" and so on. These events are manually caused by men, e.g. by an employee or by actions of the system. These events are needed by the workflow engine for flow controlling. These events, however, are no events, which are filtered from the event cloud or from an event stream by a CEP- or ESP-system and processed, e.g. for the visualization in a BAM. It is the job of a business process modeler to design the process model by interviewing the process-owner or the business department as a chain of events and actions in such a way that the process model can be executed by a BPEL-based workflow-engine (see fig. 3). These thousands and presently archived eEPC-models at banks and other users are at present not designed as "executable workflows" and have to be redesigned more fine grained – also under SOA-prospects.

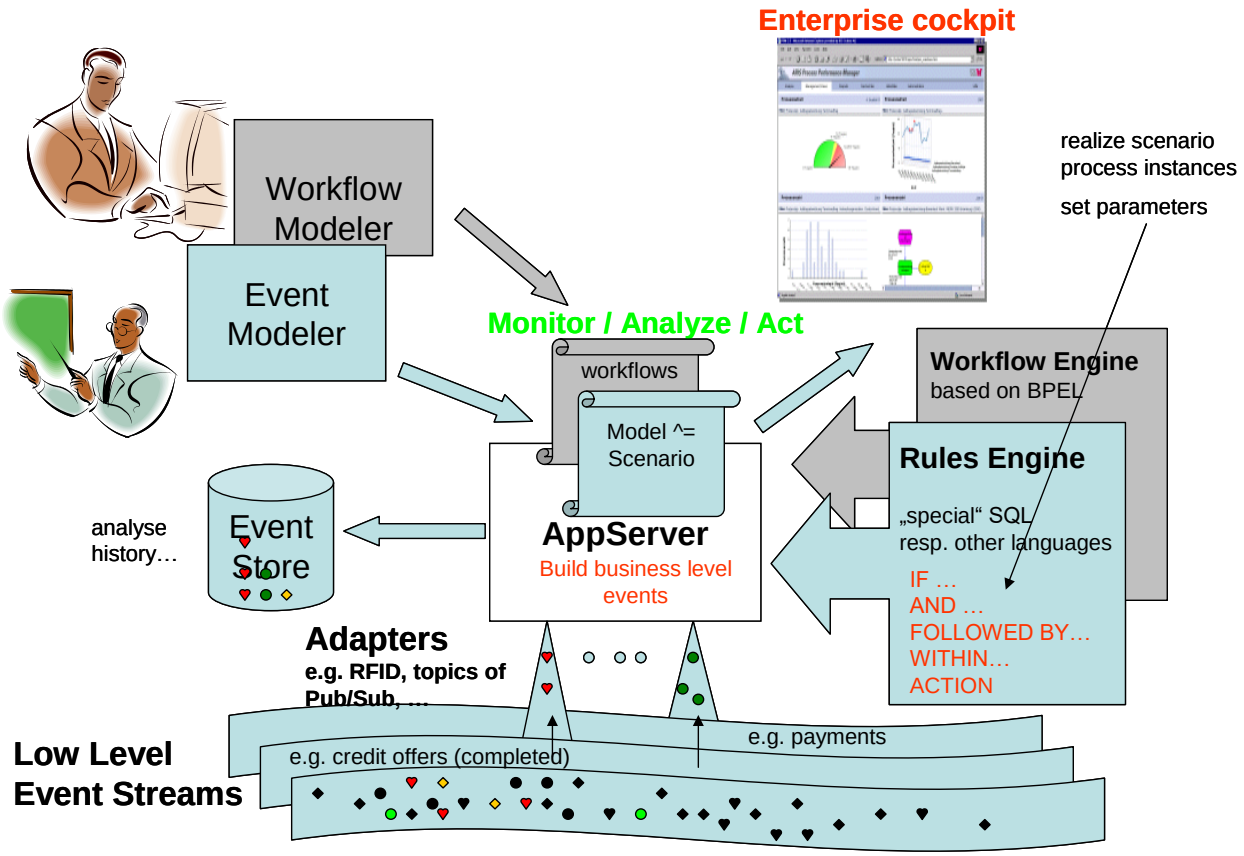


Figure 3: The technology challenge and the principle of BPM/BAM/CEP, e.g. for "Next Generation Instant easyCredit System"

The Business Process Management System (BPMS) (see [14]) generates at executing the business process instances partly autonomously BAM- respectively CEP-relevant informations, e.g. time stamps for each single process step, whereby in the BAM throughput times can be monitored and analyzed. Additionally specific events can be generated through appropriate implementation of actions, e.g. for those in chapter 4 mentioned monitoring-views "establishing the absolute amount of applications within the single pipeline sections at defined times", "establishing, which application objects have newly reached the corresponding pipeline section within a defined time interval", or the calculation of a credit sum or of the interest rate. This can be realized in the Java EE environment via JMS as the Publish-/Subscribe-method or with CORBA analogically via the "notification service" and so on.

Altogether there can be on the network level a very large amount of events of different types in a certain time window (1 minute, 1 hour, 1 day and so on) – metaphorically as an unordered "event cloud" or transformed as a chronologically ordered "event stream".

The event modeler (see fig. 3) decides, which of these events have to be filtered for which BAM-view from one or more streams, if necessary have to be aggregated and correlated as higher business level events (see chap. 4, e.g. pipeline progress, wait time monitoring) and in which time window these events have to be held and stored (see in detail [9]). According to CEP/ESP-systems, just entering the market, those event scenarios again can be generated very quickly and modifiable at any time by means of special high level tools without IT experts.

There are special, often already prebuilt adapters for the filtering of each event type (see. fig. 3). Examples are SNMP-, email- and log file-adapters for searching for specified strings. The aggregation and correlation take place according to the event scenarios through the used CEP/ESP-system by means of their Event-Processing Language (EPL). The CEP-discipline, just

coming up, is presently discussing - still controversial -, how an appropriate language or an appropriate user interface for this task shall be designed. At present the first CEP-platforms offer SQL-like languages, which, however, are extensively enhanced, and process events - precompiled and "in memory" – in a highly performing way (e.g. the "Event Query Language" (EQL) of the Open Source Platform Esper [6] or the "Continous Computational Language" (CCL) of Coral8 [4]). Other systems provide an user interface without any programming as far as possible (e.g. APAMA [2]).

In the future an important, new task will be the modeling of appropriate event patterns by the new role of the event modeler.

6. Conclusion

Though from the perspective of the next generation easyCredit the BAM-views have to be defined first, which are to be monitored – if necessary in real time –, as well as all required actions and alerts have to be determined. After that the necessary event types on the network level (in the event stream) have to be decided and aggregated or correlated in appropriate models. Furthermore it will be evaluated, which BPEL-engine shall be used and which events of which actions in the credit process will have to be generated. For the event management and processing in real time diverse CEP-/ESP-systems are being evaluated.

7. REFERENCES

[1] Ammon, R.v., Pausch, W. and Schimmer, M. Realisation of Service-Oriented Architecture (SOA) using Enterprise Portal Plattforms taking the Example of Multi-Channel Sales in Banking Domains, *Wirtschaftsinformatik 2005*. Ferstl et al. (Publ.), Heidelberg, Physica-Verlag 2005, 1503-1518.

[2] Apama, http://www.progress.com/realtime/products/apama/apama_technology/index.ssp.

[3] Brandner, M. et al., Web services-oriented architecture in production in the finance industry, *Informatik Spektrum*, Volume 27, No. 2, 2004, 136-145.

[4] Coral8, http://www.coral8.com/, downloaded 2006-04-24.

[5] eEPK – erweiterte Ereignisgesteuerte Prozesskette, http://de.wikipedia.org/wiki/Erweiterte_ereignisgesteuerte_Prozesskette, downloaded 2006-04-24

[6] Esper, http://esper.sourceforge.net, downloaded 2006-04-24.

[7] Greiner, T. Lachenmayer, P. Bereitstellung einer neuen technischen Plattform als Grundlage für eine moderne Internetfilialbank, *Banking and Information Technology*, Institut für Bankinformatik und Bankstrategie an der Universität Regensburg, Regensburg, 2002, 53-61.

[8] Greiner, T., Düster, W. Monitoring von Geschäftsprozessen mit OpenSource Produkten aus Endkundensicht, *Banking and Information Technology*, Institut für Bankinformatik und Bankstrategie an der Universität Regensburg, Regensburg, 2005, 49-61.

[9] Luckham, D., *The power of events*, Addison Wesley , Boston, San Francisco, New York et al., 2002.

[10] Luckham, D. The Beginnings of IT Insight: Business Activity Monitoring http://www.ebizq.net/topics/bam/features/4689.html, 2004, downloaded 2006-04-24

[11] Kochar, H. Business Activity Monitoring and Business Intelligence http://www.ebizq.net/topics/bam/features/6596.html, 2005, downloaded 2006-04-24

[12] Meinhold, G. EAI und SOA: Die Komponenten fallen nicht vom Himmel, *Objektspektrum*, No. 2, 2004, 33-36.

[13] OASIS, Web Services Business Process Execution Language Version 2.0, Draft, Dec. 2005, http://www.oasis-open.org/committees/download.php/16024/wsbpel-specification-draft-Dec-22-2005.htm, downloaded 2006-04-24.

[14] SixSigma, BPMS, http://www.isixsigma.com/dictionary/BPMS-536.htm, downloaded 2006-04-24.

[15] SOAP, http://de.wikipedia.org/wiki/SOAP, downloaded 2006-04-24.

[16] Woods, D., *Enterprise Services Architecture*, O'Reilly, Gravenstein, 2003.