Integration of Business Process Management and Complex Event Processing

Dr. Manuel Goetz[1]

[1] iTransparent GmbH, IT Consulting, Bergstraße 5, 90403 Nuremberg, Germany
manuel.goetz@itransparent.de

Abstract. Business Process Management (BPM) and Complex Event Processing (CEP) are two technologies with growing market shares. Additional to their inherent strengths, they also complement well. This paper will present two general scenarios for integrating both technologies together with an example for each scenario. These examples are realized using ActiveVOS as BPM suite and Esper as CEP suite. Both suites are already integrated to a high degree and offer a huge potential.

Keywords: Business Process Management, BPM, Complex Event Processing, CEP, integration

1 Introduction

Business Process Management (BPM) and Complex Event Processing (CEP) are terms for two classes of technologies. Both terms exist for some time already and a certain amount of literature can be found describing them [10][6].

Especially BPM is used extensively in many enterprises in the meanwhile. The goal of BPM is the support and improvement of processes and their execution. Key features are therefore modeling, IT-supported execution and analysis of processes in many industries like financial services or telecommunications. Through modeling a process, a certain structure of this process is defined; bad structures can be identified and corrected which improves the process itself. IT-supported execution of processes exonerates process attendees from time-consuming tasks like process coordination or document search. Additionally, it integrates escalation and alerting mechanisms to react to problematic situations. After execution, Business Intelligence tools can be used to analyze finished processes and data items included. Although in many use cases these features are sufficient to realize all requirements, sometimes a complex analysis option on running processes would be necessary. Using such a feature, complex problematic situations occurring during execution of a process could be identified soon enough to react immediately.

CEP is a technology that provides real-time analysis of events. Events which have to be analyzed can be defined flexibly. This enables CEP to be adapted to nearly every use case and existing IT architecture. In order to analyze events, they are collected in event streams and queries are predefined. During runtime, these queries are evaluated frequently on the streams to detect event patterns. If an event pattern is detected, a predefined reaction has to be executed. The reaction definition can be done in several ways, but especially complex reaction definitions are hard to handle. Programming languages, for instance, offer enough flexibility but are hard to maintain and document. This is a direct consequence of the programming language approach: Reactions defined using a programming language can be programmed but not configured.

In this paper, an integration of BPM and CEP will be presented which will solve both mentioned problems in BPM and CEP. This integration is already realized in tools which will also be introduced. ActiveVOS [1] is a sophisticated BPM suite which offers many features and CEP integration. Esper [5] is the leading open source CEP system which offers several approaches to be connected.

The rest of this paper is structured into several sections. Section 2 introduces BPM shortly and gives an overview on ActiveVOS as BPM suite. Section 3 contains a general CEP introduction and presents Esper as a CEP suite. Section 4 provides a conceptual overview on integration approaches for BPM and CEP. This overview includes two approaches which define BPM or CEP as leading technology. A realization approach for BPM realized with ActiveVOS using CEP for real-time analyzing purposes is presented in Section 5. Section 6 provides a realization for CEP realized with Esper using BPM as reaction definition system.

2 Business Process Management with ActiveVOS

Business Process Management (BPM) is an approach which directs to the improvement of frequently executed processes. Therefore, it defines a method and a structure for IT tools supporting BPM. ActiveVOS is a sophisticated BPM suite which realizes the defined structure and supports BPM methods. As BPM and ActiveVOS are used to present an integration approach with CEP, a general overview on BPM and an introduction to ActiveVOS as BPM suite will be presented in the following.

2.1 Business Process Management

In every company or enterprise several processes are executed many times. One typical process that appears throughout many industries is a claim process, for instance. No matter how much money one invests, there is always a hopefully small amount of claims for a product. As no company wants to lose customers, claims should be answered fast and competent in order to satisfy claim requesters as soon as possible. Additional to this example, there are many other general and more specific processes in nearly all industries. BPM offers the possibility to manage these processes, improve them in structure, quicken their execution and also save money after an initial investment.

In order to achieve those goals, existing processes are modeled in a first step. Modeling of processes unveils inefficient structures and shows potential bottlenecks. Furthermore, responsibilities can be defined for every part of a process which facilitates collaboration. If a process model is created using an IT modeling environment, additional support at execution time can be offered. Therefore, the model is transferred to an IT execution environment. Execution support through IT exonerates process attendees as several tasks can be managed by the execution environment. These tasks include coordination of single process steps or delivery of needed documents when attendees need them. Furthermore, IT execution environments generally integrate human attendees and IT systems. IT systems can be called directly by the execution environment and transparently for human attendees. This quickens a process execution and exonerates human attendees.

As IT support for process execution enables automatic and transparent storing of process execution data, analyses on executed processes are realizable. These analyses may show further inefficiencies during process execution and lead to process model modifications. Following this procedure leads to the so-called BPM lifecycle. Continuous improvement of processes is therefore not just supported in BPM, it is a goal.

2.2 ActiveVOS

ActiveVOS from Active Endpoints is a sophisticated BPM suite that contains all components to realize a complete BPM lifecycle. Additionally, it is completely based on standards which prevent customers from any kind of vendor lock. Created process models are portable regarding modeling as well as execution.

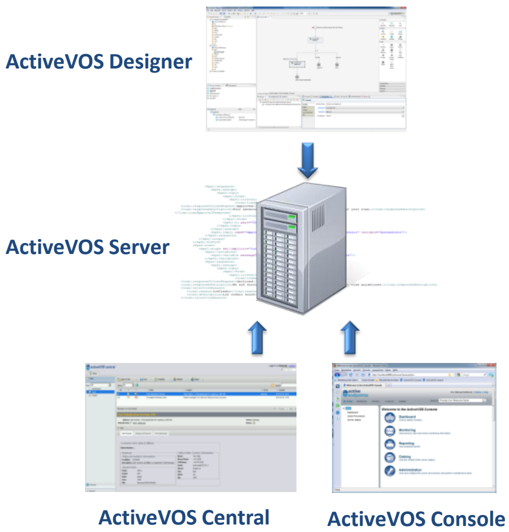

Figure 1 Components of ActiveVOS

ActiveVOS mainly consist of four components which are a Designer, a Server and two user interfaces called Central and Console (Figure 1). The ActiveVOS Designer provides the standard Business Process Model and Notation (BPMN) [9] as modeling language. Usage of a standard modeling language to create process models facilitates discussions on process models as all participants have the same understanding of a specific process model. Consequently, errors resulting from a wrong understanding of a process model's content can be minimized. Furthermore, it facilitates technical exchangeability of process models as process models created using BPMN can be im- and exported using the standardized XML Process Definition Language (XPDL) [11]. XPDL is a file storage format for BPMN process models which enables these models to be viewed and modified in different Designers. In order to execute process models, ActiveVOS Designer provides an on-the-fly transformation from a BPMN model to a model described in the standardized Business Process Execution Language (BPEL) [7]. BPEL processes can be executed by any BPEL server and provide IT support for process execution. As the ActiveVOS Designer provides the transformation from a BPMN process model into an executable form, it also provides the configuration of technical execution details. Nevertheless, these details can also be hidden to focus on process modeling only.

The ActiveVOS Server is a technically mature BPEL server that executes BPEL process models. Process models created in ActiveVOS Designer can be deployed directly to this server, but in principle every BPEL process conforming to the standard is executable. The ActiveVOS Server is more often than not connected to further systems like an LDAP user management [4] or external databases. This is necessary for advanced system interaction during runtime of process models.

Interaction with human attendees during process execution is realized by the AJAX [8] based web application ActiveVOS Central. This graphical process interface provides a user-specific worklist presenting all process steps available for execution. Furthermore, it provides forms to execute single process steps. These forms can also be designed in the ActiveVOS Designer. Like the whole ActiveVOS suite, ActiveVOS Central is also realizing standards for human interaction. These standards are BPEL4People [3] and WS-Human Task [2] which specify protocols and programming interfaces for the interaction of a BPEL server with human attendees. Additionally, reports defined in the ActiveVOS Designer can be integrated easily in ActiveVOS Central which facilitates process analysis.

The final component of the ActiveVOS suite is the ActiveVOS Console which is also an AJAX based web application. The ActiveVOS Console provides an administrator interface for the ActiveVOS server. It allows administrators the access to the server itself, all deployed process models and all running processes. Running processes can even be modified or reset if needed. Although this should not happen frequently, it is a very important feature especially in long running processes and improves efficiency.

3 Complex Event Processing with Esper

Complex Event Processing (CEP) is an approach that enables real-time analysis of events. Opposite to well-known database analyzing techniques, it does not work on a static set of data items but uses event streams instead. This way, the most current data is used as analysis base. Esper is the leading open-source CEP suite which allows very customizable and yet fast event analysis. As it is used as CEP suite in the following sections, it is introduced with its basic features.

3.1 Complex Event Processing

The main goal of CEP is the real-time analysis of events. Because of that goal, it is often compared to databases which structure and analyze data sets. Simplified, CEP can be seen as a database approach turned around. Databases establish a static amount of data and allow definition of queries which are then executed. They analyze the present amount of data and deliver a result. Therefore, the result is always just as up to date as the data basis of the query.

CEP defines event streams as structures. Event streams allow routing of arbitrary incoming events and a goal-oriented analysis. Queries are predefined on one or more event streams. Other systems like databases, for instance, can also be integrated into CEP queries for comparison reasons. As events arrive arbitrarily, a static data basis is not given in CEP. Therefore, the predefined queries are evaluated frequently and search for certain patterns in the event streams. The results can be displayed in frequently updating diagrams or some kind of trigger can be defined on occurrence of a special pattern. As events are analyzed directly after they are delivered to a CEP system, results are generally nearly reflecting a real-time status. Furthermore, this approach can be used to filter important data and define queries for use cases with a huge amount of events or data. Consequently, CEP is also applicable to scenarios where databases cannot manage to store data fast enough without losing some data sets.

In order to depict this approach, CEP realized in Esper is described in the next section.

3.2 Esper

Esper from Espertech is a sophisticated CEP suite which integrates Event Streaming and Event Analysis. It is extremely adaptable to any kind of application and can be used as an integrated part of a software (library) or as a standalone server. Both approaches provide the same core functionality.

In Esper, events and the data included can be defined. A common POJO can be an event, for instance, and therefore be added to event streams. Event streams are used to structure the potentially huge number of incoming events and apply queries more goal-oriented. For structuring, filters are used which can drop events or move them to another stream. Applying filters, a stream architecture can be developed that facilitates analysis. Consequently, there can be created analysis streams internally that realize another structure than the ingoing and outgoing streams.

Events in the streams can be analyzed with queries (Figure 2). Defined queries supervise events in selected streams and search for event patterns. The language queries are defined with is based on the Structured Query Language (SQL). This makes Esper easy to use for all people that already worked with database queries and reduces learning time. Although it is based on SQL, the Esper query language provides some new key words. This is necessary because of the stream analysis nature of Esper. A keyword for a time window, for instance, defines a time interval for a pattern appearance. As databases have a potentially huge, but nevertheless limited amount of data in their storage, queries do not have to be limited by a time interval. But a stream is endless. If a query for a pattern is not limited by a time window, all events ever arriving are taken into account. This is intended just in a few use cases. Mostly, a pattern in an event stream has to appear in a certain time interval; otherwise, the events are more often than not unrelated und do not have any information value.

Results of queries can be handled in two ways. One way is to define so-called Eventlets. Eventlets are continually updating displays showing the current results of a query, variable or engine metric via push or pull of data. They can be integrated into web applications and provide an overview for users. Data push and pull is facilitated by a data distribution service that optimizes fan-out and manages interest lists. For supervising reasons, also some kind of reaction can be defined if a special pattern is found. For instance, an alarm can be fired if this pattern indicates a dangerous status of a system. The kind of reaction definition is not further specified in Esper. Principally, any programming language or mechanism can be used and connected via several mechanisms.

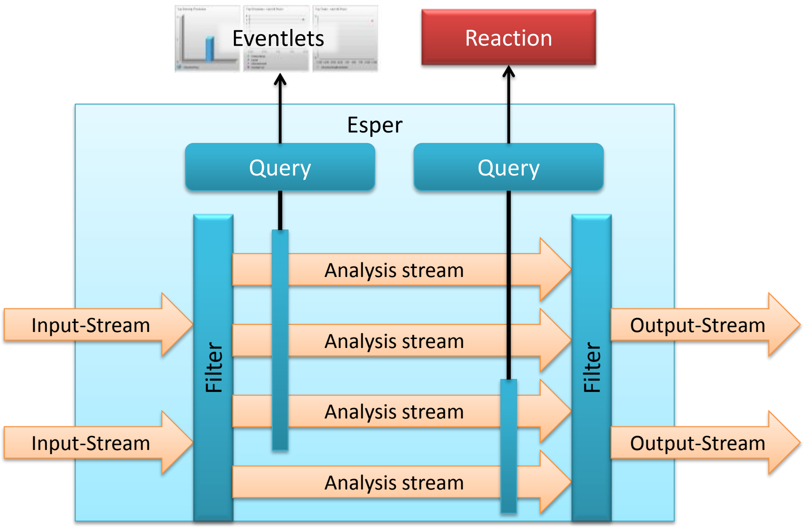

Figure 2 Overview on Esper

Esper is available in a Community and an Enterprise Edition. While the Community Edition already provides a complete CEP suite, the Enterprise comes with additional features like JMX runtime management, enhanced scalability and different support offerings.

4 General integration approaches

Generally, there are currently two integration approaches for BPM and CEP which will both be described throughout this section. A first approach is to define BPM related events in a BPM suite and deliver these events to a CEP suite. Consequently, the CEP suite supervises the BPM suite at runtime of processes. A second approach is to use a BPM suite for the definition of reactions in a CEP suite. The CEP suite itself analyzes arbitrary external events and calls the BPM suite if a pattern is detected.

4.1 Using CEP in BPM

In this approach, CEP is added to a BPM suite and used to analyze BPM events. These events as well as the queries are defined at modeling time. The stream architecture in the CEP system can be automatically generated based on the given definitions or modified manually. At runtime, defined events are created by running processes and delivered to an input stream automatically. Results of the queries can be observed via diagrams or reactions can be defined. This scenario is depicted in Figure 3.

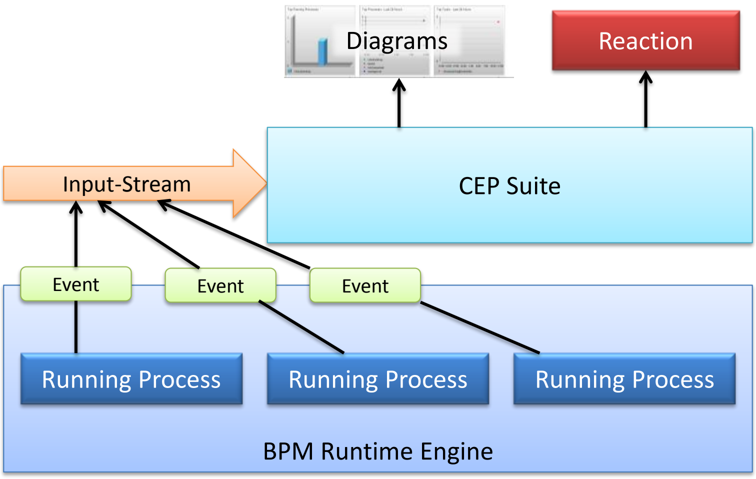

Figure 3 CEP is used to analyze BPM events

Integrating CEP into BPM this way, the BPM events can be analyzed in real-time. Consequently, running processes can be analyzed in real-time in respect to their processing behavior. Additionally, also complex queries including more than one or even all running processes can be realized. This makes also situations dependent on different processes analyzable. Trends can be found very fast, for instance, when all processes are included. As a result, interference options are given if unwished behavior takes place. Current trends can be controlled or redirected as they are recognized while they establish and not afterwards. Another example is that pure system interactions can be controlled directly. These interactions happen too fast for human observers, but coupling these interactions with events allows for continuous observing of the results without diving into the details.

The overall message is that with usage of CEP in BPM, knowledge is not derived after process executions, but during execution. This is an absolute necessity in every use case that needs fast and direct interference dependent on special constellations in a BPM server at runtime.

4.2 Using BPM to define CEP reactions

This approach assumes that a CEP suite is available and analyzes any kind of external events. These events are produced by an external software or hardware which is not further specified here. In a CEP suite, streams and queries are defined to analyze the incoming events. The intention is to search for patterns that imply a trend or situation that needs to be reacted to. The reactions are considered to be complex and include calls to different systems or humans. Therefore, the reactions are defined as processes in a BPM suite and pattern detection leads to a process start and execution (Figure 4). The process executes all actions needed. It can call other systems, send notifications via e-mail or just add process steps to a worklist, for instance.

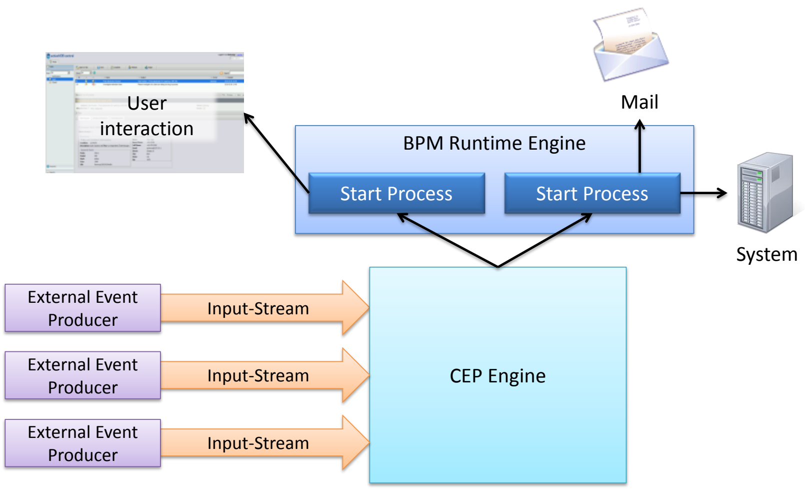

Figure 4 CEP using BPM for reaction execution

Using processes to define and execute reaction definitions provides several advantages. Processes have a graphical presentation which enhances understandability. Therefore, the reactions are easy to maintain. Furthermore, they provide a high degree of flexibility. Processes can be modified or adapted to new environments very easily. Additionally, they can be substituted by other processes in a BPM suite without effects in the CEP suite if both systems are loosely coupled. This loose coupling also enables a CEP suite to be integrated in any other system without affecting the BPM suite. All these features show that processes used to define reaction definitions in CEP can be very useful.

5 BPM with ActiveVOS using CEP

This section describes a realization of the approach introduced in Section 4.1. This realization presents the integration of the CEP suite Esper into the BPM suite ActiveVOS. In order to demonstrate the main features of this integration, an example process called Loan Application is used.

5.1 Loan Application example

For the demonstration of Esper in ActiveVOS, the Loan Application example is used (Figure 5). This Loan Application starts when a customer sends a loan request to a bank.

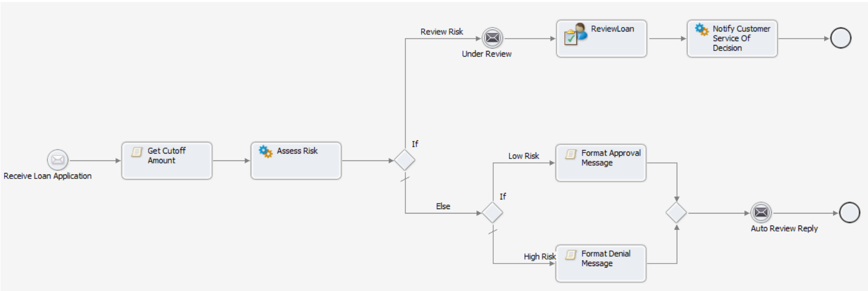

Figure 5 Loan Application example

This loan request contains some data concerning the request. Based on this data, the risk is assessed automatically first. If an unambiguous decision can be derived based on some guidelines, then the request is approved or denied automatically. Otherwise, an expert has to review the loan request and notify the customer of his decision.

5.2 Defining events

In order to use CEP in this example, the events have to be defined in a first step. These events are the BPM events and relate to situations that take place during process execution. ActiveVOS Designer provides a special wizard to support modelers in defining events.

The event defined in Figure 6, for instance, is fired every time a loan request arrives and is transferred successfully to the system. This event is named LoanRequest and is associated to a Location Path which represents an element in the process model. In this example, the event is associated to “Receive Loan Application”. Additionally, a status of the event can be defined which must be valid before the event fires. The Business Properties define the data that will be included in the event. This data has to be available in the process before.

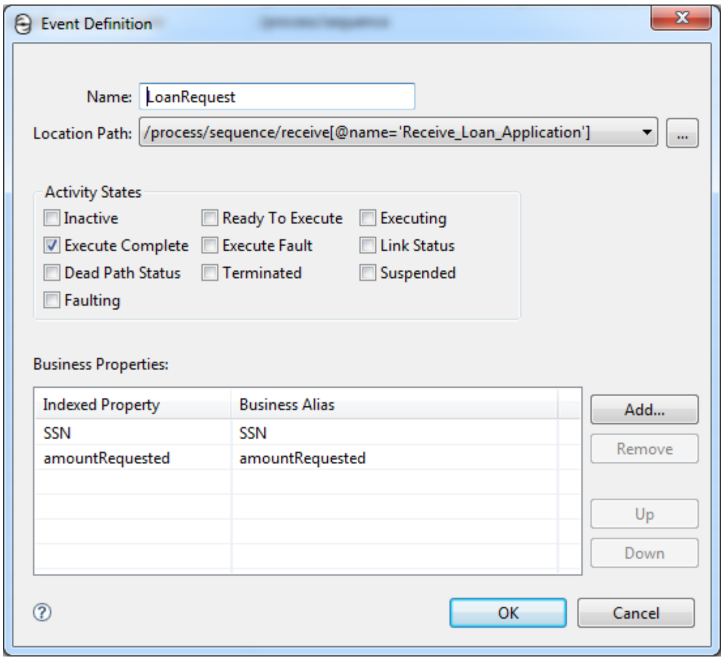

Figure 6 LoanRequest event defined in ActiveVOS wizard

5.3 Defining Queries

After definition of the events, the queries working on these events need to be defined.

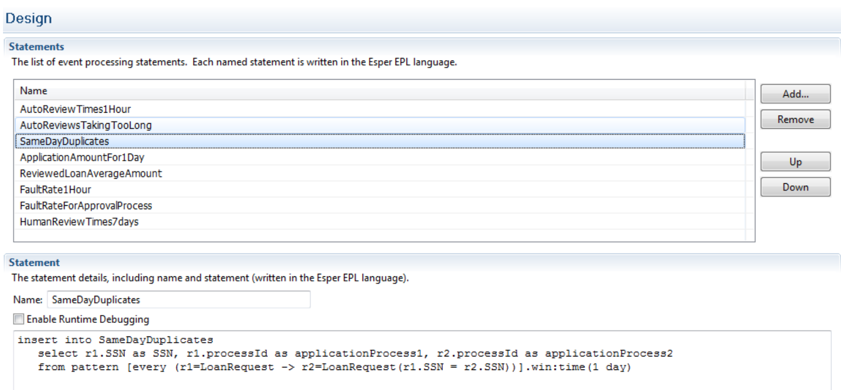

Figure 7 Query definition in ActiveVOS

Again, ActiveVOS provides a wizard (Figure 7) which supports the definition of Esper queries. The query defined eliminates duplicate loan requests, for instance. It compares all loan requests arriving at one day based on the SSN of the requesting person. If a potential duplicate arrives, it is added to another stream and can be saved there. The process working on a duplicate can be stopped. Consequently, wasting of resources is reduced.

At runtime, this query is evaluated constantly in the background and transparent for the users. They do not need to care about the integrated CEP suite if they do not want to.

5.4 Summary

ActiveVOS as BPM suite allows event definition based on modeled processes. Wizards facilitate connection of events and a process model. Furthermore, queries can be defined based on the created events. As a result, real-time event processing for running processes is enabled. This means, that additional to completed processes running processes can be analyzed. This can be applied at use cases where fast reactions on exceptions are needed, for instance. Although the presented example provided just a simple process and a simple query, far more complex use cases can be realized with this architecture.

6 CEP with Esper using BPM

This section introduces a realization scenario for the approach presented in Section 4.2. Therefore, Esper as CEP suite will be used to analyze events produced in an external system. If a predefined pattern occurs, a process modeled in ActiveVOS as BPM suite is started and executes a reaction. Again, this functionality will be described using an example.

6.1 Network example

In this example, there exists a network that routes data packages through a number of Nodes. The Nodes are connected with several cables. In order to keep it simple, the technical details are ignored as this network is just used an event source in this example. Events can be fired by Nodes. A Node fires an event if more data packages than can be routed arrive in a short time interval. The event fired is called Overload Event and contains a Node ID which identifies the overloaded Node.

The Overload Events are delivered to the Esper suite (Figure 8). In Esper, the events are analyzed according to some conditions. Principally, it is not a problem if a Node is overloaded for a short time. Data packages are not lost but rerouted to another Node. Nevertheless, if some Overload Events are fired by one Node in a short time interval, there is a reaction needed as this indicates a Node defect or any kind of failure.

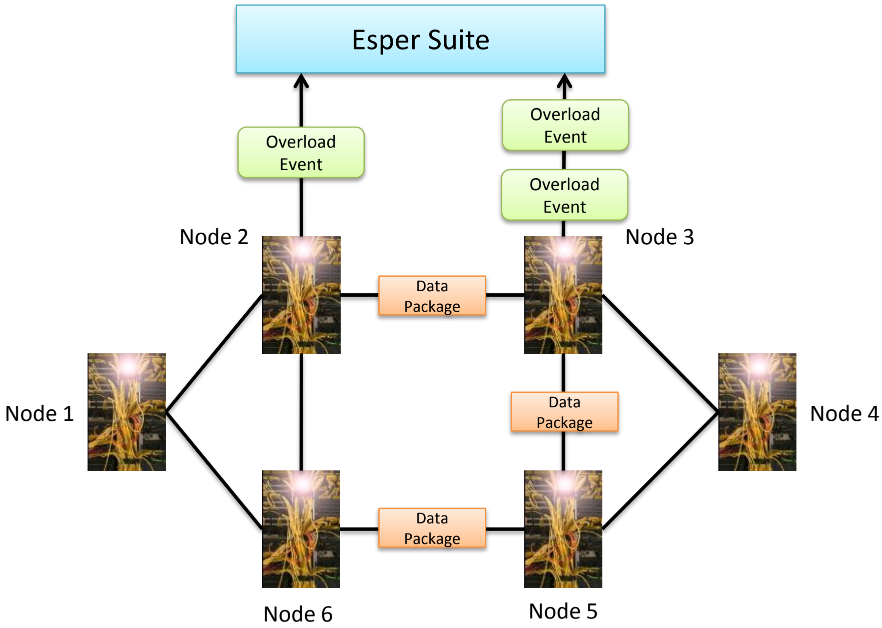

Figure 8 Network example with Esper

6.2 Defining Queries and Reactions

Following the conditions mentioned in the Network example introduction in Section 6.1, Overload Events need to be analyzed according to their Node ID and the time interval they arrive in. Whenever more than 3 Overload Events produced by one Node arrive in less than 5 seconds, a reaction needs to be started. The query expressing these conditions looks like this:

```
select NodeID
from OverloadEvent.win:time(5 sec)
group by NodeID
having count(*) > 3
```

Whenever this pattern is detected in the event stream, a web service call to the ActiveVOS server starts a process (Figure 9).

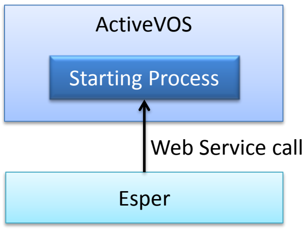

Figure 9 Esper calling ActiveVOS with a Web Service

This process defines the reaction to the Overload Events (Figure 10). First, an automatic repair routine is started by the process that tries to solve the problem with standard solution approaches. If one of these approaches solves the problem, it is just documented and the process is finished. Otherwise, this problem has to be solved manually which is defined in a subprocess. This subproces includes several notifications and user interactions. Again, after solution of the problem all actions are documented and the process is finished.

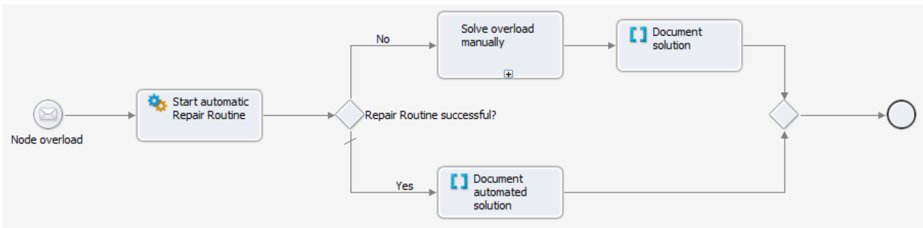

Figure 10 Reaction to pattern detection

6.3 Summary

Esper mainly focuses on Event Streaming, Filtering and Analyses and has excellent solution approaches for these problems. ActiveVOS is used to define the reactions triggered by the detection of event patterns. This improves flexibility and manageability of reaction definitions and provides all other advantages that come with a BPM suite.

For Esper, starting a process is just a call to an external system. Therefore, integration using loose coupling is easy to establish between Esper and ActiveVOS and provides advantages of and for both suites.

7 Conclusion

This paper introduced BPM using ActiveVOS and CEP using Esper. Additionally, it presented two integration approaches and demonstrated both approaches with the help of examples.

Summarized it can be said, that an integration of both technologies provides several advantages. They complement well and integration can be done in both directions as CEP can be used to analyze BPM events and BPM processes can be used to define CEP reactions. ActiveVOS already provides a strong integration of Esper and provides wizards for event and query definitions. At runtime, queries are evaluated transparently. Esper can be extended easily to call ActiveVOS processes through usage of standard web service calls.

As both technologies benefit from a tight integration, scenarios using both technologies will multiply in number in the future.

References

1. Active Endpoints: ActiveVOS Product Info. URL: http://www.activevos.com/products-productinfo.php.
2. Agrawal, A.; Amend, M.; Das, M.; Ford, M.; Keller, C.; Kloppmann, M.; König, D.; Leymann, F.; Müller, R.; Pfau, G.; Plösser, K.; Rangaswamy, O.; Rickayzen, A.; Rowley, M.; Schmidt, P.; Trickovic, I.; Yiu, A.; Zeller, M.: Web Services Human Task (WS.HumanTask), Version 1.0. URL: http://xml.coverpages.org/WS-HumanTask-V1-200706.pdf.
3. Agrawal, A.; Amend, M.; Das, M.; Ford, M.; Keller, C.; Kloppmann, M.; König, D.; Leymann, F.; Müller, R.; Pfau, G.; Plösser, K.; Rangaswamy, O.; Rickayzen, A.; Rowley, M.; Schmidt, P.; Trickovic, I.; Yiu, A.; Zeller, M.: WS-BPEL Extension for People (BPEL4People), Version 1.0. URL: http://xml.coverpages.org/BPEL4People-V1-200706.pdf.
4. Carter, G.: LDAP System Administration. O'Reilly, Sebastopol (2003).
5. EsperTech: Esper: Event Processing for Java. URL: http://www.espertech.com/products/esper.php.
6. Etzion, O.; Niblett, P.: Event Processing in Action. Manning Publications (2010).
7. OASIS: Web Services Business Process Execution Language Version 2.0. URL: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf.
8. Holdener III, A.: AJAX – The Definitive Guide. O'Reilly, Sebastopol (2008).
9. Object Management Group: Business Process Model and Notation – Version 1.2. URL: http://www.omg.org/spec/BPMN/1.2.
10. O'Connell, J.; Pyke, J.; Whitehead, R.: Mastering Your Organization's Processes – A Plaing Guide to Business Process Management. Cambridge University Press, New York (2006)
11. Workflow Management Coalition: Process Definition Interface – XML Process Definition Language – Version 2.1a. URL: http://www.wfmc.org/xpdl.html.