eChallenges e-2009 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2009
ISBN: 978-1-905824-13-7

Event-Driven Service Chain Monitoring for the Internet of Services

Krešimir VIDAČKOVIĆ, Holger KETT, Thomas RENNER
Fraunhofer IAO, Nobelstraße 12, 70569 Stuttgart, Germany
Tel: +49 711 970-5120, Fax: +49 711 970-5111,
Email: {Kresimir.Vidackovic / Holger.Kett / Thomas.Renner}@iao.fraunhofer.de

Abstract: In the Internet of Services, where web services of different providers may be composed to build decentralized and cross-company service chains, and traded over web platforms, the ability of monitoring the over-all Quality of Service becomes crucial for the service chain provider. This paper introduces a concept which envisages monitoring functionality being offered as a service by a specialized monitoring service provider and used by service chain providers without the enforcement to change their current IT infrastructure. The concept is based on the technology of Complex Event Processing and covers modeling as well as execution of monitoring with a focus on design time. The modeling of monitoring logic is accomplished in an intrusive manner on top of the business process model which is supported by an appropriate software prototype.

1. Introduction

In the context of the Internet of Services, approaches emerge where services are electronically offered and used over web platforms. These web platforms may consist of an electronic marketplace for finding and trading services as well as providing useful value-added services, such as payment or hosting. There is also the alternative that service providers host their services themselves, publish them for discovery on the web platform and execute them on demand in their own runtime environments [1]. As web services feature standardized interfaces, several services of different providers can be composed to build complex service chains which are considered as Service Ecosystems [2]. A provider of such a cross-company service chain is termed service chain provider and each web service within the service chain is referred as sub service in this paper.

The over-all Quality of Service (QoS) of these service chains depends significantly on the quality of each sub service [3], so monitoring of important sub services becomes crucial for service chain providers who are responsible for the efficient operation of the service chain. It may even be necessary for them to receive relevant status information for monitoring purposes from inside the capsulated sub services of other providers. This is for example the case in a scenario from the insurance domain where the claims settlement process of an insurance company is composed of several sub services of different providers forming a cross-company service chain. The process execution is outsourced entirely to an independent service provider who in this context appears as the service chain provider. One of the included sub services is a cost estimate service which in turn, depending on the handled claim, in some cases may be based upon a residual value computing service of another provider. Although this information is encapsulated within the cost estimate service and therefore is not directly accessible for the service chain provider, it is still necessary for him to be informed about the status of this important sub service in real-time. Only in that way he is able to get proper insight into the entire service chain offered to the insurance company.

Real-time monitoring of cross-company service chains tends to be a highly challenging task demanding specific know-how and an appropriate IT infrastructure. Particularly small and medium sized companies acting as service chain providers could therefore profit from monitoring functionality which is accessible as a service over a web platform offered by a specialized monitoring service provider. This has the advantage for them to leave the complex monitoring task to an expert and to maintain their current IT infrastructure by merely integrating external monitoring functionality instead of implementing it on their own. Besides, such an approach follows the basic idea of the Internet of Services where specialized service providers offer their services over web platforms to potential consumers [1]. Figure 1 demonstrates such a monitoring scenario.

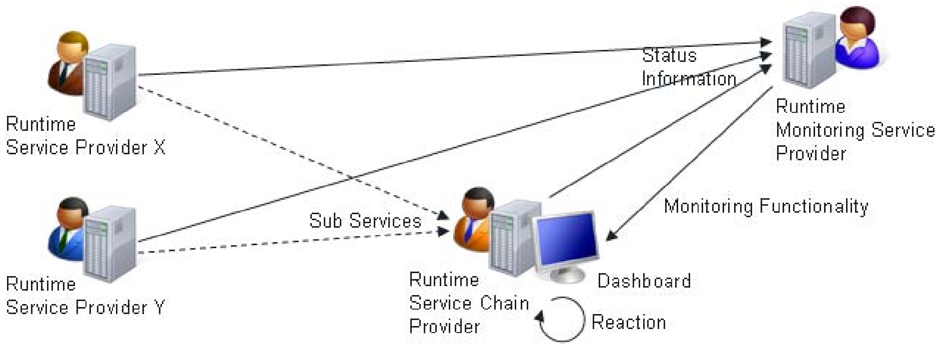

Figure 1: Monitoring Scenario with a Specialized Monitoring Service Provider

According to figure 1, the service chain provider as well as service providers, who deliver sub services of the service chain, send status information to a monitoring service provider during process execution. The monitoring service provider processes the received information in real-time and offers monitoring functionality to the service chain provider where it is visualized in the form of a dashboard and may cause real-time reactions within the execution process. Such a scenario has the following requirements:

- Real-time monitoring and reactivity functionality for distributed service chains
- Monitoring offered as a service by a specialized monitoring service provider and used by service chain providers, so that the execution of the monitored service chain and the monitoring service itself run on separate environments
- Feasibility of sending process-oriented status information from within the service chain and important sub services to the monitoring service provider

Currently, there is no monitoring approach available to fulfill these requirements (see chapter 3). Therefore, an integrated concept of Event-Driven Service Chain Monitoring for the Internet of Services is introduced in this paper.

The concept and an associated software prototype are developed in the research project Theseus/TEXO funded by the German Federal Ministry of Economy and Technology (http://theseus-programm.de/en-us/theseus-application-scenarios/texo/default.aspx).

2. Objectives

The main objective of this paper is to introduce an integrated concept of Event-Driven Service Chain Monitoring for the Internet of Services meeting the requirements mentioned above. A comprehensive solution for this issue from the perspective of a service chain provider comprises modeling of monitoring logic on top of decentralized cross-company process models, transfer to the runtime of the monitoring service provider and execution of monitoring and reactivity functionality during process execution. In this paper the entire concept of Event-Driven Service Chain Monitoring from modeling to execution is explained with a close focus on design time.

Additionally, the implemented software prototype is depicted supporting service chain providers in modeling monitoring logic on top of their process models which are represented in the Business Process Execution Language (BPEL).

3. Methodology and Related Work

In order to develop a concept that fulfills our requirements, a literature research on existing activities in the field of monitoring decentralized service chains was conducted.

Non-intrusive monitoring approaches, as presented e.g. in [4], [5] or [6], use events thrown due to regular state changes during process execution which are checked against defined monitoring rules and may lead to specified actions. Most of the current BPEL engines support an automatic, internal generation of events during process execution, e.g. at the start and the end of activities or at value changes of variables, which could be used in our context as well by forwarding the events to the monitoring service provider. Non-intrusive monitoring is advantageous in terms of performance and clean separation of process and monitoring logic. However, it does not permit sending additional, process-specific information from capsulated sub services or the service chain to the monitoring service provider as required in our scenario.

An intrusive approach for monitoring service chains is taken in [7] where an assertion language for specifying functional and non-functional properties in BPEL processes is introduced. These properties are modeled on top of the BPEL model and weaved into the process at runtime using aspect-oriented techniques. The concept is supported by a modeling environment and a runtime framework which is unified in another work to integrate different monitoring approaches [3]. A downside of this work is the enforcement to modify the source code of the used BPEL execution engine in order to extend it with monitoring functionality. As our scenario in the context of the Internet of Services envisages a specialized monitoring service provider who is not in charge of the different BPEL execution engines run by the service chain providers consuming his monitoring service, this approach will not work for our requirements.

However, our concept is built on an intrusive monitoring approach in order to enable the sending of process-specific status information from important sub services and the service chain to the monitoring service provider as required in our scenario.

In addition, we have concluded that real-time monitoring of distributed environments encountered in the Internet of Services is most suitably accomplished in an event-based manner which means in response to incoming events representing any meaningful happening from an internal or external source [8]. The usage of asynchronously occurring events as the central design concept has a lot of benefits, like the good support of loosely coupling between event producers and event consumers, the processing of different kinds of events from distributed sources, and real-time awareness [9]. The emerging technology of Complex Event Processing (CEP), primarily introduced in [10], exhibits the ability to detect relations of events from an event cloud by applying event patterns which contain event templates, relational operators and variables in order to match sets of related events [8] as well as to initiate predefined reactions in real-time. These qualities led to our decision to use CEP for the monitoring of decentralized service chains, since the monitoring service must handle a multitude of events from several service chain providers and apply process-specific monitoring rules in real-time.

In our further research, CEP techniques have been analyzed regarding the applicability for our demands, especially in terms of modeling. We identified two main shortcomings:

- Different event processing languages (EPL) for event-based monitoring logic
- Lack of process-oriented modeling functionality

Well-known CEP engines generally provide the ability of modeling event patterns and rules expressed in an engine-specific Event Processing Language (EPL), whereas no standard for an EPL is currently available [11]. Additionally, these rules are solely based on the previously declared events and do not have a direct link to business processes.

Our concept of Event-Driven Service Chain Monitoring for the Internet of Services combines CEP techniques with intrusive monitoring by generating relevant events for monitoring explicitly from within the process. Thus, the required status information from capsulated sub services and the service chain can be allocated to the monitoring service provider in the form of events. The corresponding monitoring logic is modeled by the service chain provider and expressed in the specific EPL of the CEP engine used by the monitoring service provider.

The concept has been implemented in a software prototype supporting the modeling of monitoring logic on top of the BPEL process model. Finally, the results have been verified with experts from the process optimization domain and a real-world scenario for the usage of cross-company service chains has been built within the insurance domain where the demand for monitoring functionality across the whole business processes has been identified.

4. Technology Description

During the execution of cross-company business processes, events can be used to identify process-specific situations from the perspective of the service chain provider. According to our concept, a specialized monitoring service provider receives these events from the process and handles them using CEP techniques. Figure 2 illustrates the envisaged infrastructure for Event-Driven Service Chain Monitoring.

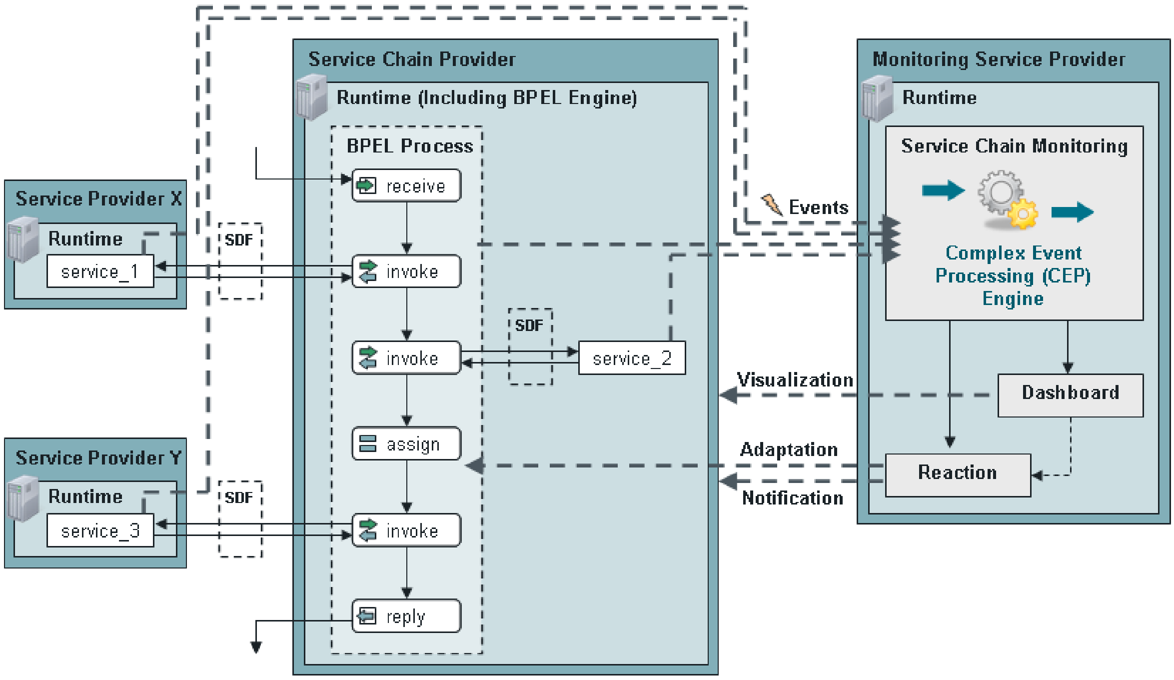

Figure 2: Infrastructure for Event-Driven Service Chain Monitoring

With CEP, passing events are analyzed continuously by performing event queries, whereas operations like data extraction, event composition, detection of temporal, causal or spatial relationships and event accumulation are performed. In that way, monitoring rules can be applied on incoming events which include deductive and reactive rules [11]. With deductive rules, simple (low-level) events are composed to form composite (high-level) events which could e.g. represent important key performance indicators (KPIs) visualized in a monitoring dashboard. Reactive rules are reaching even further by initiating predefined actions in response to detected event patterns. Exemplary reactions could be the dynamic adaptation of the service chain in order to increase the over-all QoS or the sending of notifications in the form of alert messages in case of critical situations. By applying such kinds of monitoring rules, the monitoring service provider is able to offer a dashboard and a reaction component to the service chain provider.

The events causing these monitoring rules to be triggered are generated intrusively from within the process. For this purpose, the monitoring service provider offers a generic event firing service which is invoked with the corresponding event data as input whenever a certain event is required during process execution. Thus, the first step in modeling monitoring logic on top of the BPEL process model at design time is to place such event generators in the form of invocations of the generic event firing service, whereas they could be prior to, in parallel or after a certain BPEL activity. Figure 3 shows an exemplary BPEL process instrumented with three event generators where the first one is executed prior to service 1, the second one in parallel to service 2 and the third one subsequent to service 3.

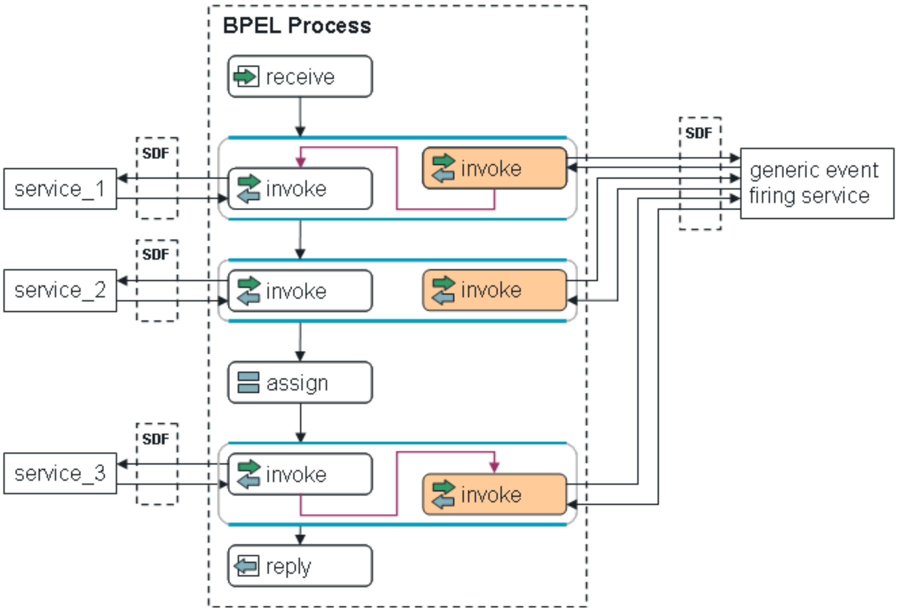

Figure 3: Exemplary BPEL process instrumented with event generators

These events from the process are then used to specify the above-mentioned monitoring rules. From the perspective of a service chain provider, monitoring rules are categorized in the following groups: temporal, quantitative, qualitative and financial information. Temporal information, e.g. average lead times of particular sub services or process-specific time intervals, as well as quantitative information, e.g. the invocation quantity of the service chain, are gathered in a rather easy way by processing the timestamps of the generated events and by counting respectively. In contrast, qualitative information, e.g. the average ratings of sub services, and financial information, e.g. the average costs of sub services, need additional information which may be obtained from the web platform where the sub service is published.

This can be accomplished by using the recently introduced Service Description Framework (SDF) which is also developed within the research project Theseus/TEXO for the Internet of Services. In figures 2 and 3, the interfaces to the considered services are already specified using the SDF. In contrast to the standardized Web Service Description Language (WSDL) which reveals only technical details, SDF covers different facets of a service, namely General Information, Functionality, Service Interaction, Data, Business, Legal, Quality of Service, Security and Trust as well as Rating, so that services in fact become tradable on web platforms [12]. In that way, if SDF is used for service descriptions, more information can be accessed for monitoring. In order to get e.g. the average rating result of a sub service, the SDF Rating attributes Community Feedback and Expert Test Rating can be extracted. The average cost of a sub service is e.g. derived from the SDF Business attributes Price and Discount. In these cases, the generated events from the process have to include the SDF attributes of the sub services they are related to.

Subsequent to the modeling of the monitoring logic on top of the business process logic by the service chain provider at design time, the models have to be transferred. At the transition from design level to runtime level, two transfers have to be accomplished:

- The extended BPEL process model instrumented with invocations of the generic event firing service at important process steps is transferred to the runtime environment of the service chain provider where it is executed inside a BPEL engine.
- The monitoring model is transferred to the runtime environment of the monitoring service provider where the comprised deductive and reactive rules on the generated events are loaded into the CEP engine ready for execution.

Figure 4 illustrates the transition from design level to runtime level with the relevant components and the required transfers between them.

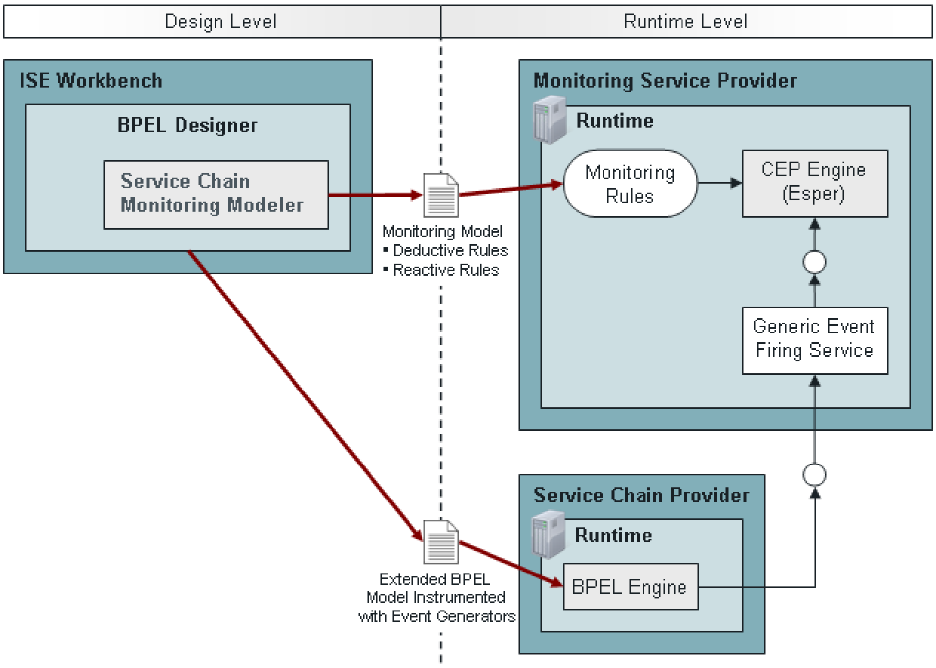

Figure 4: Transition from Design Level to Runtime Level with Required Transfers

The components in figure 4 on design level correspond to the developed software prototype for modeling Event-Driven Service Chain Monitoring. Their functionalities are explained in the following chapter.

5. Developments

The research project Theseus/TEXO introduces the Integrated Service Engineering (ISE) Methodology and ISE Workbench which supports the development of a service by provisioning a set of appropriate models. The ISE Workbench is based on the Eclipse-Modeling Framework (EMF) [13].

On the basis of the Eclipse BPEL Project (http://www.eclipse.org/bpel/index.php) which is included as a plug-in into the ISE Workbench, the Service Chain Monitoring Modeler is developed as a software prototype to enable service chain providers in modeling monitoring logic on top of the BPEL process model. This software tool supports both deductive and reactive rules on events generated during process execution.

The allocation of low-level events to KPIs is accomplished by a hybrid approach, where primarily the generation of relevant events is modeled by placing the invocation of the generic event firing service graphically at the desired position in the service chain. Depending on the process-specific application, this can be before, in parallel or after a certain BPEL activity (see figure 3). The available event types are derived from important KPIs from the perspective of a service chain provider. In order to increase usability, relevant KPIs are ready-made through an editable configuration file, whereas they are categorized in the already mentioned groups: temporal, financial, quantitative and qualitative information. Some of these KPIs require the usage of the Service Description Framework (SDF) as interface for the integrated sub services, others can directly be calculated during process execution. The previously set events are then allocated to the KPIs using predefined metrics with user interface integration. Such deductive rules would be sufficient for the usage with a monitoring dashboard visualizing meaningful information for the service chain provider.

In order to use reactive rules as well, actions can be defined that will be initiated after detecting a corresponding event pattern. From the modeling point of view, these actions are also events sent to a certain component that will actually perform them. Therefore, in our Service Chain Monitoring Modeler actions are modeled as invocations of the generic event firing service with the corresponding input attributes.

Since at runtime the open source CEP engine Esper (http://esper.codehaus.org) is used, the monitoring rules are modeled in the Esper-specific EPL which follows a syntax similar to the standardized and well-known Structured Query Language (SQL).

6. Results and Business Benefits

The results of our research are the illustration of an integrated concept for the provision of monitoring functionality by a specialized monitoring service provider which can be consumed by different service chain providers in the Internet of Services. The concept is exposed by a software prototype allowing service chain providers to model process-oriented monitoring logic on top of the BPEL process model.

Event-Driven Service Chain Monitoring increases the transparency within a decentralized service chain. The modeling of cross-company business processes and the required monitoring logic becomes a more integrative activity, since the monitoring model is linked directly to the important process steps.

By consuming Event-Driven Service Chain Monitoring, service chains providers are enabled to observe the QoS of the whole service chain and each sub service as well as to initiate real-time reactions to certain situations in order to optimize their cross-company business processes. By integrating external monitoring functionality from a specialized monitoring service provider, the current IT infrastructure can be maintained.

7. Conclusions and Future Work

Event-Driven Service Chain Monitoring as a concept facilitates monitoring functionality for cross-company business processes from a service provider's perspective, without forcing him to change his IT infrastructure, because the execution of monitoring features is accomplished by a specialized monitoring service provider and consumed as a service. This complies with the basic idea of the Internet of Services. The concept offers a flexible and process-oriented solution with real-time awareness by using the emerging technology of Complex Event Processing (CEP) and covers both design time and runtime.

The modeling of monitoring logic at design time is left to the service chain providers themselves. In that way, they are enabled to apply process-specific monitoring rules in order to monitor the Quality of Service (QoS over the whole service chain and each sub service. Not only deductive rules, but also reactive rules with predefined actions in response to detected event patterns are possible. Modeling is done in an intrusive manner on top of the BPEL process model with a direct link to the important process steps and is supported by a software prototype. Subsequently, the monitoring logic is transferred to the monitoring service provider for execution at runtime.

Our future work will be focused on runtime providing a configurable monitoring dashboard easy to integrate into the existing enterprise portal. The visualized KPIs will automatically be read out from the deductive monitoring rules and illustrated in a suitable way with the ability to be customized.

In addition, further research will be conducted on the topic of automating the modeling of monitoring logic. In this context, the BPEL process model will be analyzed and useful monitoring rules will be applied automatically if desired as well as the instrumentation of the BPEL model with the corresponding events. In that way, some supplementary out-of-the-box monitoring functionality can be offered by the monitoring service provider apart from the provision of flexible and process-specific monitoring features.

Acknowledgement

The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference "01MQ07012". The authors take the responsibility for the contents.

References

[1] Janiesch, C., Ruggaber, R. and Sure, Y. (2008). Eine Infrastruktur für das Internet der Dienste. *HMD - Praxis der Wirtschaftsinformatik*, 44(261), 71-79.

[2] Barros, A. P. and Dumas M. (2006). The Rise of Web Service Ecosystems. *IT Professional*, 8(5), 31-37.

[3] Baresi, L., Guinea, S. and Pasquale, L. (2008). Towards a Unified Framework for the Monitoring and Recovery of BPEL Processes. In *Proceedings of the Workshop on Testing, Analysis and Verification of Web Software, TAVWEB 2008* (pp. 15-19). New York: ACM Press.

[4] Spanoudakis, G. and Mahbub, K. (2006). Non-Intrusive Monitoring of Service Based Systems. *International Journal of Cooperative Information Systems*, 15(3), 325-358.

[5] Moser, O., Rosenberg, F. and Dustdar, S. (2008). Non-Intrusive Monitoring and Service Adaptation for WS-BPEL. In *Proceedings of the 17th international conference on World Wide Web, International World Wide Web Conference* (pp. 815-824). New York: ACM Press.

[6] Barbon, F., Traverso, P., Pistore, M. and Trainotti, M. (2006). Run-Time Monitoring of Instances and Classes of Web Service Compositions. In *Proceedings of the IEEE International Conference on Web Services (ICWS '06)*, pp. 63-71.

[7] Baresi, L., Bianculli, D., Ghezzi, C., Guinea, S. and Spoletini, P. (2007). Validation of web service compositions. *IET Software*, 1(6), 219-232.

[8] Luckham, D. and Schulte, R. (2008). *Event Processing Glossary - Version 1.1*. Retrieved February 19th, 2009, from epts - Event Processing Technical Society Web site: http://www.epts.com/component/option,com_docman/task,doc_download/gid,66/Itemid,84/.

[9] Mühl, G., Fiege, L. and Pietzuch, P. (2006). *Distributed Event-based Systems*. Berlin and Heidelberg: Springer.

[10] Luckham, D. (2002). *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Boston and London: Addison-Wesley.

[11] Eckert, M. and Bry, F. (2009). Complex Event Processing (CEP). *Informatik-Spektrum*, 32(2), 163-167.

[12] Scheithauer, G. and Winkler, M. (2008). A Service Description Framework for Service Ecosystems. In *Bamberger Beiträge zur Wirtschaftsinformatik und Angewandten Informatik Nr. 78*. Bamberg University.

[13] Kett, H., Voigt, K., Scheithauer, G. and Cardoso, J. (2008). Service Engineering in Business Ecosystems. In Ganz, W., Kicherer, F. and Schletz, A. (Eds.), *Proceedings of the XVIII International RESER Conference* (pp. 1-22), Stuttgart: Fraunhofer IRB.