ASPECT-MONITOR
An Aspect-based Approach to WS-contract Monitoring

Mario Freitas da Silva, Itana Maria de Souza Gimenes
Departamento de Informática, Universidade Estadual de Maringá, Av. Colombo, 5.790, Maringá, PR, Brazil

Marcelo Fantinato
Sistemas de Informação, Universidade de São Paulo, Rua Arlindo Béttio, 1.000, São Paulo, SP, Brazil

Maria Beatriz Felgar de Toledo
Instituto de Computação, Universidade Estadual de Campinas, Av. Albert Einstein, 1.251, Campinas, SP, Brazil

Alessandro Fabricio Garcia
Depto. de Informática, Pont. Univ. Católica do Rio de Janeiro, R. Marquês de São Vicente, 225, Rio de Janeiro, RJ, Brazil

Keywords: Electronic Contracts, Web services, Quality of Service, Aspects.

Abstract: Contract monitoring is carried out to ensure the Quality of Services (QoS) attributes and levels specified in an electronic contract throughout a business process enactment. This paper proposes an approach to improve QoS monitoring based on the aspect-oriented paradigm. Monitoring concerns are encapsulated into aspects to be executed when specific process points are reached. Differently from other approaches, the proposed solution requires no instrumentation, uses Web services standards, and provides an integrated infrastructure for dealing with contract establishment and monitoring. Moreover, a Business Process Management Execution Environment is designed to automatically support the interaction between customer, provider and monitor organizations.

1 INTRODUCTION

Current business competition requires that companies act quickly not to lose partnerships. The internet, Business Process Management Systems (BPMSs) (Weske, 2007) and Service-oriented Computing (SOC) (Papazoglou *et al*., 2008) are major resources towards improving inter-organizational cooperation.

In the SOC paradigm, electronic services (e-services) are part of distributed applications that enable fast software development and cost reduction. They are autonomous and platform independent software that can be either atomic or composed from lower granularity e-services. The Web services technology, a realization of SOC, allows the publication, search and discovery of e-service according to internet standards (WSDL, SOAP and UDDI). This technology has imposed additional challenges to the BPM scope. Inter-organizational cooperation requires proper regulation to ensure the Quality of Service (QoS) exchanged between collaborative partners (Fantinato *et al*., 2009).

Electronic contracts (e-contracts) are means to represent an agreement between organizations in such a way that they can be monitored throughout the business process execution. E-contracts include information about parties involved in a business process, activities to be executed as e-services and respective QoS and constraints. An e-contract life cycle includes a negotiation phase that defines contract parameters and an execution phase. In the latter, the inter-organizational cooperation must be monitored to ensure contractual clauses.

A reliable inter-organizational cooperation demands e-contract monitoring. Contract conditions, such as

Freitas da Silva M., Maria de Souza Gimenes I., Fantinato M., Beatriz Felgar de Toledo M. and Fabricio Garcia A. (2010).
ASPECT-MONITOR - An Aspect-based Approach to WS-contract Monitoring.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Software Agents and Internet Computing*, pages 5-12
DOI: 10.5220/0002863800050012

those related to QoS attributes, have to be constantly and accurately supervised, for the benefit of every party involved in the business process. E-contract monitoring involves supervising the performance of the provided services and collecting relevant auditing information so that provider's commitments can be assessed and proper penalties can be applied (Napagao *et al*., 2007).

Monitoring is not a trivial issue as it requires process instrumentation either to register service performance or to take corrective actions. The aspect-oriented (AO) approach (Filman *et al*., 2005) can be used to separate orthogonal concerns in the BPM domain (Charfi and Mezini, 2004; Braem *et al*., 2006). Concerns such as monitoring can be encapsulated into aspects and executed when well-defined point cuts are reached in the process. Nevertheless, conventional aspect mechanisms cannot be directly applied to contract monitoring.

This paper presents an AO approach to monitor QoS attributes and levels, named Aspect-Monitor. An e-contract comprises collaborative partners, an inter-organizational business process specified in AO4BPEL and QoS attributes of services in WS-Agreement. Monitoring aspects are designed to deal with Web services and their related QoS attributes and levels. Thus, aspects are not only applied to the process specification but also to enhance process execution environment.

In this paper: Section 2 presents e-contracts and monitoring; Section 3 discusses AO; Section 4 proposes an approach for e-contract monitoring; Section 5 shows an example of application; Section 6 and 7 present lessons learned and related work; and Section 8 concludes the paper.

2 ELECTRONIC CONTRACTS

Important e-contracts and monitoring concepts are discussed in the following subsections.

2.1 Electronic Contracts

A contract is an agreement between two or more parties interested in establishing a mutual relationship on business or legal obligations. An e-contract is an electronic document used to represent an agreement between partners running business using the internet (Erl *et al*., 2008).

Although e-contracts differ regarding their size, content and complexity, there are usually common elements within the same domain. The canonical elements are (Grefen *et al*., 2001): parties that represent the organizations involved in a business process; activities that represent e-services to be executed; and, contractual clauses that describe constraints on the execution of activities.

Contractual clauses can represent three types of constraints: obligations that describe what parties should do; permissions that describe what parties are allowed to do; and, prohibitions that describe what parties should not do (Marjanovic and Milosevic, 2001). Obligations may include QoS clauses which define attributes related to non-functional properties of e-services. Examples are availability, integrity, reliability, security, performance and reply time (Keller and Ludwig, 2003; Menascé, 2002; Sahai *et al*., 2002).

Different languages might be used to represent e-contracts sections. The most recent ones are related to Web standards. In this context, e-contracts are called Web services e-contracts (WS-contracts). Our approach focuses on WS-BPEL and WS-Agreement. WS-BPEL has been widely used for business process specification (Barreto *et al*., 2007). WS-Agreement, based on WSLA (Keller and Ludwig, 2003), has been used for QoS specification.

2.2 Contract Monitoring

Contract monitoring requires events and information produced by executing business processes. This includes checking whether QoS attributes defined in an e-contract are being obeyed during process execution. Passive capabilities such as raising alerts when a certain attribute is not according to specified levels are not enough; therefore, corrective actions must also be provided for running processes. Even, contract renegotiation might be necessary. Thus, monitoring goals, in the BPM domain, include not only run-time measurement of QoS attributes and KPIs (Key Performance Indicators) but also dynamic adaptation to variation in parameters and application of actions specified in contracts.

According to Benbernou *et al*. (2008), new approaches developed for monitoring service-based applications should come up with holistic and comprehensive methodologies that:

1. Integrate various monitoring techniques at all functional service-based application layers;
2. Provide a way to target all the relevant application aspects and information;
3. Define rich and well-structured modeling and specification languages capable of representing these features;
4. Allow for modeling, identifying, and propagating dependencies and effects of monitored events

and information across various functional layers and aspects in order to enable fault diagnosis.

Monitoring can be carried out by the consumer and provider organizations, or else by a third-party. Current approaches for e-contract specification used for monitoring have an excess of instrumentation in the process, many extra functions to be implemented in the client and server and lack of available tools. Our approach aims at reducing the complexity of the e-contract monitoring mechanism by applying the AO fundamental concepts. Moreover, it searches for meeting the requirements set defined by Benbernou *et al*. (2008) and presented above.

3 ASPECT-ORIENTATION

Systems are usually decomposed into modules to reduce their complexity. The Structured and Object-Oriented paradigms have been useful to provide small units of reuse. However, they do not adequately treat crosscutting concerns, such as logging. The aspect-oriented (AO) paradigm aims at encapsulating crosscutting concerns into isolated parts of the system – called aspects, thus improving development, maintenance and reuse (Filman *et al*., 2005).

AO programming languages require specific compilers. For example, the AJC compiler converts an AspectJ program into a Java bytecode program, which can be executed by any JVM. AO languages have additional elements (Kiczales *et al*., 1997) which include: aspect, join point, pointcut, advice body and inter-type declaration.

AO4BPEL is a WS-BPEL extension which introduces AO mechanisms to support e-service composition (Charfi and Mezini, 2004). AO4BPEL targets two existent problems in composition languages: i) modularity specification allowing the separation of concerns such as access control, authentication and audit from business models; and, ii) dynamic service composition. A specification in AO4BPEL includes two main parts: a business process in WS-BPEL and its respective aspects.

Aspects comprise PartnerLinks, Variables and PointcutAndAdvice as illustrated in Figure 1. PartnerLinks identify partners providing Web services; Variables are used in the Advice part; and, PointcutAndAdvice includes Pointcuts described in Xpath and Advices in WS-BPEL. A Pointcut defines the point in the process where an aspect should act. Advices are executed when a Pointcut is identified.

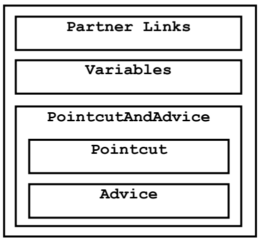

Figure 1: Aspects in AO4BPEL.

Our approach improves the application of the AO paradigm to business process domain as it complements AO4BPEL with the association of its variables with QoS specified in WS-Agreement.

4 THE ASPECT-MONITOR APPROACH

Aspect-Monitor proposes monitoring aspects as an instance of the AO4BPEL aspects. In addition to be related to the process, the new aspects are also associated with QoS attributes and levels specified in WS-Agreement. They are created to support the monitoring of services to which they refer to.

This monitoring is part of a wider BPM approach (Fantinato *et al*., 2009), including: business process negotiation, reuse, contract establishment, dynamic execution environment and the WS-contract metamodel. In this paper, we focus on the introduction of aspects concepts and the mechanisms provided for supervising QoS attributes; no coverage for aspect-orientation was provided in the previous works already published by the same authors. Next sections describe the BPM execution environment, the introduction of aspects into the e-contract metamodel and the monitoring mechanism designed.

4.1 The BPM Execution Environment

The execution of a business process and the monitoring of its Web services require inter-organizational cooperation and dynamic exchange of e-services. Figure 2 presents the elements involved in this execution environment comprising three parties: Customer Organization; Provider Organization; and, Monitor Organization (Fantinato *et al*., 2009).

The Customer Organization has the most complex structure, including: i) a structure for WS-Contract Definition which supports WS-contract negotiation and establishment; and, ii) a structure for WS-Contract Execution to support the execution of business processes specified in AO4BPEL. A SOC System is necessary in the customer organization if its own Web services are required as part of the processes. In the Provider Organization, the SOC System controls the Web services subcontracted by the consumer.

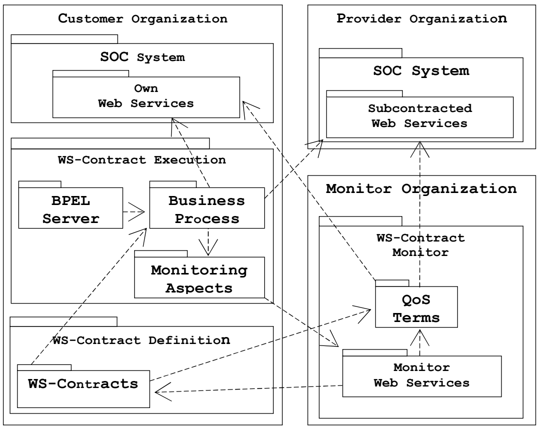

Figure 2: Execution environment architecture.

The Monitor Organization implements the Aspect-Monitor mechanisms. A third-party – the Monitor Organization, has a WS-Contract Monitor structure to control the execution of the process and hence the composed Web services, by using a set of monitor Web services and the QoS terms contained in the WS-contract as a reference. These Monitor Web services are invoked by the monitoring aspects executed in the Customer Organization whenever a join point is reached.

During the execution of a business process, the customer organization invokes Web services which can be local or subcontracted. For each invoked Web service, a set of Monitoring Aspects is triggered. The aspects are related to QoS attributes defined to the Web service. A monitoring aspect triggered during a process execution invokes correspondent monitor Web services, which will follow up the respective services execution to ensure that the contracted QoS levels are satisfied. If they are not satisfied, actions can be undertaken according to the contractual clauses. Those actions may be process cancellation, penalty application and contract renegotiation.

A prototype for the WS-Contract Monitor structure has been developed to support the execution of Web services monitoring according to the Aspect-Monitor approach.

4.2 WS-Contract Metamodel

A WS-contract metamodel represents rules to create WS-contracts and supervise them based on aspects (Figure 3). It includes concepts related to: i) Web services described by the WSDL language; ii) business processes involving Web services, described by the business-part of the AO4BPEL language; iii) QoS described by WS-Agreement language; and, iv) monitoring aspects described by the aspect-part of the AO4BPEL language.

A WS-contract is composed of four sections: WSDL definitions, BPEL Business Process, WS-Agreement, and Monitoring Aspects, as follows:

i) **WSDL Definitions Section:** contains the primary elements Message Types, Partner Link Types, Port Types and Operations - the last two describe the Web services. These elements are used to form the elements of the next sections;

ii) **BPEL Business Process Section:** describes the business process that composes Web services. It consists of: Variables, Partner Links, Activities (both Interaction Activities and Structured Activities types);

iii) **WS-Agreement Section:** describes the QoS attributes and levels regarding the Web services. Attributes and levels are described in terms of: Service Properties (including Variables) and Guarantee Terms (including Service Scope and Service Level Objectives). The WS-Agreement Name, Context and Service Description Terms elements are not included in this metamodel since there are already similar sections representing this information in the BPEL Business Process section;

iv) **Monitoring Aspects Section:** contains the aspects related to the Web services monitoring. Each aspect is described in terms of: Monitoring Partner Link, Monitoring Variable, Join point, Pointcut and Advice.

Details about the first three sections are presented by Fantinato *et al.* (2008). The fourth section is an extension of the metamodel defined for the Aspect-Monitor. The objectives of the Aspect elements as well as their relationships with other WS-contract elements are as follows:

- **Monitoring Aspect:** a module encapsulating a monitoring crosscutting concern. It is composed of monitoring partner link, monitoring variable, pointcut and advice. There is a unique monitoring aspect for each

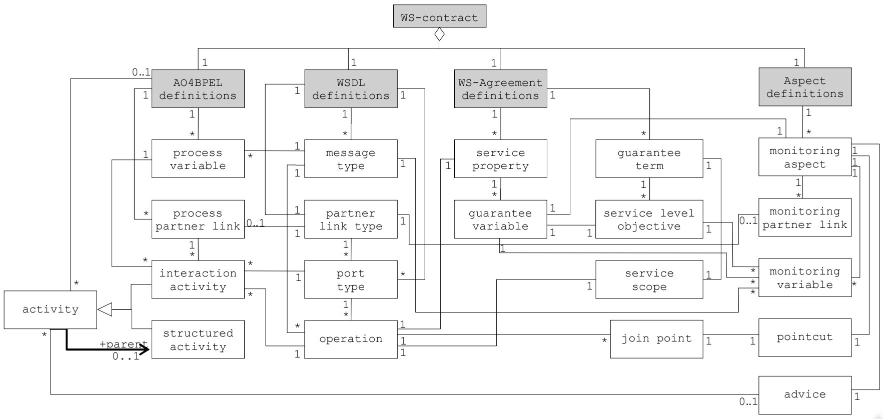

Figure 3: WS-contract metamodel.

guaranteed variable and hence for each service level objective, regarding the WS-Agreement service properties and guarantee terms.

- **Monitoring Partner Link:** represents the partners involved in the services monitoring, which are: a monitoring organization; a consumer organization; and, a provider organization.
- **Monitoring Variable:** represents variables related to the process in the advice element.
- **Join Point:** represents a Web service whose execution must trigger a monitoring aspect, thereby injecting behavior at the join point through its advice body.
- **Pointcut:** represents the Operation which identifies the Join Point – i.e. the Web service – for which the monitoring aspect must react. In this approach, the monitoring aspects have only one join point for each pointcut.
- **Advice:** contains a business process, specified in AO4BPEL, which must be executed when the join point (Web service) triggers a pointcut. This process must be used to invoke Monitoring Web services. When a join point is reached, the advice can be executed before, at the same time or after the join point.

5 APPLICATION EXAMPLE

This section presents a practical experiment undertaken on a pseudo-scenario to evaluate the application of the Aspect-Monitor approach, identifying its benefits and drawbacks.

The experiment context involves a travel agency which uses Web services, from partner organizations such as airline, car rental and hotel reservations companies. Each party, in this collaborative business process, provides a set of Web services to be used by the other parties. The travel agency system offers Web services to their customers which therefore require Web services from its partners. A WS-contract is established to regulate the business agreement. An airline company can provide a series of Web services to a travel agency, such as: timetable queries, flight booking, ticket purchase, itinerary changes and seat selection. On the other hand, a travel agency can also provide some Web services to the airline company, such as: customer notification on flight changes and advertisement on special offers.

As there is not, as yet, an AO4BPEL server available, a WS-BPEL server – ActiveBPEL (ActiveVOS, 2010) – was used instead; thus, a workaround solution has been used, in the execution environment, to execute aspect-related tasks. Accordingly, the business process specified in WS-BPEL (using here ActiveBPEL Designer) had to be instrumented in order to manually enter in the pointcuts the excerpt of process regarding the advice to be executed. Thus, the business process receives an instrumentation that simulates the use of aspects which can be performed with the WS-BPEL technology, using here the ActiveBPEL Engine.

Figures 4, 5, 6 and 7 present different excerpts of the WS-contract in which the relationship between Web services, business process, QoS attributes and levels, and monitoring aspects can be observed. In Figure 4, the Web service provided by the Airline company (identified by the operation flight-bookingOP) is defined.

```
<wsdl:portType name="flight-servicesPT">
 ...
 <wsdl:operation name="flight-bookingOP">
  <wsdl:input message="tns:flight-booking-Req"/>
  <wsdl:output message="tns:flight-booking-Resp"/>
 </wsdl:operation>
 ...
</wsdl:portType>
```

Figure 4: Excerpt from WSDL definitions.

In Figure 5, the invocation of the Web service is specified in the WS-BPEL, identified by the command invoke on the operation flight-bookingOP.

```
<bpel:invoke
 partnerLink="flight-bookingPL"
 portType=" flight-servicesPT"
 operation="flight-bookingOP"
 inputVariable="flight-booking-Request"
 outputVariable="flight-booking-Response"/>
```

Figure 5: Excerpt from WS-BPEL definitions.

In Figure 6, a QoS attribute and its level are defined for the Web service, identified respectively by the reply-timeVAR and less than 60 seconds.

In Figure 7, the monitoring aspect related to the Web service and QoS attribute is specified; when the Web service whose operation is flight-bookingOP is invoked, the respective monitoring aspect is triggered so that it can invoke a third-party monitor Web service. The following elements are present: the pointcut whose name is book-flight-invoke is defined for portType flight-servicesPT and operation flight-bookingOP (Figure 4), which means that this operation is the join point of this monitoring aspect; and the advice, of “around” type, whose actions flow is used to invoke the third-party monitoring Web service, identified by portType flight-services-monitorPT and operation flight-booking-monitorOP.

Since this monitoring aspect is related to the reply-timeVAR, its content is copied to a temporary variable so that it can be forward to the third-party monitoring Web service to be invoked. Other monitoring aspects for flight-servicesOP regarding other QoS attributes can exist as well as other ones for reply-timeVAR regarding other Web service operation.

The monitoring aspects, producing the Aspect definitions section, are generated semi-automatically based on the information from the other WS-contract sections: WSDL, WS-BPEL and WS-Agreement.

```
<wsag:ServiceProperties wsag:Name="flight-
bookingSP" wsag:ServiceName="flight-
bookingOP">
 <wsag:VariableSet>
  ...
  <wsag:Variable
   wsag:Name="reply-timeVAR"
   wsag:Metric="second">
   <wsag:Location>//wsag:SDT/[
    @portType="flight-bookingPT"
    @operation="flight-bookingOP"]
   </wsag:Location>
  </wsag:Variable>
  ...
 </wsag:VariableSet>
</wsag:ServiceProperties>
...
<wsag:ServiceScope ServiceName="flight-
bookingOP"/>

<wsag:QualifyingCondition>…</wsag:QualifyingC
ond…>
 <wsag:ServiceLevelObjective>
  <exp:Less>
   <exp:Variable>reply-timeVAR</exp:Variable>
   <exp:Value>60</exp:Value>
  </exp:Less>
 </wsag:ServiceLevelObjective>

<wsag:BusinessValueList>…</wsag:BusinessValue
List>
</wsag:GuaranteeTerm>
```

Figure 6: Excerpt from WS-Agreement definitions.

```
<aspect name="Flight-bookingOP_+_reply-timeVAR">
<partners>…</partners>
<variables>…</variables>
<pointcut name="book-flight-invoke"
contextCollection="true">
//process//invoke[
@portType="flight-servicesPT"
@operation="flight-bookingOP"]
</pointcut>
<advice type="around">
...
<bpws:sequence>
<bpws:assign>
<bpws:copy>
<bpws:from expression="reply-timeVAR"/>
<bpws:to variable="monitor-request"
part="service-property-namePART"/>
</bpws:copy>
...
</bpws:assign>
<bpws:invoke
partnerLink="third-party-monitoringPL"
portType="flight-services-monitorPT"
operation="flight-booking-monitorOP"
inputVariable="monitor-request"
outputVariable="monitor-response"/>
</bpws:sequence>
...
</advice>
</aspect>
```

Figure 7: Excerpt from Aspect definitions.

6 LESSONS LEARNED

This section presents the lessons learned from the development and the exercising of the aspect-based approach to WS-contract monitoring. In the performed analysis, the proposed approach was compared mainly to the pure WSLA, WS-BPEL and WS-Agreement approaches. Moreover, the requirements presented by Benbernou *et al*. (2008) for new approaches related to e-contract monitoring were also taken into account.

No Need for Instrumentation. The proposed approach allows the customer organization to define its business process without adding extra code for monitoring. It means that, according to the AO paradigm, the main concern (i.e. the business process) was not crosscut by the secondary concern (i.e. the contract monitoring). Other monitoring approaches do not perform such separation, requiring instrumentation of the business process with monitoring functions.

Simplified Computational Support. The proposed approach simplifies the computational support necessary to carry out e-contract monitoring when compared to other existent approaches for the same objective. No extra technology needs to be implemented to support the monitoring functions. The proposed BPM execution environment aims to support all the contract life-cycle, including monitoring, in a unique support environment.

Integrated Solution. The proposed approach presents an integrated solution in which all the main parts of a WS-contract are treated together: business process, QoS attributes and levels, and monitoring activities. Other monitoring approaches commonly deal with isolated Web services and do not provide a solution for composed Web services. This point is closer related to requirements 1 and 2 from the set proposed by Benbernou *et al*. (2008), since the proposed approach provides a broad integration in different levels – including different techniques and information.

Use of Standard Languages. No new language was defined to compose this approach. Instead, standard languages, or already existing extensions, are used, which together are capable of representing the information related to contract monitoring, as required by item 3 from the requirements set proposed by Benbernou *et al*. (2008).

7 RELATED WORK

Several works are related to e-contract monitoring in a general way, but not involving AO, from which we can highlight Cremona (Ludwig *et al*., 2004) – used as the architectural basis for the development of the Aspect-Monitor approach. Cremona is an architecture for the creation, management and monitoring service-level agreements represented as WS-Agreement documents. The monitoring module is not only used to observe and detect contract violations, but also to predict future violations and to trigger corrective actions in advance.

There are also some aspect-based works related to e-contract monitoring, but none of them applies AO both in a wide and complete BPM approach, and using specifically the WS-Agreement language as presented in this paper. Some of these works are presented as follows.

Singh *et al*. (2004) proposed a methodology based on aspects to specify functional and non-functional characteristics of Web Services. The Web Services are described in an extended WSDL named AO-WSDL and published in an extended registry mechanism named AO-UDDI.

Tomaz *et al*. (2006) introduced AO concepts to add non-functional requirements to Web services. They focus on run-time adaptation of non-functional characteristics of a composite Web service by modifying the non-functional characteristics of its component Web services. They also propose a language for representing non-functional properties.

Ortiz & Leymann (2006) claim that, although the encapsulation of non-functional services properties using aspects are present in other works, aspect implementation must be combined with WS-Policy expressing non-functional properties. This proposal maintains policies separated from the functional behavior of the service and its main description.

Narendra *et al*. (2007) proposed run-time adaptation of non-functional properties of a composite Web service by modifying the non-functional properties of its component Web services. They use AO technology for specifying and relating non-functional properties of the Web services as aspects at both levels of component and composition. WS-Policy and WSLA are used in the negotiation between Web services.

Bianculli & Ghezzi (2007) proposed an approach in which the behavior of a service is specified in an algebraic language and can be checked at run-time. A BPEL engine is extended, using AspectJ, with three components: an interceptor, a specification registry, and a monitor. The monitor is a wrapper for the evaluation of algebraic specifications, which keeps a machine-readable description of the state of the process instances which is updated when the new information from the interceptor is received. The symbolic state is then evaluated and compared against the expected one; and, in case of mismatch, certain specific actions may be performed.

8 CONCLUSIONS

This paper presented an infrastructure to define, execute and monitor WS-contracts. In the proposed approach, a WS-contract comprises collaborative partners, business process specified in AO4BPEL and QoS attributes of services defined with WS-Agreement. AO technology is used to separate concerns such as monitoring into aspects. Those aspects are executed when defined join point are reached in the process. Compared to other approaches, the proposed solution requires no instrumentation, uses Web services standards and provides an integrated infrastructure for dealing with contract establishment and monitoring.

Future work includes: i) studying and incorporating other AO properties to enrich the proposed approach; ii) concluding the development of the WS-Contract Monitor structure, from the BPM execution environment; and, iii) carrying out a broader experiment of the presented proposal to evaluate its benefits and its limitations.

ACKNOWLEDGEMENTS

This work was supported by The State of São Paulo Research Foundation (FAPESP) and The National Council for Scientific and Technological Development (CNPq), Brazil.

REFERENCES

ActiveVOS (2010). BPMS from Active Endpoints. Retrieved October 12, 2006, from http://www.activevos.com/

Benbernou, S. *et al*. (2008). State of the art report, gap analysis of knowledge on principles, techniques and methodologies for monitoring and adaptation of SBAs. *Technical Report (Project Deliverable)*. Université Claude Bernard Lyon, France.

Bianculli, D. and Ghezzi, C. (2007). Monitoring conversational web services. In *IW-SOSWE'07, 2nd Int. Workshop on Service Oriented Software Engineering*. ACM Press.

Braem, M. *et al*. (2006). Isolating process-level concerns using Padus. In *BPM'06, 4th Int. Conf. on Business Process Management*. Springer.

Charfi, A. and Mezini, M. (2004). Aspect-oriented web service composition with AO4BPEL. In *ECOWS'04, European Conf. on Web Services*. Springer.

Erl, T. *et al*. (2008). *Web sService Contract Design and Versioning for SOA*. Prentice Hall.

Fantinato, M., Toledo, M. B. F. and Gimenes, I. M. S. (2008). WS-contract establishment with QoS: An approach based on feature modeling. *Int. Journal of Cooperative Information Systems*, 17(3), 373-407.

Fantinato, M., Gimenes, I. M. S. and Toledo, M. B. F. (2009). Product Line in the Business Process Management Domain. In Kang, K. C., Sugumaran, V. and Park, S. (Eds.), *Applied Software Product Line Engineering* (pp. 497-530). Boca Raton: Auerbach.

Filman, R. *et al*., (2005). *Aspect-Oriented Software Development*. Addison-Wesley.

Grefen, P. W. P. J. *et al*. (2001). CrossFlow: Cross-organizational workflow management for service outsourcing in dynamic virtual enterprises. *IEEE Data Engineering Bulletin*, 24(1), 52-57.

Keller, A. and Ludwig, H. (2003). The WSLA framework: Specifying and monitoring service level agreements for Web services. *Journal of Network and Systems Management*, 11(1), 57–81.

Kiczales, G. *et al*. (1997). Aspect-oriented programming. In *ECOOP'97, 11th European Conf. on Object-Oriented Programming*. Springer.

Ludwig, H., Dan, A. and Kearney, R. (2004). Cremona: An architecture and library for creation and monitoring of WS-Agreements. In ICSOC'04, 2nd Int. Conf. on Service-Oriented Computing. Springer.

Marjanovic, O. and Milosevic, Z. (2001). Towards formal modeling of e-contracts. In *EDOC'01, 5th Int. Enterprise Distributed Object Computing Conf.* IEEE Computer Society.

Menascé, D. A. (2002). QoS issues in web services. *IEEE Internet Computing*, 6(6), 72-75.

Napagao, S. A. *et al*. (2007). Contract based electronic business systems state of the art. *Technical Report (Project Deliverable)*. Univ. Politècnica de Catalunya.

Narendra, N. C. *et al*. (2007). Run-time adaptation of non-functional properties of composite web services using aspect-oriented programming. In *ICSOC'07, 5th Int. Conf. on Service-Oriented Computing*. Springer.

Barreto, C. *et al*. (2007). OASIS Web Services Business Process Execution Language (WSBPEL) TC. Retrieved January 10, 2010, from http://docs.oasis-open.org/wsbpel/2.0/Primer/wsbpel-v2.0-Primer.pdf

Ortiz, G. and Leymann, F. (2006). Combining WS-Policy and aspect-oriented programming. In *AICT/ICIW'06, Advanced Int. Conf. on Telecommunications and Int. Conf. on Internet and Web Applications and Services*. IEEE Computer Society.

Papazoglou, M. P. *et al*. (2008). Service-oriented computing: A research roadmap. Int. Journal of Cooperative Information Systems, 17 (2), 223-225.

Sahai, A. *et al*. (2002). Automated SLA monitoring for web services. In *DSOM'02, 13th IFIP/IEEE Int. Workshop on Distributed Systems: Operations and Management*. Springer.

Singh, S., Grundy, J. C. and Hosking, J. G. (2004). Developing .NET web service-based applications with aspect-oriented component engineering. In *AWSA'04, 5th Australasian Workshop on Software and Systems Architecures*.

Tomaz, R. F., Hmida, M. B. and Monfort, V. (2006). Concrete solutions for web services adaptability using policies and aspects. *Int. Journal of Cooperative Information Systems*, 15(3), 415-438.

Weske, M. (2007). *Business Process Management: Concepts, Languages, Architectures*. Berlin: Springer.