(Rep 08)

[Rep08] *Nicolas Repp;* **Monitoring of Services in Distributed Workflows.** In: 3rd International Conference on Software and Data Technologies (ICSOFT 2008), INSTICC PRESS, Portugal, July 2008. Seite15-25

MONITORING OF SERVICES IN DISTRIBUTED WORKFLOWS

Nicolas Repp
Technische Universität Darmstadt, Multimedia Communications Lab, Merckstr. 25, 64285 Darmstadt, Germany
nicolas.repp@KOM.tu-darmstadt.de

Keywords: Distributed Systems, Service Monitoring, Service-oriented Computing, Distributed Workflows, Quality of Service.

Abstract: In order to implement cross-organisational workflows and to realise collaborations between enterprises the use of Web service technology and the Service-oriented Architecture paradigm has become state of the art. Here, as in every distributed system, several challenges arise in order to ensure the quality of service delivery. Especially, the monitoring of workflows and the related services at runtime is crucial in order to fulfil business requirements, e.g., as described in Service Level Agreements (SLA).
In my research, I am working towards an integrated monitoring approach for distributed service-based workflows, supporting the detection of SLA violations as well as their correction. The following research questions are subject to my work: *what?* to monitor, *where?* to monitor in a service-based environment as well as *how to react?* in case of SLA violations.

1 RESEARCH PROBLEM

Cross-organisational collaborations based on the integration of business processes and IT systems of business partners is getting more and more important to strengthen the competitiveness of enterprises. In order to implement those cross-organisational workflows, the Service-oriented Architecture paradigm (SOA) is often facilitated as an architectural blueprint. Supported by an adequate SOA implementation, services of different parties can be combined to cross-organisational workflows and therefore support collaborations between business partners. Nevertheless, the application of SOA and service-based workflows also creates various challenges enterprises have to cope with. The integration of third party services into an enterprise's workflows needs to address aspects like Quality of Service (QoS) and security to realise dependable and trusted business relationships. Therefore, Service Level Agreements (SLA) have to be defined between the participating parties addressing business requirements and responsibilities of the business partners. Furthermore, it is crucial not to simply rely on the requirements defined by the SLA but to monitor the fulfilment of those requirements during runtime as a part of both IT operations and the enterprise's IT governance strategy. Moreover, in case of deviations from requirements defined in the SLA adequate countermeasures have to be applied to re-align the overall workflow execution with the SLA. As discussed in many recent publications, the "typical" scenario for cross-organisational workflows using service-based workflows consists of a single enterprise (i.e., the service requester) and several business partners providing services in a client-server style. Monitoring and deviation handling is centralised and handled by the service requester itself. Although, centralised monitoring and alignment is broadly used in real-world applications, a centralised approach is not applicable to large service-based scenarios containing an extensive number of service requesters and providers. Here, scalability and complexity issues as well as legal and governance problems do exist, e.g., unclear responsibilities and a lack of privacy. Both quality and amount of monitoring data needed for decision making as well as the time in which the data can be provided are not sufficient for real-time deviation handling or avoidance.

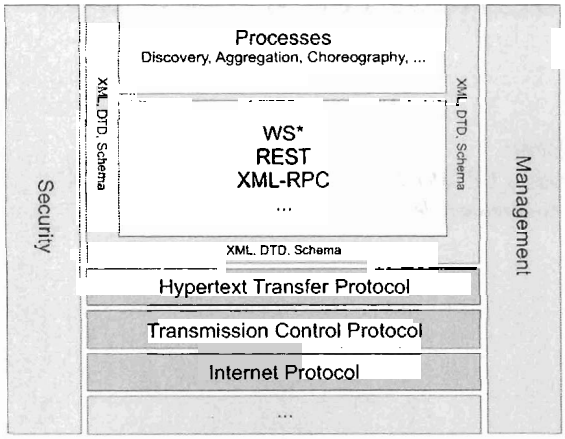

Figure 1: Layered model of Web service related standards

Currently, two overarching classes of service monitoring approaches can be differentiated in practice based on the layers they are working on. Real world enterprises often combine approaches of both classes in order to harvest a broad scale of data for decision making. The first class contains approaches, which primarily work on transport layer and below. They make strong use of existing network management systems, e.g., based on the Simple Network Management Protocol (SNMP) or the less established Remote Management (RMON) standard to monitor service execution. Here, service implementations, e.g., Web services, are seen as additional network resources which can be managed by the network management system. Approaches also using application layer knowledge define the second class of monitoring approaches, which are often part of the service runtime environment, i.e., the application server on service provider side or the execution engine on service requester side. Both classes have in common, that they are primarily implemented using a centralised architecture for information gathering, where monitoring units and decision making components are located with the service requesters, even if monitoring data is collected in a distributed fashion. As centralised management has different drawbacks with respect to scalability and performance, various enhancements of the centralised network management approaches do exist, which are in fact capable of distributing parts of the information gathering process. Nevertheless, decision making in case of SLA violations is coordinated by a central instance.

Additionally, the selection of the "right" parameters to monitor during service execution is crucial in order to reduce the amount of monitoring data needed as well as the delay resulting from analysis of the data. In addition, current monitoring approaches often lack the provision of fine grained information about service status due to the information hiding principles used in current infrastructure implementations. Here, a cross-layer approach to detect SLA violations by correlating events from various layers and sources is helpful for an efficient and proactive monitoring of services (cp. Figure 1 for a Web service related overview of layers).

The remaining part of this paper is structured as follows. In the next section, the objectives of my research are discussed in more detail. Afterwards, I present the expected outcome of my overall research separated in individual contributions. The subsequent section discusses the research methodology followed by a section containing the current stage of my research. Before the paper closes with a conclusion and outlook, the state of the art in service and cross-layer monitoring is presented.

2 OUTLINE OF OBJECTIVES

The overall goal of my research is the development of an efficient, scalable, and nevertheless easy to integrate monitoring approach for workflows in distributed service-based environments. The objectives of the work are best defined by the following research questions:

- What parameters of single services or workflows as composition of services have to be monitored?
- Where to place monitoring units in a distributed service-based environment?
- What to do in case of SLA violations of single services and complete workflows?
- How to integrate all of the monitoring and deviation handling in existing workflows, workflow/process engines, and infrastructures?

Hereby, the peculiarities of service-based environments need to be considered in particular, i.e., the large scale of the distributed system in scope, the different spheres of control being addressed (e.g., service requester, service provider, intermediaries, and network providers), and the instability/dynamics of relationships between parties.

3 EXPECTED OUTCOME

In order to create a distributed but nevertheless smoothly integrable monitoring approach, several contributions are planned or already realised. As a

foundation, a cross-layer analysis of correlations between network anomalies (e.g., time-outs, lost packets) and the behaviour of Web service quality was carried out, resulting in the definition of metrics for service monitoring (here in the context of Web service technology). In order to define not only monitoring requirements in parallel to a workflow definition, e.g., using the Business Process Execution Language (BPEL), but also adequate reactions to deviations, a policy language based on the WS-Policy Language of the World Wide Web Consortium (W3C) was defined. Furthermore, a mobile agent-based approach to the monitoring and deviation handling of distributed service-based workflows was designed and is currently beeing implemented based on Web service technologies as a proof of concept also facilitating the policy language. A special focus is on the placement of monitoring units in a service-based environment. Here, the benefits of distributed monitoring and deviation handling to improve scalability, to minimise monitoring traffic in networks, and to improve reaction times in case of deviations from given requirements will be addressed by defining appropriate distribution strategies and algorithms. Those evaluations will be based on the application of simulation approaches.

4 RESEARCH METHODOLOGY

The different contributions of my research determine different research approaches, so that not only a single research methodology can be applied. The following list gives an overview of the contribution and the research methodology / approach used:

- Conceptual work to analyse the problem space and to design the architecture as well as monitoring and deviation handling approaches
- Measurements of "real world" Web services in test-bed to derive metrics for cross-layer monitoring
- Analytical evaluation of the benefit of distribution mechanisms as a foundation for their subsequent simulation
- Simulation to evaluate improvements of reaction times, scalability, and minimised monitoring traffic by the distribution of monitoring functionality
- Prototypical implementation of the architecture as well as monitoring and deviation handling approaches

Only the combination of all methodologies allows an integrated treatment of the research questions addressed before.

5 STATE OF THE RESEARCH

In this section, the core building blocks (i.e., contributions) of my research are presented. Here, my recent work in the cross-layer monitoring of services is presented in short summary as well as the agent-based monitoring and deviation handling approach named *Automated Monitoring and Alignment of Services* (AMAS.KOM), its architecture and its underlying distribution strategies for monitoring and alignment agents in a service-oriented environment (work in progress). In addition, a policy language for the definition of requirements and countermeasures is presented.

5.1 Cross-layer monitoring of services

The idea behind cross-layer monitoring is to monitor service execution on different protocol layers. The basic assumption for this kind of monitoring is that the propagation of events through the protocol stack takes (too much) time, which otherwise could be used for the preparation of countermeasures in case of any problems. During service execution several problems can occur, which all have an impact on Web service behaviour. On network layer, routing problems, e.g., hosts which are not reachable, congestion in Internet routers as well as traffic bursts may exist. On transport layer, e.g., the retransmission of packets due to packet loss or connection setup problems generates delays. There are also some potential problems on application layer, e.g., non-existing or non-accessible resources or problems during SOAP due to incomplete or non-valid XML data (cp. Table 1 for some overall examples). Many of the aforementioned problems are already solved in modern systems. Nevertheless, the knowledge about them can be used for monitoring and deviation handling.

Protocol	Measuring Point / Parameter
IP	ICMP messages
TCP	Size of advertising window
	Roundtrip time
	Sequence numbers in use
	Flags used in packets
	Information about timers
HTTP	Header information

Table 1: Measuring points per protocol layer

In (Repp et al., 2007a) and (Repp et al., 2007b), transport layer parameters are used to derive metrics and heuristics for performance anomaly detection, e.g., the average throughput in bytes per second, the throughput based on a moving average over

a given window, or the number of gaps in sequence numbers based on a moving average over a given window size. The aggregation of single metrics in combination with the usage of appropriate thresholds allows to build heuristics to detect anomalies.

The evaluation of different combinations of metrics showed that, e.g., the roundtrip time extracted from TCP packets can be used as trend estimate for the overall response time of services. In combination with adequate thresholds, a warning to a deviation handling module can be sent before a real problem occurs on service level (i.e., application layer).

5.2 Agent-based distributed monitoring and alignment

For scalable and high-performing service monitoring and alignment the use of distributed monitoring mechanisms in combination with cross-layer data is crucial. As a foundation to design and implement such mechanisms, I apply the mobile agent paradigm to service monitoring, where agents take the role of monitoring units, capable of executing countermeasures in an autonomic manner. Monitoring and alignment agents (MAA) are automatically created based on a given process description and requirements specification and afterwards deployed using dedicated distribution mechanisms.

Generation of monitored workflow instances
This section will give a short overview about the process to automatically generate MAA based on a given workflow description and related business requirements. In my approach, which is already partially implemented in the AMAS.KOM architecture, the generation of MAA is realised by a transformation process, transforming both workflow description and business requirements into monitored workflow instances. In AMAS.KOM, an existing workflow description (e.g., described in WS-BPEL) is analysed and modified in order to integrate proxy services for the redirection of service calls to the monitoring and alignment infrastructure and the respective MAAs. The approach contains four basic transformation steps:

1. Annotation
2. Modification and Splitting
3. Generation
4. Distribution

During "Annotation", the description of a business process has to be connected to a specification of business requirements using a semiautomatic approach.

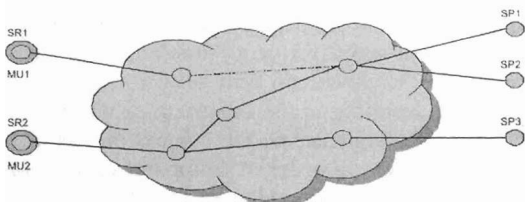

Figure 2: Classical centralised monitoring of services

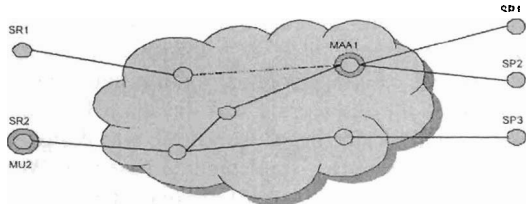

Figure 3: Monitoring of services using distributed MAA

Policy documents in machine-readable format have to be created containing the requirements of the complete workflow. For the specification of both requirements and possible reactions to deviations from the requirements, the *Web Service Requirements and Reaction Policy Language* (WS-Re2Policy) can be used, which is discussed in a later section. In the next step "Modification and Splitting", requirements for single services are derived by analysing the overall requirements. Therefore, QoS-aware planning approaches and algorithms are applied to generate feasible execution plans, e.g., as discussed by (Canfora et al., 2005) and (Berbner et al., 2006). During the "Generation" step, MAAs are created based on the policy information. Subsequent to their generation, the MAAs have to be distributed in the infrastructure during the "Distribution" step, based on the results of suitable distribution algorithms. An in-depth discussion of the single steps is subject of (Repp et al., 2008).

Distribution strategies for MAA
MAAs are responsible for certain parts of a workflow, i.e., single services or sub-workflows. MAAs can be distributed between service providers, requesters, and intermediaries based on distribution algorithms. In classical monitoring approaches monitoring is mostly carried out by every party on its own. From a service requesters perspective, the only monitoring data available is the one gathered in the domain (cp. Figure 2). Monitoring using MAA allows the placement of monitoring units at different locations in an infrastructure, e.g., after a weak link, i.e., a link with bad QoS characteristics (cp. Figure 3). Furthermore, the monitoring and alignment services offered by a MAA can be shared between different service requesters and can

also be used in combination with more centralised approaches. Here, the overall goal is the improvement of scalability, the minimisation of monitoring traffic, and the improvement of reaction times in case of deviations.

The distribution of MAA is the current focus of my work. Currently, I am investigating mechanisms and algorithms to support both the initial distribution of MAA and the continuous improvement of their location using agent mobility, cloning, and the fusion of agents. Furthermore, adequate delegation mechanisms between agents are under investigation. Both will be evaluated using simulation approaches. Therefore, I am enhancing the OMNet++ network simulator to support my agent based monitoring approach.

5.3 The Requirements and Reactions Policy Language

In order to generate and distribute MAA a description of the monitoring requirements is needed, which the agent carries during its "journey" in the infrastructure. A limitation of existing requirements languages is that they do not support deviation handling capabilities, allowing the MAA to react based on a given set of allowed reactions. Therefore, I developed the WS-Re2Policy language based on the well known WS-Policy language of the W3C. WS-Re2Policy language uses the well-founded Event-Condition-Action (ECA) rules paradigm. The elements of the language can be mapped to concepts of the ECA paradigm, i.e., events are the subjects to monitor, conditions the thresholds for monitoring, and actions the reactions to deviations.

Every WS-Re2Policy compliant document has a requirements and a reactions part. Requirements can be described in any WS-Policy compliant language. Reactions are simple, easy to understand and implementation independent control constructs. The following reactions are currently supported:

- Restart of selected services
- Renegotiation of Service Level Agreements
- Replanning of execution plans
- Selection of different services based on various criteria
- Report results to caller or different third parties
- Interruption of execution

In addition, different control constructs, e.g., iterations and sleep, are supported. More details on the WS-Re2Policy language can be found in (Repp et al., 2008).

5.4 The AMAS.KOM architecture and prototype

In order to integrate the different research contributions the AMAS.KOM architecture has been designed and implemented based on current Web service standards as a proof of concept.

The architecture consists of four core components, which contain unique functionalities realised by the AMAS.KOM architecture. The core components are:

- AMAS Controller: provides the transformation logic to create a monitored workflow and create service specific policy documents.
- AMAS Repository: used to store policies and configurations of overall system and its MAAs.
- Monitoring and Alignment Manager: responsible for the generation of MAAs and their distribution.
- Monitoring and Alignment Agents: realised as mobile software agents responsible for the monitoring and the execution of countermeasures. MAAs are extendable by plug-ins to support various monitoring interfaces, e.g., SNMP.

The implementation of AMAS.KOM uses the JADE agent development framework, enriched by Apache Axis to integrate a plethora of different Web service standards and related specifications, i.e., the Web service description language (WSDL) 1.1, SOAP 1.2, and the REST approach as transport mechanisms. Furthermore, WS-Policy 1.5 and WS-SecurityPolicy 1.1 are supported as policy formats as well as WSBPEL 2.0 for the description of workflows.

6 STATE OF THE ART

There are various approaches for the monitoring of services in distributed workflows. Approaches can be divided into centralised and decentralised as well as functional and non-functional monitoring approaches. Robinson discusses monitoring of functional requirements specified in temporal logic, which are evaluated in parallel to the workflow execution (Robinson, 2005). The handling of deviations is not supported by the approach. Spanoudakis and Mahbub also use logical constructs to describe functional monitoring requirements. In their approach, BPEL4WS code is transformed into event calculus for monitoring (Spanoudakis and Mahbub, 2006). Also, no support for the handling of deviations based upon the monitoring results is offered. (Baresi and Guinea, 2005) discuss a monitoring approach, in which monitoring requirements are embedded as pre- and post-conditions

BPEL. In order to generate a monitored instance of the workflow, a BPEL pre-processor is used for the extraction of the monitoring requirements. Again, no deviation handling is provided by the authors. In addition, there are various approaches for monitoring of non-functional requirements. (Canfora et al., 2005) describe a conceptual framework for monitoring of non-functional requirements as well as the replanning of service-based workflows, however, without any proof of concept. (Schmietendorf et al., 2005) focus on monitoring of performance and availability by independent third parties, but without support for deviation handling. (Berbner et al., 2007) provide QoS-aware service selection and replanning of workflows using centralised monitoring of non-functional requirements.

Furthermore, there are different approaches, which support decentralised monitoring instead of centralised monitoring. Some of them also make use of (mobile/software) agent technology for the distribution of the monitoring logic. An early representative from the area of network management is (Zapf et al., 1999), in which the authors present an approach for network monitoring supporting the integration of SNMP into an agent-based architecture. Also for network management purposes, (Gavalas et al., 1999) created a scalable framework based on mobile agents. Both network monitoring approaches only support monitoring and no deviation handling mechanisms. A further representative of agent-based monitoring approaches is discussed by (Liotta et al., 2001). In their work, software agents act as area managers responsibly for certain parts of a network. Instead of assigning the area at system start-up time, the monitoring agents are able to adapt to changes in the underlying network infrastructure and therefore move within the infrastructure.

There also exists a vast amount of classical approaches for network monitoring with and without applicability to Web services. Therefore, two of those representatives are shortly presented.(Gschwind et al., 2002) describes a performance analysis system focusing on transactions between Web browsers and Web servers. Here, monitoring is based on HTTP. Also of interest is the work of Feldmann, who uses cross-layer capturing and analysis of TCP and HTTP for Web performance studies, whose data is derived using packet capturing (Feldmann, 2000).

Also in scope of my research are languages to specify monitoring requirements and reactions to deviations of requirements and therefore adequate related work is provided in this paragraph. Baresi et al. provide the Web service constraint language supporting both functional and non-functional requirements (Baresi et al., 2006). It allows the specification of user, provider, and third party requirements in a WS-Policy compliant form. Ludwig et al. use WS-Policy included in WS-Agreement to specify requirements (Ludwig et al., 2004). Sen et al. do not use a WS-Policy based language. Instead they use past time linear temporal logic for the description of monitoring requirements (Sen et al., 2004). Lazovik et al. use proprietary business rules for the specification of functional and non-functional requirements (Lazovik et al., 2006).

7 CONCLUSION AND OUTLOOK

Cross-organisational workflows based on Web service technology and the SOA paradigm realising collaborations between enterprises need consequent monitoring in order to fulfil various business requirements. Therefore, an integrated monitoring approach for distributed service-based workflows is needed, which supports both the detection of SLA violations and their correction.

In my research presented in this paper I work towards such an integrated approach. During my work, I contributed or currently contribute the following:

- An approach to generate monitored workflow instances based on a process description and specification of business requirements.
- A policy language for the description of monitoring requirements and reactions to deviations of business requirements.
- An architecture facilitating mobile software agent technology for distributed monitoring.
- Mechanisms and algorithms for the efficient distribution of monitoring units in an infrastructure.

Especially, the last of the mentioned contribution is under strong investigation. Here, different approaches are currently investigated, e.g., the definition and solution of an optimisation problem for a cost and response-time efficient initial distribution of MAA. Also of importance is the continuous enhancement of WS-Re2Policy to natively support various QoS requirements. Finally, the enhancement of AMAS.KOM with security features as well as its integration with the OASIS Web Services Distributed Management standard are also open issues.

ACKNOWLEDGEMENTS

This work is supported in part by E-Finance Lab Frankfurt am Main e.V. (http://www.efinancelab.de).

REFERENCES

Baresi, L. and Guinea, S. (2005). Towards dynamic monitoring of ws-bpel processes. In *Proceedings of the 3rd International Conference on Service oriented computing*, pages 269–282.

Baresi, L., Guinea, S., and Plebani, P. (2006). Ws-policy for service monitoring. In *Proceedings of the 6th Workshop Technologies for E-Services*, pages 72–83.

Berbner, R., Grollius, T., Repp, N., Eckert, J., Heckmann, O., Ortner, E., and Steinmetz, R. (2007). Management of service-oriented architecture (soa)-based application systems. *Enterprise Modelling and Information Systems Architectures*, 1(2):14–26.

Berbner, R., Spahn, M., Repp, N., Heckmann, O., and Steinmetz, R. (2006). Heuristics for qos-aware web service composition. In *Proceedings of the 4th IEEE International Conference on Web Services*, pages 72–79.

Canfora, G., Penta, M. D., Esposito, R., and Villani, M. L. (2005). Qos-aware replanning of composite web services. In *Proceedings of the IEEE International Conference on Web Services*, pages 121–129.

Feldmann, A. (2000). Blt: Bi-layer tracing of http and tcp/ip. *Comput. Networks*, 33(1-6):321–335.

Gavalas, D., Greenwood, D., Ghanbari, M., and O'Mahony, M. (1999). Using mobile agents for distributed network performance management. In *Proceedings of the 3rd International Workshop on Intelligent Agents for Telecommunication Applications*, pages 96–112.

Gschwind, T., Eshghi, K., Garg, P. K., and Wurster, K. (2002). Webmon: A performance profiler for web transactions. In *Proc. of the 4th IEEE Int'l Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pages 171–176.

Lazovik, A., Aiello, M., and Papazoglou, M. (2006). Planning and monitoring the execution of web service requests. *International Journal on Digital Libraries*, 6(3):235–246.

Liotta, A., Pavlou, G., and Knight, G. (2001). A self-adaptable agent system for efficient information gathering. In *Proceedings of the 3rd International Workshop on Mobile Agents for Telecommunication Applications*, pages 139–152.

Ludwig, H., Dan, A., and Kearney, R. (2004). Cremona: An architecture and library for creation and monitoring of ws-agreements. In *Proceedings of the 2nd International Conference on Service oriented computing*, pages 65–74.

Repp, N., Berbner, R., Eckert, J., Heckmann, O., Schulte, S., and Steinmetz:, R. (2007a). Der einfluss von transportschicht-anomalien auf die performanz von web services. In *Proceedings of the 5. Fachtagung Kommunikation in Verteilten Systemen*, pages 117–122.

Repp, N., Berbner, R., Heckmann, O., and Steinmetz, R. (2007b). A cross-layer approach to performance monitoring of web services. In *Emerging Web Services Technology*, pages 21–32.

Repp, N., Eckert, J., Schulte, S., Niemann, M., Berbner, R., and Steinmetz:, R. (2008). Towards automated monitoring and alignment of service-based workflows. In *Proceedings of the IEEE International Conference on Digital Ecosystems and Technologies 2008*, pages 235–240.

Robinson, W. (2005). A requirements monitoring framework for enterprise systems. *Journal of Requirements Engineering*, 11(1):17–41.

Schmietendorf, A., Dumke, R., and Stojanov, S. (2005). Performance aspects in web service-based integration solutions. In *Proc. of the 21st UK Performance Engineering Workshop*, pages 137–152.

Sen, S., Vardhan, A., Agha, G., and Rosu, G. (2004). Efficient decentralized monitoring of safety in distributed systems. In *Proceedings of the 26th International Conference on Software Engineering*, pages 418–427.

Spanoudakis, G. and Mahbub, K. (2006). Non intrusive monitoring of service based systems. *International Journal of Cooperative Information Systems*, 15(3):325–358.

Zapf, M., Herrmann, K., and Geihs, K. (1999). Decentralized snmp management with mobile agents. In *Proceedings of the 6th IFIP/IEEE International Symposium on Integrated Network Management*, pages 623–635.