COSMA – An Approach for Managing SLAs in Composite Services

André Ludwig and Bogdan Franczyk

University of Leipzig
Information Systems Institute
Marschnerstr. 31, 04109 Leipzig, Germany
{ludwig,franczyk}@wifa.uni-leipzig.de

Abstract. Service provisioning is largely built on agreements specifying the mutual responsibilities of service providers and their customers with respect to functional and non-functional parameters. Current SLA management approaches, i.e. WSLA, WS-Agreement, or WSOL, provide extensive SLA language formalizations and management frameworks. However, they focus on bi-lateral service requester/provider constellations neglecting the SLA management requirements of composite service providers, i.e. managing SLAs with atomic service providers and with composite service requesters and aligning both with each other. A SLA management solution for composite services has to consider the contribution of sourced services - formalized in their (atomic) SLAs (ASLA) - in the management of the provided service - formalized in its respective (composite) SLA (CSLA). This paper presents the novel COmposite Sla MAnagement (COSMA) approach for an integrated management of atomic and composite SLAs during their entire lifecycle. It can be utilized for controlling the relationships between ASLAs/CSLAs and thus serves as the basis for managing and optimizing the SLAs involved in composite services.

1 Introduction

Service-oriented computing (SOC) has emerged as the most promising design paradigm for next-generation distributed information systems. The vision that goes along with SOC is that once standards have established themselves and become widely adopted by service providers and requesters, a globally available infrastructure for hosting and accessing services will be created [1]. This infrastructure will allow service providers to offer multiple services with individually adapted service capabilities to their changing customers that can dynamically and on-demand bind these services into their own applications; thus forming a market of services. The advent of service markets on the basis of the SOC paradigm will pave the way for a service-oriented business model which is referred to as composite service provider (CSP) [2]. A composite service provider requests services from external service providers (atomic services) and provides these services according to a process flow as composite service to service requesters.

A. Bouguettaya, I. Krueger, and T. Margaria (Eds.): ICSOC 2008, LNCS 5364, pp. 626–632, 2008.

In this constellation, a composite service provider acts as an independent, self-interested business entity, motivated to fulfil own goals, i.e. be profitable and maximize customer satisfaction.

In order to control the interface between service requesters and providers, a contractual basis in form of service level agreements (SLA) is needed. Current SLA management approaches applicable for SOC environments, i.e. WSLA [3], WS-Agreement [4], or WSOL [5], provide extensive SLA language formalizations and management frameworks. However, they focus on bi-lateral service requester/provider constellations neglecting the SLA management requirements of composite service providers, i.e. managing SLAs with atomic service providers and with composite service requesters and aligning both with each other. A SLA management solution for composite services has to consider the contribution of sourced services - formalized in their (atomic) SLAs (ASLA) - in the management of the provided service - formalized in its respective (composite) SLA (CSLA). Since composite services are created on-the-fly also their SLA management must be realized on-the-fly. Manual SLA management is not appropriate for composite service providers and automation support is required, i.e. for creation, monitoring and evaluation of SLAs.

In this paper, an approach for the management of SLAs involved in composite services during their entire lifecycle is presented (COSMA - COmposite Sla MAnagement). COSMA can be utilized for controlling the relationships between SLAs and thus serves as a basis for managing and optimizing the dynamics between SLAs of composite services. The paper is structured as follows: section 2 presents the central idea behind the COSMA approach and briefly presents its constitutional elements. Section 3 refers to a use case and demonstrator. Section 4 concludes the paper.

2 Composite SLA Management Approach (COSMA)

The central idea behind the COSMA approach is the integration of all SLAs a composite service provider has to deal with into one composite SLA management document. This composite SLA management document, which is defined as COSMAdoc, contains all contractual information of all involved SLAs and in addition the relationships and dependencies that exist between the different aspects of atomic and composite SLAs. On the basis of a COSMAdoc, the SLA management system of a CSP is able to understand how atomic SLAs contribute to the provision of the composite SLA. This knowledge enables a CSP to control and optimize its SLA management activities in providing a composite service. This includes, in particular, planning and negotiating SLAs, monitoring and evaluating SLAs (cf. SLA lifecycle as outlined in [6]). The COSMA approach consists of the following three parts:

– COSMAdoc: A generic information model that integrates contractual data, SLA management data, and elements for the expression of dependencies and relationships between SLA elements.

- COSMAframe: A conceptual framework that outlines the components that are necessary for the management of the composite SLA lifecycle on the basis of a COSMAdoc instance.
- COSMAlife: An integrated set of SLA management practices that use COSMAdoc instances to cover the phases of the SLA lifecycle.

COSMAframe, COSMAdoc, and COSMAlife are embedded into an operation and management system (OMS) of a CSP, i.e. platform as proposed in [7].

2.1 Information Model COSMAdoc

As the informational heart of COSMA, COSMAdoc provides a generic information model that encapsulates contractual and management information of SLAs. It comprises a set of SLA documents (CSLA and ASLAs) and elements for the expression of relationships and dependencies between those SLAs. For every composite service an individual instance of COSMAdoc is created and bound with the service. For all management activities defined in COSMAlife, the COSMAdoc instance is used. Different components of COSMAframe execute their tasks on the basis of the COSMAdoc instance; thus, it is the basis for performing complete, composition-wide SLA negotiations and compliance evaluations. COSMAdoc serves as an internal composite SLA management tool used by a CSP; embedded SLA documents carry only contractual data and can be exposed to involved parties without publishing COSMAdoc instance-internal management information. At the top-level, COSMAdoc consists of the following six core sections (Fig. 1):

- Header: The purpose of the Header section is to declare contextual information and parameters for the COSMAdoc instance (description, owner, version number, definition of semantic models and languages).
- ServiceComposition: The ServiceComposition section includes the orchestration script. Since the script expresses the general relationships and structure between all involved services, it governs the mapping and dependencies between SLA parameters of composite and atomic SLAs.
- SlaSetAssembly: The purpose of the SlaSetAssembly section is to capture all SLAs involved in a composite service. A SLA element carries the contractual information between the involved two parties. It carries no information that may be used by the CSP to manage the mapping of ASLA elements to CSLA elements. Hence, each SLA document can be used to represent a public agreement between the involved two parties and can be accessed by both parties. This ensures separation of concerns and provides a sound contractual basis by means of a valid SLA document. The COSMAdoc SLA model is based on the comprehensive WS-Agreement SLA model [4] for bilateral agreements. The COSMAdoc SLA model comprises of the sections Name, Context, Terms as defined in WS-Agreement and extends the specification with diverse aspects, i.e. MonitoringTerms, NegotiationTerms, and FinancialTerms in the Terms section.

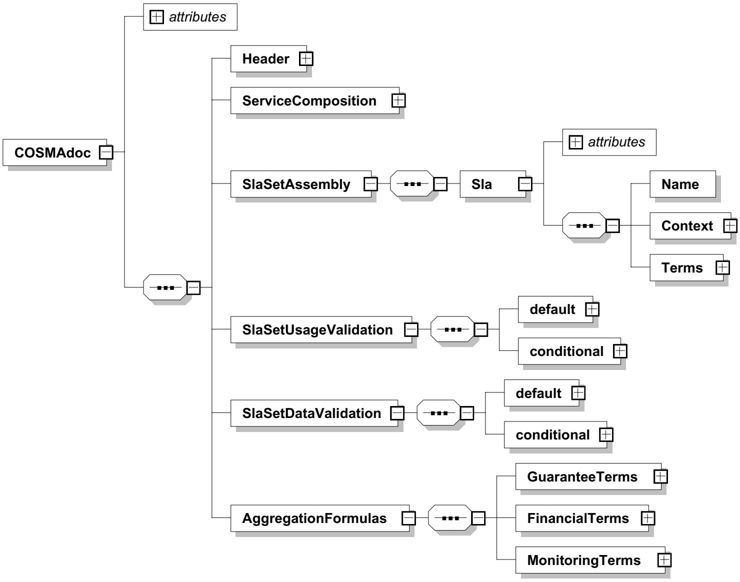

Fig. 1. COSMAdoc information model

– SlaSetUsageValidation: The SlaSetUsageValidation section is used to define specific requirements and constraints on the SLA elements of the SlaSetAssembly. These requirements and constraints regard only the content usage of the involved SLA documents (not the data). For instance, elements of a SLA can be declared as mandatory, negotiable, or optional. The technique used to identify elements of the SLA is the definition of pointers. Predicates can be defined as default or depending on a condition.
– SlaSetDataValidation: While the SlaSetUsageValidation section controls the usage of the involved SLA elements, the SlaSetDataValidation controls the data of the SLA elements. It provides means to explicitly enforce, validate, and check the data values of the involved SLAs by defining predicates on them, i.e. setMaxValue, setValueRange etc. Since most of the data validation restrictions on the contents of a CSLA result from the contents of ASLAs, predicates on CSLA elements may refer to aggregation formulas that calculate the value from other ASLA elements. Thus, a predicate with a link to an aggregation formula connects and restricts values of SLAs, i.e. defined in GuaranteeTerms, with aggregation formulas defined in the AggregationFormulas section (cf. Fig. 2).
– AggregationFormulas: Formulas stored in the AggregationFormulas section reflect the relationships and dependencies between SLA elements in composite services. Aggregation formulas can use all types of algorithmic operators and reference SLA elements using pointers. The complexity of these formulas

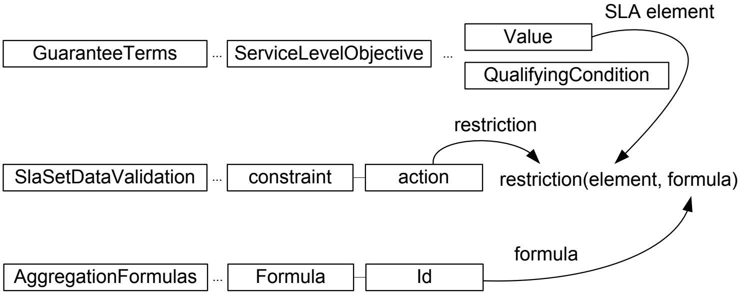

Fig. 2. Connecting SLA elements, aggregation formulas, and predicates

depends on the structure of the service composition script and the number and type of SLA parameters. Generic aggregation patterns for different SLA parameters were proposed i.e. in [9].

2.2 Conceptual Framework COSMAframe

COSMAframe is a conceptual framework that outlines the components that are required for an automated processing of COSMAdoc instances. COSMAframe presents these components in terms of their generic interfaces, basic behaviours, and functions. COSMAframe consists of the following components (cf. Fig. 3):

- The COSMA Manager is the central management component that triggers all components depending on the necessary steps within the SLA lifecycle. It is the central port of COSMAframe towards external components and provides a COSMAdoc validation interface that processes COSMAdoc aggregation formulas and predicates.
- The COSMAdoc Creator is responsible for creation of COSMAdoc instances. Based on a given generic service composition script provided by an external Service Composer, it creates composition-specific COSMAdoc instances using a composition decomposer.
- The composition-specific COSMAdoc instance is integrated by the COSMAdoc Integrator. The COSMAdoc Integrator adds different SLA contents, like quality of service parameters or financial parameters, to the included SLA documents. Afterwards, the component inserts different types of restrictions that evolve from the structure of the service composition script and the types of SLA parameters to the COSMAdoc instance.
- COSMAdoc instances are stored in the COSMAdoc Repository.
- The COSMAdoc Validator and Violation Detector compares service level measurements against service level objectives defined in ASLAs. It normalizes and aggregates the monitoring data across all ASLAs using stored aggregation formulas and is then able to state whether service levels of the CSLA are violated or not. If a SLA violation is detected, a proposal how to deal with the violation is created and returned to the COSMA Manager.

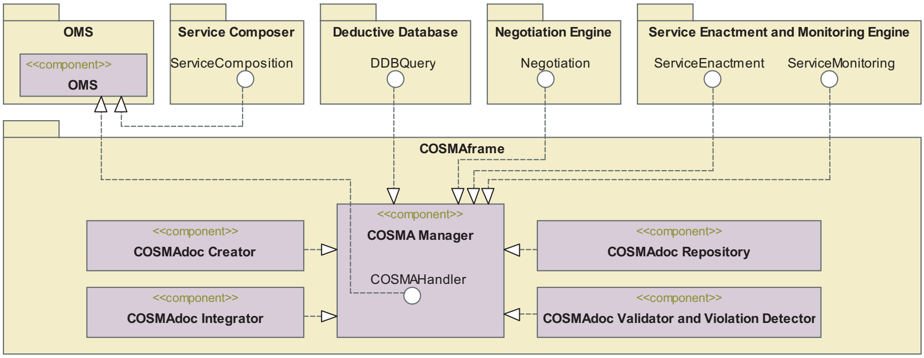

Fig. 3. Components of COSMAframe and external components

External components, i.e. Service Composer, Negotiation Engine etc., are described on the Adaptive Services Grid project website [7].

2.3 Composite SLA Management Lifecycle Mechanisms of COSMAlife

COSMAlife presents a set of composite SLA management mechanisms that cover different phases of the SLA lifecycle presented in [9]. The focus in COSMAlife is given to phases SLA creation and negotiation and SLA monitoring and evaluation. Briefly, a successful COSMAlife run-through conists of three steps: (1) creating and integrating, (2) negotiating, and (3) enacting/monitoring/validating a COSMAdoc instance. First, the COSMAdoc Creator creates a composition-specific instance from the generic COSMAdoc template. It uses an internal composition decomposer to atomize the generic service composition script into its generic composition patterns. Afterwards, the COSMAdoc Integrator integrates SLA content pre-settings, the service composition script, and SLA parameter-specific usage and data validation restrictions. Second, a Negotiation Engine determines the concrete atomic service implementations to use in the composite service (the Negotiation Engine hosts negotiation agents, whereas each negotiation agent negotiates a single SLA document). To ensure that the negotiation outcomes of ASLA negotiation processes are optimized with regard to the CSLA negotiation, the Negotiation Engine iteratively uses a validation interfaces provided by the COSMA Manager which processes predicates/aggregation formulas of the COSMAdoc instance. Third, after enactment of the composite service, monitoring and evaluation of the involved SLAs is executed. The COSMA Manager receives monitoring results from the Monitoring Engine (as specified in COSMAdoc) and sends these data to the COSMA Validator and Violation Detector for validation. In case of SLA violations, proposals for consequential actions are determined. The component uses the AggregationFormulas of COSMAdoc to detect service level violations and decide on the best type of action.

3 Use Case and Demonstrator

In order to illustrate the application of the COSMA approach and to support the understanding of COSMAdoc, COSMAframe, and COSMAlife, a use case was developed. The use case is published in [10]. In addition, a demonstrator was developed which implements the key elements of COSMA.

4 Conclusions

The paper addressed the topic of managing the dynamics of SLAs in composite services. For this, the novel composite SLA management approach COSMA was outlined in brief. With COSMA, a composite service provider can control and optimize its SLA management activities, pro-actively plan financial consequences, and dynamically calculate the expected service level objectives from dynamically varying service composition scripts. COSMA represents a conceptual approach for management of SLAs in composite services and is to be interpreted as a starting point that must be extended and adapted to individual requirements of arbitrary scenarios. Thus, the proposed, theoretical approach has to be tested extensively and employed on a number of different scenarios.

References

1. Papazoglou, M.P.: Web Services: Principles and Technology. Prentice Hall, Essex (2007)
2. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web services: concepts, architectures and applications. Springer, New York (2004)
3. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. J. of Network and Systems Management 11, 57–81 (2003)
4. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web Service Agreement Specification (WS-Agreement), http://www.gridforum.org/documents/GFD.107.pdf
5. Tosic, V., Patel, K., Pagurek, B.: WSOL - Web Service Offerings Language. In: Bussler, C.J., McIlraith, S.A., Orlowska, M.E., Pernici, B., Yang, J. (eds.) CAiSE 2002 and WES 2002. LNCS, vol. 2512, pp. 57–67. Springer, Heidelberg (2002)
6. Ludwig, A., Braun, P., Kowalczyk, R., Franczyk, B.: A Framework for Automated Negotiation of Service Level Agreements in Service Grids. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 89–101. Springer, Heidelberg (2006)
7. Integrated Project Adaptive Services Grid (ASG), http://www.asg-platform.org
8. Momotko, M., Gajewski, M., Ludwig, A., Kowalczyk, R., Kowalkiewicz, M., Zhang, J.Y.: Towards adaptive management of QoS-aware service compositions. J. of Multiagent and Grid Systems 3, 299–312 (2007)
9. Jaeger, M.C., Rojec-Goldmann, G., Muehl, G.: QoS aggregation for Web service composition using workflow patterns. In: 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC 2004), Monterey, pp. 149–159 (2004)