Model-Driven Service Level Management

Anacleto Correia, Fernando Brito Abreu

HAL Id: hal-01056646

https://hal.inria.fr/hal-01056646

Submitted on 20 Aug 2014

Model-Driven Service Level Management

Anacleto Correia[1,2], Fernando Brito e Abreu[1]

[1] Faculdade de Ciências e Tecnologia/Universidade Nova de Lisboa, Caparica
[2] Escola Superior de Tecnologia/Instituto Politécnico de Setúbal, Setúbal
accorreia@fct.unl.pt, fba@di.fct.unl.pt

Abstract. Service-level agreements (SLA) definition and monitoring are open issues within the IT Service Management (ITSM) domain. Our main goals are to propose a model-based approach to IT services SLA specification and compliance verification. The specification part will be accomplished by proposing a SLA language - a domain specific language (DSL) for defining quality attributes as non functional requirements (NFRs) in the context of ITSM. Its metamodel will be an extension of the meta-model of an adopted process modeling language for IT services. As such, it will be possible to ground SLA definition on the corresponding IT service model constructs. This will allow that SLA monitoring and compliance validation could occur at a level of abstraction that is understood by all the stakeholders involved in the service specification.

1 Introduction

Most organizations rely on Information Technology (IT) services to support their business processes. IT services are built upon the technical infrastructure (servers and network devices) as well as on systems and application software. The set of processes that allow planning, organizing, directing and controlling the provisioning of IT services is called an IT Service Management (ITSM) framework. Several ITSM frameworks have been proposed, such as the ITIL [1]. In ITIL, one of the most relevant processes is service level management [2]. In order to support the IT services, IT providers use two kind of applications: service management applications, which allow to track IT services incidents and problems; and systems management tools, for monitoring and control networks, systems or applications [3].

Among the concerns within the service level management process, are, for instance, the requirements for services availability, performance, accuracy, which are specified in terms of service-level agreements (SLA) [4]. In a SLA, such requirements, known as quality attributes, are quantitatively bounded (e.g. maximum time to recover or repair.). However, although those requirements are non-functional in nature, and many non-functional requirements techniques have been proposed in the literature [5], those techniques are not being used in SLA specification.

SLA definition and monitoring are identified as open issues within the ITSM domain [6]. In this research plan we are going to focus our attention on three specific problems in this context: The first problem arises from the current practice in SLA specification for IT services, mostly based in templates filled with natural language descriptions [7], and results not amenable to the automation of SLA compliance verification. Since SLA definition are subjective in nature it renders the second problem: it is not clear to all involved stakeholders which are the activities upon which SLA compliance should be verified, and how their non-compliance will affect the evolution in the process describing the IT service delivery. The third problem concerns a semantic gap which occurs in SLA compliance verification. This gap comes from the fact that compliance checking has been relying mostly in Quality of Service (QoS) information from systems management applications, instead of consolidated data presented at the process model representation of the IT service.

In the rest of this paper we will present a survey on the current state of the art of SLAs (section 2) and our proposal to deal with SLAs in the ITSM domain (section 2).

2 Related Work

When discussing related work, it is useful to have a common taxonomy. A taxonomy-based analysis fosters a more systematic study of different approaches and also helps identifying the issues, which are not overcome by current approaches, and our proposal intends to tackle. Our taxonomy for describing the entries of the survey of the current state of the art in SLA specification and compliance validation (see Table 1) is based in the following dimensions: **Domain**: it refers to the technological context at which the SLA is defined and evaluated. (1) *ITSM* – regarding processes of an ITSM framework; (2) *SOA* – concerning web services interacting in an SOA environment; (3) *System Management* – related with services of IT infrastructure assessed by QoS parameters. **Formalization**: it refers to the level of formalism used for SLA specification. (1) *Formal* – mathematically-based techniques amenable of rigorous proof; (2) *Semi-formal* – although the representation might not be a complete description of reality, the representation itself and the system description are equivalent; (3) *Informal* – descriptions in natural language usually using a structured template form. **Abstraction Level**: it refers to the kind of representation used in SLA. (1) *Model-based* – diagrammatic models for SLA elicitation and specification; (2) *Hybrid* – a blended approach using a diagrammatic notation supplemented by a textual language; (3) *Textual* – specification exclusively based in textual language. **Verification Compliance**: it refers to the level at which the SLA validation occurs. (1) *Model-based* – the SLA compliance is checked at the level of models; (2) *Message-based* – the compliance verification is based in the exchanged messages; (3) *Data-based* – the compliance verification is based on information stored in databases or the like repositories, by applications that deal with and support special types of IT infrastructure or specific IT processes; (4) *Absent* – there is no SLA compliance checking.

The most well-known specifications related to SLA are for web services: the language and framework *Web Service Level Agreement* (WSLA) [8] is a specification

and reference implementation developed by IBM that provides detailed SLA specification capabilities that enable the runtime monitoring of SLA compliance; the *Web Services Agreement Specification* (WS-Agreement) specification [9], defines an XML-based language for agreements as well as a protocol for advertising the capabilities of service providers, creating agreements between service consumers and providers, and monitoring agreement compliance. It also defines a protocol for negotiating and establishing agreements dynamically based on web services. Other works for the specification of quality properties are the *Web Services Offering Language* (WSOL) [10], the *Web Service Management Layer* (WSML) [11] and the *Web Service Modeling Ontology* (WSMO) [12].

Other projects related with SLA specification are: Rule-*Based Service Level Agreements* (RBSLA) [13] that provides a set of abstract language constructs based on *RuleML* to define SLA/policy rules and uses knowledge representation concepts for the specification of SLA; and the *SLAng* [14], a language for SLA, defined using a combination of *Meta-Object Facility* (MOF), *Object Constraint Language* (OCL), and natural language. All the mentioned approaches are related with SOA, therefore they only address computational processes and are not able to deal with processes where humans are involved such the ones covered of ITSM. Moreover the verification of compliance is made at an implementation instead of at a model level most suitable for common stakeholders.

There are also available, a number of service management applications that support the ITSM service level management process [15], as well as systems management tools ,[16] that monitor network, IT devices and applications at QoS parameters level. Albeit the former applications are related with the ITSM domain, they have issues, as we highlighted in section 1, such as the informal specification of SLA and the monitoring and compliance verification, based mostly in scattered and not integrated repositories of data. The latter applications have also these kinds of issues, and moreover their domain is mainly the technical infrastructure.

Table 1. Summary of current SLA approaches to specification and validation.

Approach	Domain	Formalization	Abstraction Level	Verification Compliance
[8] [9] [10] [11] [12] [13]	(2)	(1)	(3)	(2)
[14]	(2)	(1)	(2)	(2)
[15]	(1)	(3)	(3)	(3)
[16]	(3)	(3)	(3)	(3)

3 SLA specification and validation Environment

Our proposal for the specification and validation of SLA compliance of IT services is supported by an environment, the *SLA specification and validation Environment* (SLAEnv). The SLAEnv architecture addresses two main phases in the SLA life cycle: the SLA *specification phase*, when a negotiation process takes place among the

stakeholders; and the *validation phase*, which includes the monitoring, reporting, evaluating and improving of the SLA.

The negotiation step aims to define the conditions of the SLA contract. An *SLA Editor* supports this activity. The SLA Editor must be built based in the *Process Metamodel* and *SLA Metamodel*, using a *DSL Tool*. For the Process Metamodel definition the foundation is the BPMN 2.0 specification [17]. On the other hand the SLA metamodel specification is based in techniques of non functional requirements (NFRs) elicitation [18]. The output of the SLA Editor is an *SLA Contract* expressed in a SLA language, compliant with the SLA metamodel, and additional integrity constraints expressions written in the OCL.

The conditions of an SLA contract are the input for an *SLA Evaluator*. System states (snapshots of a running system) could be created and manipulated during an evaluation. Information about a system state will be attained by querying the *Process Modeler Animator* for IT service execution. For each snapshot OCL constraints can be checked. By evaluating OCL expressions we could gather detailed information about the system's SLA conformance. Graphical views (eg. dashboard or balanced scorecard) of system's state can be provided to the *SLA Visualizer*.

The *Process Modeler Animator* is a BPMN console where IT services are depicted and animated as BPMN process diagrams. An IT service specification, can be persisted in a standard format (e.g. XMI), which will be the Process Model repository. This definition is required for the SLA contract definition, since it will allow that hooks, concerning SLA quality attributes, could be associated to BPMN diagram elements (activities, gateways, data, etc.). Those hooks will allow the SLA Evaluator to trace an IT service instance execution to the corresponding SLA contract quality attributes thresholds. Therefore, the SLA Evaluator is the corner stone of the environment since it joins the specification and validation phases of an SLA. This is done by matching the SLA contract assertions with the IT service data execution, and making available the resulting snapshot to the visualization component.

This approach, unlike the mentioned in section 2, is tailored for SLA monitoring and compliance verification in the ITSM domain. Moreover, it allows a semi-formal, and model based specification and conformance checking of SLA. The validation process of this proposal will take place in an IT service management project on the domain of financial self-service systems.

4 Conclusions

Since SLA definition and monitoring are open issues within the ITSM domain, this research work intends to address it by proposing a model-based approach to IT services SLA specification and compliance verification. The specification part will be accomplished by proposing a SLA language. This language will be derived by way of a domain specific language that is intended to be able to specify the quality attributes, also known as non functional requirements, in the context of ITSM. The main contribution of this proposal is to allow SLA compliance verification to be made at the same abstraction level of specification, i.e., at a model level, filling the semantic gap between stakeholders.

References

1. ITIL3Sm, OGC-Office of Government Commerce. Summary, ITIL Version 3: ITSMF- IT Service Management Forum; 2007
2. Bon Jv, Pieper M, Veen Avd eds. Foundations of IT Service Management: Based on ITIL, ITIL Version 2: Van Haren Publishing; 2005
3. Mayerl C, Vogel T, Abeck S. SOA-based Integration of Service Management Applications. In, Proceedings of the IEEE International Conference on Web Services: IEEE Computer Society; 2005
4. Sallé M. IT Service Management and IT Governance: Review, Comparative Analysis and their Impact on Utility Computing. In: Hewlett-Packard Research Labs; 2004
5. Surhone LM, Timpledon MT, Marseken SF eds. Non-functional Requirement: Systems Engineering, Requirements Engineering, Requirement, Functional Requirements, Constraints

Betascript Publishing; 2010
6. Coyle DM, Brittain K. Magic Quadrant for IT Service Desk. In: Gartner 2009
7. Wedemeyer M, Menken I. The ITIL V3 Service Management Awareness Pocket Guide. In: The Art of Service; 2007
8. IBM. WSLA Language Specification, version 1.0. In; 2001
9. OGF, Open Grid Forum, Alain Andrieux, Czajkowski K, et al. Web Services Agreement Specification (WS-Agreement). In; 2007
10. Tosic V, Lutfiyya H, Tang Y. Web Service Offerings Language (WSOL) Support for Context Management of Mobile/Embedded XML Web Services. Telecommunications, 2006 AICT-ICIW '06 International Conference on Internet and Web Applications and Services/Advanced International 2006 156 - 156
11. Verheecke B. Web Services Management Layer (WSML). In; 2003
12. W3C. Web Service Modeling Ontology (WSMO). In; 2005
13. Paschke A. RBSLA A declarative Rule-based Service Level Agreement Language based on RuleML. Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce 2005;02:308 - 314
14. Skene J, Lamanna D, Emmerich W. Precise Service Level Agreements. proceedings of the 26th International Conference on Software Engineering (ICSE'04), Edinburgh, Scotland 2004
15. PinkElephant. PinkVERIFY 3.0 Toolsets. In; 2010
16. Gläßer L. Development and Operation of Application Solutions 2005/2006. In: Siemens Business Services; 2005
17. BPMN, OMG. Business Process Model and Notation (BPMN) In; 2009
18. Matoussi A, Laleau R. A Survey of Non-Functional Requirements in Software Development Process. In: D´epartement d'Informatique Universit´e Paris 12; 2008