

A Novel Approach to Identify Sullied Terms in Service Level Agreement

R. Parameshwari
PG scholar,
Dept. Of CSE,
SCSVMV University,
Enathur, Kanchipuram.

C.K. Gomathy
Assistant Professor,
Dept. Of CSE,
SCSVMV University,
Enathur, Kanchipuram.

ABSTRACT

A service-level agreement (SLA) is simply a file relating the level of service expected by a customer from a dealer laying out the metrics by which that examination is calculated, and the remedies or penalty, if any, must agreed-upon levels not be absolute. Typically, SLAs are between companies and external suppliers, but they may also be among two departments within a company. Service Level Agreements (SLAs) are typically used to specify regulations about the utilization of services that are agreed between the providers of the Service-Based Applications (SBAs) and their consumers. An SLA includes a list of conditions that contain the guarantees that must be fulfilled during the provisioning and use of the services. While the abuse of such guarantees may lead to the application of possible penalties, it is important to assure that the SBA behaves as predictable.

Index Terms

Service oriented Architecture, Service Based Applications, Automated Software Testing, Service Level Agreements and Coverage Criterion.

1. INTRODUCTION

The Service Level Agreements (SLAs) are used in the scope of Service Oriented Architectures (SOA) as a typical formalism to identify the conditions that control the trading between the service providers and the consumers. These contracts represent a set of guarantee terms that contain the projected Quality of Service (QoS) that must be delivered during the provision and consumption of the services. Generally, the violation of a term of the SLA leads to negative penalty for the stakeholders, for example, the payment of penalty fees. Due to this, it is main that both providers and consumers try their utmost to avoid or minimize the SLA violations and their corresponding penalty

The discoveries of these SLA violations is typically performed by observing the behavior of the Service Based Application (SBA) at runtime, recollecting data from the executions and make a choice about the valuation of the SLA. To do this, different monitoring techniques have been proposed and represent a good post mortem result in the intelligence that the troubles are detected after they have occurred in the operational surroundings Such troubles may be therefore analysed and solved so they are fewer likely to happen again. However, in particular scenarios where an SLA violation may lead to significant penalty for the stakeholders it is not recommendable to stay until the problems emerge at runtime. In these cases, the application of approaches allows the providers to cut or avoid the number of SLA violations and, hence, reduce the penalties

related to such violations. Among the fit-for-purpose tasks involved within the proactive detection of SLAs violations, testing has been identified as a challenge in the context of SOA-based research.

The objective of SLA-based testing is, on the one hand, to guarantee that the SBA satisfies the conditions specified in the SLA and, on the other hand, to declare that such SBA is able to perform properly constant when some of its constituent services violate the SLA. For example, a service may not be able to fulfil the agreed response time (perhaps because such service is down). As the response time is a condition specified in the SLA, we plan at identifying tests that exercise the situations where the service does not answer or it responds too much time to give the response. With these tests, we check that the application provides correct mechanisms to handle the sudden behaviour of the abovementioned service. At this stage, the SLA-based testing aims at anticipating as much as possible the detection of problems in the SBA and thereby avoid the penalty derived when such problems arise at runtime in the operational surroundings. In a previous work we addressed the classification of test requirements by analysing the individual guarantee terms of the SLA. In this article we offer a additional step by means of consider the whole logical structure of the SLA. As the number of test requirements may become unmanageable if the SLA is quite difficult we devise a coverage-based criterion with the plan at obtaining a compact but efficient set of test requirements.

The primary offerings of this article are:

1. Specification of a test criterion based on the MCDC coverage criterion [8] that allows the detection of test requirements by means of analysing the information represented in both the guarantee terms and their logical combinations. This condition makes use of a four-valued logic to evaluate SLAs, which is also clear
2. Definition of definite set of laws that add to avoid the identification of non-feasible situations, considering the framework of the SBA as well as the hierarchical structure of the SLA.
3. Computerization of the process that identifies the test requirements using the abovementioned condition

2. EXISTING TECHNIQUES

The existing SLAs in the engineering area seem to be quite easy today, they could become more complex by means of establish relations between the conditions or as well as information about the functional and non-functional features of the services as well as the penalties derived from the

violations of the agreed guarantees. In a previous work we deal with the detection of test requirements by means of analyzing the information contain in the individual guarantee terms. The joint drafting of SLAs, mechanism of measurement procedures and negotiation of SLAs are costly to both consumers and providers

a	b	a AND b
0	1	0
1	1	1
1	0	0

When a flips while b holds fixed, the outcome changes

When b flips while a holds fixed, the outcome changes

Fig.1. Example of application of MCDC

2.1 Service Level Agreements

Service Level Agreements (SLAs) are contracts that specify the regulations for the trading between the consumers and the Service Based Applications (SBAs) providers. Usually, these regulations specify which the essential services of the SBA that will be regulated by the agreement are, and how these services should be offered. Currently, important companies such as Google, Microsoft, AT&T, Amazon or HP, use SLAs as a guarantee for their clients to guarantee that their SBAs send the estimated Quality of Service (QoS).

2.2 WS-Agreement

WS-Agreement is an XML based language proposed by the Open Grid Forum (OGF) that specifies a set of rules for the concern of agreements between two parties. This set allows defining a hierarchical structure for the requirement of an SLA. The arrangement of an SLA using the WS-Agreement standard language is composed of three major parts. These are:

- Name, i.e., the part specifying an possible name that can be given to the contract
- Context, i.e., the part defining the parties involved in the contract and their job
- Terms, i.e., the part expressing the negotiated and granted obligations of each party.

These obligations are specified using Service Description Terms (SDT), Service Properties (SP) and Guarantee Terms (GT). The most important information of the SLA is represented by means of the Guarantee Terms, which describe the obligations that must be satisfied by a specific obligated party. A Guarantee Term (GT) contains the following internal elements: (1) the possibility specifies the list of services the term applies to, (2) the Qualifying Condition (QC) represents a requirement or statement that determines whether the term is applicable and must be considered during the evaluation process, (3) the Service Level Objective (SLO) specifies the guarantee that must be met. In WS-Agreement, the terms of the SLA can be logically and hierarchically grouped by means of Compositor elements. Specifically, there are three different types of compositor elements, namely: All, One or more and exactly One. These element types are correspondent to the logical AND, OR and XOR operators, correspondingly

2.3 Software Testing

In the context of service computing, the trying of SLA-aware SBAs has been posed as a demanding job. Generally, testing

is an movement in which the Software Under Test (SUT) is executed under specified conditions, the outcome are practical or recorded, and an estimation is made of some aspect of the system. The execution of the SUT is usually performed through the plan and execution of test cases. A test case is a set of input, execution circumstances and predictable results developed for a particular intention. The discovery of faults is addressed by means of executing the SUT and comparing the observed results with the expected results, determining whether the behaviour of the software is correct or not. It is therefore imperative that a good design of test cases should allow detecting the highest possible number of faults.

3. PROPOSED TECHNIQUE

The aim is to provide a more step by means of allowing for the whole logical structure of the SLA. As the number of test requirements may become impossible If the SLA is quite difficult we devise a coverage-based criterion with the aim at obtaining a reduced but effective set of test requirements. The testing of the SLAs requires using such documents as the test basis so we need to somehow formalize the specification of the SLA in order to automate as much as possible the obtaining of tests. Thus, in this work, we center on the syntax and semantics of WS Agreement

Methods:

3.1 Estimation of SLAS

The estimation of an SLA require analysing the information collected from the monitor and/or testers, checking the requirement of the guarantee terms and, lastly, making a choice about the fulfilment of such terms. It have outlined in Section 2 that an SLA specifies a set of terms that manage the execution of the component services of the SBA. Such guarantee terms can be hierarchically structured by means of using specific compositor elements (All, One or More, Exactly One). In this context, we identify two different levels concerning the evaluation of the SLAs.

Level I: Individual Guarantee Terms.

Level II: Composite Guarantee Terms

The first level involves making a decision about the fulfilment of each individual guarantee term represented in the SLA. The second level involves considering sets of Guarantee conditions understandably grouped by the compositor elements and determining whether the composite terms are fulfilled or not. In this section we propose a logic that allows evaluating both individual guarantee terms and composite terms, from a testing point of view, including all the potential situations derived from the task of assuring whether the SLA is being fulfilled or not.

Level I: Evaluating Individual Guarantee Terms

It focuses on each individual guarantee term in order to address the evaluation of the SLA. The evaluation of a guarantee term is usually performed in a dichotomy way, for example, depicting a two-way traffic light indicator (green/red) that indicates whether the term has been fulfilled or violated respectively. The use of these two classical values is really useful when the behaviour of the SBA is monitored at runtime in order to decide whether a problem has been detected, disregarding the situation that has caused such problem. However and from a testing point of view, we need to early identify the different potential situations that may derive in problems in the SBA. Aligning this perspective with the syntax of WS- Agreement, a Guarantee Term is

specified by means of the internal elements Scope, Qualifying Condition (QC) and Service Level Objective (SLO).

After taking this syntax into account, a Guarantee Term can be evaluated as:

Fulfilled (F) - if and only if the methods of the services specified in the Scope have been executed, the QC has been met and the SLO has been satisfied.

Violated (V) - if and only if the methods of the services specified in the Scope have been executed, the QC has been met and the SLO has not been satisfied.

Not Determined (N) - if and only if their methods of the services specified in the Scope of the guarantee term, which have not been executed.

Inapplicable (I) - if and only if the methods of the services specified in the Scope have been executed but the Qualifying Condition has not been satisfied.

Level II: Evaluating Composite Guarantee Terms

After having provided a systematic way to evaluate the individual guarantee terms, now we focus on the logical combinations of the SLA Guarantee Terms. It has been previously outlined that an SLA specified in WS-Agreement represents a hierarchical structure of guarantee terms, logically combined using the specific Compositor Elements All, One Or More and Exactly One. Thus, we need to complete the logic basis that allows evaluating the individual Guarantee Terms in order to unequivocally determine the evaluation value of these compositors.

3.2 Evaluation of One or More Compositor

Likewise, a one or more compositor element with multiple Guarantee Terms is evaluated as follows.

$$ev(OneOrMore_{i=1}^n(t_i)) =$$

Fulfilled if

$$\exists i \in [1, n] : ev(t_i) = F$$

Violated if

$$(\exists i \in [1, n] : ev(t_i) = V) \wedge (\forall j \in [1, n] : (ev(t_j) = V \vee ev(t_j) = I))$$

Not Determined if

$$(\exists i \in [1, n] : ev(t_i) = N) \wedge (\nexists j \in [1, n] : ev(t_j) = F)$$

Inapplicable if

$$\forall i \in [1, n] : ev(t_i) = I$$

3.3 Evaluation of Exactly One Compositor

Finally, an Exactly One compositor element with multiple Guarantee Terms is evaluated as follows:

$$ev(ExactlyOne_{i=1}^n(t_i)) =$$

Fulfilled if

$$(\exists i \in [1, n] : ev(t_i) = F) \wedge (\exists j, k \in [1, n], j \neq i : (ev(t_j) = F) \vee (ev(t_k) = N))$$

Violated if

$$(\forall i \in [1, n] : ev(t_i) = V) \vee (\exists j, k \in [1, n], j \neq k : (ev(t_j) = F) \wedge (ev(t_k) = F))$$

Not Determined if

$$(\exists i \in [1, n] : ev(t_i) = N) \wedge (\nexists j, k \in [1, n], j \neq k : (ev(t_j) = F) \wedge (ev(t_k) = F))$$

Inapplicable if

$$\forall i \in [1, n] : ev(t_i) = I$$

4. COVERAGE-BASED TEST CRITERION

The process of testing SLA-aware SBAs can be improved by identifying test requirements from the specification of the SLAs using a criterion based on the principle of the Modified Condition / Decision Coverage (MCDC) that allows obtaining a cost-effective set of test requirements, representing situations that are interesting to exercise regarding the SLA and the SBA. This set contains a reduced number of test requirements to be exercised in order to uncover problems in the SBA. Typically, MCDC is applied to a specification of the SUT. In our case, the specification that says how the SUT must behave is the SLA. In such SLA there are guarantee terms that represent conditions that can be satisfied or not. Hence, it is necessary to design tests that exercise situations in which the guarantee terms are fulfilled and situations in which not. Within this approach, these situations are obtained by means of the application of our MCDC-based criterion.

4.1 Four-valued MCDC Test Criterion

Once the logic to evaluate the compositor elements of an SLA has been defined, we obtain the test requirements by combining the potential evaluation values of the terms included in the compositors. Considering that each term can be evaluated with four different values, the amount of test requirements would grow exponentially with the number of terms if we applied all the possible combinations. Hence, our objective is to avoid the exponential growth of test requirements in order to obtain a reduced but cost-effective set of test requirements and we achieve it by using MCDC.

Modified Condition Decision Coverage (MCDC), de-fine studied structural coverage criterion it has also been used for test suite reduction and prioritization because it provides a linear increase in the number of test requirements. MCDC is a criterion that falls between condition/decision and multiple condition coverage. This criterion has been shown to represent a good balance of test-set size and fault detecting ability simultaneously. MCDC is defined as a conjunction of the following requirements:

- Every point of entry and exit in the program has been invoked at least once.
- Every condition in a decision in the program has taken all possible outcomes at least once.
- Every conclusion in the program has taken all possible effects at least once.

5. RESULTS

The below fig shows the design for user can view the test results of updated service level agreements.

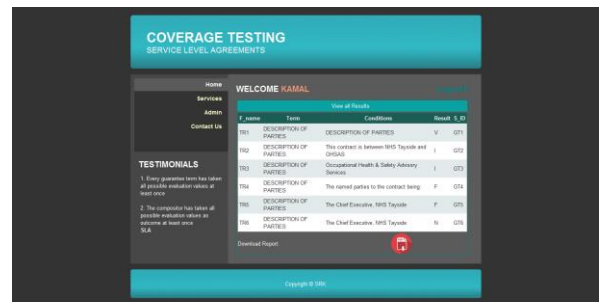


Fig.2: updated service level agreements.

The below fig shows the design for downloading results after qualifying conditions

ID	DESCRIPTION OF PARTIES	DESCRIPTION OF PARTIES	GTI
ID1	DESCRIPTION OF PARTIES	DESCRIPTION OF PARTIES	V
ID2	DESCRIPTION OF PARTIES	This contract is between NHS Topside and CHSAS	I
ID3	DESCRIPTION OF PARTIES	Occupational Health & Safety Advisory Services	I
ID4	DESCRIPTION OF PARTIES	The named parties to the contract being	F
ID5	DESCRIPTION OF PARTIES	The CHM Executive, NHS Topside	F
ID6	DESCRIPTION OF PARTIES	The CHM Executive, NHS Topside	N

Fig. 3 Download test results.

6. CONCLUSION

In this research paper focused to have addressed the estimation and testing of the logical composition of guarantee terms in a Service Level Agreement (SLA). It has defined a four-valued logic that allows evaluate both individual guarantee terms and compositor elements. This logic is the basis for the SLACDC (SLA Condition / Decision Coverage) criterion we have devised in order to identify a set of test requirements that combine different evaluation values of the terms involved in a compositor. This criterion is based on MCDC criterion and it provides a linear increase of test requirements regarding the number of guarantee terms included in the compositor.

In addition to this identification of test requirements, we have to deal with non-feasible situation due to the semantics of the SLA terms. To address this matter we have define a set of system, which are mechanically applied, that allow remove the non-feasible test requirements or, if possible, the change of such requirements in order to obtain feasible situations. The automation of the approach allows falling the tester's effort required to design and specify aligned with the SLA specification. Moreover, the analysis and exercitation of the test requirements also allow detecting wrong SLA specifications regarding the relationships between the guarantee terms.

7. REFERENCES

- [1] K. Mahbub and G. Spanoudakis, "Monitoring WS-Agreements: an Event Calculus Based Approach," *Test and Analysis of Service Oriented Systems*, Springer V., 2007, pp. 265-306.
- [2] C. Muller, M. Oriol, M. Rodriguez, X. Franch, J. Marco, M. Resinas, and A. Ruiz-Cortés, "SALMonADA: A Platform for Monitoring and Explaining Violations of WS-agreement-Compliant Documents" *ICSE Workshop on Principles of Engineering Service Oriented Systems (PESOS)*, pp. 43-49, 2012.
- [3] F. Raimondi, J. Skene, and W. Emmerich, "Efficient Online Monitoring of Web-Service SLAs," *Proc. 16th ACM SIGSOFT Int. Symp. on Foundations of Software Engineering (FSE-16)*, 2008.
- [4] L. Baresi, N. Georgantas, K. Hamann, V. Issarny, W. Lamersdorf, A. Metzger, and B. Pernici, "Emerging Research Themes in Services-Oriented Systems," *SRII*

- Global Conference (SRII), 2012 Annual*, vol., no., pp.333-342, 24-27 July 2012.
- [5] M. Palacios, J. García-Fanjul, and J. Tuya, "Testing Service Oriented Architectures with Dynamic Binding: a Mapping Study," *Information and Software Technology*, vol. 53 (3), pp. 171-189, 2011.
- [6] G. Canfora and M. Di Penta, "Testing Services and Service-Centric Systems: Challenges and Opportunities," *IT Professional* 8 (2), pp. 9-17, 2006.
- [7] M. Palacios, J. García-Fanjul, and J. Tuya, "Identifying Test Requirements by Analyzing SLA Guarantee Terms," *Proc. 19th Int. Conf. on Web Services*, pp. 351-358, 2012.
- [8] RCTA Inc. DO-178-B: Software Considerations in Airborne Systems and Equipment Certification. Radio Technical Commission for Aeronautics (RTCA), 1992.
- [9] PLASTIC European Project. <http://www.ist-plastic.org>
- [10] C.K.Gomathy,Dr.S.Rajalakshmi "A Software ability network in Service Oriented Architecture", *International Journal of science and Technology Education Research(IJSTER)*, Volume 5,Issue II, June 2014,P.No:7-14, ,ISSN : 2141-6559.
- [11] M. Autili, P.D. Benedetto, and P. Inverardi, "Context-Aware Adaptive Services: The Plastic Approach," *Proc. 12th Int. Conf. In Fundamental Approaches to Software Engineering (FASE)*, York, UK, March 22-29, 2009. *Proc. LNCS*, vol. 5503, pp. 124-139.
- [12] A. Bertolino, G. De Angelis, L. Frantzen, and A. Polini, "Model-Based Generation of Testbeds for Web Services," *Proc. Testing of TESTCOM/FATES*, LNCS, vol. 5047, 2008, pp. 266-282.
- [13] Google Apps SLA: <http://www.google.com/apps/intl/en/terms/sla.html>
- [14] Amazon EC2 SLA: <http://aws.amazon.com/ec2-sla/>
- [15] E. Di Nitto, M. Di Penta, A. Gambi, G. Ripa, and M.L. Villani, "Negotiation of Service Level Agreements: an Architecture and a Search-Based Approach," *Proc. 5th Int. Conf. Service-Oriented Computing (ICSOC)*, September 17-20, pp. 295-306, 2007.
- [16] F.H. Zulkernine and P. Martin, "An Adaptive and Intelligent SLA Negotiation System for Web Services," *IEEE Trans. Services Computing*, vol. 4, no. 1, pp. 31-43, Jan.-March 2011.
- [17] C.K.Gomathy,Dr.S.Rajalakshmi, "Software Architecture design using Service oriented on Quality Metrics", *Australian Journal of Computer Science (AUJCS)*, Volume I,Issue I March 2014,P.No:09-16,ISSN:2251-3221
- [18] M. Palacios, L. Moreno, M.J. Escalona, and M. Ruiz, "Evaluating the Service Level Agreements of NDT under WS-Agreement. An Empirical Analysis," *Proc. 8th Int. Conf. on Web Information Systems and Technologies*, Porto, Portugal, April 2012.
- [19] D.M. Quan and L.T. Yang, "Parallel Mapping with Time Optimization for SLA-Aware Compositional Services in the Business Grid," *IEEE Trans. Services Computing*, vol. 4, no. 3, pp. 196-206, July-Sept. 2011. doi: 10.1109/TSC.2011.27

- [20] H. Wada, J. Suzuki, Y. Yamano, and K. Oba, "E3: A Multiobjective Optimization Framework for SLA-Aware Ser-vice Composition," *IEEE Trans. Services Computing*, vol. 5, no. 3, pp. 358-372, Third Quarter 2012. doi: 10.1109/TSC.2011.6
- [21] M. Di Penta, G. Canfora, G. Esposito, V. Mazza, and M. Bruno, "Search-Based Testing of Service Level Agreements," *Proc. Annual Conference on Genetic and Evolutionary Computation (GECCO 07)*, London, ACM, New York, 2007, pp. 1090-1097.
- [22] M. Palacios, "Defining an SLA-aware Method to Test Service-Oriented Systems," *Proc. 9th Int. Conf. on Service Oriented Com-puting (ICSOC)*, PhD Symposium, G. Pallis et al. (Eds.): ICSOC 2011, LNCS 7221, pp. 164--170. Springer, Heidelberg 2012.
- [23] C.K.Gomathy,Dr.S.Rajalakshmi, "A Software design Pattern for Bank Service Oriented Architecture", *International Journal of Advanced Research in Computer Engineering and Technology(IJARCET)*, Volume 3,Issue IV, April 2014,P.No:1302-1306, .ISSN : 2278-1323
- [24] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, "Web ser-vices agreement specification (WS-Agreement)," 2010.
- [25] ISO/IEC 24765, *Software and Systems Eng. Vocabulary*, 2006.
- [26] IEEE STD 610.12-1990, *IEEE standard glossary of software engineering terminology*.