NEGOTIATION AND MONITORING OF SERVICE LEVEL AGREEMENTS*

Thomas B. Quillinan
D-CIS Lab
Thales Research and Technology
Delft, The Netherlands
thomas.quillinan@d-cis.nl

Kassidy P. Clark, Martijn Warnier, Frances M.T. Brazier
Systems Engineering
Faculty of Technology, Policy and Management
Delft University of Technology
The Netherlands
[k.p.clark, m.e.warnier, f.m.brazier] @tudelft.nl

Omer Rana
School of Computer Science/Welsh eScience Centre
Cardiff University, UK
o.f.rana@cs.cardiff.ac.uk

Abstract Service level agreements (SLAs) provide a means to define specific Quality of Service (QoS) guarantees between providers and consumers of services. Negotiation and definition of these QoS characteristics is an area of significant research. However, defining the actions that take place when an agreement is violated is a topic of more recent focus. This paper discusses recent advances in this field and propose some additional features that can help both consumers and producers during the enactment of services. These features include the ability to (re)negotiate penalties in an agreement, and specifically focuses on the renegotiation of penalties during enactment to reflect ongoing violations.

Keywords: monitoring, penalties, negotiation, automation, ws-agreement

*This paper extends preliminary work reported at the 5th International Workshop on Grid Economics and Business Models [12]

1. Introduction

A Service Level Agreement (SLA) is an agreement between clients and providers in the context of a particular service provision. SLAs may be between two parties, for instance, a single client and a single provider, or between multiple parties, for example, a single client and multiple providers. SLAs specify Quality of Service (QoS) properties that must be maintained by a provider during service provision – generally defined as a set of Service Level Objectives (SLOs). Often an SLA is only relevant when a client *directly* invokes a service (rather than through an intermediary – such as a broker). Such direct interaction also implies that the SLOs need to be measurable, and must be monitored during the provision of the service.

Significant work exists on how SLOs may be specified and monitored, such as [7]. Furthermore, some work has focused on actually identifying how SLOs may be impacted by the choice of specific penalty clauses [3, 12, 13]. A trusted mediator may be necessary to resolve conflicts between involved parties. The outcome of conflict resolution depends on the situation: penalties, impact on potential future agreements between the parties and the mandatory re-running of the agreed service, are examples. While it may seem reasonable to penalize SLA non-compliance, there are a number of concerns when issuing such penalties. For example, determining whether the service provider is the only party that should be penalized, or determining the type of penalty that is applied to each party.

Automating conflict resolution processes can provide substantial benefits. In general, there are two main approaches for contractual penalties in SLAs: reputation based mechanisms [1, 5] and monetary fines. It is useful to note that often obligations within an SLA are primarily centered on the provider towards the client. An SLA is therefore an agreement between the provider to offer particular QoS to a client for some monetary return. This paper does not consider scenarios where there is also an obligation on the client towards the provider. An example of such a scenario could be where a provider requires the client to make input data available by a certain time frame to ensure that a particular execution time target is met. If the client is unable to meet the deadline for making such data available, the penalty incurred by the provider would no longer apply. However, similar techniques to those outlined in this paper will apply. Moreover, this paper assumes the Grid's client/provider division of tasks, but could also be extended to apply to agreements between two independent entitities.

An aspect of penalising violations of existing agreements that has not received much attention is how agreements could, or should, be renegotiated during the enactment of the service. A graduated response to violations, as proposed by [12], functions as a post facto reaction to the violation that allows some level of violations with a lesser penalty. Alternatively, providing a renegotiation mechanism has specific advantages: for example, graduated responses can lead to implicit incentives for bad behaviour, up to the threshold. For example, if the provider had the option to provision a second consumer knowing that the graduated penalty would be less that the reward for a second service provisioning. In contrast, a renegotiation mechanism allows a more reactive system, where if conditions have altered, producers and consumers can alter the agreement to match those conditions.

In this paper, a number of recent proposals (from [3, 8, 12, 13]) in the area of service violations and penalities negotiation, as well as suggesting approaches that could be used to support renegotiation of SLAs during enactment. The remainder of this paper is organised as follows: Section 2 discusses WS-Agreement and how it can be used to formulate agreements between clients and providers. Monitoring these agreements is discussed in Section 3, where different methodologies are examined. Violations of SLAs is examined in Section 4 and a number of proposals for the future of SLA negotiation and penalites are offered. Finally, Section 5 concludes the paper.

2. Background

WS-Agreement [2] provides a specification for defining SLAs, and is undergoing standardisation by the Open Grid Forum (OGF). WS-Agreement is an XML document standard, that is, interactions between clients and providers are performed using an XML standardized format. There are two types of XML documents in WS-Agreement: *templates* and *agreements*. One basic element is that agreements need to be confirmed by both parties. Including penalties in a WS-Agreement, for example, cannot be one-sided. The WS-Agreements needs to be confirmed by the client. The existing WS-agreement specification, however, will need to be extended to include this step. Mobach *et. al.* [9] proposed such an extension in the context of the WS-Agreement specification.

Figure 1 shows the extended interactions between a service provider (SP) and a consumer (C) described by [9]. The advertisement phase uses WS-Agreement template documents; the request and offer phases also use WS-Agreement agreement documents. Templates describe the different services that the provider supports. When a negotiation takes place, the service provider sends these templates to the consumer. The consumer then makes an offer to the provider and, if acceptable, the agreement is created by the provider based on the offer. In Figure 1, the initial template is generate by the provider, in accordance with the WS-Agreement specification.

Templates and agreements both use the concept of negotiation *terms*. Terms define the service description and guarantees about the service. Guarantees are

1. SP → C : Advertisement
2. C → SP : Request
3. SP → C : Offer
4. C → SP : Acceptance/Rejection

Figure 1. Negotiation using WS-Agreement

made relating to the service, such as the quality of service and/or the resource availability during service provision.

Agreements have a name defined by the provider and a context that contains meta-information about the agreement. This meta-information can include identifiers for the service provider and the agreement initiator; the name of the template that the agreement is based on; references to other agreements, and the duration of the agreement [9], as agreements have a fixed period when they are valid. Functional and non-functional requirements are specified in the *Terms* section. This is divided into the *Service Description Terms* (SDT) and *Guarantee Terms* (GT). A SDT holds the functional requirements for the delivery of services, and may refer to one or more components of functionality within one or more services. There may be any number of SDTs in a single agreement. GTs hold a list of services that the guarantee applies to, with the conditions that this guarantee applies, and any potential pre-conditions that must exist.

3. Monitoring Violations in SLAs

Monitoring plays an important role in determining whether an SLA has been violated, and thereby determine the penalty clause that should be invoked as a consequence. From a legal point of view, monitoring appears as a pre-requisite for contract enforcement. The basic requirement is a set of ‘consequences’ for breaching the agreed SLOs. Service clients base their trust in service providers largely on the provided monitoring infrastructure.

Monitoring facilitates a direct and automated SLA enforcement mechanism at run-time and without undue delay (that is, once a SLA violation is recorded, the agreed sanction can be automatically triggered). Monitoring also facilitates a more traditional enforcement. In either case, if the provider or the client contests the automatic sanction imposed, it can use monitoring data to argue its case. It is therefore vital to monitor all those metrics that have legal relevance and to give the parties the possibility to retrieve such data in a format that is admissible as evidence.

Identification of violations is either discovered through online monitoring or post facto auditing of the service enactment. However, while auditing allows definitive decisions to be made, it is necessary that accurate logs are maintained by the parties.

3.1 Online Monitoring

Monitoring an agreement requires periodically testing whether the agreement terms have been met by all relevant parties. Depending on the agreement terms, this either entails testing a specific variable, such as network latency, or logging communication between consumer and provider. Monitoring intervals are specified appropriately, such as daily or hourly, depending on the duration of the agreement and the nature of the agreement terms. Monitoring must also support both simple and complex evaluation formulae. For instance, some requirements can be verified by measuring a single variable, such as '*Host is reachable*'. However, other requirements can only be verified once a set of measurements have been performed and their results stored, such as '*Host uptime is greater than 99%*'.

A monitoring mechanism must take accurate measurements and be secure against malicious parties, including the parties with whom agreements have been reached. For instance, a log of communication should not be write accessible to the parties involved: a secure logging mechanism [4] is required. Non-repudiation is also of importance to prove that certain messages were sent by a certain party, and a mechanism to prevent forged messages containing false timestamps or false measurements from being inserted into the message log is also required.

Furthermore, where a monitoring module is placed has to do with trust and objectivity. [12] distinguishes three possible locations for monitoring:

- **Trusted Third Party (TTP)**: an independent module that can monitor (and log) all communication between consumers and service providers. Once the SLA is successfully completed, both parties receive a signed ticket from the TTP that can be used for non-repudiation and/or reputation building of the service provider. However, a TTP cannot measure the internal state of either the consumer or provider.

- **Trusted Module at service provider**: functionally equivalent to a TTP but with access to the internal state of the service provider. However, the provider may not reveal all of the internal state or may report incorrect information to the monitor. A module at this location can show that the provider attempted to avoid violations and dealt with them responsibly when they occurred.

- **Trusted Module on the consumer site**: functionally equivalent to a TTP but it can be difficult to distinguish between provider delay and network delay. A module at this location is not particularly useful for measurements, but only for establishing the trust level for certain providers.

An alternative to this online monitoring approach is to reactively monitor agreements [8]. Reactive monitoring takes place when one of the parties involved in the enactment 'complains' to the monitor that violations are taking place. Such a scheme has the advantage of allowing an immediate response to violations, without the overhead of monitoring every active service provisioning. In the worst case using the approach of [8] entails the same overhead as an online monitoring mechanism.

Passive monitoring is an offline monitoring scheme that uses cryptographic primitives to prove that specified checkpoints in the enactment have been reached correctly. Reactive monitoring extends such a passive monitor, in that, should a checkpoint not be reached, the monitor reacts and starts actively monitoring the enactment from that point forward.

3.2 Violations and Penalties

When a violation occurs, typically a penalty is incurred as a consequence. Penalties can be as simple as terminating the current agreement and finding a different provider, or more complex reputation or monetary based penalties [11]. These penalties are commonly used for service provisioning [6]. In these systems, reputation is a community-wide metric of an entity's trustworthiness. This metric increases if the entity completes transactions without violating the agreement. Conversely, the metric decreases if a term is violated. Reputation based penalties utilise the notion that consumers prefer providers with a higher reputation and try to avoid providers with a lower reputation. In contrast, monetary based penalties operate on the assumption that consumers pay less for poor service and more for better service.

Both of these mechanisms require additional infrastructure and security measures [6]. A reputation based system requires a persistent record of all transactions, both successful and violated. A monetary based system requires a secure means of payment, whether in currency or credit, that has actual value to the users of the system. Both of these approaches require a means of guaranteeing that identities are unique, persistent and legitimate, as well as a conflict resolution process. For instance, underlying authentication mechanisms using a PKI can verify that users are indeed whom they claim to be.

Deposits with a jointly agreed TTP can be used in a monetary based system to implement penalties if needed. In the event of violation, the deposit can be used to effectuate penalty payment. The exact penalty terms can be separately negotiated during SLA negotiation or according to known policies, such as the following [12]:

- **All-or-nothing provisioning**: provisioning of a service must meet all SLOs. ALL of the SLO constraints MUST be met to satisfy the SLA;
- **Partial provisioning**: provisioning of a service must meet some SLOs. SOME of the SLO constraints MUST be met to satisfy the SLA;
- **Weighted Partial provisioning**: provision of a service meets SLOs that have a weighting GREATER THAN a [user specified] threshold.

For example, the SLA framework in AgentScape [10] has been extended to support monitoring and penalty enforcement [3]. A trusted monitoring module is required to measure the provided services and ensure that the GTs in the SLA are being fulfilled by both parties. In addition to this trusted monitoring module, the SLA document must be extended to include monitoring and penalty clauses, similar to those described in [14]. This includes specifying the item to be measured, time constraints, and the method to be used for measurement as described the following example.

Negotiation of the violation policy is also required to determine, for example, the severity of a violation and appropriate action using the policies introduced above. [12] proposes negotiating this violation policy as a separate SDT during the negotiation phase.

4. Negotiation of Penalties

While negotiations can be managed in the existing WS-Agreement framework, this does not adequately reflect the complexity of penalty negotiation. For example, if a mutually trusted third party cannot be agreed upon by both consumer and provider, there is little point in proceeding with the SLA negotiation. Similarly, if an SLA cannot be agreed upon, there is no need to negotiate the penalty clause. Therefore it is instead proposed to separate these three stages into distinct negotiation steps. Each of these steps follows the same steps as shown in Figure 1: Advertisement; Request; Offer, and Acceptance/Rejection. These steps can be considered negotiations for three separate services.

For example, negotiations to select a TTP proceeds as follows: In the *Creation Constraint* section of the WS-Agreement template, the TTPs trusted by the service provider are listed. When the consumer receives this template, it creates an agreement offer specifying the TTP that they have selected. The offer is then processed by the provider. If it is acceptable, the provider produces the agreement document. This is passed to the consumer for acceptance/rejection. Negotiations for the SLAs and penalties are handled using the same process.

One concern with this approach is the verification that a SLA template refers to the TTP agreement previously negotiated and, similarly, the penalty template to the SLA and TTP agreements. This is achieved by the use of the references to the prior agreements within the *context* section of proceeding templates and agreements. Each penalty agreement then contains references to the TTP and SLA agreements. This ensures that a verifiable link is maintained throughout the service negotiation and provision.

Another approach to the multi-step process could be to specify the template and agreement documents as a single document, with separate services for each of the three stages. This would eliminate the need for three separate negotiations. However, this approach would make the templates more complicated.

4.1 Multiround Negotiation

While negotiation of penalites may become a standard part of SLA negotiations, it is an area where multiple rounds of negotiation has potential to be particularily useful. Several proposals are under consideration by the GRAAP working group of the Open Grid Forum (OGF) towards supporting multiround negotiation. One possible approach is to allow 'negotiation offers'. Such offers would form a non-binding template offering suggestions what *might* be acceptable to the offering party. Such offers would form part of a session where multiple rounds of offers could be provided by one or both parties without altering the fundamental principles of WS-Agreement. Such an approach has the advantage of allowing a rollback mechanism. As each offer forms part of a session and each session has a unique identifier within the negotiation, if negotiations are diverging, the parties would have the option to revert to a previous session.

Multiround negotiations have the potential to allow agreement to be reached on both the service QoS details as well as any penalty clauses. Session based negotiation also has the advantage of allowing both the functional and non-functional aspects to be negotiated seperately, where options could be negotiated, such as, "service *x* with deadline *200*, penalty *$50*, price *$200*" or "service *x* with deadline *500*, penalty *$10*, price *$100*". This would allow much more flexibility to both the provider and consumer. However, no such multiround negotiation mechanism exists within the current WS-Agreement specification.

4.2 Renegotiation

While current work focuses on penalising violations of SLAs, one alternative approach is to *renegotiate* the SLA during enactment. For example, such an approach would allow the producer and consumer to alter the SLA towards providing a more realistic deadline for the consumer and potentially reducing any penalties that the producer would otherwise be subject. Such a mechanism could take advantage of multiround sessions that formed part of the original negotiation. For example, if a previous round had a longer deadline, at a lower price, the renegotiation might take the form of both parties agreeing to select this SLA session as a replacement. However, this would entail both parties storing the entire set of negotiation sessions until after provisioning has been completed.

Another approach to renegotiation would be for a new round of negotiations to take place during the enactment of the service. An example of this approach would use a version of the existing WS-Agreement negotiation framework, with the initial positions reflecting the current state of the enactment. This renegotiation process could form the penalty associated with an existing SLA. This would allow the enactment to proceed without explicit penalities. Such an approach has the advantage of allowing implicit *penalties* to be negotiated only when required and making these penalties reflect the exact situation rather than the more abstract penalties that would be determined beforehand.

5. Discussion and Conclusion

The use of penalties in SLAs has obvious benefits for both clients and service providers. Monetary sanctions (and optionally reputation based mechanisms) can be used as, pre-agreed, penalties. Both of these approaches require the participation of a Trusted Third Party. The types of monitoring infrastructure that can be used to validate SLOs during service provisioning are identified. As monetary sanctions are the *de facto* standard in industry for penalty clauses, these are preferred over reputation based solutions, though the latter can be used if so required.

While explicit penalties can be specified within the WS-Agreement framework, they lack flexibility when unexpected events interrupt enactment. This paper discusses the use of both multiround negotiation and runtime renegotiation of SLAs towards improving the experience for both service providers and consumers. While such mechanisms are, as yet, undefined, they indicate an interesting area of future research and usability of service level agreements.

Acknowledgments

This work supported by the NLnet Foundation (`www.nlnet.nl`) and the EU FP7-IST-215890 “ALIVE" project. We are grateful to Dana Cojocarasu (Research Center for Computers and Law, University of Oslo, Norway) for contribution and discussion regarding European law for electronic contracts.

References

[1] *The EigenTrust Algorithm for Reputation Management in P2P Networks.*, Budapest, Hungary, 2003. ACM Press.

[2] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. Web services agreement specification (ws-agreement).

[3] K. P. Clark, M. Warnier, T. B. Quillinan, and F. M. T. Brazier. Secure monitoring of service level agreements. In *Proceedings of the Second International Workshop on Organizational Security Aspects (OSA 2010)*. IEEE, March 2010.

[4] L. Gymnopoulos, S. Dritsas, S. Gritzalis, and C. Lambrinoudakis. GRID security review. *Lecture Notes in Computer Science*, pages 100–111, 2003.

[5] J. Sabater and C. Sierra Social regret, a reputation model based on social relations. *SIGecom Exch.*, 3(1):44-56, 2002.

[6] A. Jfisang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[7] A. Keller and H. Ludwig. The wsla framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management*, 11(1):57–81, March 2003.

[8] D. Khader, J. Padget, and M. Warnier. Reactive monitoring of service level agreements. In *In the Service Level Agreements in Grids Workshop proceedings*, 2009.

[9] D. G. A. Mobach, B. J. Overeinder, and F. M. T. Brazier. A ws -agreement based resource negotiation framework for mobile agents. *Scalable Computing: Practice and Experience*, 7(1):23–36, 2006.

[10] B. J. Overeinder and F. M. T. Brazier. Scalable middleware environment for agent-based Internet applications. In *Applied Parallel Computing*, volume 3732 of *Lecture Notes in Computer Science*, pages 675–679. Springer, Berlin, 2006.

[11] T. B. Quillinan, B. C. Clayton, and S. N. Foley. GridAdmin: Decentralising grid administration using trust management. In *Proceedings of the Third International Symposium on Parallel and Distributed Computing (ISPDC04)*, Cork, Ireland, July 2004.

[12] O. Rana, M. Warnier, T. B. Quillinan, and F. M. T. Brazier. Monitoring and reputation mechanisms for service level agreements. In *Proceedings of the 5th International Workshop on Grid Economics and Business Models (GenCon)*, Las Palmas, Gran Canaria, Spain., August 2008. Springer Verlag.

[13] O. Rana, M. Warnier, T. B. Quillinan, F. M. T. Brazier, and D. Cojocarasu. Managing violations in service level agreements. In *the Proceedings of the Usage of Service Level Agreements in Grids Workshop*, September 2007.

[14] A. Sahai, V. Machiraju, M. Sayal, L. J. Jin, and F. Casati. Automated sla monitoring for web services. In *IEEE/IFIP DSOM*, pages 28–41. Springer-Verlag, 2002.