Temporal-Awareness in SLAs: Why Should We Be Concerned?*

C. Müller, A. Ruiz-Cortés, and P. Fernández

Dpto. Lenguajes y Sistemas Informáticos
ETS. Ingeniería Informática - Universidad de Sevilla (Spain - España)
41012 Sevilla (Spain - España)
{cmuller,aruiz,pablofm}@us.es

Abstract. Traditionally, Service Level Agreements have been decomposed in two sets of properties: functionals (what) and non-functionals (how). However, in our opinion, there has been a third key element that has had a minor attention from academy: temporal awareness (when). We believe temporality is a main concern that should be addressed in realistic scenarios. In doing so, this position paper discuss our experience in extending the specification WS-Agreement with a temporal Domain Specific Language; importantly, main aim of the paper is to provoke a debate about the importance of temporality in SLAs.

1 Introduction

As service-oriented computing (SOC) [18] paradigm is being widely adopted in industry, new challenges need to be considered to address the problems of complex scenarios. The core infrastructure stack behind SOC has been proved as solid pilar to articulate new specifications to deal with more abstract business requirements. In particular, a key element is that gaining importance is the traditional service level agreement (SLA) idea to enrich the basic SOA platform. In doing so, SLAs will help to express and guarantee the appropriate quality of service (QoS) demanded by the different participants of a SOC scenario.

However, in spite of the extensive work carried out on SLAs in the context of SOC [7,8,15,16], we believe there has been not needed minor attention to an important issue that must be promoted as a first-class element of SLAs: the temporality.

Traditionally, SLAs are expressed using a set of properties in order to specify "*which*" service is offered and "*how*" it is offered. That is to say, it includes requirements and guarantees about functional (FP), and non-functional properties (NFP) of the services. However, another important question about services is "*when*". In our opinion, temporal awareness is top concern in order to deal with realistic scenarios because they usually have strong temporal restrictions at different levels that should be met: the entire agreement (e.g. *the agreement*

* This work has been partially supported by the European Commission (FEDER) and Spanish Government under CICYT project Web-Factories (TIN2006-00472).

E. Di Nitto and M. Ripeanu (Eds.): ICSOC 2007 Workshops, LNCS 4907, pp. 165–173, 2009.

expires on 2007/05/31), any FP of the service (e.g. *this operation of the service is available from 8:00h to 18:00h*), and/or any NFP that appears in the SLA (e.g. *the response time is 30 ms from 8:00h to 17:00h and 15 ms from 17:00h to 8:00h*).

In order to deal with the previous issues, in [13] we propose a framework to express temporal-aware SLA by extending the specification WS-Agreement (WS-Ag) [7]. This specification is a proposed recommendation of the Open Grid Forum working group (OGF) that provides both a structural schema for specifying SLAs and a protocol for creating them based on a mechanism of templates. It is important to highlight that, due to compatibility and complexity reasons, WS-Ag specification only defines a general structure of the agreement; complementary, in order to enrich the core structure, WS-Ag leave extensibility points for domain-specific extensions like [1], [17] and [14].

Based on the experience of the authors in creating the temporal-awareness extension to WS-Ag, the main goal of this position paper is to analyse the key topics of temporalities in SLAs: *when it is useful?* or *which is the correct approach?* are some of the questions we raise. Additionally, with this discussion we aim to boost a debate to make temporality the main concern to be addressed by the SLA community.

The rest of the paper is structured as following: In section 2, we outline the main issues of our temporal-aware DSL for extending Ws-Ag. Next, in section 3, we analyse from a critical point of view our experience. In section 4, we summarise a set of discussion topics based on our experience. Finally, we present the conclusions.

2 Improving Temporal-Awareness of WS-Ag in a Nutshell

2.1 WS-Agreement in a Nutshell

WS-Ag specifies an XML-based language and a protocol for advertising the capabilities of service providers, creating agreements based on agreement offers. The structure of an agreement for WS-Ag is comprised of:

- **Name**: identifies the agreement and can be used for reference.
- **Context**: it includes information such as the name of the parties and their roles of initiator or responder of the agreement. Additionally, it can refer to an agreement template if needed. In this element, an agreement lifetime can be defined by means of an element called "ExpirationTime".
- **Terms**: agreement terms are wrapped by term compositors, which allow simple terms or sets of terms to be denoted by "ExactlyOne", "OneOrMore", or "All". The two main types of terms are: (a) *Service terms*: they provide information to instantiate or identify services and operations involved in the agreement. Additionally, it can comprise of information about the measurable service properties. (b) *Guarantee terms*: they describe the service level objectives (SLO) agreed by the parties. It also includes the scope of the term (e.g. a certain operation of the service or the whole service itself);

a "QualifyingCondition" that specifies the validity conditions under which the term is applied; and information about business properties in the "BusinessValueList" element such as "Importance", "Penalty" or "Reward" and "Preference".

In order to create agreements, WS-Ag allows us to specify templates with the above structure, but including agreement "Creation Constraints" that describe the variability allowed by a party and it should be taken into account during the agreement creation process. The element "Item" is used in case we need to describe a creation constraint for a specific part of the agreement.

2.2 Temporal-Awareness of WS-Agreement

WS-Ag identifies two locations to include temporal awareness. On the one hand, lifetime for the entire agreement must be included in Context into the "ExpirationTime" element (i.e. the last instant where the agreement is valid). On the other hand, WS-Ag recommends the use of "QualifyingCondition" elements for describing validity periods of terms and/or the party preferences. However, the specification document leaves open the specific way that temporal awareness must be exposed for reasons of compatibility and complexity.

2.3 Our Temporal DSL for WS-Agreement

Figure 1 shows an UML class diagram which represents the schema we proposed in [13] for describing temporal properties in SLAs. An interval is the basic element, different non-disjoint intervals can be grouped together so that more

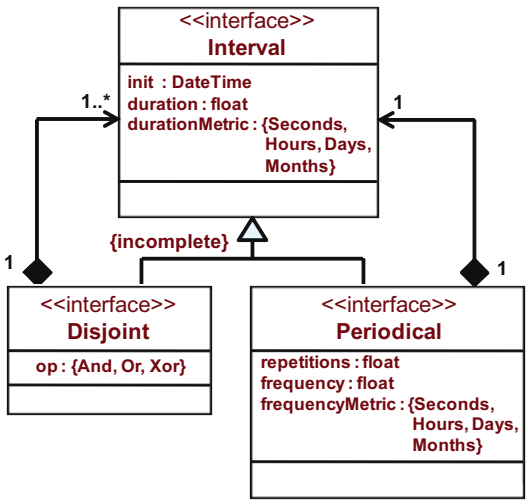

Fig. 1. Schema for Temporal Intervals

complex intervals can be composed. The three interfaces denote the different types of intervals:

- **Interval**: stands for the basic element; is comprised of an initial time and a duration (which can be infinite) expressed in seconds, hours, days, or months.
- **Disjoint**: stands for disjoint intervals constituted of a set of intervals related by a logic operator (or, and, or xor).
- **Periodical**: stands for periodic intervals, either disjoint or non-disjoint. Its periodicity is comprised of the number of period repetitions (which can be infinite) and a frequency expressed in seconds, hours, days, or months, which denotes the time between two consecutive intervals.

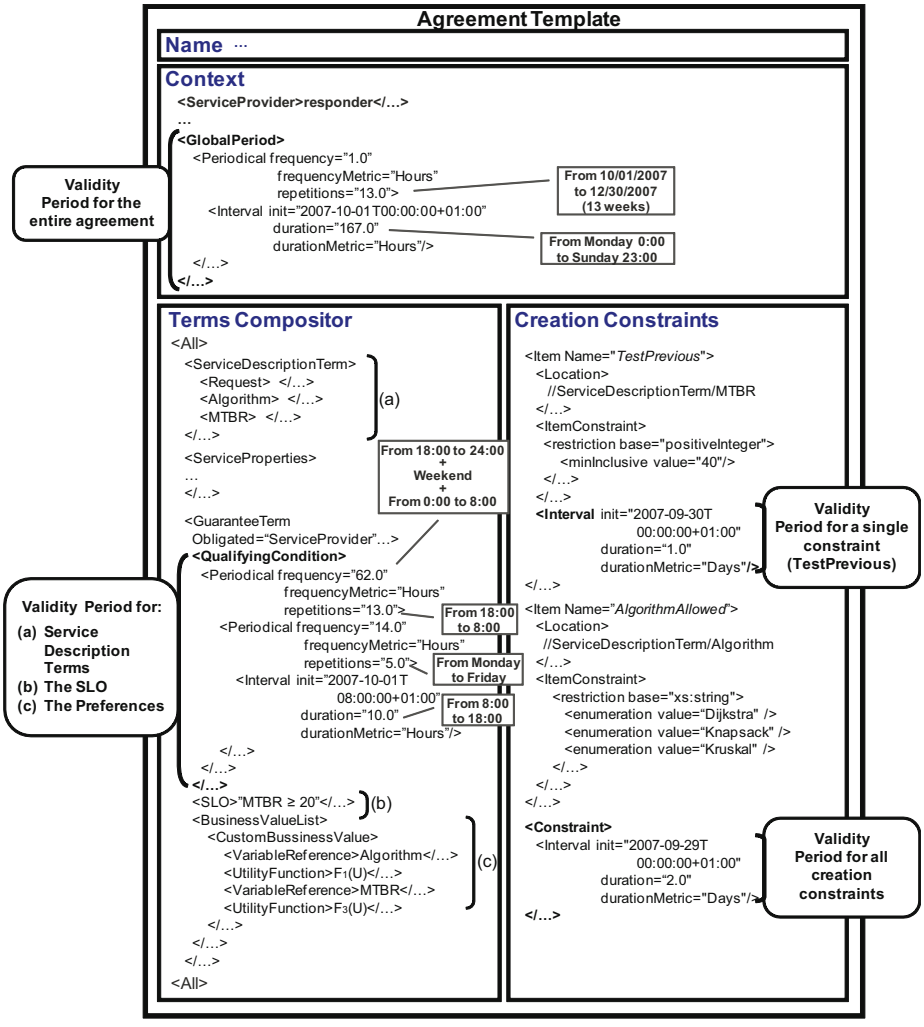

Fig. 2. An example of temporal-aware WS-Ag Template

Our proposal allows to include temporality regarding several aspects of agreements. Figure 2, shows a particular case study where a provider offers computing services to other organizations with a template. The Figure includes where we improve the Temporal-Awareness into the WS-Ag structure.

In this scenario, a provider is likely to be looking for an optimization in the usage of its resources. Thus, agreement offers should vary in a certain period by the mean time between two consecutive requests (MTBR) within the period.

- The global validity of the SLA is from october 1/2007 to december 30/2007.
- All Sundays at 23:00h. servers are down for an hour due to maintenance. This affects the global validity period.
- The Provider needs part of its server resources for its own computing necessities from Mondays to Fridays, from 8:00h to 18:00h. Therefore, in such time period, the provider requires that consumers specify in their service requests a MTBR greater or equal to 20 seconds.
- The provider allows requests with "Dijkstra", "Knapsack", or "Kruskal" algorithm allowing execution tests *48 hours before agreement initiation date* with any value of the MTBR. But *24 hours before agreement initiation date* the provider constrains these tests with a minimum MTBR of 40s.

3 Complexity on Defining Our DSL

Most of the time, design is a challenging task with a high degree of uncertainty. In particular, we face the challenge of creating a DSL to express the temporal requirements that we could find in real-world SLAs. At the end of the designing process we found that, choosing the WS-Ag gives us a consistent framework. In this section, we analyse with a more critical point of view, the advantages and drawbacks of our approach in order to spread our know-how in the design of the temporal extension so it could be used as a basis for further debate on temporal-aware SLAs.

The expressiveness and succinctness of a DSL are key factors for considering it as a good proposal or not. Our temporal DSL is the result of several works where we have already studied temporality on web services: In [11], we presented a constraint-based approach to temporal-aware web services procurement. In [12], we elaborated a study about expressiveness of temporal descriptions for web services. And in [13], while improving temporal-awareness of WS-Ag, we reviewed the kinds of temporal periods defined in several proposals from semantic and traditional web service specifications and additionally we studied the periods defined by the *IETF RFC 3060* [20]. We concluded that the expressiveness of our proposal is similar to theirs.

In relation to succinctness it is important to remark that as we show in [13], we consider the periodical validity periods as a main concern in the agreements domain rather than simply using a set time intervals (choice made by most of the approaches found in the literature [2,3,4,6,9,10,21,22]). In this context, to the best of our knowledge the only proposal with explicit periodic terms is IETF RFC 3060 which includes a large amount of properties such as "MonthOfYearMask",

"DayOfMonthMask" and other syntactic structures that allows the definition of this kind of period as concise as our definition. On the contrary, other proposals that do not include periodic time periods can be defined as *less succinct* due to the amount of effort that must be carried out to express a periodic behaviour. As an example, we would need 1000 period definitions for describing a single periodic interval from our proposal with a value of 1000.0 for repetitions property.

The effort to express periodic behaviour even increases if the repeated period of time is discontinuous. Our proposal allows a simple definition of periodic and discontinuous periods of time by mean of a set of intervals defined inside the disjoint element over which the periodicity is defined. Additionally, a common limitation found in other proposals is related with the possibility of specifying restrictions only over certain parts of the agreement; this limitation can be described as a main drawback in the expressiveness of the DSL.

Another important issue is related with parties preferences for an SLAs. In general, each of the parties that sign a particular agreement, have to specify in a concrete manner its preferences. In WS-Ag, this is made by means of an agreement template that should be completed for the other party. In this preference context, we found a little set of proposals , among those which are temporal-aware, that have taken the preferences into account. However, to the best of our knowledge, none of them have studied temporality on preferences and creation constraints.

We distinguish two ways to declare the preferences: (1) by comparing the "degree of similarity" between values of service properties from different agreement offers and templates; for example, if a provider specifies in the agreement template that it prefers a value of MTBR of 30s, an agreement offer which requires a MTBR of 32s will be more similar to the template than another offer requiring 20s; and (2) by comparing utility values given by utility functions defined on the service properties, just as we have described above. Both alternatives use weights as a means of incorporating the degree of importance among service properties to the preferences. UDDIe [2], EWSDL [3] and METEOR-S [17] have based their preferences on the degree of similarity, whereas other proposals have only mentioned their interest. Regarding utility functions, WSMO/WSML [5], Gonzalez et al. [6] and METEOR-S [17] are currently working on incorporating this feature to their proposals. Therefore in preferences our proposal is as expressive as most expressive proposals.

In addition, a key point addressed by our DSL has been to express a suitable semantic for determining the consistency of an agreement offer/template, the conformity between templates and offers, and the selection of the optimal offer for a given template.

In [11], we described the semantic of these operations applied to web services specifications and matchmaking processes but in this paper we consider their on the SLAs establishment context. More formally, we identify the following three operations that are highly common in fault-tolerant scenarios: (i) *consistency* (i.e an agreement document is consistent if it does not have contradictory terms); (ii) *conformity* (i.e. two agreement documents conform if the guarantee terms

from each party support the requirements to the other party); (iii) *optimality*, an agreement offer is optimal for a SLA when its guarantee terms give the best support to the requirements of the other party and vice versa.

Finally, while describing our temporal DSL this has lead us to discover a wide range of uses for temporality in WS-Ag, it is important to highlight the explicit inclusion of a temporal feature made in the specification; that is the lifetime for the whole agreement. As a complement, we can summarise the new range of possibilities when using our approach; depending on the target we could have temporal-aware *creation constraints* or *preferences over service properties*: On the one hand, we can express temporal creation constraints in the agreement creation process, e.g. "Provider must allow execution tests with a minimum MTBR of 40s, *48 hours before agreement initiation date* in order to create the agreement". In [13], we give more details for temporal creation constraints on single or multiple constraints. On the other hand, we allow the specification of temporal preferences over service properties.

Additionally, we define a schema for using any kind of utility functions over any set of service properties, instead of only constant float functions as described in WS-Ag. With this approach, we now formulate the preferences of the parties over a service property (or a set) versus other properties with any kind of mathematical function, leaving open the concrete language to specify those functions.

4 Discussion

Based on the experience of developing an improvement of temporal-awareness of WS-Ag we believe that some important topics must be discussed. In this section, we propose a set of key points that must be analysed in order to deal properly with temporality in real-world scenarios. Main aim of this list is to establish a rich discussion rather than expose an exhaustive analysis.

- To develop concrete DSL for describing temporal properties in SLAs based on real scenarios.
- To have a temporal DSL as expressive as possible in order to allow the description of a rich set of temporal constraints.
- To express this succinctly, temporality is made easier for its user allowing it to be more understandable and clearer for both parties.
- To apply the operations of consistency, conformity and optimality on the SLA establishment boosts an automation of the process.
- To allow the definition of variable preferences based on temporal-aware utility functions for supporting scenarios in which the parties preference may vary at times, such as "an increasing utility function $f(x)$ will be used in rush hour and a decreasing utility function $f'(x)$ will be applied at any other instant". This gives more expressiveness or more succinctness to our proposal.
- To allow the definition of temporal creation constraints. Thus, we could describe constraint such as "In order to create the agreement you must have paid the 50% of the total price a week before the agreement initation date".

5 Conclusions

In this paper, we show our experience in the extension of WS-Ag as the starting point to propose a set of key topics to start a discussion about the convenience of to consider temporal properties as a main concern in SLAs. As a conclusion we can enumerate the following issues: We consider the importance to extend the temporal-awareness WS-Ag by defining an appropriate domain-specific language. Based on the discussion presented, we believe the temporal DSL must have a wide expressiveness in order to allow as many temporal properties as possible. Although the succinctness is not essential for the DSL it makes it easier and more understandable to an user.

Additionally, to boost an automation of the process based on previous works [11,19] we should apply the operations of consistency, conformity and optimality on the SLA establishment. Furthermore, to support scenarios in where the parties preference which may vary from time to time we should allow the definition of variable preferences based on temporal-aware utility functions; e.g. the provider would vary its preferences according to the usage level of the resources. These kinds of preferences give more expressiveness to our proposal in a succinct way because a complex utility function described on a concrete period of time can be defined as several constant utility functions on consecutive instants.

Finally, we need to describe temporal constraints in the agreement creation process in order to allow that variability of the parties would vary from time to time too. This is why are concerned about temporal-awareness in SLAs. Our proposal is quite expressive, but what about efficiency in the automation of the process?. So, expressiveness on one side and efficiency on the other, can actually represent two sides of the same coin (You can't have one without the other). In order to obtain a good solution we should get a good balance between them.

References

1. Aiello, M., Frankova, G., Malfatti, D.: What's in an agreement?An analysis and an extension of WS-agreement. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 424–436. Springer, Heidelberg (2005)
2. Ali, A.S., Al-Ali, R., Rana, O., Walker, D.: UDDIe: An Extended Registry for Web Services. In: Proc. of the IEEE Int'l Workshop on Service Oriented Computing: Models, Architectures and Applications at SAINT Conference. IEEE Press, Los Alamitos (2003)
3. Chen, Y., Li, Z., Jin, Q., Wang, C.: Study on qoS driven web services composition. In: Zhou, X., Li, J., Shen, H.T., Kitsuregawa, M., Zhang, Y. (eds.) APWeb 2006. LNCS, vol. 3841, pp. 702–707. Springer, Heidelberg (2006)
4. Chen, Z., Liang-Tien, C., Bu-Sung, L.: Semantics in Service Discovery and QoS Measurement. IT Pro - IEEE Computer Society, 29–34 (2005)
5. de Bruijn, J., Feier, C., Keller, U., Lara, R., Polleres, A., Predoiu, L.: WSML Reasoning Survey (November 2005)
6. González-Castillo, J., Trastour, D., Bartolini, C.: Description Logics for Matchmaking of Services. Technical Report HPL-2001-265, Hewlett-Packard (2001)

7. OGF Grid Resource Allocation Agreement Protocol WG (GRAAP-WG). Web Services Agreement Specification (WS-Agreement), v. gfd.107 (2007)
8. IBM. Web Service Level Agreement (WSLA) Language Specification (2003)
9. Li, L., Horrocks, I.: A Software Framework for Matchmaking based on Semantic Web Technology. In: Proc. of the 12^{th} ACM Intl. Conf. on WWW, pp. 331–339 (2003)
10. Lodi, G., Panzieri, F., Rossi, D., Turrini, E.: SLA-Driven Clustering of QoS-Aware Application Servers. IEEE Transactions on Software Engineering 33(3), 186–196 (2007)
11. Martín-Díaz, O., Ruiz-Cortés, A., Durán, A., Müller, C.: An approach to temporal-aware procurement of web services. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 170–184. Springer, Heidelberg (2005)
12. Müller, C., Martín-Díaz, O., Resinas, M., Fernández, P., Ruiz-Cortés, A.: A WS-Agreement Extension for Specifying Temporal Properties in SLAs. In: Proc. of the 3^{rd} Jornadas Científico-Técnicas en Servicios Web y SOA (2007)
13. Mller, C., Martín-Díaz, O., Ruiz-Cortés, A., Resinas, M., Fernández, P.: Improving temporal-awareness of WS-agreement. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 193–206. Springer, Heidelberg (2007)
14. Di Modica, G., Tomarchio, O., Vita, L.: A framework for the management of dynamic SLAs in composite service scenarios. In: Proc. of the 1^{st} Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop (NFPSLA-SOC 2007), Vienna, Austria. Springer, Heidelberg (2007)
15. Molina-Jiménez, C., Pruyne, J., van Moorsel, A.: The role of agreements in IT management software. In: de Lemos, R., Gacek, C., Romanovsky, A. (eds.) Architecting Dependable Systems III. LNCS, vol. 3549, pp. 36–58. Springer, Heidelberg (2005)
16. OASIS and UN/CEFAT. Electronic business using XML, ebXML (2007)
17. Oldham, N., Verma, K., Sheth, A., Hakimpour, F.: Semantic WS-Agreement Partner Selection. In: 15^{th} International WWW Conf., pp. 697–706. ACM Press, New York (2006)
18. Papazoglou, M.P., Georgakopoulos, D.: Service oriented computing. Commun. ACM 46(10), 24–28 (2003)
19. Ruiz-Cortés, A., Martín-Díaz, O., Durán, A., Toro, M.: Improving the Automatic Procurement of Web Services using Constraint Programming. Int. Journal on Cooperative Information Systems 14(4) (2005)
20. The Internet Society. Policy Core Information Model - v1 Specification (2001)
21. Tian, M., Gramm, A., Naumowicz, T., Ritter, H., Schiller, J.: A Concept for QoS Integration in Web Services. In: Proc. of the IEEE Int'l Web Services Quality Workshop (WISE 2003), pp. 149–155 (2003)
22. Trastour, D., Bartolini, C., González-Castillo, J.: A Semantic Web Approach to Service Description for Matchmaking of Services. Technical Report HPL-2001-183