

# Service Level Agreements Modeling and Monitoring Using Ontologies

Kaouthar FAKHFAKH<sup>1,2,3</sup>, Tarak  
CHAARI<sup>1</sup>, Said TAZI<sup>1,2</sup>, Khalil  
DRIRA<sup>1</sup>

<sup>1</sup>LAAS-CNRS ; Université de  
Toulouse ; 7, avenue du Colonel  
Roche, F-31077 Toulouse, France

<sup>2</sup>Université de Toulouse ; UT1  
+33 (0)5 61 33 63 22. 31077

{kchaari,tchaari,tazi,khalil}  
@laas.fr

Mohamed JMAIEL<sup>3</sup>  
<sup>3</sup>ReDCAD;

Département d'Informatique et de  
Mathématiques Appliquées  
Ecole Nationale d'Ingénieurs de Sfax,  
Tunisie

+21698644167. 3038

mohamed.jmaiel@enis.rnu.tn

## ABSTRACT

Performance monitoring and optimization in the service-oriented architectures (SOA) have an important role in the provider/consumer relationship. The SLA model (Service Level Agreements) defines the quality of service (QoS) obligations and the guarantees when violations occur. However, the contract establishment and the automatic obligation monitoring become difficult to be achieved because (i) the knowledge differences between suppliers and customers, (ii) the measurement process complexity in case of composed QoS parameters and (iii) the diagnosis implementation difficulty to localize the errors. In this paper, first, we study and analyze existing SLA-related models. Then, we elaborate a complete, generic and semantically richer ontology-based model of SLA. We used the Semantic Web Rule Language (SWRL) to express SLA obligations in our model. This language facilitates the SLA monitoring process and the eventual action triggering in case of violations. We used this model to automatically generate semantic-enabled QoS obligation monitors. We have also developed a prototype to validate our model and our monitoring approach. Finally, we believe that this work is a step ahead towards the total automation of an SLA management process.

## Categories and Subject Descriptors

H1 [Models and Principles]: General; C4 [Performance Of Systems]; E2 [Data Storage Representation]; D2.2 [Software engineering]: Design Tools and Techniques.

## General Terms

Measurement, Design, Management

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

JFO 2008, December 1-2, 2008, Lyon, France.

Copyright 2008 ACM 978-1-60558-373-0/08/0003.\$5.00.

## KEYWORDS

Service Level Agreements, Monitoring, Quality of services, Ontology, Service Oriented Architecture.

## 1. INTRODUCTION

De nos jours, l'informatique ubiquitaire nous permet de garantir un échange collaboratif et riche d'informations en utilisant des machines et des services communicants. Ces machines peuvent être des machines légères comme des téléphones portables ou des PDAs ou même des grandes machines de production. Beaucoup d'efforts actuels sont investis afin d'informatiser et de rendre ces machines suffisamment intelligentes pour pouvoir communiquer avec d'autres machines et collaborer d'une façon de plus en plus autonome pour effectuer des activités communes. L'interopérabilité de ces machines nous mène à des systèmes complexes ; difficiles à gérer et à maintenir. Dans ce contexte, le monitoring et l'administration de ces machines et des services qu'elles offrent prennent une importance cruciale. En effet, elles sont utiles pour le maintien des accords de qualités de services élaborés entre les services et leurs clients humains ou logiciels. Notre travail s'intéresse aux modèles et aux outils logiciels nécessaires pour réaliser la surveillance des obligations spécifiées dans les contrats de qualités de services et de leur administration afin de réagir aux problèmes de violations de contrats ou de pannes dans le système. Dans cet article, nous nous focalisons sur l'élaboration d'un modèle de SLA basé sur une ontologie [1] générique pour aider à l'élaboration des contrats de QoS et surveiller les accords et les obligations spécifiés dans ces contrats. Le choix d'une ontologie est dû principalement aux possibilités qu'elle offre pour faciliter l'interprétation des informations des humains d'une part, et des machines d'une autre part. Ses implémentations sont basées sur des standards ouverts comme OWL [2] et SWRL [3] qui servent à faciliter l'expression des obligations du contrat et des inférences à fin de prendre les actions nécessaires face aux violation du contrat. Dans la section 2 de cet article, nous commençons par définir les principes de SLA et ses implémentations. Dans la section 3, nous présentons les modèles principaux existants relatifs aux SLAs. Dans la section 4, nous détaillons notre modèle générique des accords de qualité de services que nous appelons *SLAOnt*. Dans la

section 5, nous expliquons comment utiliser ce modèle pour surveiller les instances de ses obligations et comment réagir automatiquement faces aux problèmes de violation du contrat. Nous présentons aussi un simple exemple d'instanciation de notre modèle et nous montrons comment surveiller ses obligations à travers un prototype de surveillance de SLA que nous avons développé. Avant de conclure, dans la section 6, nous présentons une discussion comparative pour mieux situer notre modèle par rapport à l'existant présenté dans la section 3.

## 2. PRINCIPES DES SLAs

La littérature propose un grand nombre de définitions pour le terme SLA [4]. Retenons qu'un SLA est un contrat définissant les engagements de l'hébergeur quant à la qualité de sa prestation, et les pénalités engagées en cas de manquement. Cette qualité doit être mesurée selon des critères objectifs acceptés par les deux parties (Par exemple, le temps de rétablissement du service en cas d'incident).

Plusieurs langages ont été proposés pour implémenter les spécifications des SLA. Nous citons WSOL (Web Service Offerings Language) [5], GXLA [6]. WSML (Web Services Management Language) [7]. SLAang [8]. Ws-Agreement [9]. Ws-Negotiation [10] et WSLA (Web Service Level Agreement) [11]. Parmi ces langages, la contribution la plus aboutie et la plus répandue est celle du langage WSLA proposé par IBM<sup>1</sup>. WSLA est une spécification flexible, utile et assez simple pour être utilisée dans des cas réels. Elle est extensible et applicable sur plusieurs types de services (Services Web, stockage...). Cependant, l'élaboration du contrat reste une tâche assez difficile à réaliser. En effet, les fournisseurs et les clients n'ont pas le même degré de connaissances et peuvent ne pas partager le même langage. D'un autre côté, le suivi du contrat et ses violations éventuelles restent des tâches très peu explorées aussi. En conséquence, nous analysons les travaux existants relatifs aux modélisations de SLA qui présentent plus de richesse sémantique et plus de structuration au niveau de l'expression des obligations des SLA et de leur suivi.

## 3. MODELES EXISTANTS RELATIFS AUX SLAs

Nous avons exploré les modèles existants relatifs aux SLAs. Nous avons remarqué un grand intérêt à la modélisation de la qualité de service qui est un élément principal dans la spécification des contrats. La littérature présente une panoplie de modèles de spécification de qualité de service. Le modèle OWL-QoS [12] est un modèle de description des QoS. Il présente l'avantage de la réutilisation de l'ontologie standard de la description de services OWL-S [13]. Ce modèle est caractérisé par sa description formelle de la diffusion et de la consommation de qualités de services. Cependant, il présente quelques insuffisances. Les métriques sont spécifiées sans préciser à quoi elles servent et dans quel contexte elles sont valables. Il ne bénéficie pas aussi des relations d'inférences de l'ontologie puisqu'il utilise des chaînes de caractères non structurées pour la spécification de contraintes de QoS (par exemple le temps de réponse <100ms). Le modèle QoSOnt [14] est un autre modèle de QoS. Il est défini par le langage OWL [2]. Il présente l'avantage d'être interopérable avec d'autres ontologies existantes telque OWL-S. Il est plus riche que l'ontologie OWL-QoS au niveau de la définition des métriques et

de la correspondance des exigences de qualités de services avec les métriques. Cependant, les types de données utilisés pour les métriques sont définis dans un langage XML spécifique. En conséquence, ce travail perd l'avantage de l'utilisation d'OWL pour les interprétations et les inférences d'informations sémantiques riches. Le modèle SL-Ontology [16] est une autre tentative de modèle de QoS. Il différencie entre les offres du fournisseur et les demandes du client. Il spécifie une partie de transformation d'unités de mesure à fin de résoudre les disparités de langages entre les clients et les fournisseurs. Cette résolution n'est spécifiée qu'au niveau des unités dans ce modèle. WS-QoS [17] est un framework qui utilise un modèle de QoS à base d'ontologies pour la sélection dynamique de services Web selon les exigences de performances d'exécution et de réseau. Ce modèle est caractérisé par l'utilisation de métriques spécifiques qui doivent être connues à l'avance par tous les services. Il utilise aussi un langage XML spécifique pour la description des métriques d'où la perte du raisonnement et des inférences sémantiques offertes par OWL. FIPA [18] est un autre modèle de représentation de QoS à base d'ontologies. Il est complet mais malheureusement il reste trop spécifique aux couches basses du modèle OSI. MOQ [19] est une autre tentative de modélisation de QoS qui définit des exigences de QoS composites mais échoue dans la spécification de clauses logiques complexes. Tous les modèles que nous venons de citer présentent des avantages et des lacunes les uns par rapport aux autres. En effet, peu de modèles existants définissent la notion de contexte dans la qualité de service. Cette notion est importante pour nous pour pouvoir gérer le cycle de vie d'un contrat de QoS d'une façon automatique. En plus, certaines contributions (comme WS-QoS) utilisent des formats XML spécifiques pour l'implémentation complète ou partielle de leurs modèles. Ceci réduit considérablement l'intérêt de l'utilisation d'une ontologie qui offre des possibilités d'inférences et d'interprétation sémantique en se basant sur le langage OWL. En outre, certains modèles sont soit spécifiques à un domaine particulier (comme le cas de FIPA-QoS qui est spécifique aux couches basses du modèle OSI) où présentent des incomplétudes diverses (comme l'absence de spécifications de contraintes logiques composées sur les QoS pour le modèle MOQ). Enfin, tous ces travaux se focalisent sur la modélisation de qualité de services sans détailler les obligations et les accords entre les acteurs impliqués. Cette dernière constatation nous a encouragée à élaborer un modèle d'accords de qualité de services en s'inspirant des modèles que nous venons d'explorer.

## 4. SLAOnt: PROPOSITION D'UN MODELE DE SLA BASE SUR UNE ONTOLOGY

Notre contribution dans ce domaine est caractérisée par le développement d'un modèle d'accords de qualité de service (SLA) à base d'ontologie. Nous avons effectué ce choix pour (i) faciliter l'élaboration des contrats entre des entités qui ont des niveaux de connaissances différents (fournisseurs et clients) (ii) avoir un modèle qui offre une richesse sémantique facilitant sa compréhension par des êtres humains d'une part et son exploitation et son suivi d'une façon automatique d'autre part. Notre modèle (voir figure 1) définit une ontologie décrivant les différents concepts et propriétés nécessaires constituant un contrat de qualité de service.

---

<sup>1</sup> <http://www.research.ibm.com/wsla/>

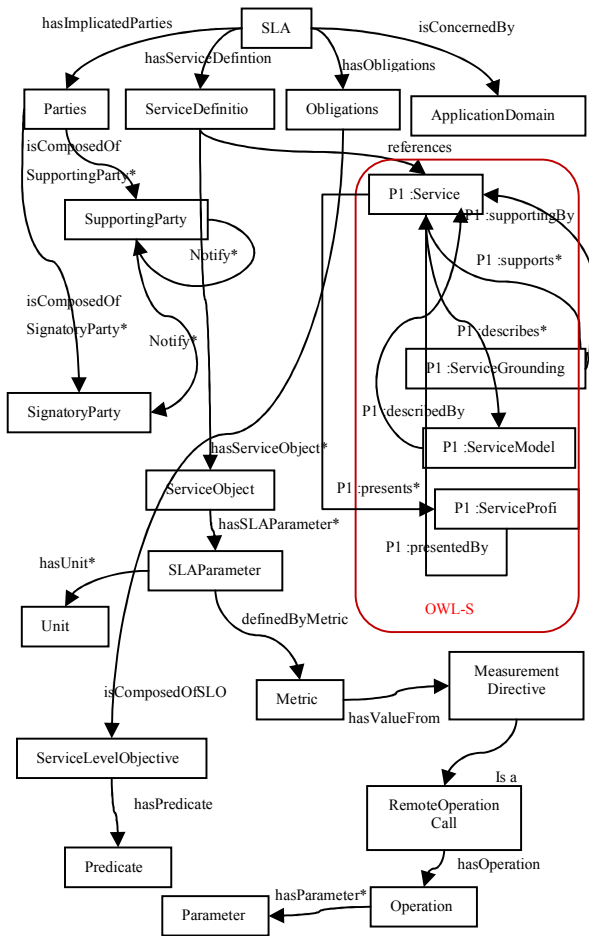


Figure 1. SLAOnt: Structure générale du contrat de QoS

La racine de ce modèle est le concept SLA. Il représente le concept des contrats qu'on peut instancier à partir de notre modèle. Ce concept est composée des concepts suivants : *Parties*, *Obligations* et *ServiceDefinition*. Le premier concept définit les parties impliquées dans le contrat : les parties signataires et les parties tierces. Les parties signataires sont généralement les fournisseurs des services et leurs clients. Les parties tierces fournissent les entités nécessaires pour la mesure de la qualité de service requise définie dans le contrat. Le deuxième concept définit les obligations de qualité de service que les parties s'engagent à respecter. Le troisième concept décrit les services fournis concernés par ces obligations. Le contrat est défini dans un domaine d'application bien déterminé. Ainsi, nous avons défini un concept *ApplicationDomain* pour décrire le contexte du contrat. Ce concept peut être un lien vers une ontologie définissant les terminologies utilisées dans ce contexte. Les variables nécessaires pour la définition de la qualité de service évaluée par le contrat sont modélisées dans *SLAOnt* par le concept *SLAParameter*. Un *SLAParameter* est associé à une ou plusieurs métriques (ou des agrégations de métriques) qui définissent la qualité de service à mesurer dans le contrat. Ces métriques, modélisées par le concept *Metric* peuvent être agrégées par une fonction (*Function*) mathématique ou algorithmique de calcul. Par

exemple, cette fonction peut être la moyenne ou un pourcentage calculé à partir d'un ensemble de mesures. Un *SLAparameter* représente une variable qui a une unité de mesure (*Unit*) qui peut être des secondes, des minutes, des pourcentages, etc. Chaque fonction est définie par un opérateur (comme avg pour calculer une moyenne) et un opérande qui représente une ou plusieurs mesures d'une métrique de QoS. Ces mesures sont obtenues à l'aide de directive de mesure (*MeasurementDirective*). Ces directives sont généralement des appels d'opérations distantes qui renvoient les valeurs des métriques. Nous avons modélisé ces appels par le concept *RemoteOperationCall*. Il décrit l'opération à appeler avec son protocole d'invocation. Chaque opération est associée à un *Handler* qui représente l'objet ou la classe distante qui l'héberge. En outre, nous décrivons les paramètres d'entrées de l'opération à l'aide du concept *Parameter*. Ce dernier encapsule les valeurs à passer à l'opération distante pour son invocation. Le concept *Predicate* définit les obligations de qualité de service qui doivent être respectées dans le contrat. Chaque prédicat est exprimé par une règle SWRL définie dans *SLAOnt*. Chaque règle SWRL est définie par une partie tête (swrl:head) et une partie corps (swrl:body). Dans la partie tête, on spécifie les actions à prendre en cas de détection de violation du contrat. La partie corps spécifie les conditions qui déclenchent la violation du contrat. La règle *PredicateEvaluationRule* présentée dans la section 5.1 montre un exemple concret d'un prédicat dans *SLAOnt*.

## 5. LE MONITORING DES INSTANCES DE SLAOnt

Dans cette section, nous présentons l'utilité de notre modèle SLA pour générer automatiquement un processus de surveillance qui contrôle les obligations définies dans le SLA. Pour valider notre modèle d'accords de qualités de services, nous avons créé une instance du modèle SLAOnt en utilisant l'outil "protégé".

### 5.1 Exemple d'instance Flight SLA

Dans cette section, nous présentons un simple exemple de contrat entre un fournisseur de service de réservation d'un vol et ses clients. Ce service doit fournir un temps de réponse moyen inférieur à 100 millisecondes pour un nombre déterminé de clients.

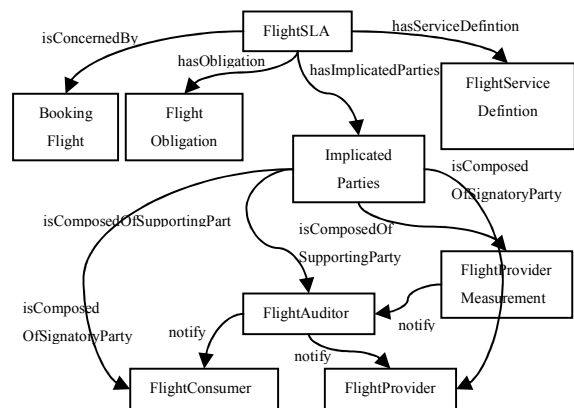


Figure 2. Exemple d'instanciation de SLAOnt

La figure 2 illustre l'instance *FlightSLA* correspondant à notre exemple. Le fournisseur de service de ce contrat est appelé

*FlightProvider* et son client est nommé *FlightConsumer*. Dans cet exemple, nous avons défini les parties tierces impliquées dans ce contrat : *FlightProviderMeasure* et *FlightAuditor*. Le service *FlightProviderMeasure* doit fournir les mesures du temps de réponse du service de réservation en ligne de billets d'avion. Le service *FlightAuditor* doit notifier les parties signataires des violations éventuelles du contrat. Le fournisseur de service doit respecter l'objectif défini par l'instance *FlightSLO*. Cette instance spécifie le prédicat principal qui doit être évalué pour vérifier cette obligation. Les prédicats sont définis dans les règles SWRL comme montré ci après pour faciliter le processus de monitoring du contrat. SWRL (Semantic Web Rule Language) est un langage de règles pour le web sémantique, combinant des sous-langages OWL (OWL DL et OWL Lite) avec ceux du Rule Markup Language (Unary/Binary Datalog). L'obligation définie dans *FlightSLA* est :

**PredicateEvaluationRule:**

$$\text{hasEvaluation(average\_response\_time, ?x) } \wedge \\ \text{swrlb:greaterThanOrEqual(?x, 100.0) } \rightarrow \\ \text{slaont:disseminateViolation(lessThan100msPredicate, "false",} \\ \text{average\_response\_time, ?x)}$$

Dans cette règle, nous vérifions si le temps de réponse moyen du service surveillé est supérieur ou égal à 100 millisecondes (partie corps de la règle). Dans ce cas, on déclenche un message de violation aux parties signataires du contrat.

## 5.2 Le monitoring du Flight SLA

Pour implémenter le processus de surveillance des instances de *SLAOnt*, nous avons développé 3 services : le service de mesures de métrique, le service de mesures des paramètres de SLA et le service de surveillance des obligations. Le premier service est instancié automatiquement pour chaque métrique défini dans le SLA. Il génère les valeurs de métrique en invoquant leurs directives de mesures correspondantes selon une fréquence spécifique définie dans le SLA. Le deuxième service est instancié aussi automatiquement pour chaque paramètre de SLA défini dans le contrat. Il applique les fonctions d'agrégations aux valeurs mesurées par le premier service. Par exemple, dans le cas de *FlightSLA*, il calcule la valeur moyenne de la dernière réponse selon une période spécifique définie dans le SLA. Le dernier service est automatiquement instancié pour chaque obligation dans le contrat. Il utilise les valeurs calculées dans les *SLAParameters* pour vérifier si elles sont satisfaites aux conditions spécifiques définis dans l'obligation. Nous avons développé aussi un programme principal qui accède à l'ontologie (*SLAOnt.owl*) pour retrouver la fréquence de mesure pour chaque métrique et sa période d'agrégation et de générer les services de mesures des métriques et les services de mesures de *SLAParameter* dans le but de surveiller les instances des SLAs fournis. Ce programme génère un service de surveillance des obligations pour chaque obligation définie dans ces instances. Puis, le programme déploie et exécute tous les services générés. Une fois déployé, le service de surveillance des obligations exécute les règles SWRL spécifiées dans les prédicats pour chaque obligation. Nous utilisons le moteur d'inférence Jess [20] pour évaluer ces prédicats et déclencher automatiquement les

actions nécessaires en cas de violations. Dans l'exemple de *FlightSLA*, ces actions consistent à une dissémination de messages pour les parties signataires de SLA. Ces messages contiennent les valeurs des paramètres qui ont déclenché la violation. Notre programme principal est une interface d'utilisateur Swing ou un administrateur peut importer un fichier OWL contenant les instances de SLA et commence le monitoring. Les valeurs mesurées les *SLAParameters* calculées sont affichées sur cette interface d'utilisateur. Dans le cas de violations de contrat, en plus des messages de notifications envoyés aux parties signataires, notre prototype affiche un message d'alerte à l'administrateur (figure 3).

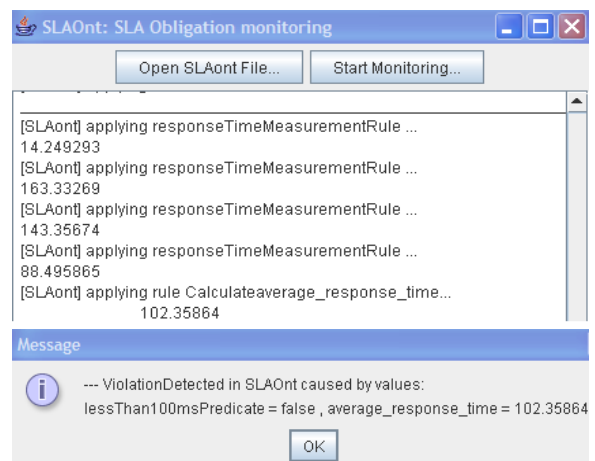


Figure 3. Interface du prototype du monitoring de SLAOnt

## 6. COMPARAISON ET DISCUSSIONS

Afin de pouvoir situer notre contribution dans les travaux existants, nous avons fait une étude comparative entre les différents modèles présentés dans la section 3. Le tableau 1 présente une comparaison de ces modèles selon trois critères. Le premier critère "Portée" illustre le degré de complétude de chaque modèle en listant leurs concepts principaux. Le deuxième critère "Implémentation" montre si des exemples concrets ont été élaborés afin de valider ces modèles. Le troisième critère "Facilité d'utilisation automatique" illustre le degré de structuration des informations dans les modèles afin de faciliter leurs interprétations et leurs utilisations automatiques dans des applications de monitoring et d'administration d'accords de qualité de service. Sur le premier critère, et selon notre comparaison, *SLAOnt* est le modèle le plus complet en termes de spécifications d'accords et d'exigences de qualité de service. En effet, la majorité des modèles existants se focalisent sur la spécification et la mesure des QoS. Peu de modèles s'intéressent aux accords et contrats de QoS. Ceci donne généralement des spécifications vagues et incomplètes pour exprimer les obligations des acteurs impliqués dans les contrats. Notre ontologie *SLAOnt* présente aussi l'avantage de la réutilisation de l'ontologie OWL-S pour la description des services évalués dans les SLA. Sur le deuxième critère, nous avons cherché des exemples concrets utilisant les différents modèles existants. Nous avons aussi essayé de réaliser la même instance de notre exemple « *FlightSLA* » avec ces modèles. Nous avons rencontré des difficultés diverses avec la plupart d'eux à cause des incomplétudes au niveau des spécifications des obligations et des contraintes de QoS dans le

contrat. Par exemple, le modèle de MOQ est d'ordre très générique ce qui écarte la distance entre les modèles et leur implémentation. Un effort conceptuel considérable doit être fourni pour pouvoir décrire des contrats concrets avec ce genre de modèles. Même si nous supposons que les instances des modèles existants ont été déjà élaborées, leur maintenance et leur utilisation automatiques dans des applications de gestion de contrats de QoS restent difficiles (troisième critère du tableau 1). On peut citer l'exemple de WS-QoS qui utilise un langage XML spécifique pour la description des métriques ce qui rend le développement d'outils d'analyse automatique de ces modèles difficile et lié à ces langages propriétaires.

**Tableau 1: Situation de SLAOnt dans les modèles existants**

	Portée	Implémentation	Facilité d'utilisation automatique
OWL-QoS [12]	Métrique, Unité, Fonctions de mesure, Profil de QoS, accords, Acteurs, Service	Présence d'exemples d'implémentation partielle	Contraintes de QoS en chaînes de caractères
QoSOnt [14]	Profil du Service, Profil de QoS, Métrique, Unité, Acteurs, Fonctions de mesure, Service	Présence d'exemples d'implémentation partielle	Langage spécifique
SL-Ontology [16]	Unité, Métrique, QoS, Services	Présence d'un exemple d'implémentation partielle	Modèle bien structuré pour l'établissement des contrats mais pas pour leur suivi
WS-QoS [17]	Métrique, Fonctions, QoS, accords, Acteurs	Ne dispose pas d'implémentation OWL	Langage spécifique
FIPA QoS [18]	Quality of Service Description, Rate Value, Probability Value, Transport Protocol Description	Ne dispose pas d'implémentation OWL	Langage spécifique
MOQ [19]	Exigences de QoS, Traçabilité, Administration	Fournit un fondement théorique sans implémentation	Interprétation ad hoc des prédicats
SLAOnt	SLA, Parties, OWL-S Contextualisation, Obligations, Fonctions, Métriques directives de mesures, Unités, Prédicats, Expressions,	Présence d'un exemple d'implémentation complet	Modèle structuré et riche en relations pour des interrogations automatiques faciles

Un effort considérable d'ingénierie doit être mis en œuvre afin de pouvoir exploiter d'une façon automatique les instances de ces modèles dans des applications de gestion de contrats. Enfin, nous pensons que *SLAOnt* présente un compromis entre la complexité du modèle et sa complétude. En effet, l'exemple que nous avons élaboré dans la section 5 montre la facilité d'implémentation du modèle, d'une part, et sa complétude au niveau de la structuration des informations décrivant les contrats, d'une autre part, contrairement à certains modèles existants (comme OWL-QoS) qui spécifient les prédicats sous la forme de chaînes de caractères.

## 7. CONCLUSION ET PERSPECTIVES

Le nouveau marché de développement de logiciels exige la spécification précise des qualités des prestations offertes par les fournisseurs de services. Beaucoup d'efforts ont été investis dans la modélisation des qualités de ces prestations pour que le client puisse choisir d'une façon automatique (ou semi-automatique) les services qui offrent la meilleure qualité. Nous avons exploré les différents modèles existants dans ce domaine. Nous avons remarqué le manque d'un modèle complet et générique pour la spécification d'accords de qualité de service et leur monitoring

pour détecter la violation de ces accords. Ainsi, nous avons élaboré une ontologie qui modélise ces accords (SLA) pour faciliter l'élaboration des contrats entre les clients et les fournisseurs, d'une part, et pour automatiser la gestion et l'administration de ces contrats, d'autre part. Dans cet article, nous avons présenté la structure de notre modèle basé sur l'ontologie avec un exemple d'instanciation qui consiste à surveiller le temps de réponse moyen d'un service de réservation de billets d'avion. Nous avons donné aussi une idée sur un prototype que nous avons développé pour surveiller automatiquement les obligations de QoS définie dans les instances de notre modèle. Actuellement, nous continuons à travailler sur ce prototype pour fournir des interfaces d'utilisateurs adaptatives pour illustrer l'état d'évolution des services surveillés dans les instances de notre modèle. Nous planifions d'utiliser les valeurs des QoS mesurées dans ces instances pour analyser, détecter les dégradations du système et de prévenir aux violations de SLA. Cette issue sera très utile pour évoluer d'un simple message de notification à une assistance d'une action corrective.

## 8. REFERENCES

- [1] R.Studer, R.Benjamins, D.Fensel. Knowledge engineering: principles and methods. *IEEE Transactions on Data and Knowledge Engineering*, 25 (1-2) pp.161-197, March 1998.
- [2] D.L. McGuinness, F. van Harmelen. OWL Web Ontology Language Overview. <http://www.w3.org/TR/owl-features/> February 2004.
- [3] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, and M. Dean. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission, 21 May 2004.
- [4] M. Debusmann, R. Kroger, and K. Geihs. "Unifying Service Level Management Using an MDA-based Approach". *IEEE Network Operations and Management Symposium*, pp.801-814, 2004.
- [5] Tomic, V., Pagurek, B., Patel, K., Esfandiari, B., and Ma, W. 2005. Management applications of the web service offerings language. *Inf. Syst.* 30, 7 (Nov. 2005), 564-586.
- [6] Badis Tebbani, Issam Aib. "GXLA a Language for the Specification of Service Level Agreements". *Autonomic Networking*, 201-214, 2006.
- [7] A. Sahai, V. Machiraju, M. Sayal, A. van Moorsel, and F.Casati. "Automated SLA Monitoring for Web Services". *IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, pp.28-41, 2002.
- [8] J. Skene, D. Lamanna, and W. Emmerich. "SLAng: A Language for Service Level Agreements". *Workshop on Future Trends in Distributed Computing Systems. IEEE Computer Society*, 2002.
- [9] A. Andrieux, A. Dan K. Czajkowski, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu. "Web Services Agreement Specification (WSAgreement)". Specification draft, Global Grid Forum (GGF), September Version 09/2005.
- [10] P. C. K. Hung, H. Li, and J-J Jeng. "WS-Negotiation: An Overview of Research Issues". *Hawaii International Conference on System Sciences (HICSS)*, 2004.

- [11] 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), 2003.
- [12] Chia, Bu-Sung Lee. "QoS Measurement Issues with DAML-QoS". IEEE InterChen Zhou, Likang-Tien national *Conference on e-Business Engineering (ICEBE'05)* pp. 395-403.
- [13] OWL-S. An OWL-based Web service ontology. <<http://www.w3.org/Submission/2004/07/>> November 2004.
- [14] G. Dobson, R. Lock, I. Sommerville. "QoSOnt: a QoS Ontology for Service-Centric System", *EUROMICRO Conference on Software Engineering and Advanced Applications*, Porto, Portugal, Aug. 2005.
- [15] C. Zhou, L. Chia, and B. Lee, "DAML-QoS Ontology for Web Services", *Proceeding of the International Conference on Web Services 2004 (ICWS04)*, San Diego, California, USA, July 2004.
- [16] Steffen Bleul, Thomas Weise, Kurt Geihss. "An Ontology for Quality-Aware Service Discovery". *Special Edition Editorial: Engineering Design and Composition of Service-Oriented Applications, Computer Systems Science & Engineering*, Volume 5, Number 21 – 2006.
- [17] Tian, M., Gramm, A., Ritter, H., and Schiller, J. "Efficient Selection and Monitoring of QoS-aware Web services with the WSQoS Framework". *IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, Beijing, China, 2004.
- [18] Foundation for Intelligent Physical Agents "FIPA Quality of Service Ontology Specification", Geneva, Switzerland, Nov. 2002.
- [19] H.M. Kim, A. Sengupta and J. Evermann. "MOQ: Web Services Ontologies for QoS and General Quality Evaluations", *European Conference on Information Systems*, Regensburg, Germany. May 2005.
- [20] M. J. O'Connor, H. Knublauch, S. W. Tu, B. Grosz, . Dean, W. E. Grosso, M. A. Musen. Supporting ULE System Interoperability on the Semantic Web with SWRL. Fourth *International Semantic Web Conference (ISWC2005)*, Galway, Ireland, 2005.