

Automating Cloud Service Level Agreements using Semantic Technologies

Karuna Pande Joshi and Claudia Pearce

CSEE department, University of Maryland Baltimore County
Baltimore, MD, USA

karuna.joshi@umbc.edu, iclaudiapearce@gmail.com

Abstract— Cloud related legal documents, like terms of service or customer agreement are usually managed as plain text files. Hence extensive manual effort is required to monitor the cloud service performance by cross referencing the metrics and measures agreed upon in these documents. We have significantly automated the process of managing and monitoring cloud Service Level Agreements (SLA) using semantic web technologies like OWL, RDF and SPARQL. In this paper, we describe in detail the cloud SLA ontology and the prototype that we have developed to illustrate how the SLA measures can be automatically extracted from legal Terms of Service that are available on cloud provider websites.

Keywords—Cloud computing, semantic cloud services, cloud service level agreements, semantic web

I. INTRODUCTION

Organizations are increasingly adopting cloud-based services to address their information technology (IT) needs for software, hardware or network bandwidth [28]. While cloud based solutions are attractive for their cost savings and rapid provisioning/scaling; consumers are finding it difficult to monitor the cloud service level agreements (SLA) [18] that define the service performance measures. The service contracts or SLA documents are legal cloud documents that define broadly the service data, delivery mode, service agent details, quality metrics and cost of the service including penalty terms, if any. These documents are currently managed as text documents and so large manual effort is required to manage them as well as to map these to the main service performance indicators.

In our discussions with large organizations interested in acquiring cloud services, especially from public cloud providers, we have observed that a key barrier preventing organizations from successfully managing virtualized services on the cloud is the lack of an integrated methodology for service creation and deployment that would provide a holistic view of the service lifecycle on a cloud. We have developed a methodology [1] to address the lifecycle issue for virtualized services delivered from the cloud. We use semantically rich descriptions of the requirements, constraints, and capabilities

that are needed by each phase of the lifecycle. This methodology is complementary to previous work on ontologies, like OWL-S, for service descriptions in that it is focused on automating processes needed to procure services on the cloud. While our work has been concentrated on organizations as consumers, our methodology can also be applied to enterprise cloud users. We have also developed a protocol to automate the negotiation process between the cloud service consumer and provider described in [1]. This negotiation process results in a service contract comprising of a Cloud Service Level Agreement.

The key reason to have a semantically rich approach to describe cloud attributes and SLA is to permit distributed clients and cloud service providers to “automate” the process of acquisition and consumption of services. Without a semantic approach that can permit the providers and consumers to understand each other, which is the present state of the practice, the acquisition process is done manually, and the consumption/monitoring process also requires significant manual input. For instance, National Institute of Standards and Technology (NIST) has identified ambiguity in cloud SLAs currently offered by cloud providers as one of the factors that prevent broad cloud adoption by large organizations, especially federal agencies. It is very difficult to compare SLAs offered by two cloud providers to determine who is offering the better deal. Also, existing cloud SLAs (for instance [29]) are provided as a text document making it open to interpretation and very difficult to monitor SLA performance and adherence by the cloud provider. Additionally, survey of industry sources also indicates overall dissatisfaction among cloud users of existing cloud SLA.

In this paper, we initially discuss the background and related work in this area. In section III, we present our approach towards automating service level agreements and describe the ontology we have developed for the same. We end with conclusions and future work.

II. RELATED WORK

In a virtualized service-oriented environment, consumers and providers need to be able to exchange information, queries, and requests with some assurance that they share a common meaning. This is critical not only for the data but also for the policies followed by service consumers or providers. The handling of heterogeneous policies is usually not present in a

closed and/or centralized environment, but is an issue in the open cloud. The interoperability requirement is not just for the data itself, but even for describing services, their service level agreements, quality related measures, and their policies for sharing data.

One possible approach to this issue is to employ Semantic Web techniques for modeling and reasoning about services related information. We have used this approach for automating Cloud service level agreements. The Semantic Web deals primarily with data instead of documents. It enables data to be annotated with machine understandable meta-data, allowing the automation of their retrieval and their usage in correct contexts. Semantic Web technologies include languages such as Resource Description Framework (RDF) [16] and Web Ontology Language (OWL) [17] for defining ontologies and describing meta-data using these ontologies as well as tools for reasoning over these descriptions. These technologies can be used to provide common semantics of Service information and policies enabling all agents who understand basic Semantic Web technologies to communicate and use each other's data and Services effectively.

Another approach has been the using Web Services Definition Language (WSDL) [2] which is a W3C standard for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. However, WSDL doesn't allow a means to express the policies that the service supports or adheres to. Hence additional proposals like WS-Policy [7] and WSLA [8] have been made to allow for the expression of additional nonfunctional attributes. Turner et al. [11] have proposed a service technology layer for creation and deployment of web services. They have compared the existing protocols and technology available to implement web services and have also noted gaps that need to be researched.

Web Services Agreement Specification (WS-Agreement) [3] is a web services protocol for establishing agreement between two parties using an extensible XML language for specifying the nature of the agreement, and agreement templates to facilitate discovery of compatible agreement parties. WS-Agreement limits the ability of matching the agreements to syntactical matching and is also very limited in matching non-functional attributes that define policies pertaining to data and security, compliance issues, data quality levels, etc.. Oldham et al. [4] have proposed semantically rich extensions to extend the WS-Agreement. Aiello et al. [5] have proposed a formal definition of "Agreement" based on WS-Agreement. WS Negotiation [6][9] is proposed as an extension of WS-Agreement to allow negotiation capabilities. WS-Negotiation consists of three parts - negotiation message, negotiation protocol and negotiation decision. In this basic model for WS-Negotiation, the emphasis is on how to negotiate and what to negotiate along with negotiation strategies is not defined. It also doesn't demonstrate how a requester's enterprise policies can be used to automate the negotiation process. Bui and Gachet [12] have described a broker capable of offering negotiation and bargaining support to facilitate the matching of Web Services. The negotiation protocol and decision-making

mechanisms for negotiation have not been described. Skogsrud et al. [14] propose trust negotiation framework that supports policy lifecycle management for web services. However, this framework is limited to managing user identity of customers accessing the service. Yao et al. propose flexible strategies for Web Services negotiation in [15]. Their approach doesn't use policy management to automate service negotiation. There has also been work done on negotiating between web services [13], however, this work does not relate to the negotiation needed between service provider and consumer.

For our cloud services lifecycle framework [1], we have used semantic web technologies like OWL instead of WS-Agreement and WS-Negotiation protocol as we were able to more richly define the cloud SLA ontologies, thereby allowing us to incorporate different descriptions of the same SLA measure.

III. ONTOLOGY FOR CLOUD SERVICE LEVEL AGREEMENTS

In our discussion with our collaborators we found that the negotiation of SLA for the cloud services procured is the most time consuming portion of the cloud service procurement process. Automation of this process using semantic technologies is itself a performance improvement over the existing human-based negotiation. The service negotiation phase of our cloud services lifecycle [1] covers the discussion and agreement that the service provider and consumer have regarding the service delivered and its acceptance criteria. The service to be delivered is determined by the specifications laid down in the Request For Service (RFS) issued by the consumer. Service acceptance is usually guided by the Service Level Agreements (SLA) [18] that the service provider and consumer agree upon. SLAs define the service data, delivery mode, agent details, quality metrics and cost of the service. While negotiating the service levels with potential service providers, consumers can explicitly specify service quality constraints (data quality, cost, security, response time, etc.) that they require.

At times, the service provider will need to combine a set of services or compose a service from various components delivered by distinct service providers in order to meet the consumer's requirements. The negotiation phase also includes the discussions that the main service provider has with other component providers. When the services are provided by multiple providers (composite service), the primary provider interfacing with the consumer is responsible for composition of the service. The primary provider will also have to negotiate the Quality of Service (QoS) with the secondary component providers to ensure that SLA metrics are met.

The key deliverable of cloud service negotiation is the service contract between the service consumer and service provider. The SLA is a key part of this service contract and will be used in the subsequent phases to compose and monitor the service. The service negotiation phase of our lifecycle consists of the discussions and agreement that the Service provider and consumer have regarding the Service. This phase

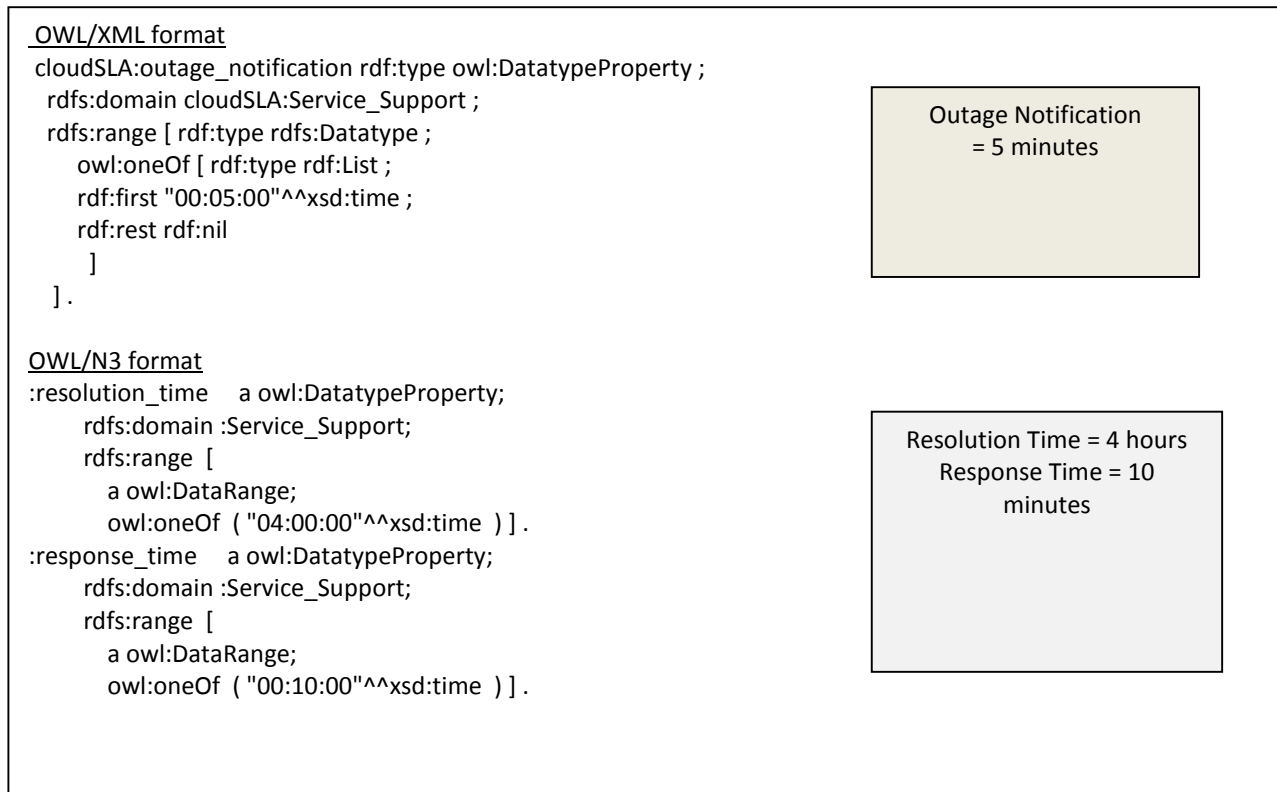


Figure 1: Section of the SLA specifying service support constraints.

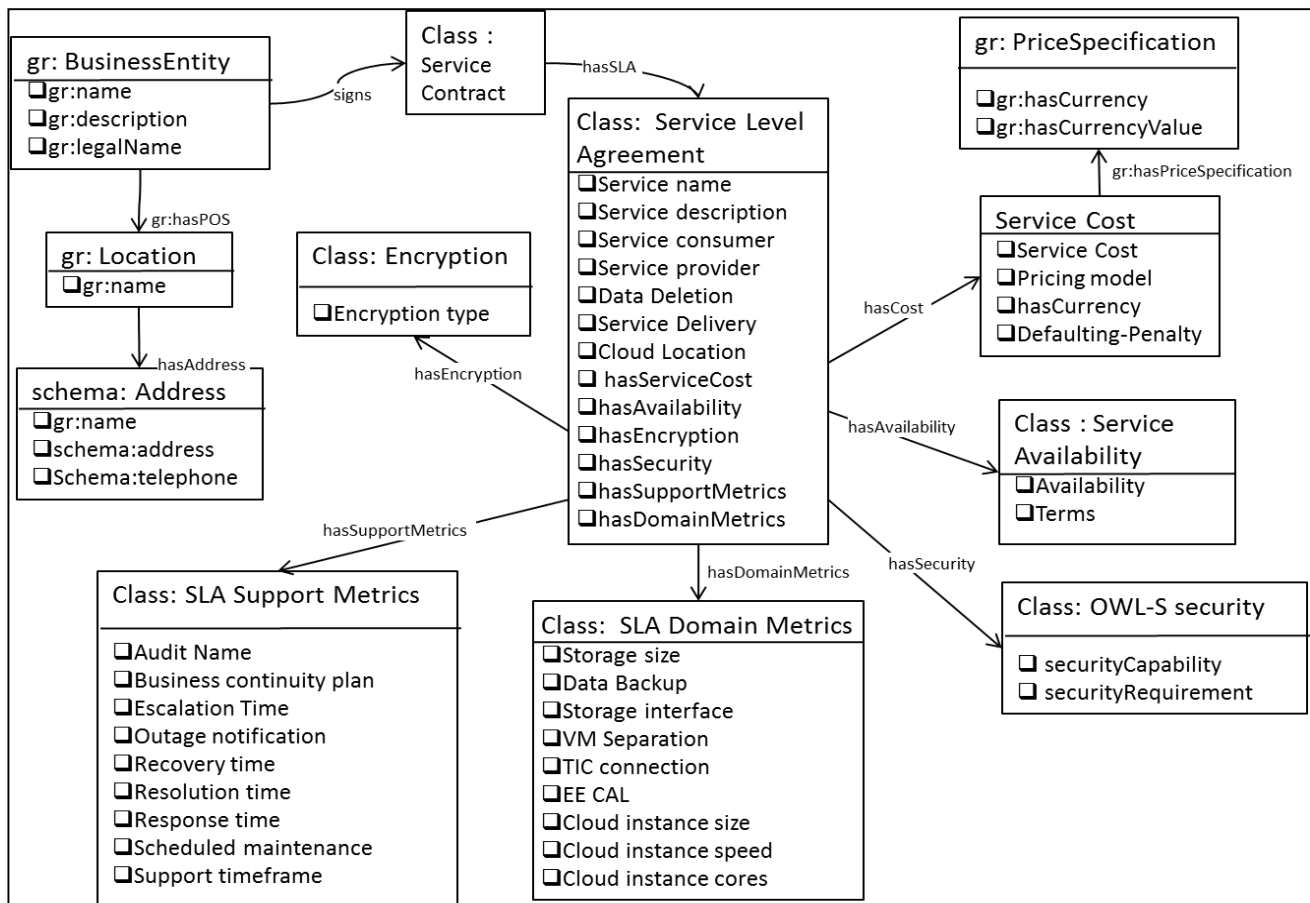


Figure 2: Ontology for Cloud Service Level Agreement (SLA)

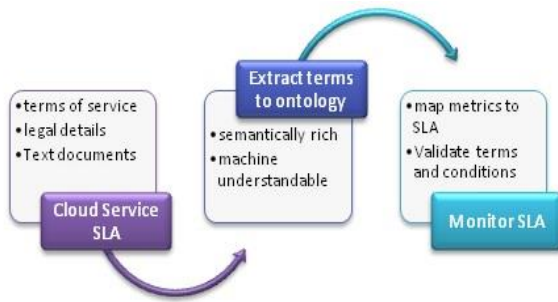


Figure 3: Technical approach to extract SLA measures from 'Terms of Service'

results in a service contract, which is comprised of the SLA that has been finalized between consumer and provider and QoS agreements between primary provider and component providers. Some of the measures that are part of the SLAs of large organizations, who outsource their IT functionality, can also be included as measures in cloud services.

Negotiation for Cloud SLAs will also include agreement on service support metrics. Some key support metrics that we were able to compile after discussions with a large financial organization include –


- Availability timeframe of services,
- Scheduled maintenance times,
- Contingency or business continuity plans,
- Timeframes for notification and recovery following an unplanned service disruption or a security incident,
- Problem resolution and escalation procedures, etc.

We have incorporated these support metrics in the SLA ontology that we have created using OWL language [17]. Figure 1 illustrates a small section of the ontology that specifies the desired support metrics.

We referred to industry best practices, NIST use case 3.9 [19] and actual enterprise data from a large international financial organization to identify the cloud SLA components. Figure 2 illustrates the detailed cloud SLA ontology. The Service Level Agreement class consists of properties that are common across all cloud applications. These include service fields like service delivery mode, and service availability; and security related attributes like cloud location, data encryption and data deletion. The SLA class consists of *hasDomainMetrics* property whose domain is the SLA Domain Metrics class which consists of properties that are specific to the service domain. For our ontology we have included properties of cloud storage domain like storage size, backup, etc. The *hasSupportMetrics* property in the SLA class has domain class SLA Support Metrics, which consists of service support properties like outage notification, resolution time, response times, etc. These properties are used to determine the performance of the service, and so can be used in the consumption phase to automate service monitoring.

Cloud consumers will have a service contract and SLA measures for each service that they purchase on the cloud. Each of these SLAs can be stored as an instance of our proposed cloud SLA ontology. These SLAs can be stored and managed as an RDF graph, which can be automatically queried by semantic technologies like SPARQL [24]. Any policy updates and changes to the SLA will need to be stored as a new RDF graph. One of the benefits of this would be to have a record of all SLAs signed for the service and will allow organizations to track their service usage over the years and use that to plan for the future. We have developed a system that automates this entire process and it is described in section IV.

This ontology is available in public domain and can be accessed at [26]. This ontology forms part of the integrated ontology [25] that we developed for our integrated cloud lifecycle framework [1].



Automatic Extraction and Monitoring of Cloud Service Level Agreements

SELECT CLOUD PROVIDER

Provider Name:

Display Terms of service/SLA

Extract SLA terms

ENTER SLA KEYWORDS TO SEARCH ON (SEPARATED BY COMMAS)

Availability, availability, available, Availabile, Compute, Services, services, Cloud, downtime, Downtime, uptime, Uptime, penalty, SLA, Connectivity

Select these keywords

To get the Terms of Service for a service -

1. Select the Provider name from the list and then press the "Display Terms of Service/SLA" button.
2. The terms of service available on the web will be displayed on this window.

To find the SLA measures in a SLA document -

1. Select the Provider name from the list and then press the "Extract SLA Terms" button.
2. The SLA terms will be displayed on this window.

To modify the keywords file -

1. Enter the SLA terms to search for (separated by commas) and then press the "Select these keywords" button.
2. The keywords that will be searched will be displayed on this window.

Figure 4: Prototype to automatically extract SLA terms from cloud terms of service documents

SLA terms that were found in the legal document HP_compute_SLA.txt include

Line No	word position	SLA term	Sentence
4	8	available	that HP Cloud Compute will be available 99.95% or more of the time
5	1	Availability	Given Region (Not Total Bill) Monthly Availability % (per Region) Credit to Bill
13	81	Availability	and consists of multiple physically separate Availability Zones. An "Availability Zone" is a
13	109	Availability	can choose to create instances. "Monthly Availability %" is calculated per Region on
13	165	downtime	as the sum of each instance's downtime minutes, during the course of a
19	32	availability	be deemed to have had 100% availability for that customer for the given
19	61	Availability	run instances in more than one Availability Zone within a Region during the
22	68	availability	end of the month in which availability was not met by contacting HP
22	111	downtime	credit if it cannot verify the downtime or you cannot show that you
31	22	availability	remedy for any failure to meet availability of HP Cloud Compute.

Maps to "Service Availability" Class of the cloud SLA ontology

Figure 5: SLA terms identified in the terms of service document that can be mapped to the ontology

IV. AUTOMATE EXTRACTION OF SLA MEASURE FROM TERMS OF SERVICES

We have developed a prototype system to automatically extract key SLA measures from the legal 'Terms of Service' documents. Figure 3 illustrates our technical approach for this prototype. We begin by downloading publically available SLAs or customer agreement documents that are posted by cloud providers on their website. Next, we automatically extract terms from the document, like uptime, availability, payment, etc., and then map it to our cloud SLA ontology (described in section III). Once we have identified the SLA terms, we save it as a RDF graph which is machine understandable and so can be used to automate the monitoring of SLA compliance of the service. We used Semantic Web technologies like OWL [17], RDF [16] and SPARQL[24] to develop this prototype. We used the Jena Apache Fuseki [27] server graph store as a store of the cloud SLAs.

Figure 4 illustrates the main screen of the prototype which allows users to select a provider and their publically available terms of service documents that are a legal agreement with their consumers. For our prototype, we have incorporated publically available 'terms of service' documents from Google for Google Apps [20], from Microsoft for MS Azure [22], from Amazon for EC2 [23] and from Hewlett Packard for their compute cloud

[21]. As part of our ongoing work, we are expanding to include other providers.

Users can select the provider, from a drop down list, whose SLA measures they want to automatically extract. After they click the 'extract SLA Terms' button, the system parses the entire document to discover the key measures in the document. Figure 5 illustrates the output screen for the HP compute cloud SLA document. We can observe that the system has identified all the places that word 'available' exists. This can now be mapped to the 'Service Availability' class in our ontology (see figure 2).

Once we have identified the SLA measures, we parse the text document again to identify the value associated with that measure. For instance, the HP Compute service SLA has availability as 99.95% on line 4 (see figure 5). Thus we set the SLA measure for service availability to 99.95%. At the same time, the HP compute SLA has a term called 'Availability Zones' in line 13 and also calculates a percentage of 'Monthly Availability' in line 13. These are also terms we should capture in the ontology instance.

The main challenge we faced while scraping SLA terms was that different terms are used by cloud vendors to express the same measures. For instance, the measure 'Availability' is referenced as 'Uptime' by another vendor's SLA. We are attempting to address this by adding all descriptions of the same measure in our ontology. Currently, we are only storing the numeric measures, and are working on also storing non-

numeric measures in the SLA instance. We store the SLA instance in the form of an RDF graph using the SPARQL CONSTRUCT. This SLA graph is stored in a Fuseki [27] graph store, which can be easily queried for continuous SLA monitoring since it is in a machine-readable format. For our prototype, we used the SLAs for the same service type and so all vendors' SLAs fit our framework. In our ongoing work, we are going to explore other service domains.

V. CONCLUSION AND ONGOING WORK

Currently cloud related legal documents, like terms of service or customer agreement documents are managed as text files. As a result extensive manual effort is required to monitor the metrics and measures agreed upon in these SLAs. We have worked on significantly automating this process using semantic web technologies like OWL and RDF. In this paper we have described in detail the cloud SLA ontology that we have developed. We have also described the prototype that we have developed to illustrate how the SLA measures can be automatically extracted from legal Terms of Service or customer agreement document that are available in the public domain. For our initial study, we selected four providers that are main providers of cloud based services.

As part of our ongoing work, we are including many more providers in our study to determine the scalability of our approach. We are also looking at other legal documents, like privacy policy documents, etc., to see if we can automatically compare the privacy measures desired by cloud consumers with the privacy measures and controls implemented by the cloud providers.

REFERENCES

- [1] Karuna Pande Joshi, Yelena Yesha, and Tim Finin, "Automating Cloud Services Lifecycle through Semantic technologies", IEEE Computer Society Press, pp.109-122, Jan 2014.
- [2] E. Christensen, F. Curbera, G Meredith, S. Weerawarana, Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/wSDL>, March 2001
- [3] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, M. Xu, WebServices Agreement Specification (WS-Agreement), <http://www.ogf.org/documents/GFD.107.pdf>, March 2007
- [4] N. Oldham, K. Verma, A. Sheth, and F. Hakimpour, Semantic WS-agreement partner selection, in proceedings of the 15th international conference on World Wide Web (WWW '06), pp 697-706, 2006.
- [5] M. Aiello, G. Frankova, and D. Malfatti, What's in an Agreement? An Analysis and an Extension of WSAgreement, in proc. 3rd ICSOC, 2005
- [6] O. Waeldrich et. al. , WS-Agreement Negotiation Version 1.0, January 2011
- [7] The Web Service Policy Framework, <http://www.w3.org/Submission/WS-Policy/>, April 2006
- [8] H. Ludwig, A. Keller, A. Dan, R. King, R. Franck, The WSLA Specification, <http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf>
- [9] P. Hung, Haifei Li and Jun-Jang Jeng, "WS-Negotiation: an overview of research issues," System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on , vol., no., pp. 10 pp., 5-8 Jan. 2004
- [10] G. E. Kersten and S. J. Noronha. "WWW-based Negotiation Support: Design, Implementation, and Use," Decision Support Systems, vol. 25, no. 2, pages 135-154, 1999.
- [11] Turner, M, et al, Turning software into a service, Computer, 36(10), 38, 2003.
- [12] Tung Bui and A. Gachet, Web Services for Negotiation and Bargaining in Electronic Markets: Design Requirements and Implementation Framework, Proceedings of the 38th Annual Hawaii International Conference on System Sciences, pp 38 - 38, 03-06 Jan, 2005
- [13] Kumar, S. and Mishra, R. B., An approach to multi-attribute negotiation between semantic web services, International Journal of Web Engineering Technology, 5, 2 (Sep. 2009), 162-186.
- [14] Skogsrud, H., Benatallah, B., and Casati, F. 2004. Trust-serv: model-driven lifecycle management of trust negotiation policies for web services. In Proceedings of the 13th international Conference on World Wide Web, WWW '04. ACM, New York, 53-62, 2004.
- [15] Yonglei Yao, Fangchun Yang, and Sen Su, Flexible Decision Making in Web Services Negotiation, Lecture Notes in Computer Science, Volume 4183/2006, 108-117, 2006
- [16] O. Lassila, R. Swick and others, Resource Description Framework (RDF) Model and Syntax Specification, WWW Consortium, 1999.
- [17] D. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C recommendation, World Wide Web Consortium, 2004.
- [18] Whats in a Service Level Agreement?', SLA@SOI, <http://sla-at-soi.eu/?p=356>, retrieved on March, 13 2012.
- [19] NIST Cloud Computing Use Case 3.9: Query Cloud-Provider Capabilities and Capacities, http://www.nist.gov/itl/cloud/3_9.cfm, retrieved on February 25, 2012
- [20] Google Apps Business Agreement , http://www.google.com/apps/intl/en-in/terms/premier_terms.html, retrieved on 12/10/14
- [21] Service Level Agreement for HP Compute Cloud, <http://www.hpcloud.com/sla/compute>, retrieved 12/10/14
- [22] Microsoft Azure SLA, <http://www.microsoft.com/en-us/download/details.aspx?id=44584>, retrieved 12/10/14
- [23] Amazon EC2 Service Level Agreement, <http://aws.amazon.com/ec2/sla/>, retrieved 12/10/14
- [24] E. Prud'hommeaux and A. Seaborne, SPARQL Query Language for RDF, , W3C recommendation Jan 2008, <http://www.w3.org/TR/rdf-sparql-query/>, retrieved on April 27, 2011
- [25] Karuna Joshi, Ontology for Services on the Cloud, <http://ebiquity.umbc.edu/resource/html/id/318/Ontology-for-Services-on-the-Cloud>, June 2011
- [26] K. Joshi and T. Finin, Ontology for Cloud Services SLA, <http://ebiquity.umbc.edu/resource/html/id/344/Ontology-for-Cloud-Services-SLA-Service-Level-Agreement->, April 2012
- [27] Jena Apache Fuseki: serving RDF data over HTTP, http://jena.apache.org/documentation/serving_data/index.html
- [28] Gartner Predicts Infrastructure Services Will Accelerate cloud Computing Growth (2013, February 19). Retrieved April 12, 2014 from <http://www.forbes.com/sites/louiscolumbus/2013/02/19/gartner-predicts-infrastructure-services-will-accelerate-cloud-computing-growth/>
- [29] Amazon EC2 SLA, <http://aws.amazon.com/ec2-sla>.