

Iterating Cloud Service Lifecycle Using Semantic Technologies

Archana Chauhan

M.tech (Software Engineering) JNTUH, India

Abstract: Managing virtualized services efficiently over the cloud is an open challenge. We propose a semantically rich, policy-based framework to iterate the lifecycle of cloud services. We have divided the IT service lifecycle into the five phases viz Requirements, Discovery, Negotiation, Composition, and Consumption. We detail each phase and describe the high level ontologies that we have developed to describe them. Our research complements previous work on ontologies for service acquisition on the cloud with focus on supporting negotiation for the key deliverable of IT services.

Keywords: Cloud Computing, services lifecycle, iteration.

1. RESEARCH MOTIVATION

Information Technology (IT) products and services which were previously either in-house or outsourced are now being replaced by a new delivery model where business-ess purchase IT components like software, hardware or network bandwidth or human agents as services from providers who can be based anywhere in the world. Such services will readily be acquired “on demand”, and composed on the fly by combining pre-existing components. In such scenario, multiple providers will collaborate to create a single service and each component service will be virtualized and might participate in many composite service orchestrations. The service, in effect, will be virtualized on the cloud. In any organization service acquisition will be driven by enterprise specific processes and policies that will constraint this acquisition.

The academic community has recently become interested in iterating steps needed to acquire services from the cloud. However, researchers have concentrated on issues like service discovery or service composition. For our research, we have worked on iterating the entire lifecycle of services on the cloud from discovery, negotiation to composition and performance monitoring. We outline our proposed methodology in below section.

2. MOTIVATION

Currently, providers decide how the services are bundled together and delivered to the consumers on the cloud. This is typically done statically and followed manual process. There is a need to develop reusable, user-centric mechanisms that will allow the service consumer to specify their desired quality constraints, and have automatic systems at the providers end control the selection, configuration and composition of services. This should be without requiring the consumer to understand the technical aspects of services and service composition. The U.S. National Institute of Standards and Technology (NIST) has identified [18] on-demand self-service as an essential characteristic of the cloud model where a consumer can unilaterally provision computing capabilities as needed automatically, without requiring human interaction with each service’s provider.

3. OUR APPROACH

We believe that a semantically rich, policy-based framework will facilitate the automation of the lifecycle of virtualized services. We have developed an integrated methodology that encompasses the entire data and process flow of services on the cloud from inception, creation/composition to consumption/ monitoring and termination. We have identified processes and policies to iterate this lifecycle.

We have used semantically rich descriptions of the requirements, constraints, and capabilities that are needed by each phase of the lifecycle. This methodology is complementary to, and leverages, previous work on ontologies, like OWL-S [14], for service descriptions in that it is focused on automating processes needed to procure services on the cloud. We have concentrated on enabling multiple iterations of service negotiation with constraints being relaxed during each iteration till a service match is obtained. Using Semantic web technologies like OWL [8] and Jena[15], we have also created high level ontologies for the various phases. These can be reasoned over to automate the phases guided by high level policy constraints provided by consumers, service customers, or service providers. Semantic web technologies also address the interoperability and portability issue of cloud computing.

4. EVALUATION

As this is a new line of research, there is currently no success criteria based on comparisons with previous work. We are developing a system as a proof of concept and will validate it against example enterprise policies obtained from various organizations. Towards that we are collaborating with NIST, a federal government organization. In collaboration with NIST, we are designing a prototype of our proposed methodology which will demonstrate automatic acquisition of services on the cloud. This is being developed by using Semantic Web technologies like OWL [8], RDF [10], and SPARQL [11].

IT services. We will calibrate our system to demonstrate how distinct processes of service acquisition can be captured and how enterprise policies can be expressed using our ontology and other policy languages to show that the service acquisition process can be iterated. One of our research goals is to develop appropriate ways to evaluate our system. One measure is existential -- can we create a system that will iterate the service acquisition process via our lifecycle's realization. As that our approach relies on semantically rich policies, another measure is how well our ontologies and policy mechanisms handle a given real-world policy. Given the large and diverse organization from which we are seeking policies, we hope to get a good representative sample to determine which enterprise policies can be automated for service acquisition and which cannot be. So our evaluation might also enable us to determine how useful such an iterated service procurement system will be in the real world. These are our preliminary ideas on evaluating our research which we plan to refine further.

5. RELATED WORK

Most approaches to iterate the acquisition services on the cloud have been limited to exploring a single aspect of the lifecycle like service discovery, service composition or service quality. There is no integrated prototype for the entire service lifecycle covering service planning, development and deployment in the cloud. In addition, most of the work is limited to the software component of the service and does not cover the service processes or human agents which are a critical component of IT Services.

Papazoglou and Heuvel [1] have proposed a methodology for developing and deploying web services using service oriented architectures. Their approach, however, is limited to the creation and deployment of web services and does not account for virtualized environment where services are composed on demand. Providers may need to combine their services with other resources or providers. Services to meet consumer needs. Other methodologies, like that proposed by Bianchini et al. [2], do not provide this flexibility and are limited to cases where a single service provider provides one service. Zeng et al. [3] address the quality based selection of composite services via a global planning approach but do not cover the human factors in quality metrics used for selecting the components. Maximilien and Singh [4] propose an ontology to capture quality of a web service so that quality attributes can be used while selecting a service. While their ontology can serve as a key building block in our system, it is limited by the fact that it considers single web services, rather than service compositions. Black et al. [5] proposed an integrated model for IT service management, but it is limited to managing the service from only the service provider's perspective. Paurobally et al. [12] have described a framework for web service negotiation using the iterated Contract Net Protocol [13]. However their implementation is limited to pre-existing web services and does not extend to virtualized services that are composed on demand. Our negotiation protocol, as detailed in [21], accounts for the fact that the service will be composed only after the contract/SLA listing the constraints is finalized. Good Relations [17] is an ontology developed for E-commerce to describe products. While this ontology is useful for describing service components that already exist on the cloud, it is difficult to describe composite virtualized services being provided by multiple vendors using this ontology. Researchers like Sbodio et. al [16] have proposed algorithms for service discovery using SPARQL language. We are using SPARQL

and other associated Semantic Web technologies to allow complex negotiation process between service providers and service consumers.

The Information Technology Infrastructure Library (ITIL) is a set of concepts and policies for managing IT infrastructure, development and operations that has wide acceptance in the industry. The latest version of ITIL lists policies for managing IT services [7] that cover aspects of service strategy, service design, service transition, service operation and continual service improvement. However, it is limited to interpreting “IT services” as products and applications that are offered by in-house IT department or IT consulting companies to an organization. This framework in its present form does not extend to the service cloud or a virtualized environment that consists of one or more composite services generated on demand.

6. RESEARCH CONTRIBUTION

Service Lifecycle Ontology:

We have developed a new methodology which integrates all the processes and data flows that are needed to automatically acquire, consume and manage services on the cloud. We divide this IT service lifecycle on a cloud into five phases. In sequential order of execution, they are requirements, discovery, negotiation, composition, and consumption. We have described these phases in detail along with the associated metrics in [19]. Figure 1 is a pictorial representation detailing the processes and data flow of the five phases. We have developed the ontology for the entire lifecycle in OWL 2 DL profile, available at [22].

The IT service lifecycle on a virtualized cloud comprises five main phases:

1. Requirements
2. Discovery
3. Negotiation,
4. Composition and
5. Consumption

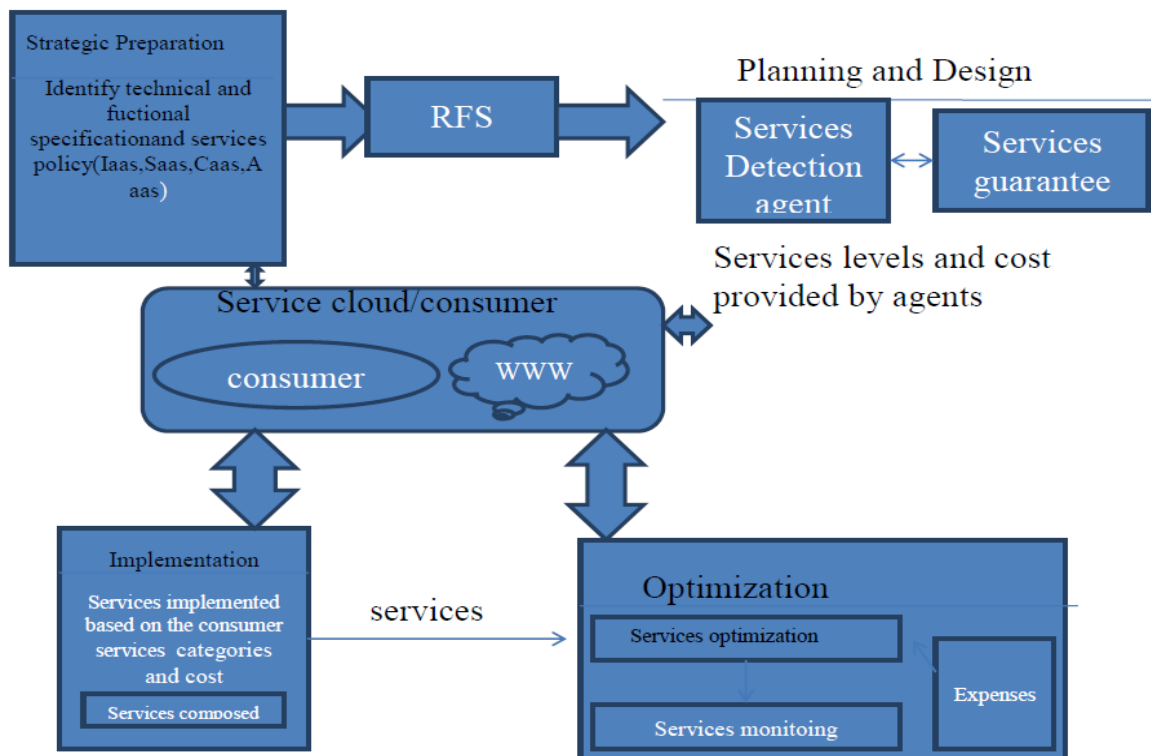


Figure 1: The IT service lifecycle on a cloud

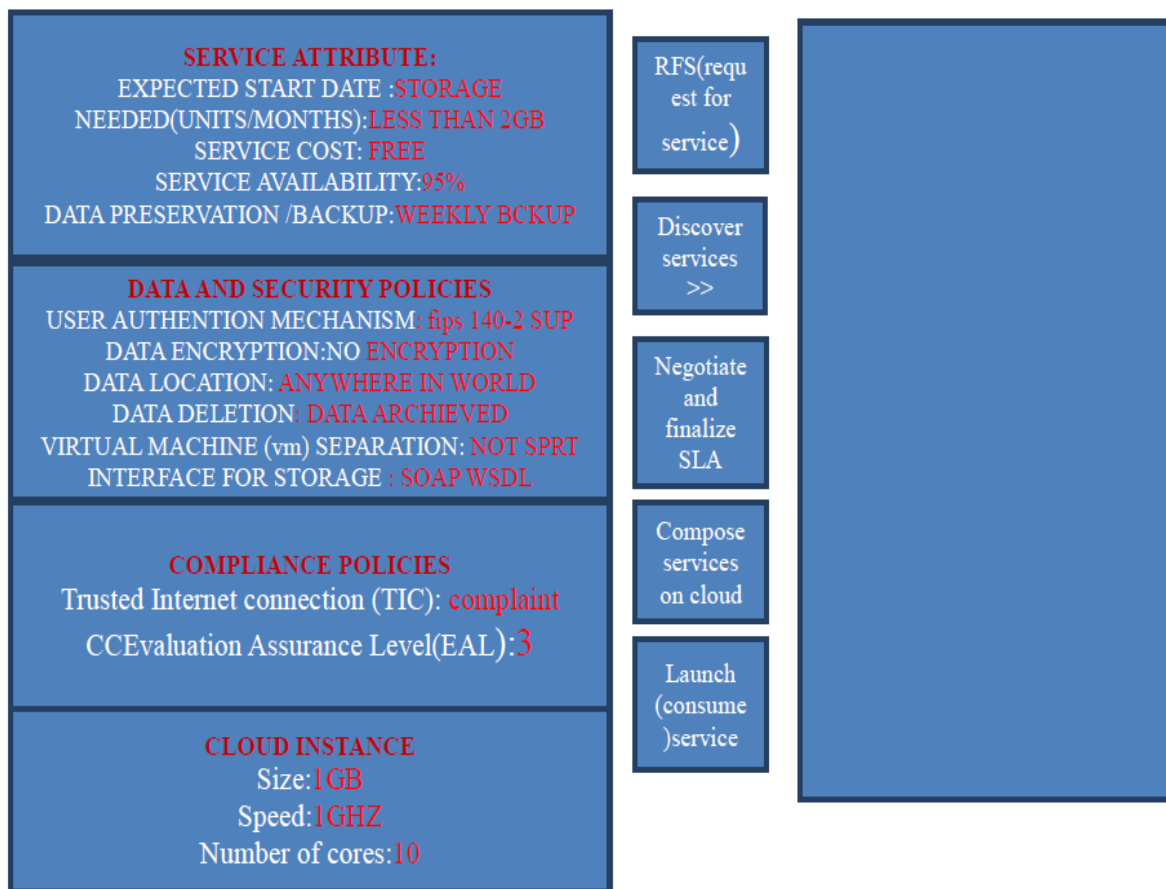


Figure 2: Prototype template for the RFS

Service Templates Includes:

1) Service Level Agreement:

A service level agreement is a formal negotiated agreement which helps to identify expectations, clarify responsibilities, and facilitate communication between two parties, typically a service provider and its customers. At the side of service provides, initially they have to define their service SLA templates in order to provide the services. Here we have focused on the cloud storage service and its template creation.

2) Requirement Analysis:

In the service requirements phase, the consumer details the technical and functional specifications that a service needs to fulfill. While defining the service requirements, the consumer also specifies nonfunctional attributes like characteristics of the human agent providing the service, constraints and preferences on data quality, and required security policies for the service. Service compliance details like certifications needed, standards to be adhered to, and so on, are also identified. The technical specifications lay down the hardware, software, application standards, and language support policies to which a service should adhere. Once the consumers have identified and classified their service needs, they issue a Request for Service (RFS). This RFS can be generated in a machine readable format using Semantic Web technologies.

3) SLA Discovery:

In the service discovery phase, providers are discovered by comparing the specifications listed in the RFS with service descriptions. The discovery is constrained by functional and technical attributes defined, and also by the budgetary, security, compliance, data quality, and agent policies of the consumer. An organization can release the RFS to a limited preapproved set of providers. Alternatively, it can search for all possible vendors. While searching the provider, service search engines or cloud brokers can be employed. A cloud broker role has been identified in the NIST reference architecture which we use in our ontology. This cloud broker runs a query against the services registered with a central registry or governing body and matches the service layer, domain, data type, compliance needs, and functional and

technical specifications and returns the result with the service providers matching the maximum number of requirements listed at the top.

4) SLA Negotiation:

The service negotiation phase covers the discussion and agreement that the service provider and consumer have regarding the service delivered and its acceptance criteria. 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 SPARQL queries is itself a performance improvement over the existing human-based negotiation. The service to be delivered is determined by the specifications laid down in the RFS. Service acceptance is usually guided by the service level agreements 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.

5) SLA Finalization:

Service is delivered to the consumer based on the delivery mode (synchronous/asynchronous, real time, batch mode, etc.) agreed upon in the negotiation phase. After the service is delivered to the consumer, payment is made for the same based on the pricing model agreed to in the SLA. The consumer then begins consuming the service. In a cloud environment, the service usually resides on remote machines managed by the service providers.

6) SLA Monitoring:

Administering, managing, and monitoring the service lies with the provider. In this phase, consumer will require tools that enable service quality monitoring and service termination if needed. This will involve alerts to humans or automatic termination based on policies defined using the quality-related ontologies. The service monitor measures the service quality and compares it with the quality levels defined in the SLA. This phase spans both the consumer and cloud areas as performance monitoring is a joint responsibility.

7) PHASES OF CLOUD SERVICES:

7.1 Service Requirement Phase:

In this phase, the consumer details the technical and functional specifications that a service needs to fulfill and also non-functional attributes such as characteristics of the providing agent, constraints and preferences on data quality and required security policies for the service. Service compliance details like required certifications, standards to be adhered to etc. are also identified. Depending on the service cost and availability, a consumer may be amenable to compromise on the service quality. Functional specification describe in detail what functions/tasks should a service help automate. The technical specifications lay down the hardware, software, application standards and language support policies to which a service should adhere. Once the consumers have identified and classified their service needs, they issue a Request for Service (RFS). This request could be made by directly contacting a few service providers for their quotes. Alternatively, consumers can use a service discovery engine or service broker on the cloud to procure the service.

The two top classes of this phase are the Specification class and the "Request for Service" class. The Specification class consists of six main classes that define the functional specifications, technical specifications, Human agent specifications, security policies, service compliance policies and data quality policies. The technical specs contain information about the Hardware, Operating System and other compatible services/applications that the desired service should conform to. Human Agent specifications also list the technical and domain expertise that the service providing agent should have. Part of our ongoing work is to use existing ontologies that have been developed for classes like standard hardware, operating systems and computer applications. Semantic Web policy languages can be used to describe service specifications and constraints in machine-processable format. For instance, a consumer may opt for a service with poor data quality to take advantage of low cost of the service.

7.2 Service Discovery Phase:

Service providers are discovered by comparing the specifications listed in the RFS. The discovery is constrained by functional and technical attributes defined, and also by the budgetary, security, compliance, data quality and agent policies

of the consumer. While searching the cloud, service search engines or service brokers can be employed. This engine runs a query against the services registered with a central registry or governing body and matches the domain, data type, compliance needs, functional and technical specifications and returns the result with the service providers matching the maximum number of requirements listed at the top.

One critical part of this phase is service certification, in which the consumers will contact a central registry, like UDDI [6], to get references for providers that they narrow down to this phase uses the RFS class from the requirements phase to search for service providers and generate a list of providers with which to begin negotiations. The class Service certification validates the providers' credentials. If the consumers find the exact service within their budgets, they can begin consuming the service immediately upon payment. However, often the consumers will get a list of providers who will need to compose a service to meet the consumers. Specifications. The consumer will then have to begin negotiations with the service providers which is the next phase of the lifecycle. Each search result will return the primary provider who will be negotiating with the consumer.

7.3 Service Negotiation phase:

The service negotiation phase covers the discussion and agreement that the service provider and consumer have regarding the service delivered and its acceptance criteria. The service delivered is determined by the specifications laid down in the RFS Service acceptance is usually guided by the Service Level Agreements (SLA) [9] 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 required.

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 the 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 service providers to ensure that SLA metrics are met. The negotiation steps are detailed in [21].

This phase uses the RFS class from the requirements phase and the provider's list class from the discovery phase to negotiate the contracts between consumer and primary provider and between the various component providers themselves. The key deliverable of this phase 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. Another deliverable of this phase are the service sub contracts between the service provider and component (or dependent services) providers. The QoS are the essential part of the service sub-contracts and are used in the consumption phase to monitor service performance.

7.4 Service Composition phase:

In this phase one or more services provided by one or more providers are combined and delivered as a single service. Service orchestration determines the sequence of the service components.

The main class of this phase is the Service class that combines the various components into a single service. We include the OWL-S [14] Composite Process class ontology. The Service class takes inputs from the Specification, Service Contracts and Service Level Agreement classes defined in the earlier phases to help determine the orchestration of the various components.

7.5 Service Consumption/Monitoring phase:

The service is delivered to the consumer based on the delivery mode (synchronous/asynchronous, real-time, batch mode etc.) agreed upon in the negotiation phase. After the service is delivered to the consumer, payment is made for the same. The consumer then begins consuming the service. In a cloud environment, the service usually resides on remote machines managed by the service providers. Hence the onus for administrating, managing and monitoring the service lies with the provider. In this phase, consumer will require tools that enable service quality monitoring and service termination if needed. This will involve alerts to humans or automatic termination based on policies defined using the quality related ontologies. The Service Monitor measures the service quality and compares it with the quality levels defined in the SLA.

This phase spans both the consumer and cloud areas as performance monitoring is a joint responsibility. If the consumer is not satisfied with the service quality, s/he should have the option to terminate the service and stop service payment. The composite service is composed of human agents providing the service, the service software and dependent service components. All the three elements, agents, software and dependent services, must be monitored to manage the overall service quality. For the service software providers have to track its performance, reliability, assurance and presentation as they will influence customer's satisfaction rating (CSATs). Since the dependent services/components will be at the backend and will not interface directly with the consumers, the service provider only needs to monitor their performance. We have proposed a framework to manage quality based on fuzzy logic for such composed services delivered on the cloud in [20].

7. SUMMARY

In this paper, we have defined an integrated prototype for processes needed to iterate IT services life cycle on the cloud. To the best of our knowledge, this is the first such effort, and it is critical as it provides a virtualized view of steps involved in deploying IT services on to the cloud. Our approach complements previous work on ontologies for service descriptions that it is focused on iterating the processes needed to procure services on the cloud. The methodology can be referenced by organizations to determine what key deliverables they can expect at any stage of the process. It is the most cost effective and fast emerging standard in the industry. Services on the cloud are iterated and available on the cloud for different sectors of the industry based on the required service and the budget .The prototype that we built demonstrated how our methodology can be used significantly to iterate the acquisition and consumption of cloud-based discovered services thereby reducing the large time required by companies to discover and procure cloud-based services. We are currently refining the ontology to capture the steps and metrics we have identified in the lifecycle using semantic web languages. We are also creating the prototype by using Semantic web technologies like SPARQL, OWL and RDF.

REFERENCES

- [1] M. Papazoglou and W. Van Den Heuvel, Service-oriented design and development methodology, International Journal of Web Engineering and Technology, Volume 2, Number 4, 2006, pp. 412 – 442
- [2] D. Bianchini, V. De Antonellis, B. Pernici, P. Plebani, Ontology-based methodology for eservice discovery, International Journal of Information Systems, The Semantic Web and Web Services, Volume 31, Issues 4-5, June-July 2006, pp 361-380
- [3] L Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q. Sheng, Quality driven web services composition, Proceedings of the 12th international conference on World Wide Web, 2003, pp 411 - 421
- [4] E. M. Maximilien, M. Singh, A Framework and Ontology for Dynamic Web Services Selection, IEEE Internet Computing, vol. 8, no. 5, pp. 84-93, Sep./Oct. 2004
- [5] J. Black et al, An integration model for organizing IT service Management, IBM Systems Journal, VOL 46, NO 3, 2007.
- [6] S Ran, A model for web services discovery with QoS, ACM SIGecom Exchanges, Vol 4, Issue 1, 2003, pp 1-10, 2003.
- [7] J Van Bon et. al., Foundations of IT service management based on ITIL V3 , Van Hatem Publishing, 2008
- [8] D. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C recommendation, World Wide Web Consortium, 2004.
- [9] Whats in a Service Level Agreement?., SLA@SOI, <http://sla-at-soi.eu/?p=356>, retrieved on July 30, 2009.
- [10] O. Lassila, R. Swick and others, Resource Description Framework (RDF) Model and Syntax Specification, World Wide Web Consortium, 1999.
- [11] E. Prud'hommeaux and A. Seaborne, SPARQL Query Language for RDF, W3C recommendation, <http://www.w3.org/TR/rdf-sparql-query/>, retrieved on April 27, 2011
- [12] S. Paurobally, V. Tamma and M. Wooldridge, A Framework for Web Service Negotiation, ACM Transactions on Autonomous and Adaptive Systems, Vol. 2, No. 4, Article 14, November 2007.
- [13] R. Smith, The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, IEEE Transactions on computers, Volume C-29, Issue 12, 1980, pp 1104-1113.

- [14] D. Martin, et al., Bringing semantics to web services: The OWL-S approach, Lecture Notes in Computer Science, volume 3387, pp. 26-42, 2005, Springer.
- [15] Jena– A Semantic Web Framework for Java, <http://jena.sourceforge.net/>, retrieved on May 10, 2011.
- [16] M. L. Sbodio, D. Martin, and C. Moulin, “Discovering Semantic Web services using SPARQL and intelligent agents.” *Journal of Web Semant.* 8, 4 (November 2010), 310-328.
- [17] M. Hepp, “GoodRelations: An Ontology for Describing Products and Services Offers on the Web”, Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW2008), Italy, 2008, Springer LNCS, Vol 5268, pp. 332-347.
- [18] NIST initiative to develop standards for Cloud Computing, <http://www.nist.gov/itl/cloud/>, last retrieved July 2011
- [19] K. Joshi, T. Finin, and Y. Yesha, "Integrated Lifecycle of IT Services in a Cloud Environment", in Proceedings of The Third International Conference on the Virtual Computing Initiative (ICVCI 2009), Research Triangle Park, NC, October 2009
- [20] K. Joshi, A. Joshi and Y. Yesha, “Managing the Quality of Virtualized Services”, in proceedings of the SRII Global conference, San Jose, March 2011.
- [21] K. Joshi, T. Finin, and Y. Yesha, "A Semantic Approach to Automate Service Management in the Cloud ", UMBC Technical Report, TR-CS-11-02, June 2011
- [22] K. Joshi, OWL Ontology for Lifecycle of IT Services on the Cloud, <http://ebiquity.umbc.edu/ontologies/itso/ 1.0/itso.owl>