A Semantic Model to Characterize Pricing and Negotiation Schemes of Cloud Resources

Giuseppe Di Modica, Orazio Tomarchio
Department of Electric, Electronic and Computer Science Engineering
University of Catania
Catania, Italy
Email: {Giuseppe.DiModica, Orazio.Tomarchio}@dieei.unict.it

***Abstract*—Cloud computing technology has reached a good level of maturity. The market of cloud resources is still dominated by proprietary solutions for what concerns resource delivering, pricing models and Service Level Agreement. The research community is working hard to define specifications that try to standardize most of these aspects. When standards will get mature, customers will no more be locked-up to any proprietary technology, and full interoperability among clouds will be a reality. In the future cloud resource market the competition challenge will be played on the real capability of providers to accommodate customers' requests in a flexible way and to supply high and differentiated QoS levels. In this scenario a mechanism must be devised to support the matchmaking between what providers offer and what customers' applications demand. In this work we propose the definition of a semantic model to support the supply-demand matchmaking process in the future cloud markets. Leveraging on a semantic description of the cloud resources' features, customers will be able to discover cloud offers that best suit their own business needs. Tests conducted on an implementation prototype proved the viability of the approach.**

***Keywords*-Cloud computing; Price model; SLA negotiation; Ontology.**

I. INTRODUCTION

Cloud computing [1] has emerged as a paradigm able to offer resources in a flexible, dynamic, resilient and cost-effective way. Following the *service-oriented* paradigm, all cloud resources, both physical and logical, are virtualized and are offered "as-a-service". The real success of the cloud is mostly due to the considerable business opportunities that it produces for both consumers and providers of virtualized resources. On the one hand, the providers see in the cloud model a way to maximize the use of their computing assets and thus minimize the maintenance cost; on the other hand, the "pay-per-use" business model allows consumers to pay for only what they actually use, without any initial investment.

However, today we are still far from an open and competitive cloud and service market, where cloud resources are traded as in conventional markets. The main technological reason for this is the lack of interoperability of existing cloud technology [2], which is also leading to the phenomenon of *vendor lock-in*.

Another not technological, yet equally important reason is that, to date, cloud resources are offered according to strict pricing models and rigid Service Level Agreements (SLA). In a future open cloud market, users (customers) will demand for flexible pricing and resources' usage schemes to meet their specific computing needs; providers will have to negotiate with the customers for differentiated levels of quality of service.

In this paper we discuss about the need of more flexible charging models for cloud resources' usage, together with advanced negotiation protocols that could better support the public cloud model. We believe that, in order to build an effective matchmaking process between supply and demand, a structured model to describe resources' business features and applications' requirements is needed. To this purpose, we propose two ontologies for describing the resources offered by cloud providers on the one hand, and the application requirements expressed by customers on the other one. The final aim is to efficiently include pricing models, negotiation capabilities and service levels into resource publish/discovery mechanisms, that can then be enriched with tools to enable providers to easily characterize and advertise their resources, and customers to easily describe application requirements. A semantic matchmaking algorithm has been devised enabling customers to search for those cloud resources that best meet their requirements. A first prototype of the semantic discovery framework has been implemented. Experimental results show that semantic technologies are a powerful means that enhance the way resources' supply and demand are expressed and matched in the cloud markets.

The remainder of the paper is organized as follows. In Section II the background context is introduced and the issues inspiring this work are discussed. Section III describes the approach proposed for the definition of a cloud service discovery framework, and provides details on the mapping and the matching processes respectively. The implementation of a system prototype and results from tests are described in Section IV. In Section V recent works in literature are discussed, outlining the novelty of the idea proposed in this paper. We conclude the work in Section VI.

ISBN: 978-1-61208-216-5

II. ANALYSIS OF CURRENT CLOUD OFFERING

The commercial success of cloud computing is witnessed by the individual success of few, very big companies that, by imposing their own proprietary solutions (e.g., Amazon's ".ami" and "EC2"), have made and are currently making huge profits by leasing their unused computational resources.

In a desirable scenario, the customer should be able to build up his own application independently of the specific cloud that it is going to run onto, define the application requirements in a standardized way, look for the cloud provider that best meets the requirements, negotiate for the service, deploy the application, monitor the application performance, move it to another cloud in the case that the service performance does not meet his expectation. However, the road that leads to cloud interoperability is long, because of several issues that still need to be addressed [2]. When such a target will be accomplished, a new scenario of business opportunities will open up to the old and the new stakeholders that will want to profit from the open market of cloud-based resources. Interoperability is the means by which also small companies can federate to each other to share their resources and propose themselves as an alternative to the big players. The European FP7 project Reservoir [3] is one of the first successful attempts to create an interoperable federation of cloud providers, spanning across different administrative domains, aiming at sharing their individual resources to respond to the customers' demand.

In the following, we analyze the panorama of the current cloud offerings by taking pricing models, negotiation protocols and service performance levels as key factors. We also take into account the customers' point of view, by analyzing how customers are used to characterize and specify application requirements under their business perspective.

A. Price Models

The main cloud paradigm's claimed strength is that resources (computing, storage, network) can be accessed on an *On-Demand* basis, and customers can be charged according to the actual resources' usage time. In particular, the CPU is usually charged by the hour, the data storage service is charged per GB/month, the data transfer service over the network is charged per GB. Providers also propose their customers an alternative pricing model based on *Resource Reservation*, which on the cloud provider's end provides an instant economic benefit (they receive an immediate payment for the reservation), and on the customer's end allows to save on the resource price provided that the resource itself is intensively used. Other cloud providers, instead of leasing "raw" computing resources, offers cloud-based services in the form of developing and execution platforms (PaaS) and applications (SaaS). Some decide to charge the customer according to the usage that the provided service make of the underlying raw resources. Others (mostly providing business oriented services) adopt a model that is more suited to those business applications that, once deployed, involve the interaction of many end users. The customer is then charged by month and by the number of end users that the application will have to serve (we refer to this model as *End-User-Based*). Finally, almost all commercial providers propose a *Free-Of-Charge* model, which is nothing but a try-before-buy strategy.

In the forthcoming cloud economy generation other pricing models might result more attractive for both the providers' and the customers' needs. In the process of optimizing the usage of internal resources, providers might want to encourage customers to access their resources during specific periods of underutilization (at night, or during the weekends), and thus would be willing to charge customers according to ad-hoc, time-oriented models. Again, in the same way like mobile phone operators do, providers might even offer their customers pre-paid packages of resources to be consumed as they like.

B. Negotiation Protocols

In the literature several proposals for the negotiation and management of SLA have appeared in the context of GRID and SOA, but many address the same issues in the cloud computing context too. Actually most of them provide limited or no support for dynamic SLAs negotiation, which we believe to be a strict requirement for the future cloud markets. As for the negotiation protocol, the OGF's WS-AgreementNegotiation [4] is the most notable standardization effort. The proposal is an extension of the former WS-Agreement recommendation, and is still in progress. It just supports the one-to-one negotiation scheme and the very simple offer/counter-offer dynamics. The approach is not efficient and flexible enough for complex application areas. Alternatives (such as **auctions**[5]) are also suggested as more appropriate for highly dynamic context. One of the objectives of the SLA@SOI European FP7 project [6] was to provide negotiation mechanisms for exchanging offers and counter offers between customers and providers in a SOA context. The implemented framework (SLAM) promises support for both one-to-one and one-to-many negotiations, allows for multiple rounds of negotiation, and can be adopted in agent marketplace as well as broker based architectures. The Vienna Service Level Agreement Framework (VieSLAF) architecture for cloud service management [7] introduces the concept of meta-negotiations to allow two parties to reach an SLA on what specific negotiation protocols, security standards, and documents to use before starting the actual negotiation.

In the actual market of virtualized resources, we notice that Amazon has launched the *Spot Instances* model, which can be seen as an example of a particular negotiation model that has been adopted to resolve the customers' competition on the provider's unused resources. Depending on the provider's business strategies and on the amount of unused resources, other negotiation models might be employed. We claim that auction-based models would bring benefits to providers and to customers as well. The latter will have the chance to search for resources according to the associated negotiation scheme that best suite their own business strategies and needs.

C. Service Performance Levels

The performance features that cloud providers advertise are usually vague, just focusing on virtual machines computation speed.The only parameter which is quantitatively expressed and granted by all the commercial providers is the *availability* of resources. All providers guarantee a very high level of resource availability (from 99% upwards), prevent any user data loss by allocating extra back-up storage, support customers to face any technical issue. The competition among the providers is played on both the price at which the resources are sold and the capability of sustaining the promised, *guaranteed service levels*. Some providers further differentiate their service offer. Besides provisioning what we call a standard *basic service level*, which is the core activity of their business, some of them also offer a *premium service level*, which provides more guarantees than the basic and adds extra services.

In the future, in order to satisfy the customers' heterogeneous and dynamic business requirements, the cloud providers might be encouraged to propose new models. To cater for more fine-grained customer requirements, providers might want to propose customizable plans of service levels, that will enable customers to build their own desired service level provisioning.

D. Application Requirements

Every application needing some computing power could technically run on a cloud. Still security is a big concern that prevents service providers from unconditionally deploying their applications on the cloud. Generally speaking, before moving an application to the cloud a cost/benefit analysis must be carefully done. The decision concerns both whether to move onto the cloud or not, and to select the cloud offer that fits.

One should verify, according to the company's business objectives to be accomplished and to how much mission-critical the application is, whether the application to be deployed requires a guaranteed service level or a best effort is enough. If the former is to be chosen, again, depending on the business requirements of the application, a choice has to be made between a basic or a premium service level. Further on, the choice of the pricing model that best fits must be made according to the *application's profile*, i.e., the application's specific usage pattern: if such pattern is "dense" (resources are continuously used within a time frame), reserved-based solutions are to be preferred; otherwise, the on-demand pricing model will result more convenient.

All the choices must be made checking that the budget they require is compatible with the company's investment capability. For example, a service level might fit a given application's profile, but might not be affordable for the company; on the contrary, a more affordable service level would make the company save money, but might not fit the strict application's requirements. In most cases a compromise must be searched for.

III. CLOUD SERVICE DISCOVERY FRAMEWORK

The previous analysis of current cloud offering has shown that provider and customer perspectives are quite different. The former seeks to maximize the profit and the utilization level of the IT asset that they have invested on. The latter just needs to make fine-tuned searches in the market in order to discover the service fitting their specific business needs. We have then designed a service discovery framework that exploits semantic mechanisms to favour the matchmaking of the providers' offer and the customers' demand. Two OWL-based ontologies have been developed to characterize respectively the provider and the customer perspectives. In particular, the first ontology semantically describes the features of the resources being offered by cloud providers (see Figure 1), the second one describes the application's business requirements demanded by customers (see Figure 2). For a detailed description of these ontologies refer to [8]. Since each ontology contains semantic concepts belonging to two different domains, we have devised a mapping process that transforms application requirements into "semantically" equivalent resource features, i.e., features that best represent the application requirements in the domain of resources. The mapping's purpose is to put application requirements and resource features on a common semantic ground (that of cloud resources) on which a semantic procedure will try to make the match.

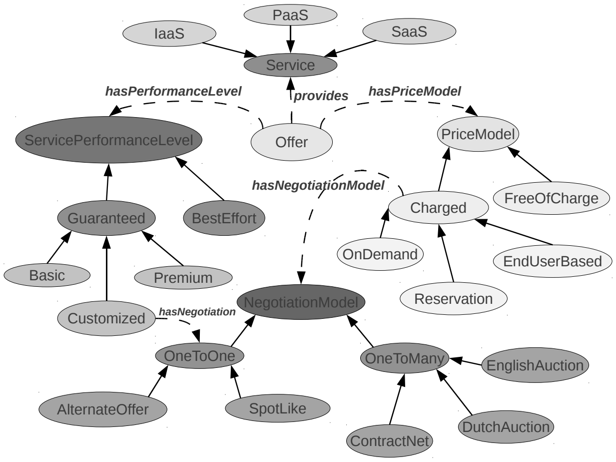

Figure 1. Resource features ontology

Figure 3 depicts the two semantic domains, along with the mapping and matchmaking processes. In the figure, the

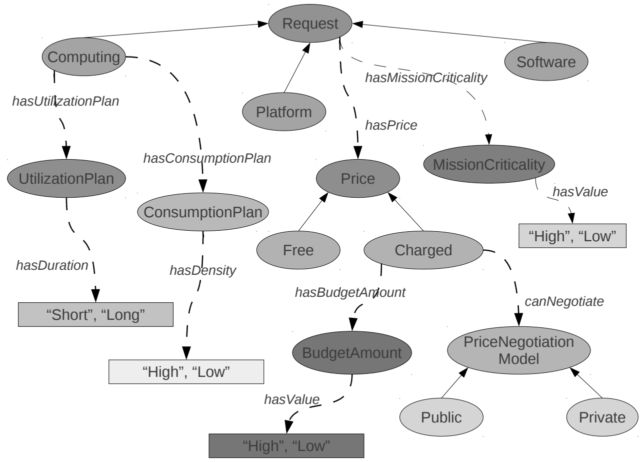

Figure 2. Application requirements ontology

filled circles represents, respectively, real requests issued by customers (within the application requirements' domain) and real offers advertised by service providers (resource features' domain). Through the mapping process the application requirement AR_4 is transformed into its "equivalent" resource feature offer RF_6 (empty circle) in the offers domain. Such resource feature does not necessarily coincide with a real offer, but rather represents the ideal offer that would perfectly match the considered application requirement. In the next step, the matchmaking procedure will explore the resource features' domain in order to search for concrete offers that show a *semantic affinity* to RF_6 (those covered by the gray area in the figure). The final outcome of the entire process will be a list of concrete offers, sorted by the semantic affinity degree, that may satisfy the needs represented by AR_4.

In the following subsections we provide some details on how the mapping and matchmaking processes work.

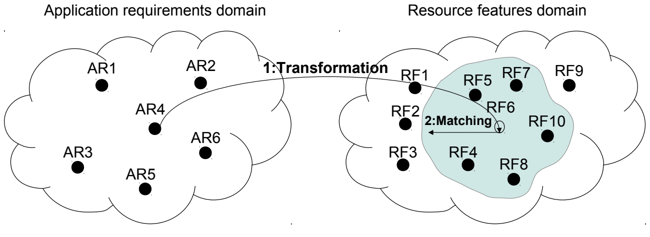

Figure 3. Mapping and matching

A. Mapping

The mapping process is a simple procedure that applies a list of mapping rules. Rules have been defined using the Semantic Web Rule Language (SWRL) [9]. SWRL was chosen since it is a W3C specification and it copes well with OWL-based ontologies. The objective of each rule is to transform a specific application requirement into the ideal, best matching resource feature.

A group of chained semantic rules drive the mapping from individuals of the Application requirements' ontology to individuals of the Resource features' ontology. A rule engine takes a request in input, applies the sequence of rules and, according on the rules that match, incrementally builds up the ideal offer. For the sake of brevity, we report only a significant subset of rules:

1) $request : Request(?request) \wedge offer : Offer(?offer)$
   $\rightarrow hasMatchedOffer(?request, ?offer)$
2) $hasMatchedOffer(?request, ?offer) \wedge$
   $request : Computing(?request) \wedge$
   $offer : provides(?offer, ?service)$
   $\rightarrow offer : IaaS(?service)$
3) $hasMatchedOffer(?request, ?offer) \wedge$
   $request : Platform(?request) \wedge$
   $offer : provides(?offer, ?service)$
   $\rightarrow offer : PaaS(?service)$
4) $hasMatchedOffer(?request, ?offer) \wedge$
   $request : Software(?request) \wedge$
   $offer : provides(?offer, ?service)$
   $\rightarrow offer : SaaS(?service)$

Rule 1 just states that, given a generic request in the application requirements' domain, a corresponding ideal offer exists in the resource features' domain. Rules 2 through 4 handle the different type of cloud services that can be requested. The rules are very intuitive, and states that a request for Computing resource is mapped onto an offer of the type IaaS, a request for a Platform resource is mapped onto a PaaS offer, and a request for a Software resource maps to an offer of the type SaaS.

B. Matchmaking

After the mapping process has elaborated the ideal offer, the matchmaking process will start exploring the domain of the real offers in order to find those ones whose features best meet the initial application requirements. In particular, for each offer advertised in the market, the matchmaking process will evaluate the *semantic affinity* between that offer and the ideal one. The semantic affinity will reveal how close a real offer is to the customer expectations. The semantic affinity will be a value in the range [0,1], being 1 the highest achievable affinity. The function that calculates the semantic affinity is the following:

$$A = Serv_a * W_{serv} + Price_a * W_{price} + Perf_a * W_{perf} + Neg_a * W_{neg}$$

The overall affinity between the ideal offer and a real offer is obtained by summing up the sub-affinities evaluated on each offer's feature: service, price model, performance level and negotiation model. So, for instance, the addendum $Price_a * W_{price}$ represents the sub-affinity evaluated on the price feature. In particular, $Price_a$ is the outcome of the semantic comparison between the price concepts exposed by the two individuals (the offers), while W_{price} is a weight factor. We plan to use the weight factor to let the customer tune the affinity algorithm according to customizable priority criteria.

We now provide some details on the semantic comparison of concepts. Let O_j be a generic offer, characterized be the semantic concepts: $Serv_{o-j}$, $Price_{o-j}$, $Perf_{o-j}$, Neg_{o-j}. In order to evaluate the overall semantic affinity of two offers O_{ideal} (the ideal offer that is the outcome of the mapping process) and O_{real} (a real offer in the market place), couples of homologous concepts must be compared.

The semantic affinity values for all the possible cases are shown in the following:

- 1, if the two concepts are semantically equivalent;
- 1, if $C_{o-ideal}$ is the father of C_{o-real};
- 0.5, if the two concepts are siblings and the father is the root concept in the considered branch;
- 0.75, if the two concepts are siblings and the father is a non-root concept in the considered branch;
- 0, if $C_{o-ideal}$ is not expressed;
- 0.5 in any other case.

The algorithm assigns the highest value to equivalent concepts, or to concepts that are in a father-son relationship. Instead, it penalizes two concepts that are direct descendants of a root concept, as in our ontology siblings concepts whose father is root usually represent opposite concepts (e.g., Charged vs FreeOfCharge, Guaranteed vs BestEffort). Conversely, siblings whose father is a non-root concept are considered different but someway "close" concepts (e.g., On-Demand vs Reservation, EnglishAuction vs DutchAuction), therefore they are given a higher grade of affinity.

IV. IMPLEMENTATION AND TESTS

A prototype of the framework has been implemented and tested. The core of the framework consists of an *ontology mapper*, that makes use of a rules engine provided by the Jess library, a *matchmaker* supported by a semantic reasoner powered with the Pellet library, and a *repository* of advertised cloud offers. Customers are then provided with a front-end tool to build and submit cloud requests, while at this stage the repository of cloud offers was populated by hand. In the future we plan to implement a tool that will help providers to build their offers and push them to the repository. For the test purpose, we generated a complete set of offers spanning the whole semantic domain of resources' features. Afterwards, several different requests were generated, each of them asking for a specific cloud service. For each submitted request, the framework replied with a list of fitting offers. In the following we describe two sample requests and analyze the corresponding results provided by the discovery procedure. In the first request the customer asks for a service of type Platform, for whose price he is willing to negotiate in the context of a public auction:

$R1(Type : Platform, PriceModel : Charged,$
$NegotiationModel : Public)$

After submission, the mapping process transformed $R1$ into the following ideal offer:

$O1_{ideal}(Service : PaaS, PriceModel : EndUserBased,$
$ServicePerformanceLevel : Guaranteed,$
$NegotiationModel : OneToMany)$

For that offer, the matchmaking process produced the results depicted in Table I.

Table I
LIST OF MATCHING OFFERS FOR REQUEST $R1$

Offer #	Service	PriceModel	SPL	NegModel	Affinity
39	PaaS	EndUserBased	Basic	ContractNet	1.0
49	PaaS	EndUserBased	Premium	EnglishAuction	1.0
48	PaaS	EndUserBased	Premium	DutchAuction	1.0
...	...	...	...	...	...
36	PaaS	EndUserBased	Customized	n.a.	0.875
35	PaaS	EndUserBased	Basic	n.a.	0.875
...	...	...	...	...	...
3	SaaS	FreeOfCharge	BestEffort	n.a.	0.5
1	IaaS	FreeOfCharge	BestEffort	n.a.	0.5

As the list is very long many results have been omitted. On the top of the list the perfectly matching concrete offers appear. The offers with an affinity value of 0.875, have a partial matching, as those offers do not provide any negotiation. The offers at the bottom do not match because of differences in both the service type and the service performance level. Here is the second request that we tested:

$R2(Type : Computing,$
$UtilizationPlan- > hasDuration : long,$
$ConsumptionPlan- > hasDensity : high)$

After submission, the mapping process transformed $R2$ into the following ideal offer:

$O2_{ideal}(Service : IaaS, PriceModel : Reservation,$
$ServicePerformanceLevel : Premium,$
$NegotiationModel : no)$

For that offer, the matchmaking process produced the results depicted in Table II.

Table II
LIST OF MATCHING OFFERS FOR REQUEST $R2$

Offer #	Service	PriceModel	SPL	NegModel	Affinity
10	IaaS	Reservation	Premium	n.a.	1.0
...	...	...	...	...	...
33	IaaS	Reservation	Premium	DutchAuction	0.9375
32	IaaS	Reservation	Premium	ContractNet	0.9375
...	...	...	...	...	...
12	IaaS	OnDemand	Basic	ContractNet	0.8125
...	...	...	...	...	...

As expected, offers proposing the reservation-based price model have the best matching; in fact, they perfectly fit the application requirements concerning the utilization plan and the resource consumption plan. Offers that propose auctions are lightly penalized. Offers proposing a basic performance level get penalized even more.

V. RELATED WORK

Several standard organizations are working hard to propose specifications that will enable future scenarios of interoperable cloud services. An exhaustive review of the research efforts dealing with interoperability issues in cloud computing systems was produced in the context of Cloud4SOA project [10]. Some attempts to design ontologies for the definition of cloud-related concepts and relationships have recently appeared in the literature. Still, there is no proposal that has reached a broad acceptance from the community, nor all features of the cloud domain have been fully covered, so far, by existing proposals. Some works([11], [12]) have tried to define taxonomies for cloud-based systems. They mostly identify and classify cloud delivery models, services and resources; some also deals with requirements like fault tolerance and security. One of the most complete cloud taxonomy is maintained and continuously updated by OpenCrowd([13]): in this project, existing cloud providers and cloud-related software are classified according to a specific scheme. In the aim of defining an open and standardized cloud interface for the unification of cloud APIs, the Unified Cloud Interface (UCI) Project [14] has proposed and released an RDF-OWL cloud data model mostly covering the definition of resources in the cloud domain. To our knowledge, the mOSAIC ontology([15]) is the most complete ontology that was proposed so far. It inherits most of the elements defined in other proposals (OCCI, NIST, IBM), and covers aspects like deployment models, service models, resources, services, actors, consumers, functional and non functional properties, languages, APIs. The ontology was developed in OWL and is used for semantic retrieval and composition of cloud services in the mOSAIC project.

The work discussed in this paper aims at discussing aspects of cloud interoperability not covered by any of the works cited above. The proposed perspective is that of a global market of cloud resources, where there is the need of a characterization of what is offered and demanded by actors in terms of business profit and utility respectively. The proposed ontology, therefore, covers a new portion of the cloud's domain of knowledge; nonetheless, it can be integrated to existing ontologies/taxonomies.

VI. CONCLUSION AND FUTURE WORK

The future market of cloud services will have to provide novel and advanced matchmaking processes in order to account for the providers' and the customers' dynamic and heterogeneous business requirements, respectively in terms of profit and utility. The work presented here aims to define a cloud offer discovery framework based on semantic technologies. A matchmaking procedure has been devised to semantically search the offers' domain in order to provide the customer with a list of most profitable offers. Tests were run on a prototype of the framework and proved the viability of the proposed model. In the future, we are planning to enhance the semantic model by extending the ontologies and accordingly enriching the semantic rules.

REFERENCES

[1] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on*, Sep. 2008, pp. 5 –13.

[2] A. Parameswaran and A. Chaddha, "Cloud Interoperability and Standardization," *SETLabs Briefings*, vol. 7, no. 7, pp. 19–26, 2009.

[3] Reservoir Consortium, "The Reservoir Project," http://www.reservoir-fp7.eu/, [Last Retrieved: July 2012].

[4] The Open Grid Forum, "WS-Agreement Negotiation Specification," http://forge.gridforum.org/sf/go/projects.graap-wg/, 2011, [Last Retrieved: July 2012].

[5] V. Krishna, *Auction Theory*. Academic Press, 2002.

[6] The SLA@SOI Consortium, "SLA@SOI Negotiation Platform," http://sla-at-soi.eu/publications/, [Last Retrieved: July 2012].

[7] I. Brandic, D. Music, P. Leitner, and S. Dustdar, "Vieslaf framework: Enabling adaptive and versatile sla-management," in *Proceedings of the 6th International Workshop on Grid Economics and Business Models*, ser. GECON '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 60–73.

[8] G. Di Modica and O. Tomarchio, "A semantic discovery framework to support supply-demand matchmaking in cloud service markets," in *2nd International Conference on Cloud Computing and Services Science (CLOSER 2012)*, Apr. 2012.

[9] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., and Dean, M., "SWRL: A semantic web rule language combining OWL and RuleML," http://www.w3.org/Submission/SWRL/, 2004, [Last Retrieved: July 2012].

[10] N. Loutas, V. Peristeras, T. Bouras, E. Kamateri, D. Zeginis, and K. Tarabanis, "Towards a reference architecture for semantically interoperable clouds," in *IEEE Int. Conf. on Cloud Computing Technology and Science (CloudCOM 2010)*, 2010, pp. 143–150.

[11] B. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*, Aug. 2009, pp. 44 –51.

[12] L. Youseff, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing," in *Grid Computing Environments Workshop, 2008. GCE '08*, Nov. 2008, pp. 1 –10.

[13] OpenCrowd, "The OpenCrowd Cloud Taxonomy," http://cloudtaxonomy.opencrowd.com/, [Last Retrieved: July 2012].

[14] Cloud Computing Interoperability Forum, "UCI Cloud OWL Ontology," http://code.google.com/p/unifiedcloud/, [Last Retrieved: July 2012].

[15] F. Moscato, R. Aversa, B. Di Martino, T. Fortis, and V. Munteanu, "An analysis of mOSAIC ontology for Cloud resources annotation," in *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*, Sep. 2011, pp. 973 –980.