A SEMANTIC DISCOVERY FRAMEWORK TO SUPPORT SUPPLY-DEMAND MATCHMAKING IN CLOUD SERVICE MARKETS

Giuseppe Di Modica and Orazio Tomarchio
*Dipartimento di Ingegneria Elettrica, Elettronica e Informatica, Università di Catania,
Viale Andrea Doria 6, 95125 Catania, Italy*

Keywords: Cloud Computing, Pricing Models, Negotiation, Ontology, Semantic Rules.

Abstract: To date a few, big providers dominate the market of Cloud resources. They provide proprietary solutions through inflexible pricing and SLA schemes. On the research side, the community is working to define specifications and standards on several aspects of the cloud technology. When standards will get mature, interoperability among clouds will be a reality. Customers will be no more locked-up to any proprietary technology and new players will have the chance to enter the 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 market 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 that helps customers and providers to characterize their demands/offers, and provide semantic tools performing the matchmaking in such a way to maximize both the provider's profit and the customer's utility.

1 INTRODUCTION

Cloud computing has recently emerged as a new paradigm for delivering utility computing services (Buyya et al., 2008). In cloud environments, infrastructures, platforms and application services are available on-demand and companies are able to access their business services anywhere and whenever they need. 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.

In addition, cloud computing is making the shift from product to service-oriented economy a reality: from a technical point of view, composition of (dynamic) cloud services allows to provide customized complex services to consumers; from an economic perspective, value is just created by the interconnection of various distributed service providers that jointly contribute to an integrated solution that meets individual customers needs. A complex services typically involve the composition of several component services (and eventually computing resources) offered by a multiplicity of providers. These technological and organizational trends drives competition between different service platforms and service (cloud resources) market places.

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 lies in the lack of interoperability of existing cloud technology (Parameswaran and Chaddha, 2009). 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. Automated negotiation mechanisms should be adopted allowing customers to bargain with providers for differentiated levels of quality of service (Badica et al., 2007).

In this paper we discuss about the need of more flexible charging models for cloud resources' usage,

Di Modica G. and Tomarchio O..
A SEMANTIC DISCOVERY FRAMEWORK TO SUPPORT SUPPLY-DEMAND MATCHMAKING IN CLOUD SERVICE MARKETS.
DOI: 10.5220/0003929305330541
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 533-541
ISBN: 978-989-8565-05-1

together with advanced negotiation protocols to 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 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.

The remainder of the paper is organized as follows. In Section 2 the background context is introduced and the issues inspiring this work are discussed. Section 3 describes in details the approach proposed for the definition of a cloud service discovery framework. In particular Section 3.1 discusses the two ontologies for the characterization of cloud service demands/offers, Section 3.2 and 3.3 provides details on the mapping and the matching processes respectively. We conclude our work in section 4.

2 RATIONALE

The commercial success of cloud computing is witnessed by the individual success of few, very big companies that in the past years have made, and are currently making, huge profits by virtualizing and selling their unused computational resources. Amazon has imposed as the main IaaS provider on the market. Microsoft and Google (respectively with the products *"Azure"* and *"App Engine"*) dominate among the PaaS providers. SalesForce.com provides the most famous example of business application delivered through the SaaS model. These players are dominating the market by imposing their own proprietary solutions (e.g., Amazon's ".ami" for the virtual machines' format specification and "EC2" API for the management of virtualized resources), which as a matter of fact have become the de-facto standards, thus fostering what in economics is known as the phenomenon of *vendor lock-in*. It is not easy for the users to put their application on a commercial cloud, and then migrate it to another cloud in the case that they are not satisfied with the provided service performance level.

The idea of putting together unused computational resources to be organized and delivered according to an "on-demand" basis has also spread among open, non commercial contexts. For instance, *volunteer computing* (also called *Global computing* or *Public computing*) uses computers, voluntarily offered by their owners, as a source of computing power and storage to achieve high-performance distributed scientific computing (Anderson and Fedak, 2006). Some (Aversa et al., 2010) have proposed to build a cloud infrastructure upon voluntarily shared computing resources, claiming that the cloud computing paradigm is applicable also at lower scales, from the single contributing user (sharing his desktop) to research groups, university campus, public administrations, social communities that make available their distributed computing resources to the Cloud. For obvious reasons, the *service level* offered by this specific kind of clouds could never compete with that of commercial clouds (contributing peers can voluntarily join/leave anytime); nevertheless the chance to access resources at no charge (or at very low rates) will attract many users. Again, in order for this volunteer-based cloud paradigm to be successful, interoperability among the open contexts must be accomplished.

The road that leads to cloud interoperability is long. In a desirable scenario, the user should be able to build up its own application independently of the specific cloud that it is going to run onto, define the application requirements (both functional and not functional) in a standardized way, search for the cloud provider that best meets his requirements, negotiate for the service, deploy the application, monitor the application performance, moving it to another cloud in the case that the service performance does not meet his expectation.

Several issues need to be addressed in order to pose the basis for fully interoperable scenarios. The research community is focused on several cloud management's aspects: provisioning, metering and billing, privacy, security, identity management, quality of service (QoS), service level agreements (SLA). Some works propose real specifications, and promote their adoption as standards by the cloud community, while others advice best practices that should be adopted when dealing with specific cloud management issues. The Ditributed Management Task Force (DMTF) has developed the Open Virtualization Format (OVF) specification (Distributed Management Task Force, 2010), which is a packaging specification designed to address the portability and deployment of virtual appliances across multiple virtualization platforms. The aim is to define a standard way for packaging together virtual machines, data and applications that should be deployed and run on any cloud provider. Within the Open Grid Forum (OGF), the Open Cloud Computing Interface (OCCI) Working Group is developing practical solution to interface with cloud infrastructures exposed as a service (IaaS). The group has defined the OCCI specification (The Open Grid Forum, 2011), which is an extensible API that allows the deployment, monitoring and management of virtual workloads (like virtual machines). The Storage Networking Industry Association (SNIA) has created the Cloud Storage Technical Work Group for the purpose of developing an architecture related to system implementations of cloud storage technology. The main output of the group's activity is the Cloud Data Management Interface (CDMI) (The Storage Networking Industry Association, 2011), a functional interface that cloud applications can use to create, retrieve, update and delete data elements from the cloud. The National Institute of Standards and Technology (NIST) push towards the adoption of the computing cloud model in industry and government. NIST aim is to foster cloud computing systems and practices that support interoperability, portability, and security requirements that are appropriate and achievable for important usage scenarios. It has identified a set of twenty five use cases (The National Institute of Standards and Technology, 2010) that seek to express portability, interoperability and security concerns that cloud users may have. The Cloud Security Alliance (CSA) was created to promote the use of best practices for providing security assurance within cloud computing. CSA is very much concerned with issues like identity and access management, that are thoroughly discussed in an official document that has been released (The Cloud Security Alliance, 2011). The Open Cloud Consortium (OCC) is an organization supported by several enterprises and academic partners that develops reference implementations, benchmarks and standards for cloud computing. It has developed the MalStone Benchmark (The Open Cloud Consortium, 2011) for large data clouds and is working on a reference model for large data clouds. The Cloud Computing Use Cases group works to define use cases for cloud computing. It is a collaboration of cloud consumers and cloud vendors trying to make steps towards keeping cloud computing open. The group has published a white paper (The Cloud Computing Use Cases group, 2011) that highlights the capabilities and requirements that need to be standardized in a cloud environment to ensure interoperability, ease of integration and portability. The provided use cases demonstrate the performance and economic benefits of cloud computing, and are based on the needs of the widest possible range of consumers. The latest version of the paper focuses on four main topics: how consumers use the cloud; how applications are built in the cloud; security in the cloud; requirements for Service Level Agreements in the cloud.

Independently of the specific aspects addressed by these research efforts, they all share the same objective: **interoperability** among clouds. 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 the cloud-based resources. The European FP7 project Reservoir (Reservoir Consortium, 2011) is one of the first successful attempts to create an interoperable federation of cloud providers, spanning across different administrative domains, IT platforms and geographies, aiming at sharing their individual resources to respond to the users' demand. 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. In the forthcoming interoperable scenarios the competition for acquiring new customers will be shifted to the ground of the overall QoS that cloud providers are able to offer. The challenge for the cloud providers will be answering at best the real needs of their customers. This will yield to a diversification of the offerings in terms of (to cite a few) adopted security models, performance monitoring tools, SLA templates, resource negotiation protocols and pricing models.

In this paper we focus on the last two of the just cited aspects. In particular we address the need to support customers when choosing the "best" cloud offer among the wide range offers that they are presented with. On the other side cloud providers, when offering their resources, should be able to differentiate at best their offer, by presenting the customers with a wide selection of flexible negotiation and pricing models. Such an offering diversification will yield benefits to cloud providers too.

3 CLOUD SERVICE DISCOVERY FRAMEWORK

The future cloud economy will have to take care of the vast set of requirements (functional and non functional) of those applications that potentially might get benefits if "cloudified". In order to best match the demand of these applications, the market of cloud services will have to broaden its range of offerings. On the cloud providers end, a great competitiveness factor will be the capability of realizing such a differen-

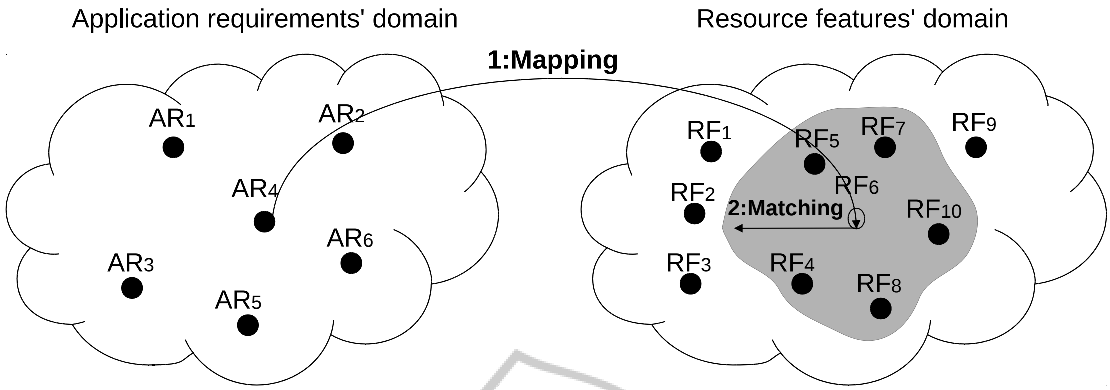

Figure 1: Mapping and matching.

tiation.

One of the key elements on which the competition will be based is the **pricing model** applied to charge the cloud service's usage. 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. Providers also propose their customers an alternative pricing model based on the *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 (densely) used. Other providers (mostly providing services for the business management) adopt a price model that takes into account the number of end users of the service. 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 (some refers to this model with the name ”freemium”).

In the forthcoming cloud economy generation other pricing models might result more attractive for both the providers' and the customers' needs (e.g., customized models, time-oriented models). Another key element on which competition will focus is the **negotiation protocols**' schemes offered to carry on a service transaction. The market of virtualized resources is today dominated by very big commercial providers that impose their charging models, price quotes, resource availability level, data security level for the resources that they put for sale. As for the resource cost aspect, for instance, customers can acquire a resource just at the price that it is advertised at, with no chance for them to negotiate on the price. Amazon has launched the *Spot Instances* model. This model enables the customer to bid for unused Amazon capacity. Depending on the provider's business strategies and on the amount of unused resources, other negotiation models (e.g., auction-based) might be employed. We argue that, similarly to what happens in some goods markets, providers should be given the possibility to tune their selling strategies according to the resources' demand and supply levels. Finally, in the future the cloud offers will strongly differentiate on the **service performance level**. The performance features that cloud providers advertise are usually vague, sometimes focused on virtual machines computation speed, ignoring other aspects such as the storage and network services performance (like, e.g., access time to stored data and network bandwidth).

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, provide them with feedbacks on the performance of the delivered services. Their commercial offers are built on top of the just cited QoS parameters. The competition among the providers is played on both the price at which the resources are leased 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*, 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 scheme.

Let us now change perspective, and imagine to look at things with the customer's eye. Customers would like to have the possibility to characterize and specify application requirements under their business perspective, i.e., in a form that is aligned to the objectives being pursued by the their own business policies. According to the customer's business objectives to be accomplished, and to how much mission-critical the application is, the application to be deployed may require either a guaranteed service level or a best effort. 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 intensively used within a timeframe), reserved-based solutions are to be preferred; otherwise, the on-demand pricing model may result more convenient. Finally, the **budget availability** is the most compelling among the requirements. All the choices must be made checking that the budget they require is compatible with the customer's investment capability. For example, a service level might fit a given application's profile, but might not be affordable for the customer; on the contrary, a more affordable service level would make the customer save money, but might not fit the strict application's requirements. In most cases a compromise has to be searched for.

The provider and the customer perspectives are quite different. The former seeks to maximize the profit and the utilization level of the IT asset that they have invested in. The latter just needs to make fine-tuned searches on 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 ontologies have been developed to characterize respectively the provider and the customer perspectives. In particular, an ontology semantically describes the features of the resources being offered by cloud providers, the other one describes the application's business requirements demanded by customers. 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 depicts the two semantic domains, along with the mapping and matchmaking processes. In the figure, the 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 an overview of the two ontologies, respectively describing the application requirements and the resource features domains, and some details on how the mapping and matchmaking processes work.

3.1 Ontology Design

We have designed two simple and extensible **OWL-based** (W3C, 2009) ontologies that may be of help to semantically characterize the resources offered by cloud providers and the requirements expressed by customers. This is a very first attempt to define the base for a more comprehensive and complete ontology framework, but we believe it is sufficient to give the reader a rough picture of the idea being proposed in this work.

3.1.1 Resource Features' Ontology

In its current state the resource features' ontology focuses on the concepts that, according to us, best represent the cloud resources from the providers' business perspective, i.e., the pricing model, the service performance level and the negotiation model.

The figure 2 depicts a comprehensive view of the developed ontology. Ellipses represent ontology classes. A solid arrow models an *is-a* relationship between two classes, while arrows with dashed lines are *object properties* defined among classes. Basically, a cloud **Offer** is the main concept of the ontology. An offer regards the provisioning of a cloud **Service**, that can be in the form of IaaS, PaaS of SaaS (Infrastructure, Platfom, Software - as a Service). Among the features that an offer can exhibit, we have depicted the **Pricing Model** that will be used to charge the

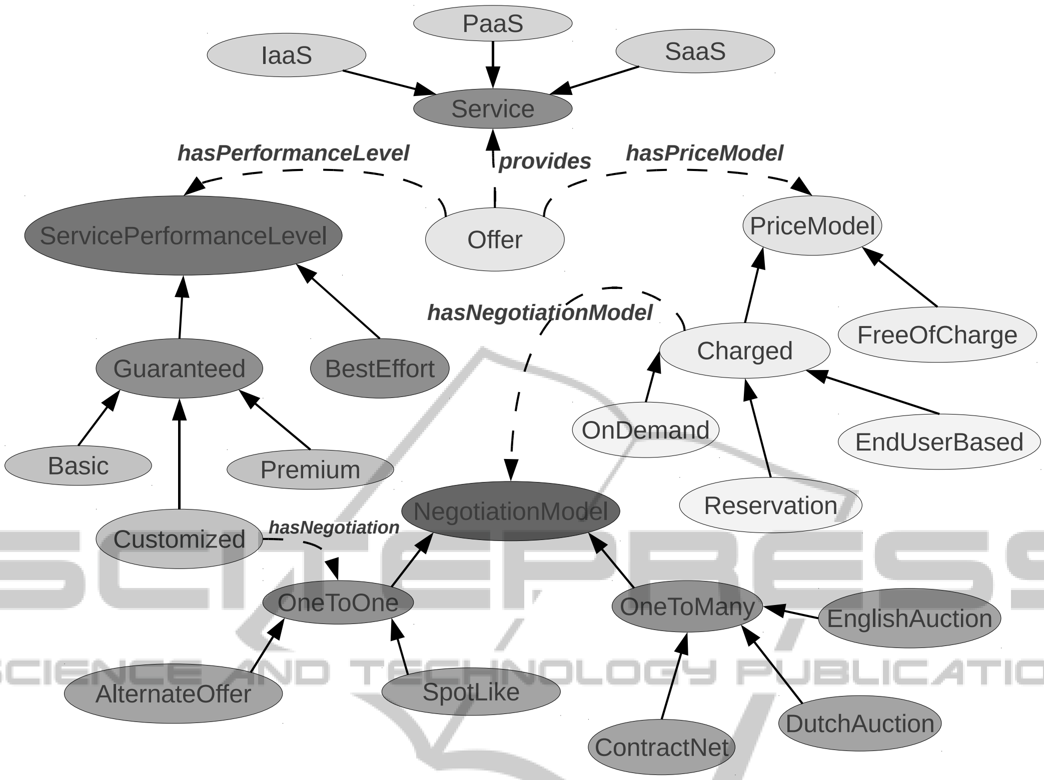

Figure 2: Resource features ontology.

customer, and the **Service Performance Level** associated to the provisioning of the service that the offer refers to.

The Pricing Model is partitioned into two disjointed concepts(classes): Free of Charge and Charged. Free of Charge represents the category of services offered for free (see "try-before-buy" strategies). Charged embraces all pricing models that charge the customer for the service usage. In this tentative ontology, Charged is furtherer subclassed by the On-Demand, Reservation and End-user-Based concepts. The three sub-concepts represents the pricing categories currently adopted by commercial cloud providers, as earlier discussed at the beginning of the section. As depicted in the figure, the Charged class is linked to the **NegotiationModel** class by means of the hasNegotiationModel property. This link models the negotiability of the price of any non-free offer. The service provider is allowed to select the negotiation model that accommodate at best its vending strategy's objective. In particular, there are two big categories of negotiation models: the OneToOne, which models private negotiation models between a customer and a provider, and the OneToMany, representing all the public negotiation schemes in which many customers compete to acquire the resource offered by one provider.

As regards the Service Performance Level class, it is subclassed by the Best Effort and Guaranteed classes. The former represents the category of services for whose performance no guarantee is provided; the latter represents the services for which a basic guarantee is offered, those for which a premium level guarantee is provided, and those allowing for a customization of the performance level to be delivered. The object property, linking the Customized class to the OneToOne class, models the possibility for the provider to select the preferred negotiation scheme to be used when contracting with the customer for the QoS to be delivered.

3.1.2 Application Requirements' Ontology

In the Figure 3 the ontology defining the application requirements is depicted. A dashed arrow originating from a concept (ellipse) and pointing to a box represents a concept's *data property*, whose range of values is indicated within the box itself. Basically, a customer can request **Computing** resources, a **Platform** or a **Software** application. The Price branch models the fact that, for a request, the customer can specify if he is willing to pay or he is just looking for a free-of-charge service. In the first case, the customer may

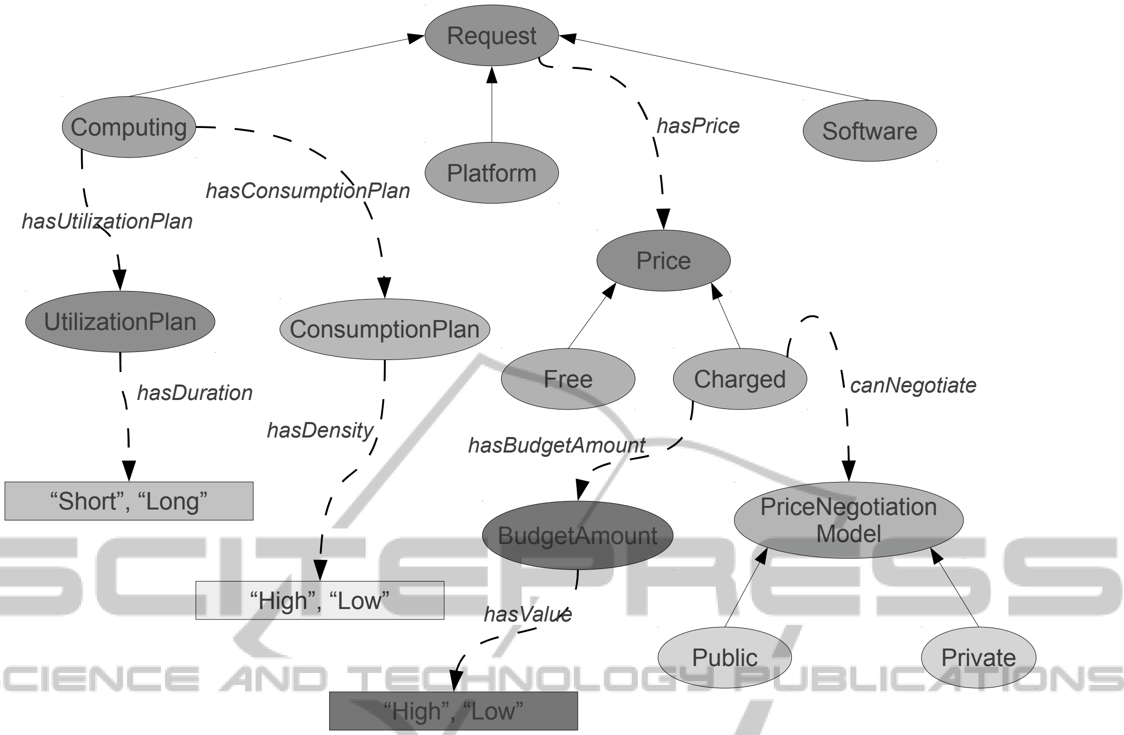

Figure 3: Application requirements ontology.

want to also specify the amount of the budget at his disposal. At this stage, two qualitative values are allowed ("High" and "Low") but a more detailed specification for the budget requirement will be introduced in the future. Also, the customer may specify if he is looking for services that are either privately or publicly negotiable.

In the case of Computing resources, the customer is given the possibility to specify whether he has a plan for utilizing the resource (UtilizationPlan concept), and a timeline for that plan (hasDuration data property). A resource consumption plan, indicating how "dense" will be the consumption of the resource for the duration of the plan, may also be specified. These parameters aim to model what in section we earlier referred to as application profile.

3.2 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) (Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., and Dean, M., 2004). The objective of each rule is to transform a specific application requirement into the ideal, best matching resource feature. As mentioned before, the mapped feature does not have to necessarily correspond to a real feature of a concrete offer. The task of searching for the real feature that best matches is committed to the matchmaking process, of which we provide details in the next section.

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 incrementally builds up the ideal offer. For the sake of brevity, we report only a subset of these 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. $hasMatchedOffer(?request, ?offer)$
   $\wedge request : Software(?request)$
   $\wedge offer : provides(?offer, ?service)$
   $\rightarrow offer : SaaS(?service)$
5. $hasMatchedOffer(?request, ?offer)$
   $\wedge request : hasPrice(?request, ?price)$
   $\wedge request : canNegotiate(?price, ?negotiation)$

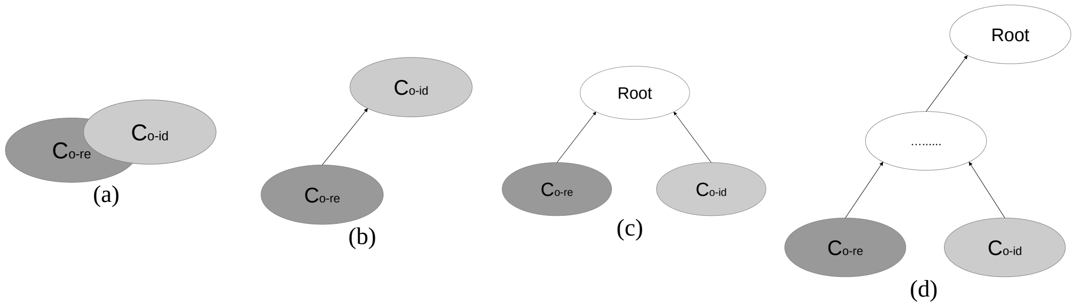

Figure 4: Concepts comparison.

$\wedge request : Public(?negotiation)$
$\wedge offer : hasPriceModel(?offer, ?priceMod)$
$\wedge offer : NegotiationModel(?negModel)$
$\rightarrow offer : hasNegotiationModel(?priceMod, ?negModel)$
$\wedge offer : OneToMany(?negModel)$

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. Rule 5 maps a request for a generic cloud service, for whose price the customer is willing to negotiate in a public auction.

3.3 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 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 offer. 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. This will allow us to present the customer with a list of concrete offers (sorted by the affinity criteria) that will ease the task of selecting the best offer. 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 four components which, respectively, represent 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 that can be employed to privilege the price criteria in the evaluation of the overall affinity.

We now provide some details on the semantic comparison of concepts. Let $O_i(Serv_{oi}, Price_{oi}, Perf_{oi}, Neg_{oi})$ be a generic offer, where $Serv_{oi}$, $Price_{oi}$, $Perf_{oi}$,Neg_{oi} represent the semantic concepts characterizing the offer. In order to evaluate the overall semantic affinity of two offers O_{id} (the ideal offer that is the outcome of the mapping process) and O_{re} (a real offer in the market place), couples of homologous concepts (i.e., referring to the same feature) must be compared. So, $Serv_{o-id}$ will be compared to $Serv_{o-re}$, $Price_{o-id}$ will be compared to $Price_{o-re}$, and so on. In general, the concept C_{o-id} must be compared to its homologous C_{o-re}. Let us now focus on that branch of the ontology tree to which the generic concept/feature C refers to.

In the figure 4 we have depicted the four different cases that are relevant to our evaluation. Let us analyze all the possible cases:

- the two concepts are semantically equivalent (figure 4(a)): they are assigned an affinity of 1;
- C_{o-id} is the father of C_{o-re} (figure 4(b): their affinity will be 1 as well;
- the two concepts are siblings and the father is the root concept in the considered branch (figure 4(c)): their affinity will be 0.5;
- the two concepts are siblings and the father is a non-root concept in the considered branch (figure 4(d)): their affinity will be 0.75;
- in any other case, the concepts' affinity will be 0.5.

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., OnDemand vs Reservation, EnglishAuction vs DutchAuction), therefore they are given a higher grade of affinity.

4 CONCLUSIONS

Cloud computing has reached a good state of maturity and is a very well accepted technology. Commercial providers are trying to take advantage of the business opportunities that this paradigm has brought. In the future, when interoperability among clouds will be fully achieved, there will be the need to rethink the way by which the provision of virtualized resources have to meet the applications’ demand. The market of cloud resources will have to provide novel and advanced matchmaking processes that account for the providers’ and the customers’ dynamic and heterogeneous business requirements, respectively in terms of profit and utility.

The work presented in this paper is a first attempt to defining a cloud service discovery framework. In particular, two ontologies have been developed to characterize respectively the cloud resource’s features and the application requirements, and a set of semantic rules has been defined to let customers’ requests be mapped onto the cloud offers’ domain. Finally, a matchmaking procedure has been devised to semantically search the offers’ domain and provide the customer with a list of most profitable offers. A prototype of the framework has just been implemented. In the future we are planning to run intensive tests in order to evaluate the viability of the proposed model.

ACKNOWLEDGEMENTS

The work described in this paper has been partially supported by the MIUR-PRIN 2008 project *“Cloud@Home: a New Enhanced Computing Paradigm”*.

REFERENCES

Anderson, D. P. and Fedak, G. (2006). The computational and storage potential of volunteer computing. In *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*, CCGRID ’06, pages 73–80, Washington, DC, USA. IEEE Computer Society.

Aversa, R., Avvenuti, M., Cuomo, A., Di Martino, B., Di Modica, G., Distefano, S., Puliafito, A., Rak, M., Tomarchio, O., Vecchio, A., Venticinque, S., and Villano, U. (2010). The Cloud@Home Project: Towards a New Enhanced Computing Paradigm. In *Workshop on Cloud Computing: Project and Initiatives (in conjunction with EUROPAR2010)*, Ischia (Italy). Springer Verlag.

Badica, C., Ganzha, M., and Paprzycki, M. (2007). Implementing rule-based automated price negotiation in an agent system. *Journal of Universal Computer Science*, 13(2):244–266.

Buyya, R., Yeo, C. S., and Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *10th IEEE International Conference on High Performance Computing and Communications (HPCC ’08)*, pages 5 –13.

Distributed Management Task Force (2010). The Open Virtualization Format. http://www.dmtf.org/standards/ovf/.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., and Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. http://www.w3.org/Submission/SWRL/.

Parameswaran, A. and Chaddha, A. (2009). Cloud Interoperability and Standardization. *SETLabs Briefings*, 7(7):19–26.

Reservoir Consortium (2011). The Reservoir Project. http://www.reservoir-fp7.eu/.

The Cloud Computing Use Cases group (2011). Moving To The Cloud. http://cloudusecases.org/.

The Cloud Security Alliance (2011). Security Guidance for Critical Areas of Focus in Cloud Computing. https://cloudsecurityalliance.org/.

The National Institute of Standards and Technology (2010). Effectively and Securely Using the Cloud Computing Paradigm. http://csrc.nist.gov/groups/SNS/cloud-computing/.

The Open Cloud Consortium (2011). The Open Cloud Testbed. http://opencloudconsortium.org/.

The Open Grid Forum (2011). The Open Cloud Computing Interface. http://occi-wg.org/.

The Storage Networking Industry Association (2011). The Cloud Data Management Interface. http://www.snia.org/tech_activities/ standards/curr_standards/cdmi/.

W3C (2009). OWL 2 Web Ontology Language. W3C Recommendation.