

RESOURCE AND SERVICE DISCOVERY IN SOAs: A P2P ORIENTED SEMANTIC APPROACH

GIUSEPPE DI MODICA, ORAZIO TOMARCHIO, LORENZO VITA

Department of Electrical Engineering, Electronics and Computer Science
University of Catania, V.le Andrea Doria 6, 95125 Catania, Italy

e-mail: {Giuseppe.DiModica, Orazio.Tomarchio, Lorenzo.Vita}@dieei.unict.it

An intense standardization process is favouring the convergence of grids and Service Oriented Architectures (SOAs). One of the benefits of such technological convergence is that grid resources and applications can be virtualized by services and offered through the SOA paradigm. In the broad and interoperable scenarios enabled by the SOA, involving the participation of several grid infrastructures across many administrative domains, service discovery can be a serious issue. In this paper we present a P2P-based infrastructure that leverages semantic technologies to support a scalable and accurate service discovery process. The key concept of the presented idea is the creation of an overlay network organized in several semantic groups of peers, each specialized in answering queries pertaining to specific applicative domains. Groups are formed by clustering together peers offering services that are semantically related. The architecture details of the proposed solution are presented. A system prototype has also been implemented and validated through a case study deployed on the PlanetLab testbed.

Keywords: SOA, web service discovery, semantic, P2P, JXTA.

1. Introduction

Today, key technologies in distributed systems are the service oriented architecture (SOA), grid and cloud computing, all of which are seeing significant investment in standardization and increasingly rapid adoption by organizations of all types and sizes. In recent years it has become clear that there is a considerable overlap between the goals of grid computing and the benefits of an SOA. The rapid advances in web service technology and standards (WSDL, BPEL, WS-RF) have contributed to moving the architecture of current grids to the standardized, service-oriented, enterprise-class grid of the future (Papazoglou and van den Heuvel, 2007; Papazoglou *et al.*, 2008).

In such an SOA-enabled grid environment, services are used both to virtualize resources and to provide other grid functions and applications. The more resources and applications get gridified, the faster the number of virtual services grows. Such a service layer demands scale-free and performing discovery mechanisms. Current Web Service (WS) discovery standards (OASIS UDDI Specification Technical Committee, 2005) are based on centralized repositories of services descriptions. Interoperability is provided just at the syntactic level, thus limiting the discovery process to essentially keyword-based search,

which may lead to inaccurate and/or incomplete search results. Furthermore, as the number of services grows, current WS discovery standards suffer from typical problems that affect centralized approaches, i.e., scalability and availability. To overcome these, many are calling the P2P paradigm into question (Sahin *et al.*, 2005; Banaei-Kashani *et al.*, 2004). The idea behind the P2P approach is to get rid of the central repository of information in favour of a distributed infrastructure over which WS descriptions are spread and managed according to some well-known schemes.

Several works have tried to combine the P2P paradigm and semantic technologies (Sahin *et al.*, 2005; Banaei-Kashani *et al.*, 2004; Mandreoli *et al.*, 2007; Li *et al.*, 2007a). On the one hand, the purpose is to achieve more accuracy in search results through the application of semantic-based algorithms to the query match. On the other hand, the distributed nature of the infrastructure provides a high degree of scalability and availability. Among the possible P2P systems, the *hybrid* approach seems to be the one that best fits the described objectives.

The main idea of our approach is to cluster nodes providing services that are semantically related. When a query is issued, it will be forwarded to the cluster (or the

group) which is more related to it from a semantic point of view, thus maximizing the probability to find a service that matches the query. The adopted overlay network is based on a hybrid P2P architecture, where leaf peers are clustered into groups managed by some super-peers: the former delegate to the latter the burden of resource searching and query routing throughout the overlay network to reduce the traffic overhead.

The remainder of the paper is organized as follows. In Section 2 related works are presented along with the main requirements for the design of a P2P framework for semantic service discovery. Section 3 discusses the proposed architecture and provides technical details of overlay management procedures. Section 4 describes a simple case study deployed on the PlanetLab testbed (PlanetLab Consortium, 2003) and discusses the results of some tests that have been conducted. Finally, we conclude the work in Section 5.

2. Rationale and related work

Automated service discovery means finding a service that matches a given set of requirements from a repository of service descriptions.

Most of earlier works in the literature are based on the syntactic description of the service (WSDL) and rely on keyword matching techniques to search for services. DHTs have been proved to be useful for keyword-based searches but are not suitable for semantic-based searches. Basically, they fail in catching the real meaning of the query, so often the domain of candidate services is not thoroughly explored. Furthermore, sometimes the discovered services do not fully meet the original search criteria, so additional time has to be spent to further filter the discovered set of services.

Some works (Sahin *et al.*, 2005; Banaei-Kashani *et al.*, 2004) have already demonstrated that the combination of P2P architectures and semantic technologies could provide the right solution to the above described problems. The use of P2P systems would make it possible to ensure decentralization, scalability and fault tolerance, while the publication of semantically described services would improve and automate the discovery process. If there is a chance to semantically annotate services, the relationships between the peers and their offered services can be better characterized, and the service discovery process can greatly benefit from this. The semantic overlay approach seems to be one of the most attractive ones for the P2P research community. Basically, such an approach prescribes that the entire network is partitioned according to the similarity of services that the peers offer. Whenever a query is posed, it is forwarded to the partition(s) of peers that are semantically “close” to the query itself, i.e., to peers that publish services which are semantically similar to the content of the query.

Several efforts have tried to combine the P2P paradigm with the Semantic Web technologies. Hypercube (Schlosser *et al.*, 2002) is an example of the usage of ontologies to optimize the position of the peers in the network topology. SPiDer (Sahin *et al.*, 2005) employs ontologies in a DHT-based P2P infrastructure. WSPDS (Banaei-Kashani *et al.*, 2004) makes use of the WSDL-S language (which has then evolved to SAWSDL) to semantically describe services in a semantic overlay network.

In the work by Mandreoli *et al.* (2007) a hybrid approach is proposed. A P2P infrastructure is employed to distribute the semantic knowledge (expressed in OWL-S) of what specific services are available in the system, while the knowledge of what objectives (goals) may be satisfied within the system is kept centralized. Each peer is a representative of an administrative domain and is responsible for advertising services belonging to that specific domain. A centralized/decentralized approach is also presented by Li *et al.* (2007a). Here the authors employ a P2P infrastructure to distribute service semantic descriptions but rely on a centralized entity to carry out the service discovery process. Another work (Vu *et al.*, 2005) builds on decentralized registries to distribute the semantic descriptions of web services. Services, whose capabilities are described in WSMO, are grouped according to the semantic concepts that they represent: service partitioning is demonstrated to bring benefits to the process of service discovery.

The combination of P2P systems and semantic technologies has been interpreted by authors in different ways. What differs in the various interpretations is the concept of *semantic similarity* or *affinity*. In the work of Cohen *et al.* (2007), similarity is interpreted as the interest of a peer towards a particular domain. Other works (Chirita *et al.*, 2007; Bhogal *et al.*, 2007; Song *et al.*, 2007) adopt the query expansion approach, in which semantic reasoning is locally performed by the peer that originates the query.

Li *et al.* (2007b) propose a dynamic semantic overlay network to manage service advertisements. Services are semantically characterized, and the indexing mechanism takes into consideration the services’ semantic features. The authors propose to code the semantic information into numeric values, and then apply a DHT-based routing of advertisements and queries. In our paper we propose a different approach to evaluate the similarity of services and peers, which also take-into account the occurrences of semantic concepts. Furthermore, we adopt group-oriented overlay management, which we believe to be most appropriate to deal with semantic services that are logically clustered in semantic domains.

Our system differs from the others in that it is capable of dealing with any of the main semantic languages being used for semantic annotation of web services (OWL-S (OWL-S Technical Committee, 2004), WSMO (WSMO

Technical Committee, 2005), SA-WSDL (Farrell J. and Lausen H., 2007)). The semantic concept representation adopted within our framework is indeed not tied to any specific language used for the semantic annotation, thus enabling better service integration.

Based on the analysis of the works cited so far, we propose some steps that should be basically taken in the design of a P2P framework for the discovery of Semantic Web services (Bisignano *et al.*, 2009):

1. Definition of the peers' knowledge, built from the semantic description of the provided services.
2. Creation and maintenance of a semantic overlay network, where peers are dynamically grouped according to their semantic affinity, i.e., to the semantic characterization of the web services they offer. What most of the works do, instead, is organizing clusters according to the profiles/interests of the peers. Furthermore, they just provide a static overlay structure.
3. Design of efficient and semantic-based routing mechanisms. In the majority of the works, semantic reasoning is performed just by the peer that poses or receives the query, while query delivering procedures are still keyword based.

The system we propose tries to cater the above requirements.

3. Service discovery infrastructure

The service discovery infrastructure that is proposed in this paper is built on top of JXTA (JXTA Community Project, 2007). JXTA provides a two-layer peer infrastructure, in which super-peers (rendez-vous peers) populating the upper layer connect each other to form a distributedly managed index of the information pushed onto the network by individual peers, which in turn form the lower layer. The obvious benefit brought by this approach is that the signalling generated by the service discovery mechanism to reply to a query is bound to the super-peers overlay and does not affect at all the lower layer connections. JXTA also provides support for peer group creation and management, allowing peers to group together according to specific interests. The combination of the just described features ensures rational dissemination of the information and grants for a robust and scalable service discovery mechanism.

We then rely on the described P2P functionality offered by JXTA to build up an infrastructure for publication and discovery of semantically described web services. Peers here are responsible for just the publication (advertisement) of services. Service implementation and the related deployment strategies are out of the scope of this work.

3.1. Semantic overlay management. The grounding idea is to exploit the semantic information carried by the web services published by peers to create semantically characterized groups. The aim is to leverage on the semantic "knowledge" of each peer (made up of the semantic information embedded in the web services to be published) to derive the position the peer will have to occupy in the P2P overlay. Peers are grouped together in the network space according to their characterization in the semantic space. A peer may also belong to several groups, as the services it publishes can span different semantic domains.

Within the overlay each peer is therefore characterized by the semantics of the services it offers. Similarly, the peers' semantic characterization contributes to determining the semantic characterization of the group(s) they belong to. In this paper we will refer to semantic characterization of services, peers and groups with the term *expertise*.

As for the discovery phase, service queries are also semantically characterized (i.e., queries exhibit semantic expertises as well). When a query is posed, it is semantically processed within the group where it originates from. If none of the group's peers is able to reply (i.e., none of the peers' expertises match that of the query), it is routed to the semantic group(s) that will most likely be able to successfully match the query. Once there, the query is delivered to all the peers belonging to the group(s). The proposed select-and-flood routing mechanism avoids flooding the entire network whenever a query is issued.

The overall management of the P2P infrastructure relies on semantics. Semantic procedures are carried out to determine

- the birth of a new group,
- the join/left of a peer to/from a group,
- the routes of queries.

At the heart of such procedures is the *Semantic Affinity (SA)* function that is used to evaluate the degree of similarity between entities, i.e., web services, peers, groups and queries. The SA function maps the expertises of pairs of entities into real values bounded in the interval $[0,1]$. The closer it is to 1, the more semantically similar the two entities are. Since a perfect similarity is not always possible, for the purposes of overlay management we define two entities to be "similar" when their similarity degree is greater than a given *Similarity Threshold (ST)*. Thus, for instance, a peer is allowed to join a group only if the similarity degree evaluated for the peer and the group overcomes the *Membership Threshold (MT)*. A semantic query is routed to a group only if their degree of similarity overcomes the *Query Membership Threshold (QMT)*.

Figure 1 depicts how peers would organize in semantically characterized groups. Each cloud ideally represents

the semantic domain that the contained peers are experts in. Clouds can also overlap: this happens when a peer is responsible for the publication of services that span across several semantic domains (peer P_{13} in the figure).

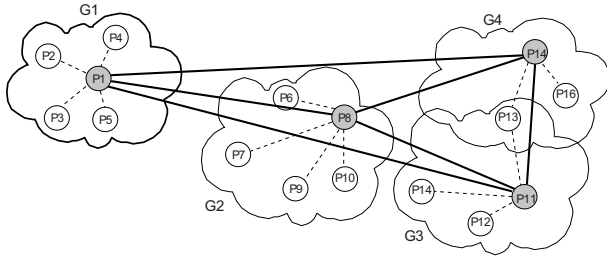


Fig. 1. Semantic groups.

Groups are created whenever new arriving peers cannot be assigned, according to their own expertise, to any of the already existing groups. For each group, a rendezvous peer (*RDV*) is appointed as being responsible for group management. In order for a peer to become an *RDV*, no special requirements are needed: the first-coming peer that creates a new group is appointed an *RDV* for that group. Should the appointed *RDV* fail (i.e., stop working as the group manager), an *RDV* re-election occurs: in that case, the peers detecting the *RDV*'s failure issue a re-election call, which will eventually end up with the appointment of a new *RDV*. The re-election mechanism allows more than one peer to compete for the *RDV* role, but again the first peer detecting the *RDV*'s failure will be appointed the new *RDV*. *RDVs* manage peers' join and leave operations, and group information maintenance and sharing. They are responsible for both intra-group and inter-group communications. *RDVs* can adopt different communication strategies. Any information that has to be delivered from outside the group to inside the group is broadcast to all group peers. A selective mechanism is instead preferred when information has to be communicated outside the group. Further details on the calculation of the *SA*, the join and leave procedures and the query management can be found in the work of Bisignano *et al.* (2009).

3.2. Peer architecture. Figure 2 depicts the peer's architecture and details the components devoted to the management of the semantic overlay.

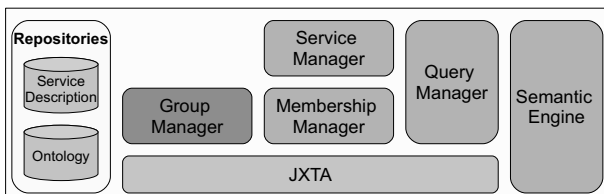


Fig. 2. System architecture.

The *JXTA* layer provides the basic functionality to manage the hybrid P2P network, on which the rest of the components rely. For the purpose of our work, the architecture makes use of *JXTA* API to create/join/leave groups and create/publish/discovery resource advertisements. Components that directly interact with the *JXTA* framework are the *Group Manager*, the *Membership Manager* and the *Query Manager*. User applications can access the architecture's API offered respectively through the *Query Manager* and the *Service Manager*. The *Semantic Engine* offers tools, accessible by all architecture components, to process the semantic information. The *Repositories* persistently store the semantic data useful for the peer to take part in the distributed process of semantic overlay management.

3.3. Service discovery. A Semantic Query (*SQ*) must be routed to groups that are close to the semantic characterization of the query itself. A query (*SQ*) is semantically worth being answered by a given group (*G*) if $SA(SQ, G) > QMT$, where *QMT* (*query membership threshold*) identifies the minimum degree of affinity that a group and a query must have in order for the group to be worth answering that query.

Therefore, in the semantic context that we have depicted the aim is to route an *SQ* to the group(s) that are most relevant for the query itself. Those groups will likely provide satisfactory results in terms of response precision.

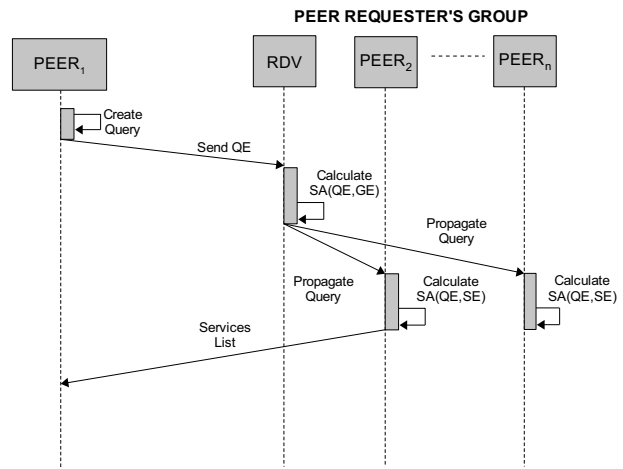


Fig. 3. Query propagation in the group.

In detail, the peer sends the query to the *RDV* of the peer's group. The *RDV* calculates $SA(QE, GE)$. Depending on this value, the query can be either propagated to the peers belonging to the group itself or routed to other *RDVs* in the network as described in the following. If $SA(SQ, G) > QMT$, the query is broadcast to the group's peers through the *JXTA propagateInGroup* com-

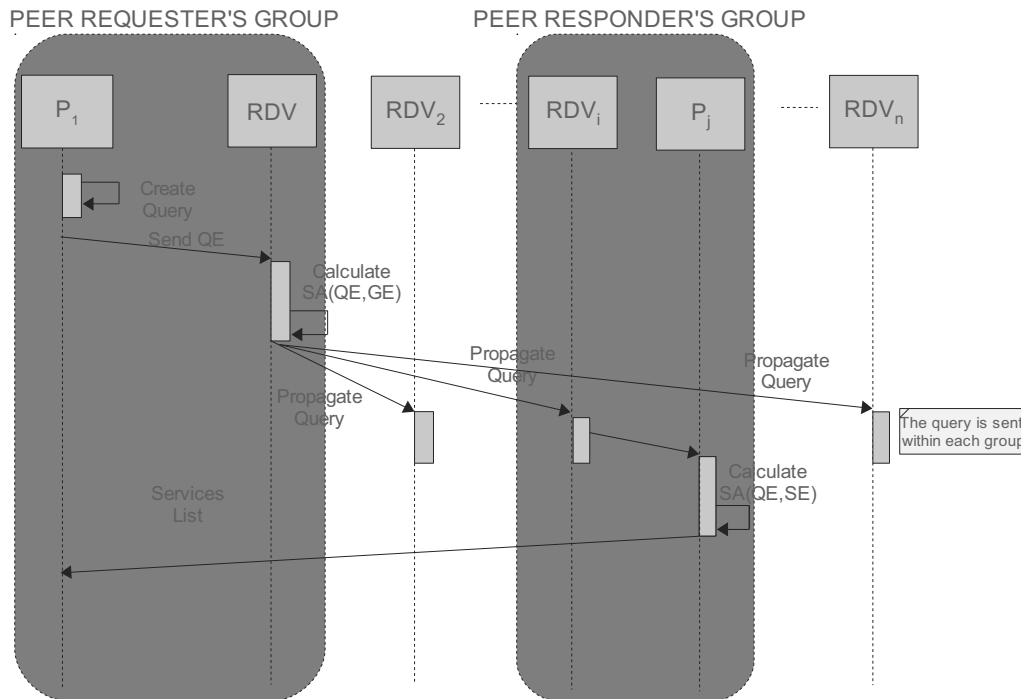


Fig. 4. Forwarding a query to other groups.

munication pattern. The peers receiving the query search for the service(s) that best match the query. The matches are then returned directly to the requesting peer.

In Fig. 3 a sequence diagram of the query process for this use case is depicted. P_1 sends a query to the *RDV* of the group that it belongs to: $SA(SQ, G)$ is greater than the *QMT*, so the query is propagated to all the peers belonging to the group (P_2, \dots, P_n). The peers evaluate the semantic affinity between the query and the service expertise of the services they offer. P_2 finds a service matching the query, so it forwards the service description to P_1 . As for the match between a semantic query and a Semantic Web service, the affinity will be evaluated again by invoking the *SA* function respectively on the expertise of the query (*QE*) and the expertise of the service (*SE*).

If $SA(SQ, G) < QMT$, the *RDV* forwards the query to other groups, trying to identify the ones whose Group Expertise (*GE*) best matches the query. For this reason the *RDV* makes use of the JXTA *walk* communication pattern to forward the *SQ* to other *RDVs*. Each *RDV* receiving the query, will in turn broadcast it to the peers of its group. The requesting peer will eventually receive the response directly from the peers whose services best match the query.

In Fig. 4 a sequence diagram of the query process for this use case is depicted. The peer P_1 sends a query to the *RDV* of the group that it belongs to: $SA(SQ, G)$ is lower than the *QMT*, so the query is propagated to other *RDVs* (RDV_2, \dots, RDV_n). Each *RDV* receiving the

query will broadcast it to all the peers belonging to its group: each peer evaluates the semantic affinity between the query and the service expertise of the services it offers. In the figure, the peer P_j , belonging to the group G_i , finds a service matching the query, so it forwards the service description to the requesting peer P_1 .

4. Case study

A prototype of the framework has been implemented in order to test the viability of the proposed solution. The framework relies on JXTA v2.5 functionality to implement the overlay. JXTA offers mechanisms and tools to easily advertise and search for resources in a P2P fashion. The proposed framework makes use of just these basic JXTA tools to build and maintain a semantic overlay.

Specifically, the object of the investigation are both the peers' grouping and the query routing mechanisms. As for the grouping, the objective was to test the proposed mechanism by comparing the automatically generated grouping against an ideal (made-by-human) grouping. As for the query routing, we evaluated classic performance parameters such as the overall hit-rate and the average response time. A further objective of the tests is to assess the amount of time spent by peers on performing semantic operations, which are fundamental for the calculus of the peers' affinity and the queries' match. Semantic operations are known to be memory greedy and CPU consuming; the tests that we have run also aimed at assessing the quantitative impact of such operations on the peers'

computational resources.

Experiments were conducted on Planet-Lab (PlanetLab Consortium, 2003), which is a real P2P testbed on the Internet. Peers' services were chosen from a wide collection of publicly available Semantic Web services (BBN Technologies, 2010). The tests were run with 100 peers and a collection of more than 300 Semantic Web services covering 20 different interest domains. Table 1 shows a list of some of the web services employed in the tests, grouped by the domain of interest.

Table 1. Semantic Web services.

Interest domain	Service name
Book	AuthorBookpriceService
	BookAuthorpriceNovelService
	BookAuthorbookTypeService

	BookCheapestpriceService
Videomedia	TitleObtainableVideomediaService
	TitleVhsService
	TitleVhsdvdService

	TitleVideomediaRecommendedPriceService
Transport	BicycleAutoPriceService
	AutoPriceService
	AutoCycleMaxpriceService

	CheapCarPriceService
.....
Restaurant	CoffeeRecommendedpriceService
	MaxpriceWhiskeycolabeerService
	PreparedFoodPriceService

	BenService

In Table 2 we report a sample of the distribution of the web services among the peers responsible for their publication. This is the first configuration setting on which the first test was conducted. In our experiment several tests were run. The rest of the tests were conducted on different configuration settings, obtained just by randomly re-distributing web services among the peers.

We then tested the framework's grouping mechanism. The experiment was conducted in the following way. At a given moment in time, all peers activate in the network and collaborate to build the overlay. Initially, there is a brief transient phase during which peers exchange their expertise and search for the semantic group to join. In this phase, peers evaluate their own semantic characterization and discover other peers with which to form consisting groups. Peers periodically evaluate their position in the overlay, thus it may happen that new groups are created and peers change their position in the overlay. The higher the number of peers, the longer this phase.

Table 2. Peers' services.

Peer ID	Service name
Peer01	ComedyfilmActionfilmService
	ComedyfilmFantacyfilmService
Peer02	FilmvideomediaDiscoveryChannel
	TitleActionfilmService
	TitleComedyfilmBFService
Peer10	TitleVhsService
	TitleVhsDvdService
	TitleVideomediaMMService
	TitleVideomediaRecommendedpriceService
.....
Peer32	BookCheapestpriceService
	BookPersonCreditaccountBeaService
	BookPersonCreditcardaccountBShopService
Peer33	BookPriceService
	BookPriceReviewBookService
	BookPriceSizeBookService
.....
Peer61	activityTownService
	adventureUrbanareaService
Peer63	sightseeingTownService
	surfingFarmlandService
	surfinghikingDestinationPFservice
.....

When the transient phase is over, the actual grouping is observed. A snapshot of the observed groups and the relative peers is depicted in Fig. 5. If we were to perform a manual grouping, one could notice that the grouping of Fig. 5 almost matches with the one obtained from the intuitive manually made grouping. Again, we outline that the grouping mechanism allows peers to simultaneously belong to more than one group (e.g., *peer02* and *peer04* in the figure).

The query routing mechanism was tested once the overlay had become stable. Two different sets of queries were tested: one requesting services of categories covered by the actual overlay (for which positive replies are expected), and one requesting services for which none of the groups has expertise at all (query misses are then expected). As for the first set, it was observed that queries were correctly routed to the groups "semantically" entitled to answer: once delivered to the group, hits or misses may occur, depending on whether the given, specific, requested service is owned whether by any of the peers in the group. The second set of queries, as expected, were not delivered to any of the existing groups; therefore a hundred percent of misses was observed. Furthermore, for both the sets, no orphan query (i.e., indefinitely travelling the network and receiving no answer, be it positive or negative) was observed.

In Table 3 we report values of the overall time spent by the semantic overlay to process join requests from

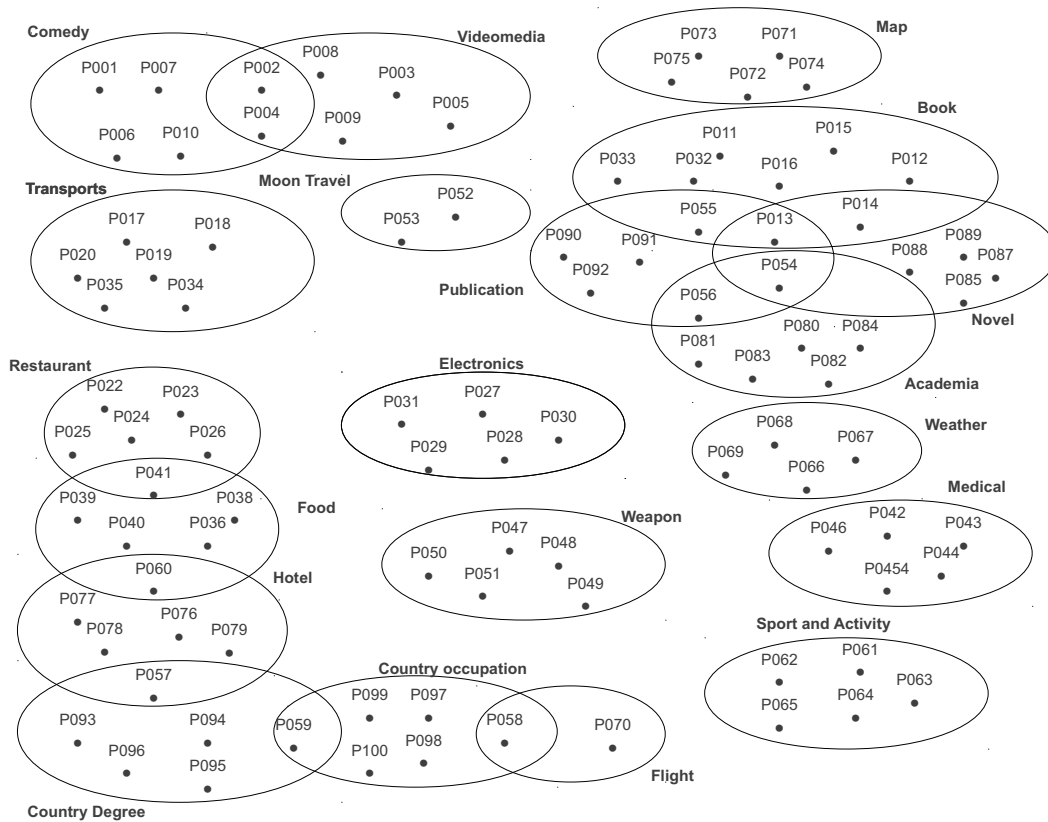


Fig. 5. Grouping observed after the boot phase.

peers. We recall that a request can be routed to many groups (and, thus, can undergo as many semantic affinity evaluations) before it gets correctly answered.

Table 3. Time to satisfy a join request.

MAX (ms)	30500
MIN (ms)	8047
AVERAGE (ms)	12000

On average, it takes 12 seconds for a join request to be answered. The more semantically “distant” is the peer from the group that evaluates its join request, the longer it takes for the peer to get an answer. Almost the same values, of course, were observed for queries. Indeed, in our framework the same mechanism is called upon to satisfy either a join request or a query.

For the test purpose, the semantic affinity of two service expertises (each containing 6–7 semantic concepts) was evaluated. To validate the process, several tests were run: for each test, two new expertises are compared (being the number of concepts per expertise constant throughout the tests). Given two expertises, the same operation (calculus of the affinity) was performed consecutively twenty

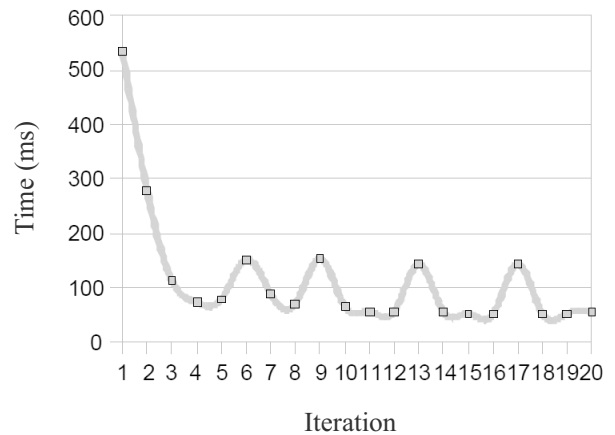


Fig. 6. Semantic affinity execution time.

times. As can be observed in Fig. 6, the very first time the execution time is a bit more than 500 ms; afterwards it establishes around 100 ms. The result for the first execution is due to the need to load in memory the semantic concepts of the expertises involved, which then are cached for subsequent computation.

Finally, tests were conducted in order to evaluate the

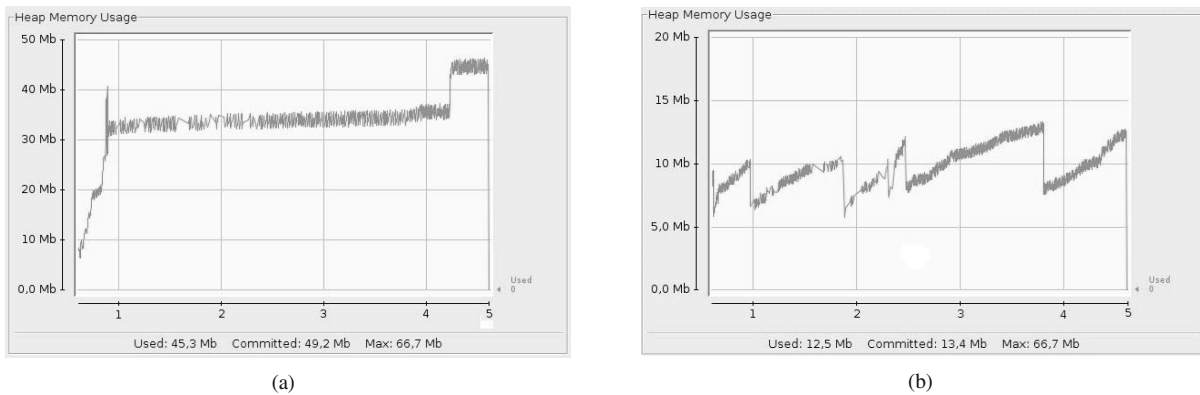


Fig. 7. Memory utilization for *Rendez-Vous* (a) and *Edge* (b).

consumption of peer's resources (namely, memory and CPU) due to the run of semantic operations. Semantic operations are the base of the calculus of affinity, which is called upon by both the join procedure and the query match one. Semantic operations make use of the API provided by the Pellet semantic reasoner (Sirin *et al.*, 2007). Resource utilization was evaluated for peers running under each of the two possible configurations: *Edge* (*Ed*) and *Rendez-Vous* (*Rdv*). Given the amount of semantic operations that the *Rdv* peer must perform, we naturally expect it to be more stressed, from a computational point of view, than the peer running as *Ed*. Basically, the memory load is due to the elaboration of ontologies performed by the Pellet semantic engine. As can be observed in Fig. 7(a), a *Rdv*'s memory load is higher during request processing (join or query). The majority of the load is due to the elaboration of the expertises. As shown in Fig. 7(b), an *Ed* consumes less memory than *Rdv* does. Indeed, it is never directly involved in the routing of queries or join requests; its task is just to answer queries that are directly forwarded to the group(s) it belongs to. Figure 8 shows the change in memory utilization of a peer that switches from *Ed* to *Rdv*.

Memory consumption can be a severe issue for the performance of the peers and thus, for the overall performance of the framework. Semantic evaluations are very memory-greedy procedures and are requested to run very often. In the future, we are aiming at improving the overlay maintenance by caching the most popular semantic concepts.

The last objective of the test was to monitor the average CPU load for both *Rdv* and *Ed* configurations. In Figs. 9(a) and (b) we report CPU utilization, respectively of *Rdv* and *Ed*, versus time. As one can notice, there is no significant difference between the *Ed* and the *Rdv* case. The CPU is stressed in a bursty style, i.e., just when requests must be served; therefore we can conclude that CPU consumption is not an issue for the overall performance of peers.



Fig. 8. Increase of memory utilization because of the switch from *Ed* to *Rdv*.

5. Conclusion and future work

Semantic technologies have attracted the interest of IT researchers and industry players over the past ten years. In the SOA research field these technologies have been widely adopted to devise scalable and efficient solutions to the service discovery issue. This paper proposed an architectural model for the discovery of services based on a hybrid P2P approach. The adopted strategy was to partition the P2P network into several semantic groups of peers, each specialized in answering queries pertaining to specific applicative domains.

What characterizes this approach is the clusterization of nodes providing services that are semantically related. Nodes providing "similar" services are logically grouped in such a way that the group can expose a strong expertise in a given semantic domain. When a semantic query is issued, the underlying mechanism will forward the query to the cluster (or group) which is more related to it from a semantic point of view, thus maximizing the probability to find a service that matches the query.

Furthermore, the described semantic approach differs from others in that a virtual, language-independent, semantic infrastructure layer is used as the base for P2P

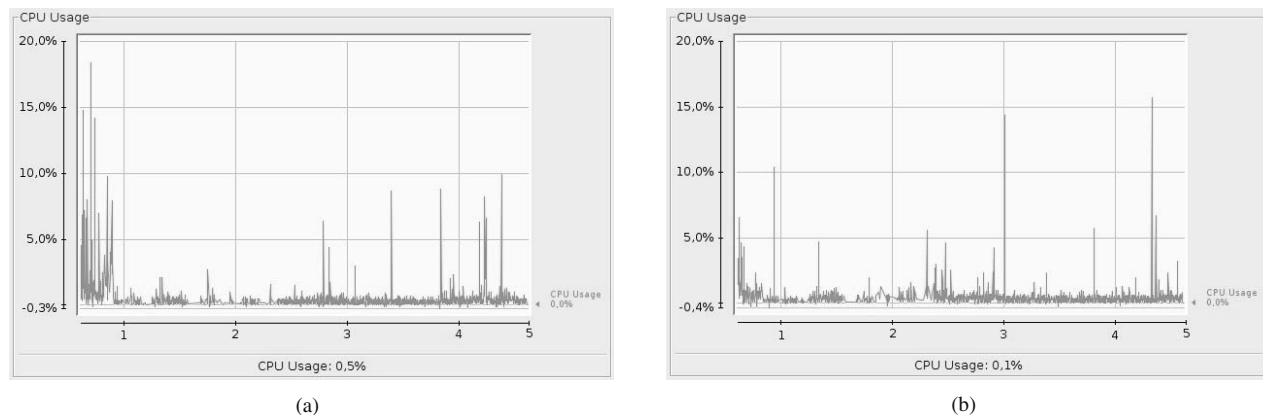


Fig. 9. CPU utilization for *Rendez-vous* (a) and *Edge* (b).

overlay management. This approach enables the integration of services described in any of the main semantic annotation languages. A prototype of the system was implemented and a simple case study (conducted on the Planet-Lab testbed) presented to show the viability of the proposed approach. In the future, we are planning to improve the group partitioning mechanism in order to grant more scalability to the overall approach. In particular, the uncontrolled growth of groups will be limited by splitting groups that count too many members. Furthermore, to avoid extreme fragmentation, micro-groups (made of a few members) will be merged to semantically similar groups.

References

- Banaei-Kashani, F., Chen, C.-C. and Shahabi, C. (2004). WSPDS: Web Services Peer-to-peer Discovery Service, *Proceedings of the International Conference on Internet Computing, Las Vegas, NV, USA*, pp. 733–743.
- BBN Technologies (2010). SemWebCentral, Open Source tools for the Semantic Web, <http://www.semwebcentral.org/>.
- Bhagal, J., Macfarlane, A. and Smith, P. (2007). A review of ontology based query expansion, *Information Processing and Management* **43**(4): 866–886.
- Bisignano, M., Di Modica, G. and Tomarchio, O. (2009). Jaxson: A semantic P2P overlay network for web service discovery, *3rd International Workshop on Web Service Composition and Adaptation (WSCA-2009), ICWS-2009, Los Angeles CA, USA*, pp. 438–445.
- Chirita, P.A., Firan, C.S. and Nejdil, W. (2007). Personalized query expansion for the web, *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands*, pp. 7–14.
- Cohen, E., Fiat, A. and Kaplan, H. (2007). Associative search in peer to peer networks: Harnessing latent semantics, *Computer Networks: The International Journal of Computer and Telecommunications Networking* **51**(8): 1861–1881.
- Farrell J. and Lausen H. (2007). SAWSDL: Semantic annotations for WSDL and XML schema, <http://www.w3.org/TR/sawSDL/>.
- JXTA Community Project (2007). The JXTA project, <https://jxta.kenai.com/>.
- Li, D., Xu, R. and Wang, H. (2007a). A Semantic Web based peer to peer service discovery mechanism for intelligent business process, *IEEE International Conference on E-Business Engineering (ICEBE2007), Hong Kong, China*, pp. 730–737.
- Li, Y., Yang, F., Shuang, K. and Su, S. (2007b). A semantic peer-to-peer overlay for web services discovery, *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM '07), Harrachov, Czech Republic*, pp. 750–760.
- Mandreoli, F., Perdichizzi, A.M. and Penzo, W. (2007). A P2P-based architecture for Semantic Web service automatic composition, *International Workshop on Database and Expert Systems Applications (DEXA2007), Regensburg, Germany*, pp. 429–433.
- OASIS UDDI Specification Technical Committee (2005). UDDI: Universal Description, Discovery, and Integration, <http://www.oasis-open.org/committees/uddi-spec>.
- OWL-S Technical Committee (2004). OWL-S: Semantic markup for Web services, <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>.
- Papazoglou, M.P., Traverso, P., Dustdar, S. and Leymann, F. (2008). Service-oriented computing: A research roadmap, *International Journal of Cooperative Information Systems* **17**(2): 223–255.
- Papazoglou, M.P. and van den Heuvel, W.-J. (2007). Service oriented architectures: Approaches, technologies and research issues, *The VLDB Journal* **16**(3): 389–415.
- PlanetLab Consortium (2003). PlanetLab: An open platform for developing, deploying and accessing planetary-scale services, <http://www.planet-lab.org/>.

- Sahin, O.D., Gerede, C.E., Agrawal, D., Abbadi, A.E., Ibarra, O.H. and Su, J. (2005). Spider: P2P-based web service discovery, *Proceedings of the 3rd International Conference Service-Oriented Computing (ICSOC 2005), Amsterdam, The Netherlands*, pp. 157–169.
- Schlosser, M., Sintek, M., Decker, S. and Nejdil, W. (2002). A scalable and ontology-based P2P infrastructure for Semantic Web services, *Proceedings of the 2nd International Conference on Peer-to-Peer Computing (P2P 2002), Linköping, Sweden*, p. 104.
- Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A. and Katz, Y. (2007). Pellet: A practical OWL-DL reasoner, *Web Semantics: Science, Services and Agents on the World Wide Web* 5(2): 51–53.
- Song, M., Song, I.-Y., Hu, X. and Allen, R.B. (2007). Integration of association rules and ontologies for semantic query expansion, *Data & Knowledge Engineering* 63(1): 63–75.
- Vu, L.-H., Hauswirth, M. and Aberer, K. (2005). Towards P2P-based Semantic Web service discovery with QoS support, in C. Bussler and A. Haller (Eds.) *Business Process Management Workshops: BPM 2005 International Workshops*, Lecture Notes in Computer Science, Vol. 3812, pp. 18–31.
- WSMO Technical Committee (2005). WSMO: Web service modeling ontology, <http://www.w3.org/Submission/WSMO/>.

Giuseppe Di Modica received his degree in computer engineering in 2000 from the Engineering Faculty of the University of Catania (Italy). Since then he has been engaged in research on distributed systems with the Department of Computer and Telecommunications Engineering of Catania University. In 2005 he received a Ph.D. in computer science and telecommunications engineering from the University of Catania. His research interests include mobile agents, ad-hoc networks, wireless networking, mobile P2P computing, grid and service oriented architectures. He is currently a research assistant at the Engineering Faculty of the University of Catania.

Orazio Tomarchio received his degree in computer engineering from the Engineering Faculty of the University of Catania (Italy) in 1995. Since then he has been engaged in research on distributed systems with the Department of Computer and Telecommunications Engineering of Catania University. In 1999 he received Ph.D. in computer science from the University of Palermo. The scientific activity of Dr. Tomarchio has been focused on studying distributed systems, particularly with regard to programming and management techniques. His main research interests include mobile agents, network and QoS management techniques, mobile P2P computing, middleware for MANETs, grid and service oriented architectures. He is currently an assistant professor at the Engineering Faculty of the University of Catania.

Lorenzo Vita received the electrical engineering degree from the University of Catania in 1979. Since 1980 he has been cooperating in many research fields with the Institute of Computer Science and Telecommunications of Catania University, where he was an associate professor. His initial scientific activity was focused on parallel processing, distributed and parallel systems, multimedia systems and network management. Recently, his research interests also include mobile agents, grid and service oriented architectures. He is currently a full professor of software engineering at the University of Catania.

Received: 7 September 2010

Revised: 9 December 2010

Re-revised: 25 January 2011