Michele Amoretti, Rosa Bertolazzi, D. Prandi, Francesco Zanichelli, Gianni Conte (2006):
Semantic Discovery of Services in Peer-to-Peer Based Inter-Firm Grids.
In Bernhard Nett, Marleen Huysman, Markus Rohde, Charles Steinfield, Volker Wulf (Eds.),
International Reports on Socio-Informatics (IRSI),
The Role of ICT in Interfirm Networks and Regional Clusters, Workshop Documentation COOP '06 Conference
(Vol. 3, Iss. 2, pp. 82-94)

Semantic Discovery of Services in Peer-to-Peer Based Inter-Firm Grids

M. Amoretti, R. Bertolazzi, D. Prandi, F. Zanichelli, G. Conte
Distributed Systems Group
Dipartimento di Ingegneria dell'Informazione
University of Parma, Italy
{amoretti, bertolazzi, prandi, zanichelli, conte}@ce.unipr.it

1 INTRODUCTION

The concept of inter-firm network, *i.e.* a group of related organizations that partner and/or cooperate with each other in order to provide expanded products and services, in our opinion fits nicely with the Grid computing concept of coordinated resource sharing in dynamic, multi-institutional virtual organizations [14]. Effective, ICT-enabled cooperation among enterprises and other institutions calls for seamless interoperability and strong security at each layer involved, from the lower network-level protocols up to the application-level data-model. Virtualization and standardization are of paramount importance to overcome the differences in models, processes and tools adopted by each enterprise, for example regarding the key issue of identity management. By leveraging upon service orientation, federated security and dynamic provisioning of resources, modern Grid middleware has the potential to enable scalable, flexible, yet controlled information and service sharing across s enterprise boundaries.

Unfortunately, this Grid vision has only been partially realized by currently available Grid architectures. To fill the gap between aspiration and practice in Grid computing, the application of Semantic Web technologies both on and in the Grid has been advocated [7]. Semantic Web gives to the Grid a standard ontology language for information interchange, while the Grid provides the Semantic Web with an extensive and flexible middleware platform for heterogeneous resource/service integration. Semantic Grid architectures should enable automatic location, selection, employment, composition, and monitoring of services. To establish a framework within which computer-interpretable service descriptions are made and shared, W3C is developing OWL-S [16], which is an OWL-based Web Service ontology supplying a core set of markup language constructs for describing the properties and capabilities of Web services in unambiguous form.

In this context, our research and development efforts focus on making it easier to design, configure, deploy and maintain semantically enriched service-oriented applications. The peculiarity of our approach lies into a flexible peer-to-peer overlay network architecture, which improves the performance of semantics based service discovery while maintaining a high degree of node autonomy and fault tolerance. Recently, there has been an increasing interest for a possible convergence between the Grid and Peer-to-Peer (P2P) technologies [13], to the purpose of progressing beyond traditional client/server approaches and centralized service provisioning, and of developing robust and highly decentralized, knowledge-based Grid systems.

The application of the peer-to-peer paradigm to the problem of service sharing and discovery could allow for greater flexibility and robustness with respect to what it can be achieved with industrybacked centralized registries such as the Universal Description, Discovery, and Integration (UDDI) project [22]. Notwithstanding their ample technical features, and the outstanding commercial weight of their maintainers, their centralized nature exhibits evident shortcomings in terms of rigidity, faulttolerance and performance. Our Peer pattern [4] has been defined to cope effectively with the needs of Virtual Organizations (VOs), *i.e.* network-enabled, transitory communities made of individuals and institutions, which are mostly related to information and service sharing/discovery, resource scarcity as well as security and trust. In a Peer-based distributed system, *i.e.* peer-to-peer network, all nodes have the same structure (regardless of the type and quantity of local resources) and are not only potential users but also potential resource providers.

We applied the Peer pattern to realize the *Service-oriented Peer-to-Peer Architecture (SP2A)* [5], a lightweight framework for the development of service-oriented peers for efficient and robust Grid environments. In writing the functional and technical specification of SP2A, we considered the Service-Oriented Architecture Reference Model (SOARM) proposed by OASIS [21]. In particular we respected the principle that a service is a set of functionalities provided by one entity for the use of others, and it is invoked through a software interface but with no constraints on how the functionality is implemented by the providing entity.

A service is opaque in that its implementation is hidden from the service consumer except for (1) the *data model* exposed through the published service interface, and (2) any information included as *metadata* to describe aspects of the service which are needed by service consumers to determine whether a given service is appropriate for the consumer’s needs. Consistent with the axiom of opacity, a service consumer cannot see anything behind the service interface and does not know if one service is actually consuming and aggregating other services.

SP2A allows to share services which implement the mechanisms for managing the resources provided by each peer (*e.g.* computational power, storage, contents, sensors). Currently, these services are implemented as Web Services, and expose both WSDL and OWL-S interfaces. Remote services can be searched using different strategies, which are illustrated in this paper. The most significant is SDDM, *i.e.* semantic discovery with distributed matching. As we demonstrate with experimental results, the matching process of semantic profiles is time-consuming. For this reason, it is unreasonable to use a single server to collect all service profiles and perform matching analysis with requested profiles. Distributing service knowledge and computational workload among peers is the key concept of the SDDM solution provided by SP2A.

Logistics is a natural application for the service sharing model proposed by SP2A. Logistics is the geographical repositioning of raw materials, work in process, and finished inventories where required at the lowest cost possible, through the integration of information, transportation, inventory, warehousing, material handling, and packaging. In this context, we participate to a regional project named STIL ("Strumenti Telematici per l'Interoperabilità delle reti di imprese: Logistica digitale integrata per l'Emilia-Romagna", *i.e.* telematic tools for inter-firm networks interoperbility: digital logistics for the Emilia-Romagna region).

The paper is organized as follows. Section 2 illustrates two strategies for resource sharing based on the semantic approach. Section 3 describes the implementation of the SP2A framework as a Java API. Many details are given about two basic modules, *i.e.* the Semantic Extractor and the Semantic Matcher, along with their performance evaluation. Section 4 discusses related work in the Semantic Grid research field. Section 5 illustrates our background and the STIL project, to which we contribute with SP2A. Finally, an outline of open issues concludes the paper.

2 DEPLOYMENT AND DISCOVERY OF RESOURCE PROVISION SERVICES

The SP2A framework allows users to share their resources, *i.e.* publishing local *Resource Provision Services (RPSs)*, as well as locating and interacting with remote ones. SP2A design is based on the Peer pattern [4], which defines a set of modules and their interactions. The most important module is the *Router*, which defines the rules for addressing, filtering, sending, and receiving messages. Most of the peer operations, such as publication, search and delivery of resources, rely on the Router. Resource sharing in SP2A can be summarized by the following three phases.

1. When a resource is published, the description of its Resource Provision Service, including information such as security requirements, is made available to the Router. On resource discovery, (possibly complex) user queries are delivered to the Router. Messages for other peers are the third kind of input a Router can receive.
2. Based on the received input, the Router creates new messages and computes their destination.

   On resource discovery, queries can be either syntactically or semantically evaluated. This process can affect the computation of next destination peer.
3. The Router sends query-related messages to the computed destination.

   Consumers are allowed to check if a discovered RPS is available, if it performs a certain function or a set of functions, if it operates under specified constraints, and if it can be invoked through a specified means, including inputs that the service requires and outputs that will form the response to the invocation. Capturing service functionalities is a difficult goal to achieve. This aspect needs to be expressed in a way that is generally understandable by service consumers, but able to accomodate a vocabulary that is sufficiently expressive for the domain for which the service provides its functionalities. This may include, among other possibilities, machine-processable interfaces. All RPSs expose an interface which specifies how to access its functionalities. This information should be represented in one or more standard, referenceable formats. The service interface prescribes what information needs to be provided to the service in order to exercise its functionalities and/or the results of the service invocation to be returned to the service requester [21]. In particular, SP2A allows to define RPS semantics, which are the shared expectations associated with the resource. RPS consumers perceive the semantics as broken in three main parts: the data model, the process model, and the behaviour. The latter is the intended real world effect of using a RPS. Semantic description of RPSs allows for fine-grain tuning of the discovery process, which should be efficient, fault tolerant, scalable and autonomic.

2.1 Semantic Discovery of RPSs

The Router allows the user to choose among the following strategies for searching remote RPSs:

- <attribute, value> discovery with distributed matching (AVDDM)
- semantic discovery with local matching (SDLM)
- semantic discovery with distributed matching (SDDM)

In the first solution, AVDDM, the keyword set provided by the user, possibly filtered (using Boolean analysis) by a *Syntactic Extractor*, is propagated in the network. Any other involved peer activates its *Syntactic Matcher* to find local RPSs which syntactically match with the received keyword set. In the second solution, SDLM, the *Semantic Extractor* module builds a small ontology from the textual query issued by the user. For each class of the ontology produced by the Semantic Extractor, the Router sends an <attribute, value> syntactic query, and the *Semantic Matcher* compares discovered RPSs with a locally produced semantic Profile.

In the third solution, SDDM, the Router propagates the semantic Profile of the required RPS. Any other involved peer activates its Semantic Matcher to find local RPSs which semantically match with the received Profile.

Figure 1 and 2 illustrate, respectively, the SDLM and SDDM strategy in the same unstructured supernode network scenario.

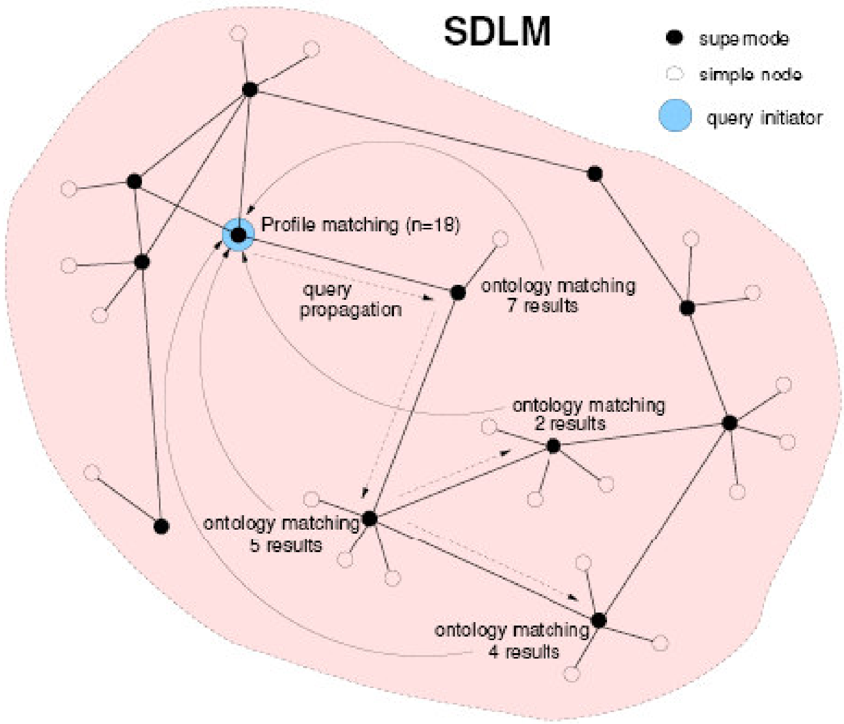

Figure 1 – Semantic discovery with the SDLM strategy.

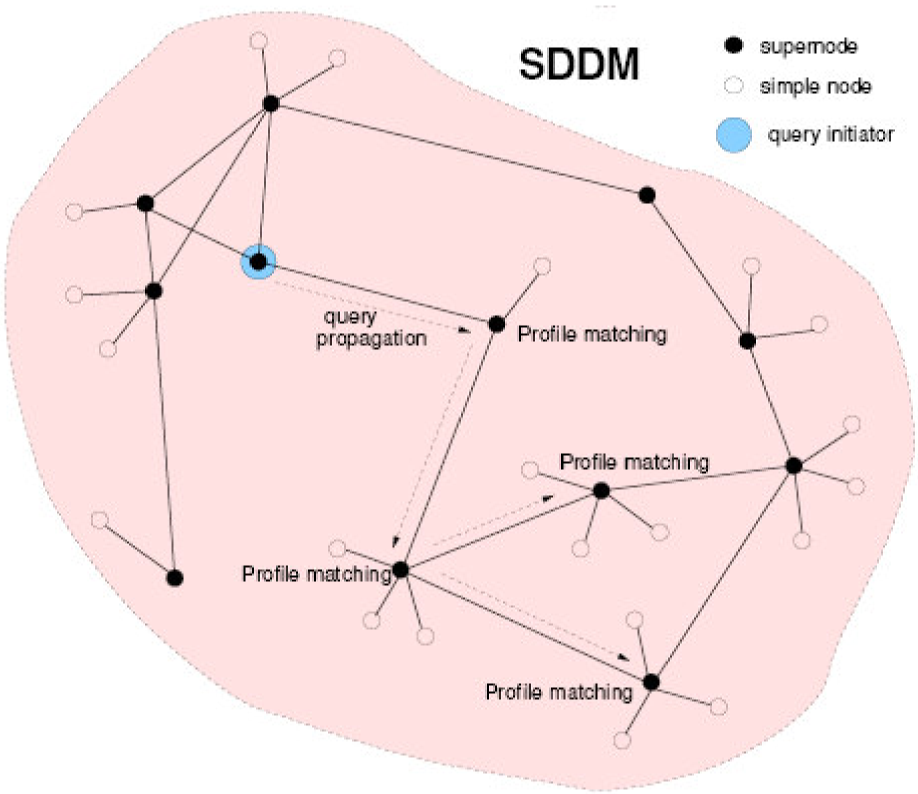

Figure 2 – Semantic discovery with the SDDM strategy.

To compare the duration of a search process based on the two strategies, we define $t_o(n_w)$ as the time for extracting the ontology from a n_w-words query, $t_{om}(n)$ as the time for syntactically matching *n* RPSs with the required ontology, and $t_{pm}(n)$ as the time for semantically matching *n* RPSs with the required Profile.

For simplicity we suppose that, for a given network topology, the path followed by a query is the same for both SDLM and SDDM strategies. Thus, propagation time can be neglected because it is the same for both approaches, and we can define semantic discovery time as:

$$t_{SDLM} = t_o(n_w) + t^d_{om}(n_{tot}) + t_{pm}(n_{om})$$
$$t_{SDDM} = t^d_{pm}(n_{tot})$$

The *d* stands for „distributed“, as ontology matching in SDLM and Profile matching in SDDM are concurrently performed among involved supernodes. Conversely, Profile matching in SDLM is performed by the query initiator peer, after the „pre-filtering“ process realized by the distributed ontology matching, which reduces the number of service to compare with the Profile. Depending on the behaviour of $t_{om}(n)$ and $t_{pm}(n)$, SDDM is more or less convenient than SDLM.

3 SYSTEM IMPLEMENTATION

The SP2A framework has been implemented as an open source Java middleware, illustrated in figure 2, which includes four components providing interfaces and classes for managing the main aspects of the service-oriented peers: state, group, security, and rps (the latter is the acronym of *resource provision service).*

Peer-to-peer connectivity is based on JXTA [23], *i.e.* Sun's open initiative. Each RPS is implemented as a Web Service, and exposes both WSDL and OWL-S interfaces which are collected in a JXTA advertisement.

JXTA-SOAP [8], an official JXTA-related project we are currently managing, is the core of our SP2A system. JXTA-SOAP allows to share Web Services in JXTA peer-to-peer networks. Recently, we provided JXTA-SOAP of Web Service Security (WSS) support. We are now planning to extend the package to support the Web Service Resource Framework (WSRF), to make JXTA-SOAP comparable with current state of the art middleware for Grid architectures based on Web Services, *i.e.* Globus Toolkit 4 [1], with the added value of native peer-to-peer support. Moreover, SP2A adopts several third-party packages for text analysis, ontology extraction and semantic matching of service profiles. In the following we illustrate how these packages have been used to implement the modules which allow for semantic discovery of resources in SP2A.

3.1 Semantic Extractor

The Semantic Extractor (SE) module generates an ontology from a textual descriptions of the desired services. The SE is made of three sub-modules: the Information Extractor, the Word Disambiguator, and the Ontology Creator.

The *Information Extractor* uses GATE [18] to produce a *Corpus* which can be analyzed by a language-specific *Tokenizer*, exploiting a *Gazetteer* to perform syntactic text analysis and to extract nouns, adverbs, conjunctions, etc.

Syntactical and structural information produced by the Information Extractor are then filtered by the *Word Disambiguator*, which is an implementation of the algorithm proposed by Navigli and Velardi in [19]. The Word Disambiguator uses WordNet [10] and JUNG [2] to create *semantic graphs*, which represent words (as nodes) and relations between words (as weighted arcs). For each given ambiguous word a semantic graph is built, connecting the word to its hypernyms, meronyms, holonyms and hyponyms. To disambiguate a word, the algorithm starts from the corresponding graph and tries to connect it to non-ambiguous word graphs. If an ambiguous word leads to many non-ambiguous words, the algorithm chooses the path with minor weight. Conversely, if an ambiguous word cannot be related to any non-ambiguous word, the algorithm chooses the meaning which is more frequently associated to the word (WordNet classifies word meanings based on their usage frequency). Weights can be assigned using several policies. We chose to assign the same weight for all kinds of relations, and to increase weights at each hop (starting from the root of the graph with *w=1*). The *Ontology Creator* extracts ontologies from semantic graphs built by the Word Disambiguator, based on an algorithm which considers hypernymy ("is a kind of") relations. To avoid excessive depth of the resulting class tree, the Ontology Creator searches for common superclasses, and for class relations. In details, for each word, the hypernym sequence is compared with those of other words, searching for the common terminal string with the greatest number of hypernyms; the chosen superclass is the the first hypernym of the common terminal string. Moreover, for each discovered superclass, the hypernyms sequence is compared with those of other classes; further superclasses are chosen with the same procedure in previous step, paying attention to recursiveness. The process can be computationally expensive, depending on the total number of words n_w in the initial text provided by the user to the Semantic Extractor.

3.2 Semantic Matcher

As described in the previous section, semantic matching can be local to the query originator, once it has received some responses, or distributed across the supernode network. In the latter case, JXTA ResolverService uses the Semantic Matcher (SM) instead of the default syntactic matcher. The SM looks for OWL-S descriptions among discovered service advertisements, comparing them with an OWL-S Profile generated upon user inputs. The SM returns a result list, ordered according to the *Similarity Degree* [16] of each discovered service with the reference profile. The user can specify one or more constraints while creating the semantic description of the desired service, such as taxonomy classification (e.g. UNSPSC), input or output types, preconditions or a global matching degree. A OWL-S Profile is created, which can be incomplete if the user does not define all service parameters. The more the OWL-S description is detailed, the more the SM is selective, at the risk of too many false negatives. Conversely, an incomplete OWL-S description can generate many unacceptable results (false positives).

To measure the usefulness of a service the SM compares the so-called *IOPE parameters* (*i.e.* input, output, precondition, and effect) among wanted and discovered OWL-S Profiles. The result of matching operation is a complex object called *matching degree*, which also considers the conformity of discovered services to the desired *Service Category*.

Profile parameters values refers to an OWL ontology set, thus they can have the following matching degrees:

1. *Equivalent(P1,P2)*, if parameter values are equivalent, *i.e.* they define the same concept; the equivalence between two concepts is not limited to "is the same as" relation, because equivalent classes can have different names;
2. *Subsume(P1,P2)*, if one parameter value generalizes the other;

3. *Fail(P1,P2)*, if parameter values are not related.

3.3 Performance Evaluation

With reference to the strategies defined in the previous section, we illustrate the results of the experiments we performed to measure the performance of Semantic Extractor and Matcher modules.

We premise that, in current SP2A implementation, the time for syntactically matching *n* distributed RPSs with the required ontology is very short, thus we can assume $t_{SDLM} = t_o(n_w) + t_{pm}(n_{om})$, where $t_o(n_w)$ is the time which is necessary to the Semantic Extractor for extracting the ontology from a n_w - words user query, and $t_{pm}(n)$ is the time in which the Semantic Matcher implementation matches *n* RPSs with a required Profile.

The semantic richness of the ontology produced by the Word Disambiguator depends on n_w. Generally speaking, the more the number of words forming the user query, the more the richness of the ontology. On the other side, n_w affects the duration of the process, as illustrated in the figure 3.

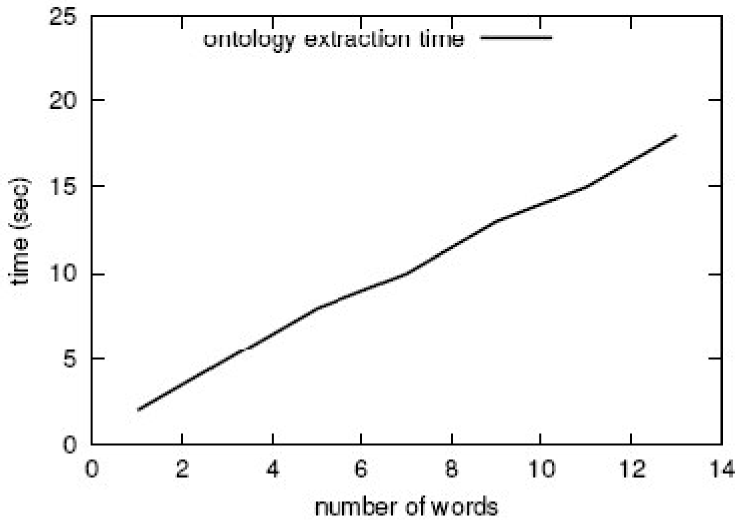

Figure 3 – Time behaviour of the ontology extraction process.

We observe that t_o increases linearly with the number of words n_w (a typical value for n_w is *5*). The Profile matching process returns an ordered list of services, through a semantic similarity measure. The duration of this process depends on the number *n* of service to order, and on the complexity of the Profile. We considered three cases: Profiles with only one parameter (the Service Category), Profiles with three parameters (Input-Output), and Profiles with complete service description (IOPE full). Figure 4 illustrates that $t_{pm}(n)$ has a linear shape, whose angular coefficient increases with the Profile complexity.

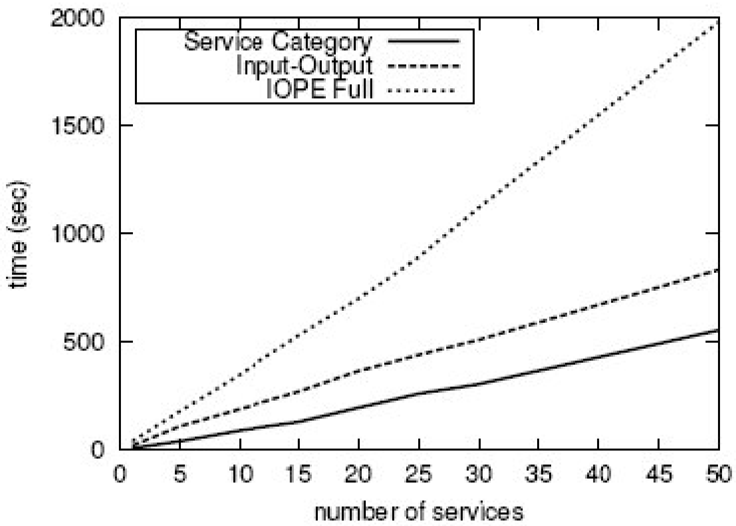

Figure 4 – Time behaviour of the profile matching process.

These results suggest that semantic Profile matching should be distributed in the overlay network, to avoid heavy workload on the query initiator peer which is already charged by disambiguation tasks. Thus the SDDM strategy should be preferred to the SDLM, in particular for supernode networks, where leaf nodes in general lack of computational power and storage. To this purpose, we are studying a possible refactoring of the JXTA routing package, to make it easy to support different routing strategies, and in particular the SDDM-based strategy.

4 RELATED WORK

After the introductory work of De Roure et al. [7], several steps have been made in the Semantic Grid research field. The Global Grid Forum has established the Semantic Grid Research Group (SEMGRID) [15] to track Semantic Web community activities and advise the Grid community on the application of Semantic Web technologies in Grid applications and infrastructure, to identify case studies and share good practice.

One of the most important topic is the choice of the language for semantic annotation of Grid services. SP2A adopts OWL-S [16], but other solutions are available. The Web Services Modeling Framework (WSMF) [11] is an alternative approach for semantically annotating Web Services, aiming at resolving interoperability problems faced by Web Service composition. Its main elements are: ontologies, goal descriptions, elementary and complex Web Services, and mediators. WSFM is an extension of the Unified Problem-solving Method Development Language (UPML) [12], which has been developed to describe, implement and semi-automatically reuse knowledge-intensive reasoning systems based on libraries of generic problem-solving components. UPML is also the basis of the Internet Reasoning Service (IRS) [9], which provides a means for ontology-based Web Service selection using reasoning.

In the peer-to-peer context, Crespo and Molina [6] propose that node connections be influenced by content, so that semantically related nodes are clustered together, forming a Semantic Overlay Network (SON). In a SON system, each query is processed by identifying which SONs are better suited to answer it. Then the query is sent to a node in those SONs and forwarded only to the other members of that SON. Our approach is completely different, because SP2A peergroups are userdriven, not self-organizing. Moreover, in SP2A queries are propagated in a particular peergroup only upon user decision, otherwise they are routed in the main group.

Our solution can be compared to the one proposed by Nakauchi *et al.* [17], whose key mechanism is query expansion at nodes that receive the query. In details, a searcher issues a query which indicates several keywords. This query is flooded (forwarded to all neighbors) with a certain TTL (time to live). A node which receives a query performs query expansion using its local KRDB, *i.e.* a thesaurus which keeps some information about the data items stored locally in the node. An interesting metadata-based P2P infrastructure for educational purposes is provided by the Edutella Project [20]. Edutella Peers use JXTA P2P primitives to form the Edutella Network, in which they can share and retrieve services. The Edutella Query Exchange Language and the Edutella common data model provide the syntax and semantics for an overall standard query interface across the heterogenous peer repositories for any kind of RDF [3] metadata. Edutella wrappers are used to translate queries and results from the Edutella common format to the local format of the peer (*e.g.* RQL, TRIPLE, SQL, dbXML, AmosQL) and vice versa.

5 DSG AND THE STIL PROJECT

The research activity of the Distributed System Group (DSG) at the Information Engineering Department of the University of Parma mainly focuses on service-oriented architectures, Grid computing, peer-to-peer architectural models, and multimedia streaming in wired and wireless networks. The DSG is involved in several projects, both national (STIL and FIRB Web Minds), and international (EU Multi-Knowledge).

In this section, we shortly illustrate the STIL project (“Strumenti Telematici per l’Interoperabilità delle reti di imprese: Logistica digitale integrata per l’Emilia-Romagna”, *i.e.* telematic tools for interfirm networks interoperability:

digital logistics for the Emilia-Romagna region). In STIL, logistics is considered in the context of interactions between providers and consumers of products and services in the e-market place. STIL's final purpose is to create a region-wide Virtual Logistic Pole (VLP) providing Enterprise Application Integration (EAI), for both business to consumer (B2C) and business to business (B2B) applications to manufacturing firms, transportation carriers and logistic hubs.

The concept of *value-chain* is applied not only to the providers-producers chain, but also to public interext, with particular emphasis on process observation and optimization for environment and quality of life safeguard.

STIL defines an ICT infrastructure which offers mechanisms and policies for the semantic integration of applications (*eServices*) which realize stategical features for the value-chain. An eService is a software entity deployed by a service provider across the Internet. eServices can be statically selected by subscribing off-line contracts, or dynamically discovered using several approaches (for example, the SP2A approach). Obviously, an eService can be selected not only for its functionalities, but also for the quality of service (QoS) it guarantees.

From the technological point of view, eServices are implemented with Web Service technologies. By focusing solely on messages, the Web Service model is completely language, platform and object model-agnostic. A Web Service can be implemented using the full feature set of any programming language, object model, and platform. A Web Service can be consumed by applications implemented in any language for any platform. For these reasons, STIL system developers are able to independently implement services using their favourite languages and tools, and to choose among several deployment architectures, either client/server oriented, such as Tomcat, or peer-to-peer oriented, such as SP2A.

6 CONCLUSIONS

In this paper we illustrated our approach to the semantic discovery of services in peer-to-peer based inter-firm Grids. We illustrated two discovery strategies for the Router component, *i.e.* Semantic Discovery with Local Matching (SDLM), and Semantic Discovery with Distributed Matching (SDDM). We described the implementation and performance evaluation of their basic modules, and demonstrated that SDDM is the best solution because it evenly distributes service information and computational workload among all peers.

We are currently working to the implementation of the SDDM strategy, which introduces ontologybased functionalities at the overlay network level and thus requires JXTA modifications to enable semantic query matching on supernodes (*i.e.* routing peers). An additional goal is related to the investigation of automatic service orchestration for SP2A based on semantic discovery and inputoutput matching.

7 REFERENCES

The Globus Alliance. Homepage http://www.globus.org.

Java Universal Network/Graph Framework. Homepage http://jung.sourceforge.net.

Resource Description Framework. Homepage http://www.w3.org/RDF/.

Amoretti, M. & Reggiani, M. & Zanichelli, F. & Conte. G. Peer: an Architectural Pattern. The 12th Pattern Languages of Programs (PLoP), Monticello, Illinois, September, 2005.

Amoretti, M. & Zanichelli, F. & Conte, G. SP2A: a Service-oriented Framework for P2P-based Grids. 3rd International Workshop on Middleware for Grid Computing, Co-located with Middleware 2005. Grenoble, France, November, 2005.

Crespo, A. & Garcia-Molina, H. Semantic overlay networks for p2p systems. Technical Report, Computer Science Department, Stanford University, 2002.

De Roure, D. & Jennings, N. & Shadbolt, N. Research Agenda for the Semantic Grid: A Future e-Science Infrastructure. Technical report, National e-Science Centre, December 2001.

Distributed Systems Group. JXTA-SOAP project. Homepage http://soap.jxta.org.

Domingue, J. & Cabral, L. & Hakimpour, F. & Sell, D. & Motta, E. IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services. *Workshop on WSMO Implementations*, Frankfurt,Germany, September, 2004.

C. Fellbaum, editor. WordNet - An Electronic LexicalDatabase. *MIT Press*, 1998.

Fensel, D. & Bussler, C. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications, 1(2), 2002.*

Fensel, D. & al. The unified problem solving method development anguage upml. *Knowledge and Information Systems, 5(1), 2002.*

Foster, I. & Iamnitchi, A. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid *Computing. 2nd International Workshop on P2P Systems (IPTPS ’03), Berkeley, CA, February, 2003.*

Foster, I. & Kesselman, C. & Nick, J. & Tuecke. S.. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications*, 15(3), March, 2001. [15] Global Grid Forum. Semantic Grid. Homepage https://forge.gridforum.org/projects/sem-rg, 2005.

Martin, D. & al. Bringing Semantics to Web Services: the OWL-S Approach. In *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, San Diego, California, USA, July, 2004.

Nakauchi, K. & Morikawa, H. & Aoyama, T. Design and Implementation of a Semantic Peer-to-Peer Network. In 7th IEEE International Conference on High Speed Networks and Multimedia *Communications (HSNMC ’04), June, 2004.*

Natural Language Processing Group. General architecture for text engineering (GATE). Homepage http://gate.ac.uk.

Navigli, R. & Velardi, P. Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Technical Report*, 2004.

Nejdi, W. & al. Edutella: a P2P Networking Infrastructure Based on RDF. *WWW2002*, May 2002.

S. O. A. Reference Model TC. Service Oriented Architecture Reference Model. *Technical report*, OASIS, June 2005.

The Stencil Group. The Evolution of UDDI -UDDI.org, *White Paper*, July, 2002. [23] Traversat, B. & al. Project JXTA 2.0 Super-Peer Virtual Network. *Technical Report*, Sun Microsystems, May, 2003.