

# Ontology-based Semantic Service Interoperability <sup>\*</sup>

Devis Bianchini<sup>1</sup> Valeria De Antonellis<sup>1</sup> Michele Melchiori<sup>1</sup> Denise Salvi<sup>1</sup>

Università degli Studi di Brescia - Dip. di Elettronica per l'Automazione  
Via Branze, 38 - 25123 Brescia - Italy  
{bianchin|deantone|melchior|salvi}@ing.unibs.it

**Abstract.** Enterprise interoperability in distributed environments has been recently improved by adopting the emerging Web Service standards and technology, making an ever-growing number of services available on the web. One of the major problems in this framework is how to make effective and efficient the process of matching service descriptions in order to find suitable available services for a given service request. Semantic service interoperability is thus a main research issue. In the paper, we propose the Semantic Driven Service Discovery approach for open P2P systems, where peers are organized in a semantic community for interoperability purposes. In the approach, ontologies are introduced to express domain knowledge related to service descriptions and to guide service discovery by mapping/matching service requests against service entertainments according to defined inter-peer semantic links.

## 1 Introduction

Enterprise interoperability in distributed environments has been recently improved by adopting the emerging Web Service standards and technology, making an evergrowing number of services available on the web. One of the major problems in this framework is how to make effective and efficient the process of matching service descriptions in order to find suitable available services for a given service request. Semantic service interoperability is thus a main research issue.

Semantics is particularly important to share (information and) services in open P2P systems where the lack of a common understanding of the world generates the need for explicit guidance in discovering available resources [1, 2, 12, 16, 17]. The use of ontologies is suggested to share descriptions of available resources and, in particular, ontology-based techniques have been defined for

---

<sup>\*</sup> This work has been partially supported by the ESTEEM (Emergent Semantics and cooperaTion in multi-knowledge EnvironmEnt [10]) PRIN Project and the TEKNE (Towards Evolving Knowledge-based interNetworked Enterprise [15]) FIRB Project funded by the Italian Ministry of Education, University and Research, and by NoE INTEROP [13] IST Project n. 508011 - 6th EU Framework Program.

service discovery [9, 14]. In fact, ontologies provide the benefits of formal specifications and inference capabilities apt to improve model-based semantic service discovery with flexible ontology-based matching.

In this paper, we propose the Semantic Driven Service Discovery approach for open P2P systems, where peers are organized in a semantic community for interoperability purposes. In the approach, ontologies are introduced to express domain knowledge related to service descriptions and to guide service discovery by mapping/matching service requests against service entertainments among peers.

The paper is organized as follows: Section 2 gives an overview of the considered application scenario; Section 3 presents a formal approach to semantic service description; Section 4 illustrates the semantic driven service discovery approach; Section 5 describes the COMPAT system for service matching and discovery; finally, in Section 6 some final considerations and future work are discussed.

## 2 The application scenario

In this section, we briefly illustrate the considered application scenario, an open networked system where peers join to form a semantic community sharing information and services. In our approach, the service sharing issues are considered and ontologies (called *peer ontologies*) are introduced to express domain knowledge related to service descriptions and to guide service discovery among peers [6].

For community constitution, we assume that a peer called promoter spreads out a manifesto containing a suitable portion of its peer ontology, expressing a core domain knowledge for possible member aggregation. Each peer that aims at joining the community matches its own peer ontology against the manifesto and replies to the promoter. Otherwise, if the peer is not interested, it forwards the manifesto to the other peers of the network. Once the semantic community is established, services can be searched and exchanged between the members of the community by means of ontology-based techniques.

First of all, each peer searches for its *semantic neighbors* by establishing semantic mappings also called inter-peer semantic links. To do this, it sends a probe service request for each service he wants to make sharable; this probe service request contains the description of the service interface in terms of required service functionalities, each of them described through the names of operations and input/output parameters. The probe service request is sent to all the other peers of the semantic community, according to the actual list of community members. When a peer receives a probe service request, it matches the service request against its own service descriptions and establishes suitable mappings by applying the matchmaking techniques explained in Section 4. Then it sends to the peer from which the probe service request came a message containing the resulting mappings.

In this way, each peer can build a map of its semantic neighbors, with the similar services and the semantic mappings with them. The community can properly evolve when new peers join it or new services are published on peers of the community.

Note that the semantic community is organized according to a P2P network, constituted by  $n$  peers, each of them carrying its own services. In the scope of the community, each peer can play different roles: (i) to search for a given service (*requester*); (ii) to propose a set of candidate services when a service request is given, through the application of matchmaking techniques (*broker*); (iii) to provide a selected service for its invocation and to publish a new service (*provider*). In an evolving collaborative P2P community, a peer can contain the description of a candidate service, while a different peer acts as a provider for that service, or a peer can be both a requester and a broker. According to this general view, each peer presents the architecture shown in Figure 1. In the following we focus on the Semantic Peer Registry and the Service MatchMaker.

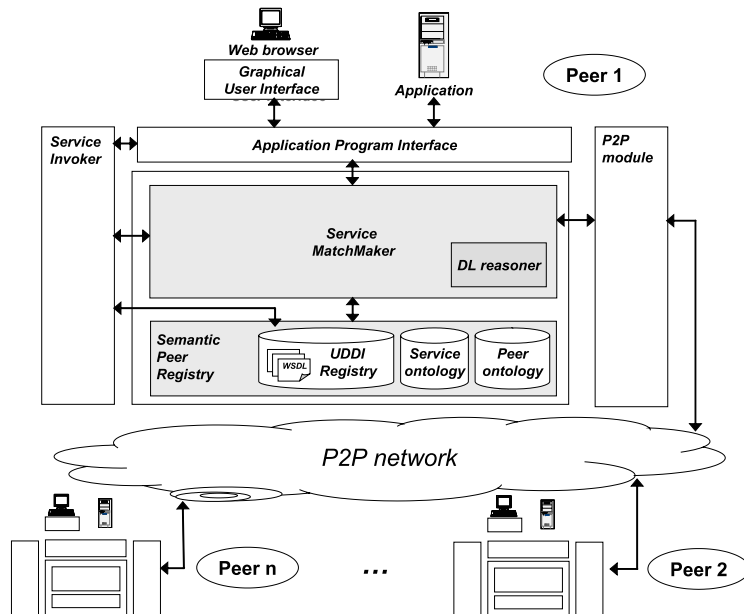


Fig. 1. Peer architecture.

### 3 Ontology-based Service Description

Service descriptions represent functional aspects of a service, based on the WSDL standard for service representation, in terms of service category, service function-

alities (operations) and input/output messages (parameters). Services are stored in an extended UDDI Registry, called *Semantic Peer Registry*, where besides the UDDI registry and WSDL descriptions, a peer ontology is added to provide semantic knowledge related to service descriptions and a service ontology is added to organize semantic service descriptions and semantic links. Service ontology also contains references to semantic neighbors through inter-peer semantic links.

The peer ontology is constituted by:

- a *Service Functionality Ontology (SFO)*, that provides knowledge on the concepts used to express service functionalities (operations);
- a *Service Message Ontology (SMO)*, that provides knowledge on the concepts used to express input and output messages (parameters) of operations.

Concepts in the peer ontology are organized according to semantic relationships, `subclass-of` and `equivalent-to`. Furthermore, the peer ontology is extended by a thesaurus providing terms and terminological relationships (as synonymy, hypernymy and so on) associated to names of concepts in the peer ontology. In this way, it is possible to enlarge matching capabilities when looking for correspondences between elements in service descriptions and concepts in the ontology. In fact, a requester can not be constrained to use exactly the concept names in the peer ontology used to describe a particular offer and, moreover, in a P2P context different peers can refer to different peer ontologies. The thesaurus is automatically derived by considering the set  $T$  of terms denoting atomic concepts in the peer ontology, the lexical system WordNet [11] and related terminological relationships.

In the service ontology, services are semantically represented by DL logic concepts [3], whose elements (service category, operation names, input/output parameter names) are properly mapped to the peer ontology. According to the Description Logic formalism, each concept is represented as a conjunction of:

- one or more concepts in the form  $\exists\text{hasCategory.CAT}$ , where  $CAT$  is an atomic concept representing an associated service category;
- one or more concepts in the form  $\exists\text{hasOperation.OP}$ , where  $OP$  is described as a conjunction of:
  - an atomic concept representing the operation name;
  - one or more concepts  $\exists\text{hasInput.IN}$ , where  $IN$  is an atomic concept representing an input parameter of the operation;
  - one or more concepts  $\exists\text{hasOutput.OUT}$ , where  $OUT$  is an atomic concept representing an output parameter of the operation.

Note that these expressions can be represented in OWL-DL, corresponding to the  $\mathcal{SHOIN}(\mathcal{D}^+)$  family of Description Logics. A service request is represented by means of the same Description Logic expression. Both peer ontology and service ontology are expressed in OWL-DL. Note that services are published in the UDDI registry by means of traditional UDDI Publish API, in order to keep compatibility with existing UDDI standard, together with their WSDL document.

## 4 Model-based service mapping

The Service MatchMaker is in charge of finding mappings between service descriptions, combining together different matching models [5]: (i) a *deductive model*, exploiting deduction algorithms for reasoning on service descriptions [4], (ii) a *similarity-based model*, where retrieval metrics are applied to measure the degree of match between services [7]. Mapping between two service descriptions is specified by the kind of match (exact, partial, mismatch) and quantified by means of suitable similarity coefficients, that is, by evaluating the service similarity. Both the matching models are based on jointed use of the peer ontology and the thesaurus by defining two basic concepts: the *Name Affinity concept* and the *Affinity-based subsumption test*.

**Definition 1 (Name Affinity coefficient).** *The Name Affinity coefficient between  $t$  and  $t'$  of the Thesaurus, denoted by  $NA(t, t')$ , is computed as follows:*

$$NA(t, t') = \begin{cases} 1 & \text{if } t = t' \\ \max_l(\tau(t \rightarrow^l t')) & \text{if } t \neq t' \wedge \\ & t \rightarrow^l t', l \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where  $t \rightarrow^l t'$  denotes a path of terminological relationships from  $t$  to  $t'$ . We consider SYN for synonymy, BT/NT for broader/narrower term and RT for related term. A weight  $\sigma_{tr} \in [0, 1]$  is associated to each kind of terminological relationship  $tr$ , in order to evaluate its implication for name affinity; in our experimentation,  $\sigma_{SYN} = 1$ ,  $\sigma_{BT/NT} = 0.8$  and  $\sigma_{RT} = 0.5$ . The function  $\tau(t \rightarrow^l t') = \prod_{k=1}^l (\sigma_{tr_k}) \in [0, 1]$  defines the strength of  $t \rightarrow^l t'$  as the product of the weights of all terminological relationships in the path. Since between two terms in the thesaurus there can exist more than one path, the one with the highest strength is chosen. We say that  $t$  and  $t'$  have name affinity ( $t \sim t'$ ) if and only if  $NA(t, t') \geq \alpha$ , where  $\alpha > 0$  is a threshold set on the basis of experimental results to select only terms with high values of the name affinity coefficient.

**Definition 2 (Affinity-based subsumption test).** *Given an atomic concept  $C$  in the peer ontology  $\mathcal{PO}$ , we define the set of terms in the thesaurus that have name affinity with a given concept as  $C_{\mathcal{TH}} = \{T \in \mathcal{TH} \mid T \sim C\}$ . Analogously, we define the set of concepts of  $\mathcal{PO}$  that have name affinity with a term  $T$  in  $\mathcal{TH}$  as  $T_{\mathcal{PO}} = \{C \in \mathcal{PO} \mid T \in C_{\mathcal{TH}}\}$ .*

*Given the peer ontology  $\mathcal{PO}$ , the thesaurus  $\mathcal{TH}$  and a pair of terms  $T^1$  and  $T^2$  used in service descriptions to denote service elements,  $T^1$  is subsumed by  $T^2$  with respect to  $\mathcal{TH}$ , denoted by  $T^1 \sqsubseteq_{\mathcal{TH}} T^2$ , if and only if there exists  $C \in T^1_{\mathcal{PO}}$  and  $D \in T^2_{\mathcal{PO}}$  such that  $C \sqsubseteq D$  is satisfied in  $\mathcal{PO}$ . Note that we pose  $T^1 \equiv_{\mathcal{TH}} T^2$  if both  $T^1 \sqsubseteq_{\mathcal{TH}} T^2$  and  $T^2 \sqsubseteq_{\mathcal{TH}} T^1$  hold.*

The Affinity-based subsumption test is exploited to determine the kind of match between a service request  $\mathcal{R}$  and a supplied service  $\mathcal{S}$ , considering separately service description components. In particular, five kinds of match are

distinguished:

- **exact** match, to denote that  $\mathcal{S}$  and  $\mathcal{R}$  have the same capabilities, that is, they have: (i) equivalent operations; (ii) equivalent output parameters; (iii) equivalent input parameters;
- **plug-in** match, to denote that  $\mathcal{S}$  offers at least the same capabilities of  $\mathcal{R}$ , that is, names of the operations in  $\mathcal{R}$  can be mapped into operations of  $\mathcal{S}$  and, in particular, the names of corresponding operations, input parameters and output parameters are in any generalization hierarchy; the inverse kind of match is denoted as **subsume**;
- **intersection** match, to denote that  $\mathcal{S}$  and  $\mathcal{R}$  have some common operations and some common I/O parameters, that is, some pairs of operations and some pairs of parameters, respectively, are related in any generalization hierarchy;
- **mismatch**, otherwise.

Similarity analysis is applied to quantify the match between services. In particular, when **exact** or **plug-in** match occur, similarity between services is set to 1 (full similarity); if **mismatch** occurs, the similarity value is set to zero; finally, when **subsume** or **intersection** match occur, similarity coefficients summarized in Table 1 are computed to quantify the degree of match between services. These coefficients are based on the Dice's metrics and have been widely experimented [8].

Once established, mappings between service request and entertainments are exploited to find out suitable available services for the request. First of all, inter-peer semantic links are exploited to extend in an efficient way service discovery over the P2P network. Given the service request  $\mathcal{R}$ , a peer  $p$  searches for candidate services in its own Semantic Peer Registry and retrieves a list  $CS = \{\langle \mathcal{S}_1, GSim_1, mt_1 \rangle, \dots, \langle \mathcal{S}_n, GSim_n, mt_n \rangle\}$  of candidate services with corresponding similarity values  $GSim_i$  and match type  $mt_i$ . If a service  $\mathcal{S}_i$  presents an **exact** or a **plug-in** match with the request, then  $\mathcal{S}_i$  satisfies completely the required functionalities and it is not necessary to forward the service request to semantic neighbors. Otherwise, if a service  $\mathcal{S}_i$  presents a **subsume** or an **intersection** match with the request, the peer  $p$  forwards the request to those peers that are semantic neighbors of  $p$  with respect to  $\mathcal{S}_i$ . Note that  $p$  does not consider semantic neighbors that presents a **subsume** or an **exact** match with  $\mathcal{S}_i$  because this means that they provide services with the same functionalities or a subset of  $\mathcal{S}_i$  functionalities and they cannot add further capabilities to those already provided by  $\mathcal{S}_i$  on the peer  $p$ . A list of semantic neighbors  $SN = \{\langle p_1, \{\mathcal{S}_1, GSim_1, mt_1, \dots, \mathcal{S}_{m_1}, GSim_{m_1}, mt_{m_1}\} \rangle, \dots, \langle p_k, \{\mathcal{S}_1, GSim_1, mt_1, \dots, \mathcal{S}_{m_k}, GSim_{m_k}, mt_{m_k}\} \rangle\}$  is obtained in this phase and is used to forward the original request. Semantic neighbors in  $SN$  can be ranked with respect to their relevance with the original request. Given a semantic neighbor  $sn \in SN$ , its relevance with respect to the request  $\mathcal{R}$  is computed according to the following equation:

$$r_{sn} = \frac{\sum_{i=1}^{m_j} \frac{2 * GSim_i * GSim(\mathcal{R}, \mathcal{S}_i)}{GSim_i + GSim(\mathcal{R}, \mathcal{S}_i)}}{m_j} \quad (2)$$

ENTITY-BASED SIMILARITY
$ESim(\mathcal{R}, \mathcal{S}) = \frac{2 \cdot A_{tot}(IN_{\mathcal{R}}, IN_{\mathcal{S}})}{ IN_{\mathcal{R}}  +  IN_{\mathcal{S}} } + \frac{2 \cdot A_{tot}(OUT_{\mathcal{R}}, OUT_{\mathcal{S}})}{ OUT_{\mathcal{R}}  +  OUT_{\mathcal{S}} } \in [0, 2]$ <p> <math>IN_{\mathcal{R}}, IN_{\mathcal{S}}</math> - sets of input parameter names of <math>\mathcal{R}</math> and <math>\mathcal{S}</math>  <math>OUT_{\mathcal{R}}, OUT_{\mathcal{S}}</math> - sets of output parameter names of <math>\mathcal{R}</math> and <math>\mathcal{S}</math>  <math>A_{tot}(IN_{\mathcal{R}}, IN_{\mathcal{S}}) = \sum_{in_{\mathcal{R}}^i \in IN_{\mathcal{R}}, in_{\mathcal{S}}^j \in IN_{\mathcal{S}}} NA(in_{\mathcal{R}}^i, in_{\mathcal{S}}^j)</math>  <math>A_{tot}(OUT_{\mathcal{R}}, OUT_{\mathcal{S}}) = \sum_{out_{\mathcal{R}}^i \in OUT_{\mathcal{R}}, out_{\mathcal{S}}^j \in OUT_{\mathcal{S}}} NA(out_{\mathcal{R}}^i, out_{\mathcal{S}}^j)</math> </p>
OPERATION SIMILARITY
$OpSim(op_{\mathcal{R}}^i, op_{\mathcal{S}}^j) = NA(name_{op_{\mathcal{R}}^i}, name_{op_{\mathcal{S}}^j}) + \frac{2 \cdot A_{tot}(IN_{\mathcal{R}}^i, IN_{\mathcal{S}}^j)}{ IN_{\mathcal{R}}^i  +  IN_{\mathcal{S}}^j } + \frac{2 \cdot A_{tot}(OUT_{\mathcal{R}}^i, OUT_{\mathcal{S}}^j)}{ OUT_{\mathcal{R}}^i  +  OUT_{\mathcal{S}}^j } \in [0, 3]$ <p> <math>IN_{\mathcal{R}}^i, IN_{\mathcal{S}}^j</math> - sets of input parameter names of the <math>i</math>-th operation of <math>\mathcal{R}</math> and the <math>j</math>-th operation of <math>\mathcal{S}</math>  <math>OUT_{\mathcal{R}}^i, OUT_{\mathcal{S}}^j</math> - sets of output parameter names of the <math>i</math>-th operation of <math>\mathcal{R}</math> and the <math>j</math>-th operation of <math>\mathcal{S}</math> </p>
FUNCTIONALITY-BASED SIMILARITY
$FSim(\mathcal{R}, \mathcal{S}) = \frac{2 \cdot \sum_{i,j} OpSim(op_{\mathcal{R}}^i, op_{\mathcal{S}}^j)}{ OP(\mathcal{R})  +  OP(\mathcal{S}) } \in [0, 3]$ <p> <math>OP(\mathcal{R}), OP(\mathcal{S})</math> - sets of operation names of <math>\mathcal{R}</math> and <math>\mathcal{S}</math> </p>
GLOBAL SIMILARITY
$GSim(\mathcal{R}, \mathcal{S}) = w_1 \cdot NormESim(\mathcal{R}, \mathcal{S}) + w_2 \cdot NormFSim(\mathcal{R}, \mathcal{S}) \in [0, 1]$ <p> <math>w_1, w_2</math> - weights introduced to assess the relevance of each kind of similarity (<math>w_1 \in [0, 1]</math> and <math>w_2 = 1 - w_1</math>)  <math>NormESim(), NormFSim()</math> - <math>ESim()</math> and <math>FSim()</math> normalized to the range <math>[0, 1]</math> </p>

**Table 1.** Similarity Coefficients between service descriptions  $\mathcal{R}$  (request) and  $\mathcal{S}$  (supply).

Ranking of semantic neighbors is exploited to constrain the forwarding according to a threshold-based mechanism. Finally, the list of candidate services is ranked with respect to the match type (**exact** > **plug-in** > **subsume** > **intersection** > **mismatch**) and the  $GSim$  value, respectively, and a threshold-based strategy is exploited to filter out searching results: only candidate services with a match type different from **mismatch** and a  $GSim$  value equal or greater than a given threshold are maintained.

## 5 The COMPAT system

The tool implementing the proposed approach, called COMPAT Service Model Mapping, has been developed maintaining full backward compatibility with existing service technologies and standards (WSDL, OWL, UDDI). COMPAT is

implemented in Java and performs automatic service discovery in P2P environments. The main features of COMPAT architecture are summarized in Figure 2.

COMPAT Service Model Mapping	
Purpose	Service2Service mapping
Source/Target Model	Service functional model
Source/Target Representation Language	OWL-DL
Mapping type	One-to-One
Matching Strategy	Similarity-based, Deductive-based
Ontology-based Matching	Yes (SMO, SFO, Thesaurus)
Optimization Strategy	Yes (inter-peer semantic links)
Ranking	Qualitative/quantitative
Search result filtering	Threshold-based
Implementation Language	Java
Degree of automation	Automatic
Compliance with standards	Yes (WSDL, OWL, UDDI)
Background	Academic

**Fig. 2.** The main features of the COMPAT Service Model Mapping

Service discovery is obtained by searching for mappings between service functional models derived by WSDL documents and expressed in OWL-DL. Mappings are established between pairs of service descriptions (*One-to-one* mappings) by combining deductive-based and similarity-based matching strategies. The jointed use of peer ontology (SMO, SFO) and thesaurus during the match-making process allows to take into account semantic aspects. Moreover, optimization of service discovery is addressed through inter-peer semantic links to forward the request on the P2P network in an efficient way. Searching results are ranked both in a qualitative way, using the kind of match, and in a quantitative way, using Gsim value, and are filtered out by means of a threshold-based mechanism.

## 6 Concluding remarks and future work

In this paper, we proposed a semantic-based approach for service discovery in a P2P scenario, where peers are organized in communities. Specific ontologies (called Service and Peer Ontologies) are used to add semantics to service descriptions and their organization inside traditional UDDI Registry enhanced with semantic aspects. Further experimentation in a P2P context will evaluate the impact of the proposed approach on several parameters such as network overloading and its analysis for concrete application (for example, support of business process design and execution in distributed environments) will be performed.

## References

1. K. Arabshian and H. Schulzrinne. An Ontology-based Hierarchical Peer-to-Peer Global Service Discovery System. *Journal of Ubiquitous Computing and Intelligence (JUCI)*, 2006.
2. F. Banaei-Kashani, C. Chen, and C. Shahabi. WSPDS: Web Services Peer-to-Peer Discovery Service. In *Proc. of the Int. Conference on Internet Computing (IC'04)*, pages 733–743, Las Vegas, Nevada, USA, 2004.
3. D. Bianchini and V. De Antonellis. Ontology-based Semantic Interoperability Tools for Service Dynamic Discovery. In *IFIP Int. Conf. on Interoperability of Enterprise Software Applications, INTEROP-ESA'05*, Geneva, Switzerland, 2005.
4. D. Bianchini, V. De Antonellis, and M. Melchiori. Capability Matching and Similarity Reasoning in Service Discovery. In *CAiSE Int. Workshop on Enterprise Modeling and Ontologies for Interoperability, EMOI 2005*, Porto, Portugal, June 2005.
5. D. Bianchini, V. De Antonellis, M. Melchiori, and D. Salvi. Semantic-enriched Service Discovery. In *IEEE ICDE Int. Workshop on Challenges in Web Information Retrieval and Integration, WIRI 2006*, Atlanta, Georgia, USA, 2006.
6. D. Bianchini, V. De Antonellis, M. Melchiori, and D. Salvi. Ontology-based Service Matchmaking in Semantic Communities. Technical Report D 3.2, ESTEEM Emergent Semantics and cooperation in multi-knowledge Environments, [http://www.dis.uniroma1.it/~esteem/docs/Deliverable\\_dic06/Brescia/Deliverable\\_D3.2\\_UNIBS.pdf](http://www.dis.uniroma1.it/~esteem/docs/Deliverable_dic06/Brescia/Deliverable_D3.2_UNIBS.pdf), 2007.
7. D. Bianchini, V. De Antonellis, B. Pernici, and P. Plebani. Ontology-based Methodology for e-Service discovery. *Journal of Information Systems, Special Issue on Semantic Web and Web Services*, 31(4-5):361–380, June-July 2006.
8. S. Castano and V. De Antonellis. A Framework for expressing Semantic Relationships between Multiple Information Systems for Cooperation. *Information Systems*, 123(3-4):253–277, 1998.
9. The OWL Service Coalition. *OWL-S 1.1 release*, November 2004. <http://www.daml.org/services/owl-s/1.1/>.
10. ESTEEM Project Home Page: Emergent Semantics and cooperation in multi-knowledge Environment. <http://www.dis.uniroma1.it/~esteem/index.html>.
11. C. Fellbaum. *Wordnet: An Electronic Lexical Database*. MIT Press, 1998.
12. P. Haase, S. Agarwal, and Y. Sure. Service-Oriented Semantic Peer-to-Peer Systems. In *Proc. of the WISE Workshop on Intelligent Networked and Mobile Systems*, pages 46–57, Brisbane, Australia, 2004.
13. The INTEROP NoE Portal. <http://www.interop-noe.org/>.
14. U. Keller, H. Lausen, and D. Roman. *Web Service Modeling Ontology (WSMO)*. WSMO Working Draft, March 2004. <http://www.wsmo.org/2004/d2/v02/20040306/>.
15. TEKNE Project Home Page: Towards Evolving Knowledge-based interNetworked Enterprise. <http://www.tekne-project.it/>.
16. I. Toma, B. Sapkota, J. Scicluna, J.M. Gomez, D. Roman, and D. Fensel. A P2P Discovery mechanism for Web Service Execution Environment. In *Proc of the 2nd WSMO Implementation Workshop (WIW'05)*, Innsbruck, Austria, June 6th-7th 2005.
17. K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, and J. Miller. METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services. *Journal of Information Technology and Management, Special Issue on Universal Global Integration*, 6(1):17–39, 2005.