SERSE: Searching for Semantic Web Content

Valentina Tamma and **Ian Blacoe** and **Ben Lithgow Smith** and **Michael Wooldridge** [1]

Abstract. Searching the Semantic Web (SW) for digital content is arguably more complex than searching the current web. In fact, a SW search system must deal with a large number of distributed, heterogeneous resources, that may reference many different ontologies.
In order to manage this complexity we have integrated different technologies such as peer-to-peer, ontologies, and multi-agent technology in the design of SERSE, a multi-agent system for searching the SW. In SERSE, agents are organised in a P2P network according to a *semantic overlay*, where the neighbourhood is determined by the semantic proximity of the ontological definitions that are known to the agents. The integration of these technologies poses some problems. On the one hand, the more ontological knowledge the agents have, the better we can expect the system to perform. On the other hand, global knowledge would constitute a point of centralisation which might potentially degrade the performance of a P2P system. In addition, maintenance of such semantic overlay can also degrade the performance.
The paper presents SERSE together with some experimental results that evaluate the performance in response to changes in the size of semantic neighbourhood and an analytical evaluation of the workload of the system due to maintenance activities.

1 Introduction

The Semantic web promises to add value to the current web, without requiring any fundamental changes to the web infrastructure that is currently in place. The added value is created, in part, by enriching digital content with annotations that reference *ontologies* — explicit and machine sharable representations of the conceptualisation abstracting a phenomenon [15].

Because of semantic annotations, the Semantic Web (SW) is intended to be primarily understood by software agents, and is thus based on machine-processable languages such as RDF(S), DAML+OIL, and OWL [1], which permit knowledge to be embedded within web pages. Software agents are able to process the knowledge expressed in these semantic markup languages, and can thus offer services which make use of or retrieve this knowledge, including search and retrieval services. In contrast, the web of today is intended primarily for human consumption, and is based on visual markup languages (such as HTML) intended for human users.

Searching the SW is arguably more complex than searching the current web. Not only must a SW search system deal with a large number of distributed, heterogeneous resources, but these resources may reference many different ontologies. Current technology struggles to deal with such complexity: in order to manage this increased complexity we have integrated in a search system different technologies that have proven successful for searching the current web in an efficient and scalable way, that is peer-to-peer, ontologies, and multi-agent technology. SERSE, (*SEmantic Routing SystEm*), is a multi-agent system for searching the SW in which *router* agents have equal capabilities and responsibilities, and which communicate on a peer-to-peer basis. Semantic descriptions of digital content are used to determine a *semantic overlay* on the network of peers, where the peers can communicate only with those peers who are within their *semantic neighbourhood*. The design and implementation of such a *semantically organised P2P* system raised a number of critical issues concerning the performance of the system with respect to the workload necessary for maintaining the semantic overlay.
In the reminder of this paper we describe the services offered by SERSE, illustrate its architecture from an agent perspective, and present some experimental results that evaluate the performance in response to changes in the size of semantic neighbourhood and an analytical evaluation of the workload of the system due to maintenance activities.

2 Desiderata for Searching the Semantic Web

The design choices we made in SERSE are motivated by some considerations and requirements for a (multi-agent) system that is able to efficiently navigate and search the semantic web. Searching the SW is arguably more complex than searching the current web. We believe that one way to overcome this increased complexity is to take the view of the SW as "fragmented knowledge": each fragment represents a specific topic or a group of similar concepts. If we have to decide the most efficient route from topic A to B, we could simply try a random direction – but this would give no guarantee to find B in a finite time. Alternatively, we can simply try to find another topic C, whose existence we are certain about, and that is reasonably close to the topic we aim to reach. By using this approach we are sure to reach the right fragment of knowledge in a limited time (the sum of the times needed to reach each topic between A and B), and if a new fragment is discovered, we can compute the route to this fragment by finding the route to the topic closest to the new one, and then between these two. We have identified a number of requirements that our system should exhibit:
1. *Decentralisation*: Efficient navigation toward a specific topic depends on the ability to identify the "right direction" in which to head. A centralised approach would imply maintaining a directory of all topics, acting as a centralised server for queries. However, this can prove too inefficient and cumbersome for the SW [10]. An alternative is to have system components each with equal roles and the capability to exchange knowledge and services directly with each other. Peer-to-peer technology (P2P) such as Edutella [12] or Morpheus [3] is a possible answer to this quest for decentralisation.
2. *Openness*: Openness is inherent in both the syntactic and the se-

[1] Department of Computer Science, University of Liverpool, Liverpool L69 3BX, United Kingdom email: {V.A.M.Tamma, I.W.Blacoe, D.B.Lithgow-Smith, M.J.Wooldridge} @csc.liv.ac.uk

mantic web. In the case of the SW, openness is enhanced, because ubiquitous software components – from computers to cell phones and TV sets – are involved in the process of using it [17].

3. *Autonomy and social ability*: P2P systems are usually comprised of simple data-storing systems, and the sharing and retrieval of data is performed on the basis of a central directory, while communication among peers simply relies on raw TCP/IP protocols. However, we require that our system components be able to interact with users and other peers, and decide which peer supervises the "fragment of knowledge" (or neighbourhood) closest to their own one. This implies that our components must exhibit autonomy and social ability: characteristic features of multi-agent systems [18].

4. *Scalability*: As with the Internet itself, the SW will grow exponentially [17]. Scalability is thus critical: the system must adapt flexibly and dynamically to the highly dynamic nature of the SW.

5. *Semantic based content retrieval*: Content retrieval can exploit the ontology-based annotation of the SW. System components must exhibit the ability to understand and process ontological descriptions. When knowledge is factored into fragments, navigation is determined by means of the ontological descriptions of the concepts to search. Semantics is used to establish the right path to a neighbourhood, by evaluating the similarity between concepts, as described above.

A key issue when searching digital content is to determine the size of a neighbourhood. This has implications on the topology of the network of peers — in particular, on the level of centralisation — and on the overlay networks, on the performance of the system, and on scalability. There is clearly a spectrum, where at one extreme a small neighbourhood means a search component has only knowledge local to few concepts of the ontology(ies), and at the other extreme, there is global knowledge (every search component has knowledge of the whole ontology or ontologies used to annotate the instances to search). Local knowledge favours a purely decentralised approach, but gives little help in determining the right path, while global knowledge creates a single point of failure that affects the performance of the system [8]. Therefore, one of the first questions we tried to answer is whether we can determine a "best" neighbourhood size: a number of concepts per neighbour that maximises the trade-off between amount of knowledge of the peers and performances of the system. Naturally, knowledge comes to a cost, and we also need to analyse the cost of maintaining this knowledge, especially in a dynamic environment such as the SW, where ontologies are likely to evolve in time (because of changes in the domain they represent or to better suit a particular application). The experiments that we present in the reminder of this paper were intended to verify the existence of this neighbourhood size, and to estimate the cost of building and maintaining such neighbourhood.

3 SERSE

The requirements and goals identified in the previous section have motivated the design of SERSE (*Semantic Routing System*). SERSE is implemented as a multi-agent system, whose routing agents share a core of equal capabilities and responsibilities, and which are capable of retrieving web resources based on the semantics of their annotations. The system is internally organised in a peer-to-peer fashion: Each router can communicate with its immediate neighbours, and the neighbourhood size is determined by the semantic proximity between the concepts known to the agents. No agent can broadcast messages to the whole system, and no agent has global knowledge of the network: this ensures decentralisation.

SERSE was implemented using JADE [6], a FIPA-compliant middleware platform. JADE is used to handle the transport aspects of agent communication: our implementation builds on JADE to provide a semantic overlay network, i.e., the *logical* organisation of the routers in a network of peers, which is based on the notion of semantic neighbourhood. Agents in SERSE have knowledge of a number of concepts forming the ontologies, that are expressed in OWL and stored in some ontology server. Agents have the ability to send FIPA messages to the agents belonging to their immediate semantic neighbourhood. Although limited, these "social abilities" permit the agents to autonomously and dynamically determine the most appropriate router agent, i.e., the agent that can retrieve instances of a concept that is identical or semantically closest to the queried concept, and to route them an unanswered query.

SERSE's is developed as part of the IST project "Esperonto" (IST 2001-34373). In Esperonto SERSE interacts with the other components in order to retrieve the SW content requested in user queries. In particular, SERSE interacts with *Notification Server* (NS), that is part of Esperonto component providing annotation and ontology services [11]: the *Semantic Annotation Service Provider* (SemASP). The NS notifies the routing system of the annotation of new resources, returning a list with the URIs of the instances of those concepts used in the annotation and that can be requested in user queries.

SERSE is composed of five types of agents, each playing different roles in the retrieval of digital content.

Portal agent: The portal agent acts as the point of entry into the network of router. It is responsible for triggering the routing process.

Notification agent: The notification agent receives XML messages from the Notification Server and parses them. It can then resend the content as an ACL message via the Portal Agent. The messages can be the notification of a new OWL ontology or of new content acquisitions. If the notification refers to a new ontology, the Notification Agent extracts the concepts definitions from the OWL file and populates the system with the routers. It also informs the routers of which agents are their immediate neighbours. If the notification refers to instances of a new concept, the notification agent creates a new router for the concept. If notification refers to instances of an existing concept, the notification agent re-sends the message into the router system to be routed to the correct index.

Router agent: Router agents are created once the system receives a notification of content acquisition regarding a new concept. Each router agent is associated with a concept in an ontology and is capable of retrieving instances of this concept. The router agents access the instances to index by reading them directly from the RDF files created by the SemASP. The locations of these files are reported to the router system by the Notification Server. The router agent also has a routing index, which contains the addresses of its semantic neighbours. When a router receives a query, it can either reply back with a FIPA `inform-ref` message, whose content is the retrieved instances, or with a `query-ref`, which re-routes the query to the agent with the least semantic distance. A router agent determines the semantic distance on the basis of the ontology. The concept for which the agent is responsible, and the set of concepts used to create the routing index are passed into the router agent as arguments when it is created.

Query management agent: The query management agent is the entry point for querying the system. It is intended to deal with the decomposition of complex queries and with the aggregation and validation of the results, but in the current implementation we support only the logical connectives AND and OR. The agent manages all the operations related to querying the system, including re-routing failed simple queries, sending multiple copies of queries in order to handle temporary unavailability, etc.

Interface agent: The interface agent acts as interface between the agents on the JADE platform and the external applications using them. The interface itself is comprised of both the agent that exchanges messages between the Query Management Agent and external applications, and of an external interface object, that connects to the agent socket and provides a simple API for generating queries and handling the answers.

The work of SERSE begins when a user poses a query by means of the system interface, which allows users to formulate a query in terms of the ontology known by the agents. (In the first prototype, we consider one ontology only, but we are currently working to support multiple ontologies.) We assume here that the system is already in the initial configuration, and that content availability messages were received by the *notification agent*, thus triggering the creation of the network of semantic routers. When a query is received by the *Query Management Agent* (QMA), it decomposes it (if it is a complex query) and then forwards the components to the *portal agent* (PA). The PA choses a random router and starts the routing process; it sends a `query-ref` message to the selected *router agent* that has as content the queried concept. The router agent consults its index to check whether it is able to access instances of the concepts. Instances are referenced by means of the URI they are accessible from. If the router agent can answer the query, it will reply back to the portal agent with an `inform-ref` whose content is the URIs of the instances. If the router agent does not have reference to the queried concept in its index, it needs to forward the query to the neighbour that is *semantically closest* to the queried concept. The router agent computes the similarity between its neighbours and the queried concept (in this prototype we evaluate the similarity in terms of *path length*), and then forwards the query to the router responsible for the concept closest to the queried one. The routing process is illustrated in Figure 1.

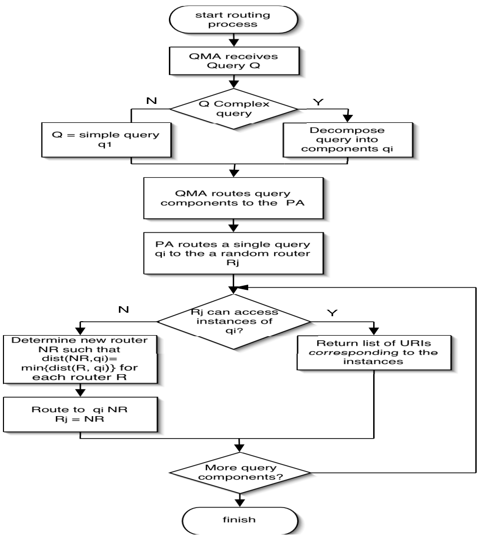

Figure 1. The routing process in SERSE

4 Experimental results

We ran a set of experiments on SERSE to test the system's *efficiency* — at this stage, we do not evaluate its *precision* or *recall*. The experiments are set in the biological domain. In order to ensure that the ontology structure was not designed to help the routing process in any way, we used an ontology available from the Protege's OWL ontology library (`http://protege.stanford.edu`). The MGED Ontology describes the domain of Microarray Gene Expression Data and it is composed of 213 concepts, but we considered only its taxonomy and not the properties of the classes.

The experiment was aimed at testing the system performance at different, increasing neighbourhood sizes. The hypothesis is that the system performance improves when we range from local knowledge to global knowledge, possibly highlighting the existence of a neighbourhood size that is the one that maximises the trade-off between performance and knowledge (and thus level of decentralisation). System performance was evaluated in terms of the *response time* spent to find the instances of the queried concept.

The semantic distance used was the *path length*. This method of evaluating semantic similarity has occasionally been criticised for not being an accurate measure [14], and for not really taking into account the semantics of the concepts in the ontology. We made this choice for efficiency reasons. The literature offers a number of similarity measures that can be used to evaluate similarity between concepts in an ontology, but none of them is widely accepted as being truly accurate, and most are computationally expensive and/or language sensitive. In our experiments we were interested in evaluating the response time of the system from the viewpoint of the routing mechanisms, and thus this choice of similarity measure permitted us to limit the burden of a heavy computation of the similarity measure, and to consider its cost negligible.

We executed 500 queries over 50 runs of the experiment. For each run we changed the agent that originated the routing process, and we queried 10 concepts changing for each the neighbourhood size by varying the path length. By varying the path length, we increased the number of concepts in the neighbourhood, thus increasing the capabilities of an agent (in terms of the concepts it can deal with), and ultimately ranging from a purely decentralised P2P system, to one where the whole ontology becomes a centralisation point. Figure 2 shows the trend in performance against the average neighbourhood size (over the 500 runs of the experiment). We can make some observations on the results of the experiments. The best results in terms of performance were obtained when the neighbourhood size was the whole ontology, or close to the whole ontology, even though this would imply that each agent maintains a copy of the whole ontology.

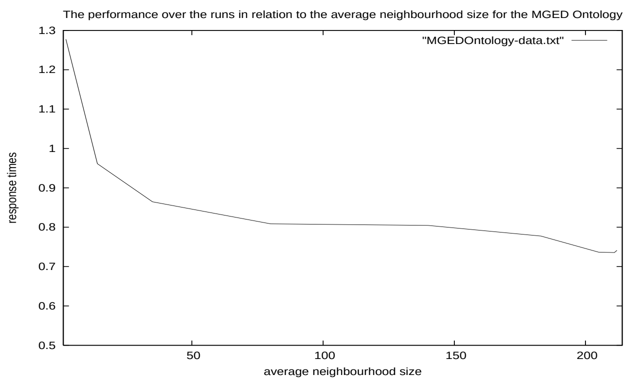

Figure 2. The performance of the system for the MGED Ontology

This would have significant implications on the system maintenance time, since each time an ontology is modified each router must update its copy of the ontology. However, a semantic neighbourhood that includes around 120 concepts (out of a possible 213) represents a good trade-off between agents knowledge and system performance (including maintenance time). These results lead us to believe that *hybrid* [19] peer to peer systems (that is, P2P systems in which some functionalities remain centralised) offer better prospects for the provision of search and retrieval services on the semantic web. Indeed, best results referred to global knowledge defeats the hypothesis that a pure peer-to-peer system could be used to search the semantic web. However, the experiments prove the existence of a region in the space of the possible neighbourhood sizes that balances knowledge vs. performance. This evidence provides us with an objective starting point for building clusters of similar peers.

5 Analytical evaluation of the workload

From the empirical evaluation of the system presented in the previous section it emerges that the SERSE works better when agents have global knowledge. However, global knowledge does not come for free, since the more knowledge the agents have, the more work it requires to maintain and organise the knowledge. Therefore, one critical issue to consider in order to evaluate the performance of SERSE is to provide an estimate of the system's workload necessary to maintain the semantic overlay network.

In general, maintenance operations can be carried out *on demand* and *routinely*. On demand activities are carried out every time the notification agent receives a notification of new content from the Notification Server. The system needs to re-organise the semantic overlay by updating the agent indices. On demand maintenance can follow two approaches, the *reactive* approach or the *proactive* approach. In the reactive approach, agents in the system regularly check their stored routing index against the current state of the system. Regular checks are performed by each agent by querying each of its neighbours for their indices, and updating the semantic distances in the index where appropriate.

The proactive approach is based on the assumption that each agent informs the neighbours of an update in the list of concepts it manages; the neighbours in turn modify their routing indices accordingly. In SERSE we have decided to opt for the proactive approach, because it permits us to reduce the number of messages during maintenance activities. Indeed, the number of content notifications that could trigger the maintenance process should be at least of an order of magnitude lower than the number of queries in the system. The maintenance process with the proactive approach can now be estimated depending on the activities involved in it:

• An agent $\mathrm{A_i}$ receives a content notification message, and thus updates its index. In particular, if the notification message concerns the addition of new digital content, $\mathrm{A_i}$ updates the instance URIs in its index, and informs its immediate neighbours $\mathrm{N^i_1}, \mathrm{N^i_2}, \ldots, \mathrm{N^i_m}$ of these changes. On the other hand, if the notification regards the less frequent, though possible event, of an ontology update, $\mathrm{A_i}$ modifies the list of concepts it manages in its index, then it also needs to contact its immediate neighbours in order to propagate the update.

• Agent $\mathrm{A_i}$ sends an `Update` message to all its m neighbours $\mathrm{N^i_1}, \mathrm{N^i_2}, \ldots, \mathrm{N^i_m}$, with the new index;

• Each neighbour $\mathrm{N^i_j}$ that receives the update message in turn updates its routing index by adding or removing concepts from those stored in relation to agent $\mathrm{A_i}$;

• If concepts are added, each of the neighbouring agents $\mathrm{N^i_j}$ needs to compute the semantic distance between the added concepts and each of the concept stored in its own index.

The considerations made above lead us to estimate the average of the time of the system due to proactive maintenance on demand caused by a single update cycle (e.g., the receipt of a content notification message) in a single agent as: $(n_{avg} \times c_{avg} \times t_{SD})$, where:

− n_{avg} is the average number of neighbours;

− c_{avg} is the average number of concepts managed by each agent;

− t_{SD} is the average time to compute the semantic distance between two concepts.

From this expression, it is clear that the more concepts are known to an agent, the higher the workload is. In estimating workload we are assuming that each agent has knowledge only of its immediate neighbours and that no lookahead mechanism is in place.

In addition to the maintenance on demand, it may be useful for the agents to periodically send random queries into the system in order to verify the stability of the semantic overlay, and whether it could benefit from an adjustment (such as grouping together two neighbourhoods because this would improve the performance). In routine maintenance, each agent sends out randomly messages that communicate the current state of the agent index, and enquire whether any of the receiving agents has similar knowledge. If this is the case, and this agent is not included in the neighbourhood, then the semantic overlay needs to be adjusted. The workload entailed by routine maintenance is estimated on the basis of the following steps:

• At regular intervals each agent $\mathrm{A_i}$ sends a random query into the system. This query includes the list of concepts indexed by $\mathrm{A_i}$;

• For each agent $\mathrm{B_i}$ receiving the query, $\mathrm{B_i}$ compares its indexed concepts with those received in the query;

• For each agent $\mathrm{B_i}$ receiving the query, $\mathrm{B_i}$ performs at most k semantic distance calculations, where k is the number of its indexed concepts;

Building on the considerations above, we can estimate the workload upper bound for a single update cycle in a single agent as: $(s/t) \times r \times c^2_{avg} \times t_{SD}$, where:

− s is the number of agents in the system;

− t is the interval between two queries;

− r is the number of agents receiving each message;

− c_{avg} is the average number of concepts managed by each agent;

− t_{SD} is the average time to compute the semantic distance between two concepts.

6 Related work

Approaches to the provision of search and retrieval services in the SW have been developed in disparate areas, from digital libraries to information retrieval. In the context of SERSE, we focus on peer-to-peer approaches for distributed knowledge management and intelligent information agents. In fact, SERSE can be seen as merging the two approaches, in that it creates a peer to peer system where the peers provide much of the functionalities characterising intelligent information agents. Therefore, much of the research carried out in our system is based on the efforts illustrated below.

Peer-to-peer systems are traditionally associated with file sharing and exchange applications, such as in Napster and Morpheus [3]. More recently, P2P systems have been used to reduce the complexity of distributed data and knowledge management applications [7, 10]. A typical example of such an application is EDUTELLA [12], a hybrid P2P architecture for sharing metadata, that implements an RDF-based metadata infrastructure for JXTA [2]. Nodes are organised into a set of thematic clusters, and a dedicated mediator performs semantic integration of source metadata in each cluster. Thematic clusters are obtained by super-peers that have semantic routing capabilities, however, there is little detail on the principles guiding the clustering of nodes and the impact of the cluster size on the system performance. Some other projects use super-peers to start the semantic routing process in the right direction.

Other approaches emphasise the use of semantics represented in ontologies. Among these there is the SWAP project [4]. In SWAP, each node is responsible for a single ontology: ontologies might represent different views of a same domain, multiple domains with overlapping concepts, or might be obtained by partitioning an upper level ontology. Knowledge sharing is obtained through ontology mapping and alignment, however mappings are not dynamically obtained.

Peer-to-Peer Semantic Web [5] builds on the research efforts carried out at the LSDIS Lab. This project aims at providing services to support distributed knowledge construction and knowledge sharing. Here the emphasis is on ontology services, such as the creation and advertisement of new ontologies and the discovery of ontology interrelationship over a P2P infrastructure, and ontology driven content retrieval. However, the P2P architecture is not explicitly defined.

There is a vast literature on intelligent information agents that is extremely relevant to SERSE, and especially the characterising features that constitute the notion of agency [18]. From this viewpoint SERSE can be classified as a cooperative information agent, such as RETSINA [16], InfoSleuth [9] and OBSERVER [13]. RETSINA is a matchmaker based information system where collaborative task execution is achieved through matching service providers and requesters over the web (and more recently, over the SW).

InfoSleuth explicitly deals and reconciles multiple ontologies by means of specialised ontology agents that collectively maintain a knowledge base of the different ontologies used to specify requests, and return ontology information as requested. Finally, OBSERVER dynamically finds mappings between partially overlapping ontologies, therefore providing agents with a way to dynamically share knowledge. Many of these features are being studied for use in the second prototype of SERSE.

7 Conclusions

The paper has presented SERSE, a system that offers search and retrieval services on the SW. SERSE builds on peer to peer, ontologies, and multi-agent technology in order to deal with the complexity of searching for digital content. In SERSE, agents have knowledge of the ontological definitions used to annotate digital content and communicate on a peer to peer basis. Routing is determined by semantic proximity of the ontological definitions that are known to the agents (semantic neighbourhood).

We have illustrated and motivated the design of SERSE, and we have presented and evaluated its implementation. The evaluation has aimed at proving the hypothesis that in this kind of systems the neighbourhood size affects the performance of the system in different ways: the more ontological definitions are known to the agents, and more easily the query can be routed to the most relevant peer. But, more ontological knowledge causes also a higher maintenance effort and a higher workload. Our experiments prove that there is a region in the space of the neighbourhood sizes that represent a good trade-off between number of concepts known to a router agent and the system performance. We have also provided an analytical estimate of the maintenance workload of such system that shows that the maintenance costs are at worst quadratic, for regular maintenance activities. Therefore, we can reorganise our network of agents objectively on the basis of this findings. This means that the network does not need to have a global centralisation point but can be organised in a number of clusters, each offering different ontological expertise and whose size is determined by the number of concepts composing the ontology.

REFERENCES

[1] OWL, web ontology language. `http://www.w3.org`.

[2] Project JXTA. `http://www.jxta.org`.

[3] Project Morpheus. `http://musiccity.com`.

[4] Semantic web and peer-to-peer. `http://swap.semanticweb.org`.

[5] M. Arumugam, A. Sheth, and I.B. Arpinar, 'Towards peer-to-peer semantic web: A distributed environment for sharing semantic knowledge on the web', in *Proc. of the International World Wide Web Conference 2002 (WWW2002)*, Hawaii, USA, (May 2002).

[6] F. Bellifemine *et al*, 'JADE a white paper', *EXP In search of innovation*, **3**(3), (September 2003).

[7] M. Ehrig *et al*, 'The SWAP data and metadata model for semantics-based peer-to-peer systems', in *Proceedings of MATES-2003: First German conference on multiagent technologies*, number 2831 in LNAI, (2003).

[8] M. Schlosser *et al*, 'A scalable and ontology-based p2p infrastructure for semantic web services', in *Proceedings of the Second IEEE International Conference on Peer-to-Peer Computing (P2P2002)*, (2002).

[9] R.J. Bayardo *et al*, 'InfoSleuth: Agent-based semantic integration of information in open and dynamic environments', in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, volume 26,2, pp. 195–206, New York, (1997). ACM Press.

[10] S. Castano *et al*, 'Ontology-addressable contents in p2p networks', in *Proc. of WWW'03 1st SemPGRID Workshop*, (2003).

[11] V.R. Benjamins *et al*, 'Esperonto application: Service provision of semantic annotation, aggregation, indexing, and routing of textual, multimedia and multilingual web content', in *Proc. of WIAMSI'03*, (2003).

[12] W. Nejdl *et al*, 'EDUTELLA: A p2p networking infrastructure based on rdf', in *Proc. of the International World Wide Web Conference 2002 (WWW2002)*, pp. 604–615, Hawaii, USA, (2002).

[13] E. Mena, A. Illarramendi, V. Kashyap, and A. Sheth, 'OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies', *Distributed and Parallel databases. An international journal*, **8**(2), 223–271, (April 2000).

[14] P. Resnik, 'Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language', *Journal of Artificial Intelligence Research*, **11**, 95–130, (1999).

[15] R. Studer, V.R. Benjamins, and D. Fensel, 'Knowledge engineering, principles and methods', *Data and Knowledge Engineering*, **25**(1-2), 161–197, (1998).

[16] K. Sycara, M. Klusch, S. Widoff, and J. Lu, 'Dynamic service matchmaking among agents in open information systems', *ACM SIGMOD Record. Special Issue on semantic interoperability in global information systems*, (1998).

[17] T.Berners-Lee, J. Hendler, and O. Lassila, 'The semantic web', *Scientific American*, (284), 35–43, (May 2001).

[18] M. Wooldridge and N.R. Jennings, 'Intelligent agents: Theory and practice', *Knowledge engineering review*, **10**(2), 115–152, (1995).

[19] B. Yang and H. Garcia-Molina, 'Comparing hybrid peer-to-peer systems', in *The VLDB Journal*, pp. 561–570, (sep 2001).