

A Semantic Information Retrieval Advertisement and Policy Based System for a P2P Network

Giovanna Guerrini¹, Viviana Mascardi², and Marco Mesiti³

¹ DI, Università di Pisa, Italy

² DISI, Università di Genova, Italy

³ DICO, Università di Milano, Italy

{guerrini,mascardi,mesiti}@disi.unige.it

Abstract. In this paper we propose a semantic based P2P system that incorporates peer *sharing policies*, which allow a peer to state, for each of the concepts it deals with, the conditions under which it is available to process requests related to that concept. The semantic routing approach, based on advertisements and peer behavior in answering previous requests, takes also into account sharing policies.

1 Introduction

In most P2P architectures, query answering is based on flooding algorithms, that propagate requests from one node to another till a given number of nodes has been reached. Typical routing protocols are based on distributed hash tables for improving routing efficiency. However, these indexes support a keyword based search rather than a *semantic* search. The advantages of a semantic routing, that keeps into account the semantics of data requests and shared resources, are well-accepted in terms of search effectiveness.

Whatever strategy is adopted for query routing, most existing systems are based on the assumption that, when connected to the network, peers are unconditionally available to share their resources with anyone interested in them. This assumption is, however, not reasonable in many contexts and for many reasons. Peers may wish to set some *sharing policies* depending on different factors such as temporal conditions (e.g., the time at which the request is received), internal state and connection conditions (e.g., the workload when the request is received), and conditions on the characteristics of the peer submitting the request (e.g., its membership to a group), that can typically be expressed through *credentials* [22]. A peer can thus customize its behavior by tailoring the general system behavior to its specific sharing needs and constraints.

In this paper, we propose a semantic routing approach in a P2P system that allows single peers to enforce their own sharing policies. The resources made available to the system may deal with many different subjects, or themes, and peers may register to one or more thematic groups. Relevant information retrieval is achieved through the use of a thematic global ontology (TGO) for each theme dealt with by the system; the TGO associates a semantics with the resources to be shared within the thematic group. All the peers that register to a thematic group share the TGO of the group. For the sake of clarity in the paper we will focus on a system with a single thematic group. Each peer associates instances of its local resource base with concepts of the TGO that better describe them. Peers actively push their expertises by sending *advertisements*, containing

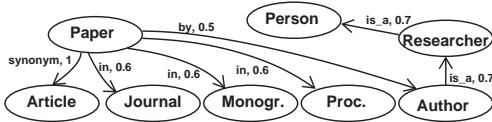


Fig. 1. A portion of a *TGO* for a computer science thematic group

the concepts of the *TGO* that better describe the resources they share. Semantic query routing is guided both by the advertised peer expertises and by the *relevance* of peer answers to previous requests. This relevance is quantified in a relevance degree associated with each concept of the *TGO*, which is updated each time a peer gets an answer to a request involving that concept.

This approach, as a novel feature, integrates in this context the sharing policy and credential notions, thus allowing a more flexible resource sharing mechanism. To allow a peer to enforce different policies for different resources, different sharing policies can be associated with different ontology concepts that the peer's user deal with. Policies associated with concepts of the ontology are actively pushed by the peer together with advertisements, so that other peers can avoid sending and forwarding requests that will not be processed. Thus, sharing policies also affect the routing algorithm.

In the remainder of the paper, Section 2 introduces basic concepts and Section 3 discusses the peer architecture and the system main functionalities. Section 4 compares our approach with related ones and concludes. For space constraints details of the developed approach can be found in [9].

2 Basic Concepts

In this section we introduce the basic notions our approach relies on. An XML format has been chosen for the representation and exchange of these components. The XML Schemas stating the exact format of each component can be found in [9].

Ontologies. In our system, all the peers that registered to a thematic group share the thematic global ontology of that group, *TGO*. *TGO* is a directed weighted graph, where nodes (V) represent concepts, arcs (E) represent relations between concepts (including the *is_a* relation), and weights, ranging in $[0, 1]$, represent *how similar* two related concepts are. Each peer P is characterized by a set of concepts of interest CoI such that $CoI \subseteq V$. For example, a portion of the *TGO* describing the computer science publication domain is shown in Fig. 1. The CoI of a peer $mike$ in this domain might be, for instance, $CoI_{mike} = \{Article, Proceedings\}$. A function Sim_C will be employed to measure the semantic distance between two sets of concepts. This function uses an auxiliary function sim_c for evaluating the similarity between a set of concepts and a single concept of the ontology. Both sim_c and Sim_C refer to the *TGO* for knowing the weights of the relations among concepts. Details of these functions are in [7].

Credentials and Policies. Credentials are a means to control resource access and to condition resource sharing to certain peer characteristics. A credential $c = (n, \{(p_1, v_1), \dots, (p_k, v_k)\})$ is a named set properties, that is, name-value pairs. The XML document corresponding to a credential is shown in Fig. 2(a). The use of credentials asserting properties of individuals raises issues related to certification of properties, their authen-

```

<Credential name="DISI@UnigeAffiliation">
  <Property name="FirstName" value="Sonia"/><Property name="LastName" value="Pini"/>
  <Property name="Position" value="Researcher"/> <Property name="Office" value="5"/>
</Credential>                                (a)

<Policy id="1">
  <TempConstDef name="TC1">
    <IntervalExpr name="sinceJan1st"> <begin> 01/01/05:00 </begin> </IntervalExpr>
    <PeriodicTimeExpr name="9to13ofWorkingDays"> <StartTimeExpr>
      <Week> all </Week>
      <DaySet><Day>2</Day><Day>3</Day><Day>4</Day><Day>5</Day><Day>6</Day></DaySet>
      <Hour> 10 </Hour> <DurationExpr> <Hours> 4 </Hours> </DurationExpr>
    </StartTimeExpr></PeriodicTimeExpr>
  </TempConstDef>
  <InternalCondition type="state" prop="PendingRequests" op="LE" value="15"/>
  <InternalCondition type="state" prop="CPUIdleTime" op="L" value="50"/>
  <CertCondition prop="Position" op="EQ" value="Researcher"/>
</Policy>                                    (b)

```

Fig. 2. (a) An example of credential and (b) an example of policy

ticity and verification. These issues are beyond the scope of this paper, thus, in our system, we assume the presence of a peer that releases and certifies credentials of peers joining a thematic group.

Peers restrict their availability to share resources through *sharing policies*. Each policy is characterized by a temporal condition stating the time instants the policy is enabled. Temporal conditions are expressed, according to [3,16], as a $\langle[\begin{begin}, \end{end}], P\rangle$ pair, where \begin{begin} , \end{end} are time instants denoting the endpoints of a time interval and P is a periodic expression of the form $P = \sum_{i=1}^n O_i.G_i \triangleright r.G_d$ where G_d , G_1, \dots, G_n are time granularities or calendars, such that G_d is finer than G_n , for $i = 2, \dots, n$, G_i is finer than G_{i-1} , $O_1 = \text{all}$, $O_i \in 2^{\mathbb{N}} \cup \{\text{all}\}$ and $r \in \mathbb{N}$. Suppose for example we wish to represent the period between 9.00 and 13.00 of working days, starting from January 1, 2005 at 00. The corresponding temporal condition is: $[2005/01/01 : 00, \infty], \text{all}.Weeks + \{2, \dots, 6\}.Days + 10.Hours \triangleright 4.Hours$.

A policy is 4-tuple (id, tC, iC, cC) , where tC is a temporal condition and iC, cC denote a conjunction of internal state/connection and credential conditions, respectively. Internal state/connection and credential conditions are of the form $(prop \ op \ value)$ where op is comparison operator in $\{\leq, \geq, <, >, =\}$. Suppose, for instance, that a peer wishes to share resources in the temporal period previously presented, but only when the pending requests are less than 15, the CPU idle time is below the 50% and the requester is a researcher. The XML representation of this policy is shown in Fig. 2(b).

A policy $p = (id, tC, iC, \{(n_1 \ op_1 \ u_1), \dots, (n_m \ op_m \ u_m)\})$ is satisfied by a credential $c = (n, \{(p_1, v_1), \dots, (p_k, v_k)\})$ and a peer P if the current time instant belongs to set of time instants described by tC , $\forall i \in [1, m] \exists j \in [1, k] (n_i = p_j) \wedge (u_j \ op_i \ v_i)$, and P internal and network property values meet iC . For instance, consider a peer P_1 that receives on Monday, July 4, 2005 at 9:30 a data request with the credential of Fig. 2(a). If P_1 enforces the policy in Fig. 2(b), does not have pending requests, and is not performing any computation, then the policy is satisfied.

Advertisement and Data Request Messages. Messages exchanged among peers can be advertisements, data requests, and answer messages. Advertisement and data request messages that are forwarded to other peers are characterized by *Time To Live (TTL)* and

```

<Advs id="AdvChi2" TTL="5" BS="0.8" PeerId = "P457" TimeSent = "27/06/05:14:13">
  <Concept name="Paper"> </Concept>
  <Policies name="Chiara"> <Policy id="1" see Fig. 2(b)> </Policy> </Policies>
</Advs>                                (a)

<DataRequest id="QD2" TTL="3" BS="1" PeerId = "P473" TimeSent = "27/06/05:14:13">
  <Query> <QueryPred op="EQ" value="Cardelli">
    <PathExpression> <Concept name="Paper">
      <Property name="by"/> <Property name="name"/>
    </Concept> </PathExpression> </QueryPred>
    <QueryPred op="EQ" value="1980">
      <PathExpression> <Concept name="Paper">
        <Property name="year"/>
      </Concept> </PathExpression> </QueryPred> </Query>
    <Credential> see Fig. 2(a) </Credential>
  </DataRequest>                            (b)

```

Fig. 3. (a) An example of advertisement and (b) an example of data request

Broad Search (BS) values, stating the maximal distance between the message sender and the last receiver, and the fraction of peers to forward the message to, respectively. Moreover, each message is characterized by an Id, by the Id of the sender peer, and by the time of the sending.

Advertisements are employed to disseminate information on expertises and sharing policies of the peer's user. An advertisement consists in concepts in the *TGO* that are related to resources the peer's user is willing to share, and a list of policies pL stating the sharing policies for resources related to these concepts. In checking satisfaction, policies in the list are considered in order, and the iC and cC conditions are checked for the first policy in the list for which *now* belongs to the set of instants described by tC . For example, Fig. 3(a) shows the XML document corresponding to the advertisement message sent by peer Chiara that shares papers under the previously presented policy.

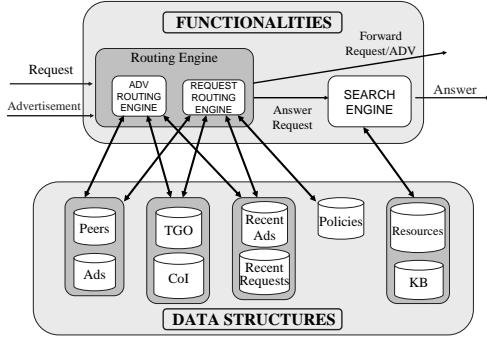
Data requests are characterized by a concept and a set of credentials. The concept belongs to the *TGO*, and may be optionally qualified with a number of predicates, interpreted as a conjunction, that allow to filter the resources of interest. Data request languages more sophisticated than ours can easily be accommodated in our framework. For example, Fig. 3(b) shows the XML document corresponding to the data request message sent by peer Sonia looking for papers published by Cardelli in 1980. The credential of Fig. 2(a) is attached to the request.

3 Architecture and Functionalities

In this section we describe the main functionalities of the system relying on the architecture graphically depicted in Fig. 4. More details can be found in [9].

Peer Registration

When a new peer wishes to register to a group of the P2P network, it connects to the “special peer”. The peer Id is inserted in the list of peers known by the special peer, and the registering peer uses this list to initialize its local *Peers* structure (initially, with null global relevance and no concept-specific relevances associated with each peer). Then, a graphical interface showing the *TGO* is presented to the peer's user who can browse



Component	Description
<i>TGO & CoI</i>	the thematic global ontology and the concepts of interest. Some indexes are kept over the TGO allowing, given a concept, to directly retrieve its more specific/general concepts and the set of its properties.
<i>Knowledge Base</i>	the set of ontology instances together with their property values. It is handled and indexed through classical database technology.
<i>Resources</i>	the set of resources the peer is willing to share with other peers. Each resource is linked by an instance in KB.
<i>Peers</i>	information on the peers the peer is aware of: peer Id, global relevance degree and concept-specific relevance degrees. Some auxiliary structures allow a direct access to the relevance value of a concept-peer pair and to efficiently get the peers ordered by global relevance.
<i>Ads</i>	information on the advertisements received by other peers: sender peer Id, advertised concept set, sharing policies for those concepts, indexed to get a direct access to the sharing policies of a concept-peer pair.
<i>Recent Requests/ Ads</i>	information on the data requests (the advertisements, respectively) the peer recently received: Data request/Adv Id, sending time, sender peer Id, indexed on the message Id.
<i>Policies</i>	local peer sharing policies, indexed on the concepts they refer to.

Fig. 4. Peer architecture

the *TGO*, read the textual explanation associated with each concept, identify her concepts of interest (*CoI*, see Section 2), and realize the concepts that better describe the local resources she wishes to share. The *TGO* is then copied locally in the peer's data structures. Now the peer's user can, when she wishes, populate the peer's local knowledge and resource bases, as well as the *Policies* structure with the sharing policies to be enforced. The peer is now ready for sending advertisements and data requests, as discussed in what follows.

Advertisement Handling

- *Sending*. A peer wishing to advertise its expertises simply sends advertisement messages, as described in Section 2, to the peers it is aware of (stored in the *Peers* structure).
- *Receiving*. A peer receiving an advertisement message first of all checks whether it has already received it looking at the *RecentAds* structure. If so, it simply discards it. Otherwise, the message is inserted in the *RecentAds* and *Ads* structures. If the sender peer was not known, it is also inserted in the *Peers* structure (with null global relevance e no concept-specific relevance). Note that all the received advertisements are stored. A graphical interface, however, allows the peer's user to browse the *Ads* advertisement database, ordered either by sending time or similarity of the advertised topics with the peer concepts of interest in its *CoI*, computed through Sim_C , and delete some of them.
- *Forward*. A received advertisement is forwarded to a set of known peers according to the *TTL* and *BS* components of the message. Specifically, if *TTL* is greater than 0, the message is forwarded to *BS* peers with the *TTL* value decremented by 1. The peers to forward the message to are chosen among the known peers in the *Peers* structure. A

fraction is randomly chosen, whereas the others are the ones whose sets of advertised concepts (as stored in Ads) are most similar to the concepts in the advertisement to be forwarded, according to the similarity function Sim_C .

Data Request Handling

- *Peer Relevance.* When a peer gets an answer to one of its requests, it updates the information in the $Peers$ structure related to the relevance of the sending peer to keep into account the new answer. The peer receiving some resources as answers to a data request evaluates them by stating which ones are relevant (and thus are accepted), and which others are not (and thus are discarded). A special *bonus* can be explicitly assigned for extremely relevant answers, through a parameter β whose default is 0. According to the evaluation of the peer P' getting a set of resources as answer to a request Q , the relevance degree got by a peer P sending the answer, related to a concept c belonging to the set of concepts appearing in Q , is a value in $[0, 1]$ computed as: $Relevance(P, c, Q) = \frac{\text{accepted_resources}}{\text{received_resources}} + \beta$. The $Relevance(P, c, Q)$ value contributes to the previous relevance of peer P and concept c , named $rel_{P,c}$, in the $Peers$ structure of peer P' , if such an entry was there. Otherwise a new entry for peer P , concept c and this value is inserted. The global relevance of a peer rel_P is the sum of the concept-related relevances $rel_{P,c}$ of the peer and is thus updated accordingly. The relevance of a peer P with respect to a set of concepts C is then obtained as $Rel(P, C) = \sum_{c \in C} rel_{P,c} + \sum_{c \in C, c \preceq c'} \alpha^d \cdot rel_{P,c'}$ where $\alpha \in [0, 1]$, \preceq denotes the *is_a* relation in the ontology, and d is the distance between c and c' in the *is_a* hierarchy of the ontology. The basic principles in using relevance, inherited from [19], are indeed the following: (i) a data request is submitted to a peer that answered well to previous requests on the same concepts; (ii) a peer that answered well on a specific concept, is likely to be quite knowledgeable on more general concepts related to the same topic; (iii) a peer that answered well to previous requests on several different concepts, is likely to be well-informed in general (on any concept).
- *Sending.* When a peer wishes to submit a data request Q to the system, it may include any of its credentials in Q . Then, it selects the peers to send the request to, taking into account the advertisements it received and the peer relevance, for the concepts the data request involves. A list of peers is computed by ordering the set of peers in $Peers$ according to their $Rel(P, C)$ value, being C the set of concepts involved in Q . This list is pruned by deleting the peers for which an advertisement has been stored for the involved concepts with associated policies whose credential conditions are not met by credentials in Q , obtaining a list L_r . A similar list L_g is obtained by taking into account the global relevance of the peer rel_P . A last list L_a is computed by ordering the peers in Ads according to the similarity of the advertised concepts and the concepts in data request Q , computed through function Sim_C , including only the peers for which the credential condition of an associated policy is met by a credential in Q . The request is sent firstly to the peers in L_r that also belong to L_a , then to other peers in L_r , then to other peers in L_a , then to peers in L_g not considered so far, till the desired number of peers is reached.
- *Receiving.* A peer receiving a data request Q first of all checks whether it has al-

ready received it looking at the *RecentRequests* structure. If so, it simply discards it. Otherwise, Q is inserted in the *RecentRequests* structure and, if the sender peer was not known, it is also inserted in the *Peers* structure (with null global relevance and no concept-specific relevance). Then, the peer checks whether it can answer Q , checking the satisfaction of its own policies associated with the concepts in Q w.r.t. the current time, its current state, and the credentials in Q . If so, its own resources satisfying the data request conditions are sent to the requesting peer. In any case, the request is then forwarded to other peers, following the same behavior adopted for advertisement forwarding, for what concerns the *TTL* and *BS* values and the choice of forwarding to a fraction of randomly chosen peers. The other peers to forward the request to are selected with the same list-based approach discussed above for request sending.

4 Concluding Remarks

We have compared our system with FreeNet (freenet.sourceforge.net), Edutella [15,14], KEEEx [4], Napster (www.napster.com), Piazza [12], the Trusted Computing P2P (TC-P2P) Architecture [18], and SWAPSTER [11,20], along the three features that characterize our proposal: *(i)* use of ontologies to answer data requests, and to better route them; *(ii)* use of advertisements to push information about a peer's expertise; *(iii)* use of sharing policies to allow a controlled flexible access to the peer's resources.

The choice of these systems has been driven by the will of considering a spectrum of heterogeneous proposals, where heterogeneity involves both the motivation and the nature of the proposal, and the intended application domain. Our comparison shows that very few systems address all the three aspects that characterize our proposal in a deep and exhaustive way, although most of them implement mechanisms to face at least two of them (see [9] for a full account of the results of our comparison). The originality of our proposal lies in addressing *all* of them into an integrated P2P system.

The system that is closer to ours is SWAPSTER, that has been used to implement two concrete applications: Bibster [11], and Xarop [20]; the developers of SWAPSTER also investigated several query routing strategies by simulation experiments.

Although it is not a P2P system, the framework developed inside the SEWASIE European project [2] shares some similarities with our proposal as far as the management of ontologies is concerned. In fact, in SEWASIE each SINode (a mediator-based system) provides a global virtual view (GVV) of the information sources managed within it, which may resemble the *TGO* of our proposal, and Brokering Agents integrate several GVVs from different SINodes into a Brokering Agent Ontology. In our proposal, *TGO* integration has not been investigated yet, but the adoption of a Brokering Agent Ontology suggested by SEWASIE could be a feasible direction to follow.

Most (although not all) of the systems that we have considered in our comparison have been tested on real applications. Although the implementation of our system is still to be completed, we have already implemented many crucial components such as those for evaluating the similarity between concepts, developed using Jena (<http://jena.sourceforge.net>). The main direction of our future work is thus completing the implementation, in order to release a first version, based on JXTA (<http://www.jxta.org>), in few months.

Acknowledgements. We acknowledge P. Bouquet, I. Clarke, E. Franconi, W. Siberski, and S. Staab for their precious advices in drawing the comparison with related work. We also thank C. Casanova for contributing with her Master's Thesis to the design of our system.

References

1. T. Andreasen, et al. On Ontology-based Querying. In *Proc. of the IJCAI Workshop on Ontologies and Distributed Systems*, 2003.
2. S. Bergamaschi, et al. The SEWASIE EU IST project. *SIG SEMIS Bulletin*, 2(1), 2005.
3. E. Bertino, et al. An Access Control Model Supporting Periodicity Constraints and Temporal Reasoning. *ACM Transactions on Information and Systems*, 23(3):231–285, 1998.
4. M. Bonifacio, et al. KEEX: A Peer-to-Peer Solution for Distributed Knowledge Management. In *Proc. of Int'l Symposium on Knowledge Management*, 2004.
5. S. Castano, et al. Semantic Information Interoperability in Open Networked Systems. In *Proc. Conf. on Semantics of a Networked World*, 2004.
6. R. Chen and W. Yeager. Poblano, a Distributed Trust Model for Peer-to-Peer Networks. TR, Sun Microsystems, 2001.
7. V. Cordì, et al. Designing and Implementing an Ontology-Based Distance between Sets of Concepts. TR, DISI TR-05-11, Uni. di Genova, 2005.
8. E. Franconi et al. A Robust Logical and Computational Characterisation of Peer-to-Peer Database Systems. In *Proc. of the VLDB Workshop DBISP2P*, 2003.
9. G. Guerrini, V. Mascardi, and M. Mesiti. A Semantic Information Retrieval Advertisement and Policy Based System for a P2P Network. TR, DISI TR-05-10, Uni. di Genova, 2005.
10. P. Haase, et al. Peer Selection in Peer-to-Peer Networks with Semantic Topologies. In *Proc. of Conf. on Semantics of a Networked World: Semantics for Grid Databases*, 108–125, 2004.
11. P. Haase et al. Bibster – A Semantics-Based Bibliographic Peer-to-Peer System. In *Proc. of Int'l Semantic Web Conf.*, 122–136, 2004.
12. A. Halevy, et al. The Piazza Peer Data Management System. *IEEE Transactions on Knowledge and Data Engineering*, 16(7), 2004.
13. W. Nejdl, et al. Ontology-Based Policy Specification and Management. In *Proc. of European Conf. on Semantic Web*, 290–302, 2005.
14. W. Nejdl, et al. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. In *Proc. of Int'l WWW Conf.*, 2003.
15. W. Nejdl et al. Edutella: A P2P networking infrastructure based on RDF. In *Proc. of Int'l WWW Conf.*, 2002.
16. M. Nieuwelt and J. Stevenne. An Efficient Symbolic Representation of Periodic Time. In *Proc. of Int'l Conf. on Information and Knowledge Management*, 1992.
17. R. Rada, et al. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989.
18. R. Sandhu, et al. Peer-to-Peer Access Control Architecture Using Trusted Computing Technology. In *Proc. of Symposium on Access Control Models and Technologies*, 2005.
19. C. Tempich, et al. Remindin': Semantic Query Routing in Peer-to-peer Networks Based on Social Metaphors. In *Proc. of Int'l Conf. on WWW*, 640–649, 2004.
20. C. Tempich et al. XAROP: A Midterm Report in Introducing a Decentralized Semantics-Based Knowledge Sharing Application. In *PAKM'04 Conf.*, 259–270, 2004.
21. G. Toni et al. Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder. In *Proc. of the Int'l Semantic Web Conf.*, 419–437, 2005.
22. M. Winslett, et al. Using Digital Credentials on the World Wide Web. *Journal of Computer Security*, 5, 1997.