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

Giovanna Guerrini[1], Viviana Mascardi[2], Marco Mesiti[3]

[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. Semantic routing of queries in P2P systems has recently gained increasing attention, since it allows to considerably improve effectiveness and accuracy of data retrieval. A common assumption of existing systems, however, is that a peer, when it is connected to the network, is unconditionally available to share its resources with anyone interested in them. In this paper we take a more flexible approach, proposing an enhanced 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 share resources and process requests related to that concept. Such conditions may include conditions on the credentials of the peer submitting the request, as well as temporal conditions and conditions on the internal and network load of the peer. The semantic routing approach, based on advertisements and peer behavior in answering previous requests, takes into account peer sharing policies as well.

1 Introduction

An important functionality of P2P systems is information retrieval. 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. To select the peers that most likely contain the requested information, routing protocols have been proposed. 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 both 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* [19]. 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 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.

The proposed 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 describe the peer resources. For instance, a computer science professor may share computer graphics articles with colleagues of any research institute without time or workload restrictions, she may share exercise solutions of its course with students of her university only during working hours, and she may share introductory computer science resources with anyone interested, only at night or during the weekends, and only if the workload is light. 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.

A peer[1] credential [19] consists of a set of property-value pairs allowing the peer to certify its membership to a group and its right to share a certain resource. For instance, a credential can demonstrate the affiliation to a computer science department or the attendance to a course. Credentials can be attached to data requests, and thus allow the system to consider the characteristics of the peer submitting a request in the release of a resource. The use of credentials asserting properties of individuals raises issues related to certification of properties, their authenticity 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 a peer joining a thematic group. However, more sophisticated solutions based on independent third party credential authorities or trust negotiation systems can be easily integrated.

The paper thus proposes a semantic information retrieval advertisement and policy based system for a P2P network. The system is based on a pure P2P architecture in which peers advertise their expertises, enforce sharing policies, and check credentials

[1] Note that credentials are actually associated with users. For the sake of simplicity in the paper we identify the notions of user and peer.

associated with requests submitted by other peers. The only centralized activity in the system is peer registration, in which peers interact with a "special peer" keeping track of all the registered peers. The special peer provides to the new peer (as well as to any peer later requesting them) the list of registered peers as well as the TGO, represented as an OWL file. During the normal system behavior, peers can send and receive advertisement and data request messages. Advertisements contain concepts of the TGO the sending peer deals with, together with the associated sharing policies. Advertisements can be stored locally by the receiving peer and forwarded to other peers. Data requests contain some concepts of the ontology, possibly some filters on properties of these concepts, and possibly credentials of the sending peers. A peer receiving a request processes it, if it shares resources satisfying the request filters and if it is compliant with its sharing policies. The relevant answers are sent directly to the requesting peer. The data request is also forwarded to other peers, according to a semantic routing. Specifically, the peers to forward the request to are selected according to the principle of choosing the peers that most likely will provide an answer to the request. Peers that have sent an advertisement related to the concepts appearing in the request, whose associated policies are met by the credentials in the request, and that have an high relevance degree associated with concepts in the request are privileged. The request is however forwarded as well to a number of randomly chosen peers to avoid the creation of "closed subcommunities" in the network. Standard limitations (e.g., time to live) on the number of node hops are enforced.

Novel features of the proposed system thus consist in the credential and policy mechanisms to support a more flexible and customized resource sharing, integrated in a semantic routing approach relying both on explicitly advertised peer expertises and the recording of the results of previous peer interactions. Both the sharing policies and the relevance degrees are handled separately for different concepts of the ontology, allowing to differentiate peer behavior on different concepts.

In the remainder of the paper, Section 2 introduces basic concepts and Section 3 sketches the peer architecture. Section 4 discusses the system main functionalities. Section 5 compares our approach with related ones, and Section 6 concludes.

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 [6].

2.1 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_P such that $CoI \subseteq V$.

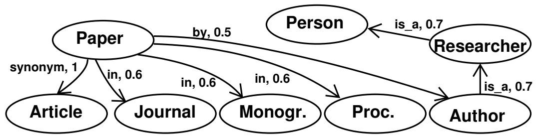

Fig. 1. A portion of a TGO for a computer science thematic group

Example 1. 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 example, $CoI_{mike} = \{Article, Proceedings\}$. □

A function Sim_C for evaluating the similarity among the concepts in two sets 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 of them refer to the TGO for knowing the weights of the relations among concepts.

As usual (see [14] for the seminal contribution, but also [1,5,7] for more recent applications), the sim_c function relies on Dijkstra's shortest path algorithm, but the weights of the paths are obtained by multiplying the weights associated with the labels of the arcs of the path, and the best path is the one with the highest value. The graph is navigated as if it were undirected (namely, as if for each relation, the inverse one, with the same weight, existed), although, from a conceptual point of view, it is a directed graph. Given two sets of concepts C_1, C_2, and the thematic global ontology TGO,

$$Sim_C(C_1, C_2, TGO) = \frac{\sum_{c \in C_2} sim_c(c, C_1, TGO)}{\mid C_2 \mid}$$

Example 2. Given $C = \{Journal, Paper\}$ and $CoI_{mike} = \{Proceedings, Article\}$, $sim_c(Article, C, TGO) = 1$, $sim_c(Proceedings, C, TGO) = 0.6$, and $Sim_C(CoI_{mike}, C, TGO) = 0.8$. □

As far as the implementation is concerned, we are implementing the sim_c and sim_C functions using the Jena semantic web framework (`http://jena.sourceforge.net/`), and we are adopting OWL as the format for representing the ontology.

2.2 Credentials and Policies

Credentials are a means to control resource access and to condition resource sharing to certain peer characteristics. A credential $c = (n, \{(p_1, v_1), \ldots, (p_k, v_k)\})$ is a named set properties, that is, name-value pairs. The XML document corresponding to a credential is shown in Fig. 2(a).

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 [3,13], as a $\langle [begin, end], P \rangle$ pair, where $begin$, 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 $G_d,\ G_1, \ldots, G_n$ are time granularities or calendars, such that G_d is finer than G_n, for $i = 2, \ldots, n$, G_i is finer than G_{i-1}, $O_1 = all$, $O_i \in 2^{\mathbb{N}} \cup \{all\}$ and $r \in \mathbb{N}$.

```
<Credential name="DISI@UnigeAffiliation">
 <Property name="FirstName" value="Davide"/>
 <Property name="LastName" value="Ancona"/>
 <Property name="Position" value="Researcher"/>
 <Property name="Office" value="102"/>
</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

Example 3. Suppose 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: $[2005/01/01:00, \infty], all.Weeks + \{2, \ldots, 6\}_{.}Days + 10.Hours \triangleright 4.Hours.$ ☐

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 $\{\leq, \geq, <, >, =\}$.

Example 4. Suppose a peer wishes to share resources in the temporal period described by the condition in Example 3, 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 $p = (id, tC, iC, \{(n_1\ op_1\ u_1), \ldots, (n_m\ op_m\ u_m)\})$ is satisfied by a credential $c = (n, \{(p_1, v_1), \ldots, (p_k, v_k\})$ 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.

Our policies are simpler than those supported by general-purpose policy languages such as [18]. Our policies are indeed specialized *sharing* policies that should be easily exchanged among peers and whose satisfaction should be extremely efficient to test. The credential condition of our policy notion is quite similar to the one discussed in [10], thus an interesting direction in the implementation could be that of integrating the PolicyTab tool in our system.

2.3 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 *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, the sender peer id, and the time of the sending. The *core* of the message is an advertisement, a data request, a set of resources, in the three kinds of messages.

Advertisements are employed to divulgate information on peer expertises and sharing policies. An advertisement consists in a set of concepts in the TGO related to which the peer has some resources it 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 wich now belongs to the set of instants described by tC.

Example 5. Fig. 3(a) shows the XML document corresponding to the advertisement message sent by peer Chiara willing to share papers under the policy of Example 4 and with no load conditions during the weekend. □

Data requests are composed by a request and a set of credentials. The request is a concept in the TGO with which resources can be associated optionally qualified with a number of predicates, interpreted as a conjunction, on the same concept, allowing to filter the resources of interest. More complex data request languages can easily be accommodated in our framework.

Example 6. Fig. 3(b) shows the XML document corresponding to the data request message sent by peer Davide looking for papers published by Cardelli in 1980. The credential of Fig. 2(a) is attached to the request. □

3 Peer Architecture

Fig. 4 graphically depicts the peer architecture. Each peer is characterized by the following functional components. A *Routing Engine* with two subcomponents, the *Adv Routing Engine*, responsible of sending and forwarding advertisements, and the *Request Routing Engine*, responsible of sending and forwarding data requests. A *Search Engine*, responsible of answering data requests locally to the peer. The peer data structures are:

- *TGO & CoI:* the thematic global ontology and the concepts of interest as described in Section 2. 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.
- *Knowledge Base (KB):* the set of ontology instances together with their property values. It is handled and indexed through classical database technology.
- *Resources:* the set of resources the peer is willing to share with other peers. Each resource is linked by an instance of the knowledge base.

```xml
<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>
  <Policy id="2">
   <TempConstDef name="TC2">
    <IntervalExpr name="sinceJan1st"><begin>01/01/05:01</begin></IntervalExpr>
    <PeriodicTimeExpr name="Weekend">
      <StartTimeExpr>
       <Week>all</Week>
       <DaySet><Day>7</Day><Day>1</Day></DaySet>
      </StartTimeExpr>
    </PeriodicTimeExpr>
   </TempConstDef>
   <CertCondition prop="Position" op="EQ" value="Researcher"/>
  </Policy>
 </Policies>
</Advs>
```

(a)

```xml
<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

- *Peers:* 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.
- *Ads:* 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.
- *Recent Requests/Ads:* 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.
- *Policies:* local peer sharing policies, associated with and indexed on the concepts they refer to.

4 Functionalities

In this section we describe the main functionalities of the system. The interested reader can find more details on the system functionalities and related algorithms in [6].

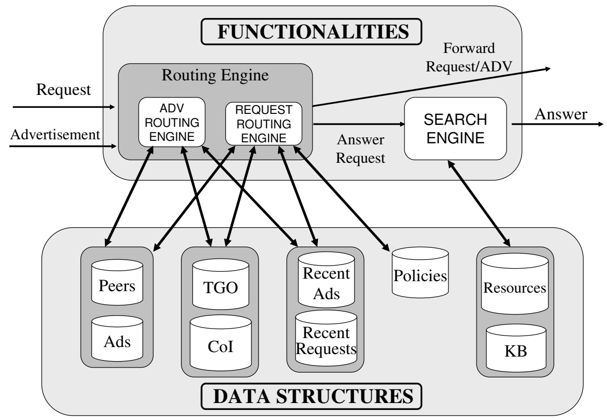

Fig. 4. Peer architecture

4.1 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 which can browse the TGO, read the textual explanation associated with each concept, identify its concepts of interest (CoI, see Section 2), and realize the concepts that better describes the local resources it wishes to share. The TGO is then copied locally in the peer. Now the peer can, when it wishes, populate its 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.

4.2 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 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 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.

4.3 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:

$$Relevance(P, c, Q) = \frac{accepted\_resources}{received\_resources} + \beta.$$

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 simply 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 can then be 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 [16], 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 it is likely to be quite knowledgeable on more general concepts related to the same topic;[2] *(iii)* a peer that answered well to previous requests on several different concepts it 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*

[2] Alternative principles could be experimented as well, such as assuming that a peer that answered well on a concept is likely to be quite knowledgeable on a related topic with a weight that depends on the distance of the two concepts, thus relying on the sim_c function. Through the use of appropriate indexes, the more general concepts are efficiently retrieved.

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 belongs 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 desidered number of peers is reached.

Receiving. A peer receiving a data request Q first of all checks whether it has already 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 in Section 4.2, 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.

5 Comparison with Related Work

In this section, we briefly survey some P2P systems born with the purpose of sharing knowledge, and we do not consider those systems developed for sharing computing resources, such as SETI@home (`setiathome.ssl.berkeley.edu`) and Avaki (`www.avaki.com`), and for allowing application-level collaboration between users, such as Groove (`www.groove.net`) and Magi P2P technology (`www.endeavors.com`).

We compare our system with Edutella [12,11], FreeNet (`freenet.sourceforge.net`), KEEx [4], Napster (`www.napster.com`), Piazza [9], the Trusted Computing P2P (TC-P2P) Architecture [15], and SWAPSTER [8,17], 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 seven systems has been driven by the will of considering a spectrum of heterogeneous proposals, where heterogeneity involves both the "birth" of the proposal (early proposals such as Napster vs. very recent ones such as the TC-P2P Architecture and KEEx), the motivation and the nature of the proposal (research projects funded by the European Community, composed by academic and industrial partners, such as SWAPSTER; opensource projects born by the will of one single person, such as FreeNet; commercial products like Napster; peer database management systems like

	Ontologies	Advertisement pushing	Access policies
Edutella	Each peer defines the supported metadata schemas; a mapping service translates between different metadata vocabularies.	In the version described in [11], peers push their advertisements to their super-peer, which aggregates this information and pushes a summary to its neighbor super-peers. This information is stored in indices and updated when peers join or leave the network.	No
FreeNet	No	No	Completely anonymous methods for storing and retrieving information
KEEx	Semantic coordination is ensured by local "contexts" (partial and approximate representations of the world)	No	Policies include conditions on the identity credentials of a peer
Napster	No	No	No
Piazza	Piazza does not integrate ontologies, but it supports an arbitrary graph of interconnected schemas described in XML	An indexing system allows peers to make their data items, summaries of collections of data items, and peer mappings, available to the other peers, and supports the simplest type of schema mappings	Data owners can specify access control policies declaratively and generate data instances that enforce them. Access control rights can be dynamically modified by exchanging cryptographic keys.
TC-P2P-A	No	No	Policies include conditions on the identity credentials of a peer, on its role, on the platform environment, on general access rights (e.g., how many times a resource can be accessed)
SWAPSTER	In the application described in [8], each peer has an ontology associated to it, that defines the content of the resources shared by the peer. The peer's ontology refers to an ontology common to all the peers in the system	In the application described in [8], advertisements include a semantic description of the peer's knowledge and are pushed by each peer in order to promote its expertise in the network	The application described in [17] provides security mechanisms, delegation of access rights, etc.

Table 1. A comparison of some existing P2P systems

Piazza), and the intended application domain (projects with well defined, pretty limited application domains such as Napster vs. general purpose ones such as SWAPSTER). Clearly, many more traditional P2P systems could have been considered, but for space constraints, we limited ourselves to analyze these seven ones. The result of our comparison is summarized in Table 1.

From Table 1, it turns out 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. The system that is closer to ours is SWAPSTER, that has been used to implement two concrete applications: Bibster [8], and Xarop [17]; 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.

6 Conclusions and Future Work

In this paper we have proposed a P2P system characterized by the following features:

- for each thematic group of peers there is one TGO; to describe its interests w.r.t. a particular subject, a peer defines a list of TGO concepts dealing with that subject;
- advertisements are pushed by peers in any moment of their life (usually, when they join the system for the first time, and when they update their resources); each advertisement contains a subset of the peer's expertise (concepts) and the policies for accessing the corresponding resources;
- sharing policies that include conditions on the credentials of a peer, on the internal state and connection type, and temporal conditions.

The analysis of the related approaches in Section 5 shows that these three aspects are definitely relevant for developing a flexible, secure, efficient, and accurate mechanism for routing and answering queries in a P2P setting. In fact, most of the systems that we have considered, face some of them. The originality of our proposal lies in addressing *all* of them into an integrated P2P system.

Most (although not all) of the systems that considered in Section 5 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. The main direction of our future work is thus completing the implementation, in order to release a first version, based on JXTA (`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, H. Bulskov, and R. Knappe. On Ontology-based Querying. In *Proc. of the IJCAI Workshop on Ontologies and Distributed Systems*, 2003.
2. S. Bergamaschi, C. Quix, and M. Jarke. The SEWASIE EU IST project. *SIG SEMIS Bulletin*, 2(1), 2005.
3. E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. An Access Control Model Supporting Periodicity Constraints and Temporal Reasoning. *ACM Transactions on Information and Systems*, 23(3):231–285, 1998.
4. M. Bonifacio, P. Bouquet, A. Danieli, A. Donà, G. Mameli, and M. Nori. KEEx: A Peer-to-Peer Solution for Distributed Knowledge Management. In *Proc. of the 4th Int'l Symp. on Knowledge Management*, 2004.
5. S. Castano, A. Ferrara, S. Montanelli, and G. Racca. Semantic Information Interoperability in Open Networked Systems. In *Proc. Conf. on Semantics of a Networked World*, 2004.
6. G. Guerrini, V. Mascardi, and M. Mesiti. A Semantic Information Retrieval Advertisement and Policy Based System for a P2P Network. Technical report, DISI, Univ. Genova, 2005.
7. P. Haase, R. Siebes, and F. van Harmelen. Peer Selection in Peer-to-Peer Networks with Semantic Topologies. In *Proc. of Int'l Conf. on Semantics of a Networked World: Semantics for Grid Databases*, pages 108–125, 2004.

8. P. Haase et al. Bibster – A Semantics-Based Bibliographic Peer-to-Peer System. In *Proc. of the Int'l Semantic Web Conf.*, pages 122–136, 2004.
9. A. Halevy, Z. Ives, J. Madhavan, P. Mork, D. Suciu, and I. Tatarinov. The Piazza Peer Data Management System. *IEEE Transactions on Knowledge and Data Engineering*, 16(7), 2004.
10. W. Nejdl, D. Olmedilla, M. Winslett, and C. Zhang. Ontology-Based Policy Specification and Management. In *Proc. of the European Conf. on Semantic Web*, pages 290–302, 2005.
11. W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Löser. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. In *Proc. of the 12th Int'l World Wide Web Conf.*, 2003.
12. W. Nejdl et al. Edutella: A P2P networking infrastructure based on RDF. In *Proc. of the 11th Int'l World Wide Web Conf.*, 2002.
13. M. Niezette and J. Stevenne. An Efficient Symbolic Representation of Periodic Time. In *Proc. of 1st Int'l Conf. on Information and Knowledge Management*, 1992.
14. R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989.
15. R. Sandhu, X. Zhang, K. Ranganathan, C. Rozas, and M. Covington. Peer-to-Peer Access Control Architecture Using Trusted Computing Technology. In *Proc. of the 10th Symp. on Access Control Models and Technologies*, 2005.
16. C. Tempich, S. Staab, and A. Wranik. Remindin': Semantic Query Routing in Peer-to-peer Networks Based on Social Metaphors. In *Proc. of the 13th Int'l Conf. on World Wide Web*, pages 640–649, 2004.
17. C. Tempich et al. XAROP: A Midterm Report in Introducing a Decentralized Semantics-Based Knowledge Sharing Application. In *PAKM'04 Conf.*, pages 259–270, 2004.
18. G. Tonti 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.*, pages 419–437, 2005.
19. M. Winslett, N., Ching, V. Jones, and I. Slepchin. Using Digital Credentials on the World Wide Web. *Journal of Computer Security*, 5, 1997.