Department of
Information Engineering
and Computer Science
DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive 14 - 38123 Povo - Trento (Italy)
http://www.disi.unitn.it

DISTRIBUTED ENTITY SEARCH

Alethia Hume

April 2011

Technical Report # DISI-11-462

Submitted as qualifying exam to the ICT PhD School of the Department of Information and Communication Technology.

Distributed Entity Search *

Alethia Hume
University of Trento
Trento, Italy
hume@disi.unitn.it

ABSTRACT

Capturing information about entities of the real world (i.e. locations, people, institutions and others) is a goal that is gaining more attention in today's web of data. We believe that this capturing would only be possible if users can contribute and interact as they do in the real world. The contribution and interaction of users may take place over a distributed network where they can publish information about known entities. We think that the contribution of users will not happen, if the network can not be considered also as a source of information for them. Therefore, this PHD thesis aims to address the problem of finding entities (i.e. information about entities) that are related to the information needs of the user, in order to consolidate a web of data, which is based on entities. These entities need to be found from a collection, which can be distributed and where each entity can be described from the points of view of different users. In this proposal we analyze different layers of abstractions in the context of an entity-centric application, which allow users to define and share their entities. Then, we describe the problem of entity search in a distributed environment considering the different abstraction layers. We end with a discussion of possible approaches for the solution, where a network architecture for the search is proposed.

Keywords

Entity Search, Peer-to-peer Search, Distributed Systems, Semantic Web

1. INTRODUCTION

The researchers that participate in the International Joint Conference on Artificial Intelligence (IJCAI) form a community (i.e. the IJCAI community). This community maintains detailed information about the scientific papers presented in the conference, the authors of the papers and the relations between papers and authors (e.g. all the papers written by the same author, co-authoring, and others). Some researchers could be interested in getting more information related to the topics of interest of the conference (e.g. related papers, reports of ongoing research or more information about authors). In order to access this information, the researchers need to find other members who have and share the information that is relevant to them. The IJCAI community is only one example, other communities interested in the area of Artificial Intelligence can be found. The members of these communities could also provide relevant information to the researchers of IJCAI community. However, the members of the other communities would need to be found and contacted by the researchers of IJCAI.

Many other examples can be found in everyday life where people exchange information about things (e.g., papers, meetings, concerts, artists, photos, locations, etc.) that are related to topics in which they are interested. We refer to these "things" that exist in the real world as *entities*. Some of these entities are of general interest (i.e. everybody knows about their existence); others are relevant for organizations of people (i.e. a limited number of individuals can provide information about the existence of these entities); there are also entities that are known at the personal level (i.e. few people are interested in these entities and they do not form an organization). In some cases different levels of detail and different aspects of an entity can be known by different people.

From the simplified examples mentioned above we can observe that the information about entities in the real world is inherently distributed and subjective. We want to address the problem of searching information about entities that are defined from different points of view and can be stored in distributed locations. In particular, the following layers of abstraction (i.e. layers of subjective view of the world) are distinguished:

- **Universal Layer**, e.g. Arnold Schwarzenegger can be considered known by everybody as an actor and a politician.
- **Community Layer**, e.g. For a group of people interested in movies, the awards received by Schwarzeneg-

*This paper has been submitted as qualifying exam for the admission to the second year of the ICT PhD School of the Department of Information and Communication Technology. I would like to thank my advisor Prof. Fausto Giunchiglia for his guidance and constant support, also to the members of Knowdive group for their valuable comments.

ger as an actor could be more relevant than other characteristics related to his political career.

- **Personal Layer**, e.g. Schwarzenegger is seen by his family as a father, husband, brother, etc.

We analyze these layers in the context of an entity-centric application that connects a set of peers (peer-to-peer network), where each peer represents a user interested in a set of topics. In this network, each peer can locally store information about its own set of entities. This information represents the way in which the peer sees the world. The local information of a peer can be shared with other peers in the network and peers with common interests can join to form communities.

In the approach we follow, the main goal is to search in a web of data that captures the dynamics with which the entities are defined in the real world. Entities can be seen as elements or resources that exist in the real world and have a name (i.e. named entities). Entities can be of different types (e.g. person, location, event and others) and are represented by a set of attributes [3]. The attributes describe the characteristics of the entities (e.g. name, latitude-longitude, size, date and others), which can depend of the type of entity. In other words, entities of different types could require different set of attributes to be properly defined.

In order to reason about entities we need to understand what the attributes of entities represent. In other words, we need to understand which characteristic of the entity is described by an attribute and the value of such characteristic. The meaning of the attributes and their values can be understood using a Knowledge Base [11]. A knowledge base contains information about terms, the concepts associated to them and the relations between the concepts. Hierarchical relations between concepts allow to organize them in a tree-like hierarchical structure from more general to more specific concepts.

In order to perform search in the described scenario, we analyze a possible architecture (in general terms) and we try to identify the main relations between peers and entities that can be exploited during search. The network obtained as a result can be seen as a source of data for the peers as well as a destination where they can publish their data. We argue that the profiles of interest of peers and communities need to be identified. Then, the relation of such profiles with the information need of the users need to be taken into consideration for the search.

Few approaches try to address the problem of entity search and they do not consider distributed scenarios. On the other hand, common techniques used for distributed search cannot be directly applied to this scenario because they are not aware of entities. Therefore, the contribution of this thesis should help to consolidate a web of data based on entities.

The remainder of this proposal is organized as follows. Section 2 presents the state of the art of the related areas. A logical architecture that defines the three layers of abstraction and the search problem are introduced in Section 3. Some hints for the solution are presented in Section 4. Section 5 concludes the proposal with some final remarks.

2. STATE OF THE ART

The problem discussed in this proposal combines the areas of entity search and peer-to-peer (p2p) systems. An approach that integrates both areas (i.e. performs search of entities over a p2p network) could not be found in the literature. Nevertheless, there are approaches from both areas which are relevant for the problem we want to address. The approaches which are aware of entities in general are relevant because the analysis performed by them can contribute with the definition of models and structures for the representation of entities. Moreover, the approaches that search for entities in a local repository can provide mechanisms to understand the information needs of the user in terms of entity attributes. On the other hand, in this proposal we discuss the entity search problem in the context of a network of peers. Therefore, current approaches to search in p2p networks can provide interesting techniques, which can be combined in order to build our solution. This section, presents an overview of the approaches from the two areas (i.e. entity aware and p2p systems).

2.1 Entity Aware Approaches

The definition of entities considered in this work is aligned with the notions of the OKKAM project [3], where the semantic web is seen as a global space into which the semantic knowledge from different sources should be integrated. In [3] an entity name system (ENS) is proposed in order to provide support for a large data collection of decentralized and independent information about the same entity. The project addresses the problem of retrieving information about the entities that are known by the repository and they can provide the links to the independent sources. Nevertheless, the problem of distributed entity search over independent repositories is not addressed. The local repository of a single user is not considered as a source of data and the users need a special access permit in order to contribute with the definition of entities.

As a first step in the goal of searching information about individual entities the work presented in [2] tries to understand the type of entity the user is looking for. A model that analyzes the attributes from the query specification and performs the disambiguation of the desired type of entity is proposed. The study performed an analysis of the kinds of attributes considered more relevant by humans to identify specific types of entities. In [23], the queries of users are analyzed in order to perform the extraction of named entities. The analysis is based on syntactic matching of patterns. The novelty of this approach is given by the fact that the user queries are analyzed instead of document collections (e.g. web pages). These approaches do not address the search, but the results of the analysis could be relevant for understanding the behavior of users during query definition.

Few approaches that retrieve entities as a search results can be found in the literature [4, 16]. An entity search engine is proposed in [4] to address the limitation of finding data entities existing in current search engines. The information about the entities is collected by a "Data Collector" and stored into a server. Then, pattern matching is performed to query the server. The entity extractor, which is

used by the data collector for the extraction of entities from web pages, is considered as a black box. Heuristic rules are used in [16] to identify entities appearing in a collection of documents. A set of features about the entities are also extracted, they are combined and weighted by a ranking model that is trained using supervised learning. Syntactic matching is used to identify the same entity in different documents. Therefore, the semantics of entities is not considered. Although the data in [4] is collected from multiple web sources (i.e. by crawling), in both approaches the search is performed in a centralized manner. There is no distribution at query time and the individual users are not considered as potential sources of information.

2.2 P2P Systems

One of the main interests of P2P networks has been file sharing and they have demonstrated to have a scalable nature in terms storage capacity [25, 20]. However, the main problem have been to find content that is relevant for peers. Several approaches have tried to address the search problem, the first attempts were using unstructured networks (e.g. Gnutella[1]) where the connections between peers do not follow any rule. Each peer maintains a list of neighbors and upon receiving a query, the request is forwarded to all the neighbors. The number of times that a query is forwarded have to be limited, first, because sending all the queries to all the peers becomes unmanageable, and second, to guarantee the finalization of the algorithm. This attempt leads to scalability problems due to the number of messages generated and does not guarantee that all answers will be found.

Some approaches use clustering techniques to group peers that have similar content [1, 5, 26, 6, 18]. To achieve scalability during the search process they changed the goal of try to reach as many sources as possible, for the goal of try to reach the best possible sources of information. Given that the best sources are queried, the quality of the result set is expected to be maintained. The goal of these approaches is to find best group to answer a query and then send the query to the peers in that group. The interest on these type of unstructured approaches is due to their simplicity and mainly because of their success.

Other attempts have proposed more structured approaches with the aim of guaranteeing the location of the content shared on the network (e.g. CAN[2], Chord[3], Pastry[4] and Tapestry[5]). They allow storing pairs of $\langle key, value \rangle$ in a Distributed Hash Table (DHT) and then retrieving the value associated with a given key. The problem with this type of systems is that they need to know the exact key that identifies the content. In search scenarios where peers want to perform lookups based on a query expressed in natural language, peers might not know the exact key of the content (in our case entities) that are relevant for the query. There are some techniques that can be used to perform multi-keyword search using DHT based approaches but they can be very expensive in terms of required storage and generated traffic (e.g. see [19]).

Hierarchical structures try to combine clustering techniques with the structure of DHTs [8, 17, 24, 9]. In [9] a two-tier DHT is examined, in which peers are organized into disjoint groups. A lookup message is first routed to the target group (using a inter-group overlay) and then is routed to the target peer (using a intra-group overlay). PCIR [24] propose a hybrid super-peer/DHT topology, which organize peers into groups. Each group is represented by a super-peer that contributes for publishing the information of the group (in batches) into a global DHT. The super-peers in this approach do not act as a point of entry for queries. A distributed clustering scheme is introduced by PCIR to form groups based on content similarity. A two layered architecture, which introduces semantics to search in a p2p network, is proposed in [17]. In [8], a paradigm called Canon is proposed to combine hierarchical structures and flat DHT approaches into hierarchically structured DHTs. They provide effective bandwidth usage and use a recursive routing structure.

The main drawback of the approaches described so far is that most of them are based on syntactic matching of words and do not deal with problems related to natural language (e.g., synonyms, homonymity, ambiguity, related concepts, complex concepts expressed by phrases). There are some approaches that try to deal with problems such as synonymity [21] and ambiguity [28] but they fail to consider more complex relations between the concepts that need to be taken into account.

Entity search is a strongly semantic task [2], therefore the techniques used to perform distributed semantic search could be considered of main relevance. Some approaches use semantic topologies to group together the peers who have interest profiles that are semantically related. Semantic overlays can offer advantages over syntactic approaches in terms of quality of results because they deal with the underlying meaning of queries. A semantic link p2p network is built by computing the semantic relationships between peers' data schemas in [30]. The routing of queries is based on semantic similarity of peers, and queries are reformulated using a schema mapping algorithm. In the Semantic Flooding approach [10] each peer can build its own classification hierarchies, which codify its interest profile. A semantic overlay is built computing the semantic relation between complex concepts specified by the nodes in the classifications of different peers. The semantic overlay is used to contact peers interested in similar or related topics and forward query requests to them.

Ontology-based p2p data management system [29] is based on ontology mapping and query processing. Edutella [22] and Bibster [14] are built on JXTA framework and aims to combine meta-data with p2p networks. Each peer is described and published using an advertisement, which is an XML document describing a network resource. Also in [15] peers advertise their expertise but in this case through semantic descriptions of their knowledge, which is based on a shared ontology.

[1] http://en.wikipedia.org/wiki/Gnutella
[2] http://en.wikipedia.org/wiki/Content_addressable_network
[3] http://en.wikipedia.org/wiki/Chord_(peer-to-peer)
[4] http://en.wikipedia.org/wiki/Pastry_(DHT)
[5] http://en.wikipedia.org/wiki/Tapestry_(DHT)

P2P approaches consider the distribution of the information but they are not aware of entities, therefore they can not be directly applied to our problem.

3. PROBLEM STATEMENT

We want to address the problem of searching information about entities that are defined according to different points of view and can be stored in distributed locations. The different points of view correspond to partial information or information about different aspects of entities from the real world, which can be stored in different locations of a network. For the discussion of the problem we try to capture the characteristics of the real world when defining, storing, and retrieving the information about entities.

We consider a network of peers, where each peer represents a user that can define the set of its local entities according to its interest and its personal point of view. Each peer has also a local knowledge base (i.e. background knowledge), which contains specific concepts about its topics of interest and shows how the peer understands the world. The peers can share their own set of entities in the network and can search over the entities shared by others.

To reason about entities on different peers, these peers need to understand each other. In real life people need to agree on some basic concepts in order to discuss, share information and learn from each other. Analogously, in this network the peers need to agree on some general knowledge, which is universally known and accepted (i.e. universal knowledge).

Peers with similar interests could store information about entities, which are related or are the same. A community of peers with similar interests can collaborate for the definition of entities related to their topics of interest. A knowledge base for this community (i.e. community background knowledge) is needed to allow the communication and understanding among the peers of the community in more specific terms.

The characteristics of this network show different layers of abstractions, which can be identified by considering how peers interact in the real world. In this section we analyze a logical architecture to represent the different abstraction layers or different views about the entities of the real world. Then, we provide further discussions of the sub-problems of searching through different abstraction layers based on the logical architecture.

3.1 Logical Architecture

Multiple layers of abstraction can probably be identified when we consider how people normally interact with each other. We distinguish between three layers in the logical architecture that is shown in Figure 1. Things that are known by everybody are in the *universal layer*, the point of view and the things that are known by organizations of peers are considered to be in the *community layer* and the things that are known only by the local peer are part of the *peer layer*. We consider this is the minimum distinction that needs to be made in order to allow the representation of the behavior of people in normal every day life. More detailed characteristics for the knowledge bases, the set of entities and the attributes of such entities at each of the three layers are described in order to better understand their implications on the problem of entity search.

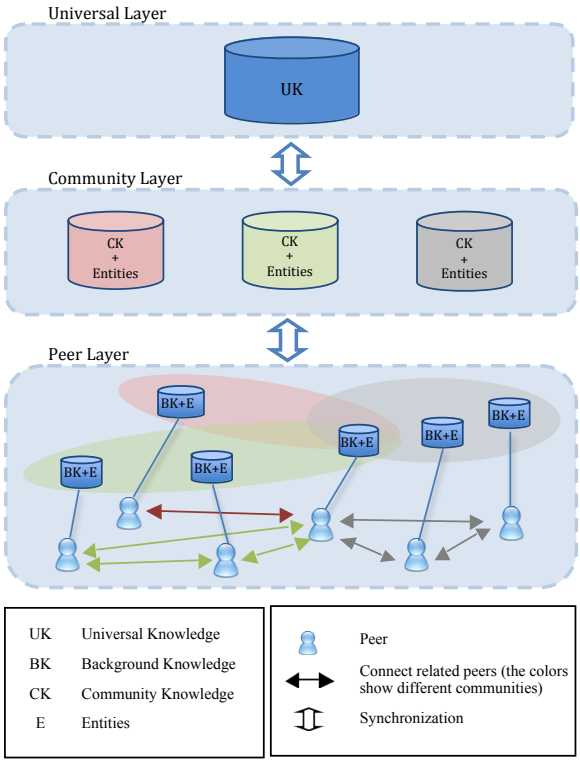

Figure 1: Logical architecture.

The Universal Knowledge (UK) and the entities defined at the universal layer are visible for all the peers in the network. The UK provides with the fundamental concepts for the definition of global entities and allows the understanding between different peers. The entity name system (ENS) proposed in [3] and the repository created by YAGO [27] could be seen as efforts to build this layer, which can represent a general source and a reference about entities of the real world.

The community layer considers the organizations of peers which are interested in the same topics. Inside a community an entity that is known at the universal layer can be defined in more specific terms by the addition of new attributes, which are relevant for that particular community. For example, the entity of "Arnold Schwarzenegger" could be defined as shown in Figure 2. Some basic attributes, which identify the person, can be globally known in the universal layer. On the other hand, a community of peers interested in movies (like IMDB, Internet Movie Data Base community) could identify and add other attributes, which are relevant for that community. The movies in which he participated and the awards he received are some examples of those relevant attributes, while other aspects of his life as politician are irrelevant for this community.

In order to define entities in more detail, each community will also need a deep understanding of terms and concepts related to the topic of interest. For example in the community of researchers that participate to the International

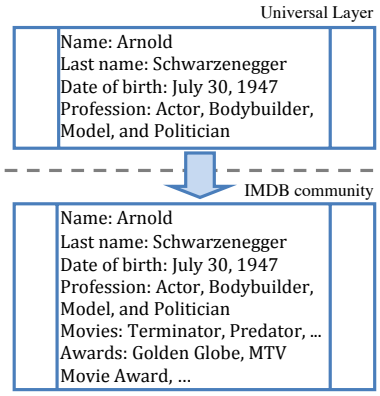

Figure 2: Example of entity extension.

Joint Conference on Artificial Intelligence (i.e. the IJCAI community), the peers use several terms to describe and classify papers. Most likely, these terms are specific to the area and some of them could be unknown outside the community. If WordNet[6] is taken as an example of a general knowledge base (i.e. UK) and we search it for the concepts of the terms Macintosh, AJAX and Apple, the results shown in Figure 3 are obtained. In the area of computer science

Macintosh(*Noun*)**:**
- S: (n) mackintosh#1, **macintosh#1** (a lightweight waterproof (usually rubberized) fabric)
- S: (n) **macintosh#2**, mackintosh#2, mac#1, mack#1 (a waterproof raincoat made of rubberized fabric)

AJAX(*Noun*)**:**
- S: (n) **Ajax#1** (a mythical Greek hero; a warrior who fought against Troy in the Iliad

Apple(*Noun*)**:**
- S: (n) **apple#1** (fruit with red or yellow or green skin and sweet to tart crisp whitish flesh)
- S: (n) **apple#2**, orchard apple tree#1, Malus pumila#1 (native Eurasian tree widely cultivated in many varieties for its firm rounded edible fruits)

Figure 3: Definitions from wordnet.

the terms Macintosh and Apple are easily associated to the well-known brands, while AJAX refers to a group of interrelated web development techniques. On the other hand, a fan of sports will associate the name AJAX to the soccer club with the same name. As can be seen in Figure 3 none of these senses are in WordNet. In the example of IJCAI the community of peers can take some subset of the UK and extend it with more specific knowledge about their area of interest to obtain the Community Knowledge (CK). In Figure 1, double arrows with the same color connect the peers from the same community.

The personal layer, where the peer has entire control over its entities and its background knowledge (BK), is called the peer layer in Figure 1. The BK of the peer can contain partial knowledge from the UK, which could be extended with more specialized knowledge (e.g. concepts created by a user, relationship between concepts that are true only in some context). A reasoning similar to the one discussed in the community layer can be applied at this layer to understand how entities can be extended with more attributes or new entities can be defined. The peer can participate in different communities and could maintain direct links (i.e., connections, references) to known peers, which are not necessarily in the same community. Each peer can decide what part of its content would be shared in the network and with whom.

[6] `http://wordnet.princeton.edu/`

3.2 Search

When the user thinks in an entity, typically he thinks in terms of a set of attributes that define the entity. For example, most of the people do not actually know (in person) Arnold Schwarzenegger, but they know some attributes about him such as his name, his physical appearance (through pictures) and the fact that he was until recently a governor in the United States. When these people think in Schwarzenegger they are actually thinking on these known attributes. In the same way, when a user wants to search for the entity of Arnold Schwarzenegger, it should be natural to specify a query in terms of the set of known characteristics about him. Something similar happens when we want to retrieve restaurants located in Trento. The fact of being located in Trento is an attribute that identifies a set of entities and most likely will be part of the specifications given in the query. In both examples, the type of the desired entities is known in advance.

The general problem of retrieving entities can be described as finding entities of a given type and whose attributes meet the constraints and characteristics described in a given query. In other words, for each attribute that is specified in the query, ideally there must be at least one attribute in the entity, which is equal to or more specific than the query attribute. The distributed environment previously described does not modify the constraints to retrieve entities, but adds additional complexity. This complexity is given by the fact that several repositories need to be explored in order to find the relevant entities. Moreover, different aspects (i.e. attributes) of the same entity can be defined in different places. In a network of small dimensions the simpler solution would be to search everywhere but in a network of the scale of the web this approach becomes impractical.

In order to search in a large scale network with the characteristics described in Section 3.1, the following aspects need to be considered:

Moving through different layers. Consider the example of a peer P_A who ask a question to a friend, peer P_B. P_B provides some answers and he also remembers other co-workers who could know more interesting answers and promises to ask them. Then, P_B go to work and asks everyone at work the same question that P_A asked him. In this way the query has, in fact, arrived to a community of peers through one of the members who knows the interest profile and the expertise of the community. One of P_B's co-workers may also have a friend P_C who is an expert on the topic of the query and the question is forwarded to P_C. The query, which was generated at the peer layer, reached a remote peer (possible unknown) from the same layer but through the community layer. These are the kinds of connections people make to get information in their real life and, to satisfy the information needs of the user, these are also the behaviors that need to be reproduced in the network.

Note that in the example above the key is to move through the different layers. The different ways in which the layers can be connected in the network need to be identified. In

our example the relationship between the two friends could be seen as a direct link between two peers. In a similar way, a link could be created between two organizations of peers (i.e. communities) if they are interested in related topics. These links allow the navigation of the respective layers. However, when the search is performed it is necessary that the layers are navigated efficiently in order to find relevant results, without generating too much load to the network. What usually happen in real life is that we do not ask to all our friends the information we need. We select among our friends the ones that could have the information that we need and we perform this selection based on what we know about them. On the other hand, the relation between P_B and its co-workers could be seen as a relation between a peer and the communities in which participates. These links allow the navigation from the peer layer to the community layer and vice-versa. Again in this case, what happens in real life is that people do no ask to random groups about the information that they need. According to the information being searched, a selection of the proper communities usually is done.

Another type of link is the one that connects entities which are related. Consider an entity of a restaurant, where one of its attribute defines its location. The value of the attribute could be the city of Trento in Italy, which is also an entity. Both entities (the restaurant and the city of Trento) are related and the connection allows the navigation from the definition of one entity to the definition of the other.

The main issue with all the relations discussed above is that they are natural to people. For an application, on the other hand, the identification of these relations is not trivial. Moreover, the selection of relevant people or organizations as sources of information is performed by humans without even thinking it. The complexity of these decisions, for an application implies a number of challenges from the point of view of representation, storing and computation of the data.

Understand the query at different layers. Once a search request is received by a data repository, a centralized search is performed over its local data. From our point of view, the centralized search can be seen as a black box that given a query specification provides as a result a list of relevant entities from the local repository. Nevertheless, the problem of understanding the query specification in terms of the local knowledge base has to be considered.

We can think an example of two friends, P_A works in informatics and P_B works in tourism. P_A ask to P_B if he has some information about "JAVA". P_B answers yes and provides catalogues, guides and tourist information about the Island of Java, while P_A was expected information about the programing language. A solution for this problem could be the use of a formal language, which provides the concept used by the initiator of the search to specify the query. But P_B will not be able to understand that Java is a programming language if he does not know what a programming language is. This mean that in order to understand the concepts at the query P_B needs to learn more information related to the context (i.e. related concepts). On the other hand, we can not expect other peers to learn new concepts just to be able to understand our query. Therefore, when considering the targets to send search requests it is important to know if the target will be able to understand the query.

One issue that should not be overlooked in our example is the fact that P_A and P_B are related (e.g. friends, co-worker, members of the same organization, interested on related topics). In the real life this relation is maintained through time and people are involved in interactions (e.g. small talk). The information about the concepts known by each other can be obtained through those interactions before the existence of any search request. Even if P_B claims to be interested in informatics issues, the level of details that he is able to understand could be used by P_A to constraint its search scope. In a similar way, part of the information needed by a peer at query time could be pre-computed in our network. Therefore, could be interesting to consider this characteristic.

Merge the results. Once the user enters a query and the search algorithm is triggered, the goal of the application is to receive the results and show them to the user. The results come from different sources and we assume each of them provides a list of entity definitions. A unique list of results needs to be showed to the final user, which imply merging the lists from different sources. Some of the definitions from different sources could be referring to the same entity. These definitions can be merged together in order to present one definition per entity in the final result list. We need to carefully analyze how to handle the merging of entities to present a coherent result set.

Consider again the example of "Arnold Schwarzenegger" and a user that searches information about him. Several entity definitions that refer to the same real world entity could be returned as part of the result list of different sources. Let us suppose that one of the definitions says that he is an Actor and contains the list of movies in which he participated. Another definition that also says he is an actor contains a list of awards he received, but does not know anything about his movies. These definitions contain partial information about the entity that complement each other. A single definition for the entity can be created by joining the information (i.e. attributes) from both definitions.

Following with the example of "Arnold Schwarzenegger", now a third definition could describe him as a politician and a model. The scenario now changed because this definition contains information about the same characteristic described by the other two definitions, but provides different attribute values. The system have the challenge of realizing that the different values are not necessarily contradictory. As a matter of fact, all of these professions are associated to Arnold Schwarzenegger in real life. It could be also the case that the attribute values are actually contradictory. For example one source could describe him as a good actor, while another may describe him as a bad one. This mean that the peers have different opinions, but do not imply that they are referring to different entities. Addressing this type of issues also represents a challenge for the system.

Ranking. The relevance of the entities for a given query specification have to be measured and the entities need to be ordered in a decreasing order of their relevance. We need to

define a similarity measure between a query and an entity. The goal is to evaluate the degree at which the entity satisfies the restrictions imposed by the query, and how much the information required by the query is contained in the entity definition.

We assume that the list of results returned by the different sources are ordered according to a ranking function and that all the peers use the same ranking function (although, probably we will need to consider also the relaxation of this assumption). Inside the result list from each source, the information about the relevance of the entities is relative to the local content and knowledge of the source. As a consequence, the ability of the different sources to evaluate the relevance of their results needs to be considered. Which means that it could be the case in which the best result from a bad source of information can be worst than the first 20 results provided by an expert.

In some approaches, the expertise of the peer who provides the information can influence the way in which such information is ranked. This could represent a problem in our approach, if one entity definition in the result set is produced by merging entity definitions from different sources (i.e. different peers and communities). If the entity definition contains attributes provided by different sources, then it is not possible to speak about a single source for the entity. Therefore, the influence of the expertise of the different sources over the same entity needs to be analyzed.

The peers in the network are assumed to be autonomous, therefore different peers can have different behaviors. The peer that always provides a number of responses is not always the best. Consider the example of the peer P_A that usually returns a number of entities that do not satisfy the needs of the peer P_B. Eventually, P_B will identify the information obtained from P_A and will assign it less importance (i.e. will be bellow in the ranking). Similarly, we need to consider the reputation of the peers, which can depend of a specific area or topic. How the ranking of an entity, which is the result of merging entity definitions from different sources, can be affected by the reputation of its different sources also needs to be considered.

4. APPROACHES FOR A SOLUTION

The first two aspects considered in Section 3.2 are related to the definition of the scope for the search. This means that the most promising sources of information need to be identified and selected. The query specification needs to be matched with the communities and peers that may have relevant answers. The other aspects refer to the management of the obtained results.

4.1 Identifying the Search Scope

In order to match the query to possible relevant sources, the relation between the query and the sources of information need to be understood. Based on these relations, a matching mechanism to identify the relevant communities and peers for a given query will be studied. The first step towards this is the definition of a profile of interest for communities and peers. These profiles must represent what they can offer to the search. This notion of profile is different from the one used in current social networks (e.g. facebook, orkut and others). In our approach, a profile contains the data to be used by the search algorithm to decide about the relevance of a given source of information (peer or community) for a given query.

In the case of peers, we believe the interest profile can be automatically bootstrapped from its local content by considering the entities that they share in the network. The automatic extraction of this information is important to avoid disturbing the users by asking them to provide it. On the other hand, if the user wants to provide explicit declaration of its interests and expertise, the system should provide with the proper interfaces in order to facilitate the task. Analogously, the content of a community can be considered in order to automatically build its profile. The main difference is given by the fact that the content of the communities could be distributed among its members. Therefore, the information needed to build and maintain the profiles of the communities should be obtained using a distributed approach.

After defining the profiles, the second step is to store, maintain and be able to map them to the query definition. Different approaches could be considered to perform search at the intra-community level. In a *centralized* approach, a server could be used to search within the community. The server could store all the data related to the community. To answer a search request, the server runs the local search and returns the entities that match the query. In this case, the server does not need to contact the peers to answer a search request. In a different approach the server could be used as a super peer. In this case the data is stored at the peers, but the profiles of the peers participating in the community are stored at the server. Search requests are received at the server, where the query specification is matched with the profiles of relevant peers. Finally, the search request is forwarded to them and the results are returned.

A *distributed* approach avoids the use of a server and the data of the community are stored by its members. In a small community (e.g. hundreds of peers or less), the profile of each peer could be flooded to all the other peers in the community; each peer then stores these profiles and uses them to select relevant peers from the community when a search request is received. A more structured approach, such as a distributed hash table (DHT) could be used in the case of a larger community (e.g. thousands of peers or more). The DHT could be used within the community to index and retrieve the profiles of its members. When a search request arrives, the given query is used to retrieve profiles of relevant peers i.e. the keys to be used in the DHT are extracted from the query specification. In both cases, the query is forwarded to the relevant peers and the results of the local search inside the peers (i.e. entities) are returned to the initiator of the search request.

As can be seen, the search within each community can be performed in a very different manner depending of the type of community (i.e. the selected approach). On the other hand, the interface seen from the outside can be very similar. The information related to the search services provided by the community and how to access them can be specified inside the profile of the community. On the contrary,

the information related to the type (i.e. centralized or distributed), the infrastructure and the search techniques of the community could be encapsulated inside the community and most likely do not need to be publicly available.

The communities themselves can be then organized in a top-level overlay network, the profiles could be published and maintained in this overlay. Given that, the availability of this information would be crucial for our query execution method, we want to avoid the problems of having a single point of failure (i.e. centralized approach). A distributed hash table (DHT) could be used to index the profiles of communities. The keys to be used for indexing the communities have to be carefully selected. The communities that are relevant for a given query need to be retrieved later. While it is true that by using a DHT we avoid having a single point of failure, its availability depends on peers that are autonomous and may have unpredictable behaviors. Therefore, the availability of DHTs can be still considered a little unstable. In order to offer further guarantees, the integration of cloud computing with a peer-to-peer (p2p) approach can be considered. For example, CLOUDCAST[7] proposes an interesting approach that includes a passive storage cloud to provide support for the distribution of content and the management of the p2p network.

The architecture proposed through this section is showed in Figure 4. As it can be seen, the information needed to find communities is encapsulated at the top-level overlay, while the profiles of peers are encapsulated at the intra-community level. It has to be noted that the three layers of the logical architecture presented in Figure 1 can be mapped into Figure 4 in a straightforward manner. When a search request is generated at the peer, the following three-step query execution method can be considered:

1. Find relevant peers within the set of known peers.
2. Find the communities that are relevant for the query.
3. Send the query to the relevant peers and communities.

4.2 Working Out Results

In order to build a ranked list of entities with the results returned by the different sources, the definitions referring to the same entity can be merged together. A list of entities where each entity appears only once will be obtained. A ranking function needs to be applied to the list obtained in order to evaluate the relevance of the entity with regard to the query specification.

Given two entity definitions, a matching approach should be able to provide the degree to which the entities are considered to be the same in the real world (i.e., identity of entities). We believe that the approaches available in the literature, such as [7, 13]) can be analyzed and used to solve this issue. Because of this, we consider the entity matching as a black box whose outcome can be used to identify definitions of entities that can be merged. Then, to merge the definitions a simple union of both set of attributes can be performed. When two attributes describe the same characteristic of the entity, their sets of values can be joined together. This approach works fine for the cases in which the attributes from the different definitions complement each other.

As has been discussed in Section 3.2, when merging the definitions of entities we risk to produce incoherent results. In order to avoid incoherent results, the attributes that describe subjective characteristics of an entity can be distinguished. These types of characteristics usually can not be associated to the identity and a contradiction of its values could be allowed. Then it becomes a matter of visualization because the user has to be informed about the source of the contradiction. Maybe even the levels of confidence assigned to each of the values could be provided.

On the other hand, attributes that describe objective characteristics of an entity can not have contradictory values. This situation could point out that the definitions are actually referring to different entities. Entity matching algorithm should already identify this problem and produce as a result a low matching degree. In such cases is better to maintain the two definitions separated.

In order to produce a ranking function for a distributed search, the entities returned by the search process and the sources that provide them, have to be analyzed. A score could be assigned to the entities to describe their relevance with regards to the query. The score of an entity could result from the combination of, (i) the evaluation of how much the entity satisfies the constraints imposed by the query, (ii) the evaluation of the degree in which the entity provides the information requested by the query, (iii) the evaluation of the score of the sources that provide information about the entity. In general, state of the art techniques for the evaluation of the relevance of an entity could be explored and combined with solutions for raking in a distributed environment in order to address these issues.

The following characteristics of an entity from the result set will be taken into account to perform the evaluations mentioned in the previous paragraph: (i) Is defined by a set of attributes which are obtained from different sources. Moreover, different values for the same attribute can be provided by different sources. (ii) The importance of each attribute could vary with regards to the query. (iii) The reputation of a source could vary with regards to the query and also depending of the initiator of the search request.

The semantic similarity [12] can be considered to evaluate the importance of an entity attribute with regard to the query. Two parts can be distinguished in the query, the constraints and the wanted information. The attributes describing characteristics that are related to the constraints imposed by the query are important, because they help to point out the right entity. Other attributes, which are also really important, describe the characteristics related to the wanted information. The values of these attributes provide the answers to the questions encoded within the query.

The expertise and reputation of the different sources (i.e. peers) can be considered to evaluate their score. Implicit

[7] `http://www.disi.unitn.it/~montreso/pubs/papers/cltoudcast.pdf`

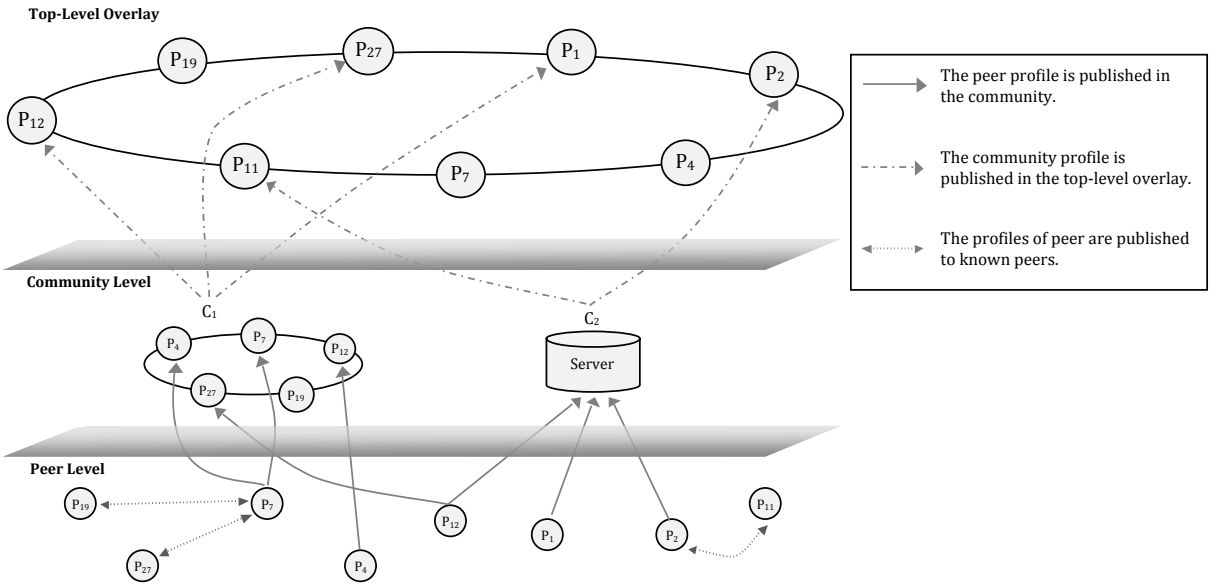

Figure 4: Network architecture for the search.

and explicit feedback from the users can be used to build the reputation of the sources. The influence of a source into the score of an entity should to be weighted according to the importance of its contributions to the final definition.

The number of sources that return the same entity in their result set could be considered as a measure of the confidence of the result, because it means that the entity is considered relevant for the query from different points of view. The same argument can be applied to entity attributes and this can help also when considering contradictory values from different sources.

5. CONCLUSIONS

This proposal has described the emerging trends of the web of data that aims at capturing information about entities of the real world. Also, the definition and characteristics of entities were discussed. Some examples were presented to show that the information about entities in the real world is inherently distributed and subjective. Moreover, three different layers of abstraction were distinguished (i.e. *Universal Layer*, *Community Layer*, *Peer Layer*) in the context of a network of peers. Then, the subproblems of moving through the different layers; understanding the query at different layers; merging the search results; and ranking entities provided by different sources were introduced.

As an approach for the solution, the need of the definition of interest profiles for peers and communities was discussed. These profiles and their relations with the information needs of the users should be considered for the selection of the scope for the search. An architecture, which can be directly mapped within the three abstraction layers, was proposed to offer support to the search process. This architecture is intended to encapsulate information about communities in a top-level overlay, while the information about peers is maintained at the community level and the peer level. To address some of the issues related to the merging of entities, the distinction between subjective and objective entity attributes has been proposed. The characteristics of the entities from the result set, which must be considered for the definition of a ranking function, were identified.

Finally, the Figure 5 shows a gantt chart that contains the research plan for the current year. The columns represent the number of the month and the rows each of the planed tasks. The tasks T1, T2 and T3 involve the formalization of profiles. The implementation and a preliminary evaluation of the proposed solutions on a real system are planned in the tasks T4 and T6. However, the details how is going to be built the proper scenario for these evaluations, are part of the research work as shows task T5. A re-planning based on the evaluation results will be done at the end of the year.

6. REFERENCES

[1] M. Bawa, G. Manku, and P. Raghavan. Sets: Search enhanced by topic segmentation. In *Proceedings of The 26th Annual International ACM SIGIR Conference*, pages 306–313, 2003.

[2] B. Bazzanella, H. Stoermer, and P. Bouquet. Searching for individual entities: a query analysis. Technical report, University of Trento, 2009.

[3] P. Bouquet, H. Stoermer, C. Niederee, and A. Maña. Entity name system: The back-bone of an open and scalable web of data. In *Proceedings of the 2nd IEEE ICSC*, pages 554–561, Washington, DC, USA, 2008. IEEE Computer Society.

[4] T. Cheng and K. C.-C. Chang. Entity search engine: Towards agile best-effort information integration over the web. In *CIDR*, pages 108–113, 2007.

[5] E. Cohen, H. Kaplan, and A. Fiat. Associative search in peer to peer networks: Harnessing latent semantics. In *Proceedings of IEEE INFOCOM*, 2003.

[6] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Stanford

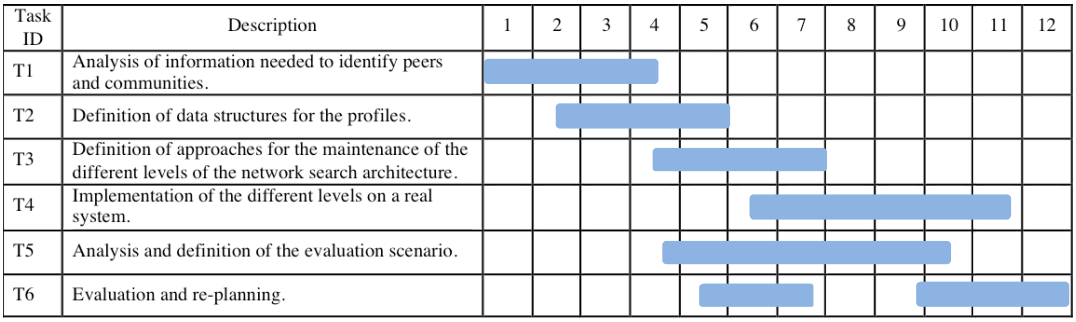

Figure 5: Research plan.

University, 2002.

[7] M. Dabrowski and P. Pacyna. Generic and complete three-level identity management model. In *SECURWARE '08*, pages 232 –237, 2008.

[8] P. Ganesan, K. Gummadi, and H. Garcia-Molina. Canon in g major: designing dhts with hierarchical structure. In *Distributed Computing Systems, 2004. Proceedings. 24th International Conference on*, pages 263 – 272, 2004.

[9] L. Garcés-Erice, E. W. Biersack, P. Felber, K. W. Ross, and G. Urvoy-Keller. Hierarchical peer-to-peer systems. In *Euro-Par*, pages 1230–1239, 2003.

[10] F. Giunchiglia, U. Kharkevich, and A. Hume. Semantic flooding: Semantic search across distributed lightweight ontologies. Technical report, University of Trento, 2010.

[11] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Discovering missing background knowledge in ontology matching. In *ECAI*, pages 382–386, 2006.

[12] F. Giunchiglia, M. Yatskevich, and P. Shvaiko. Semantic matching: Algorithms and implementation. *J. Data Semantics*, 9:1–38, 2007.

[13] N. Guarino and C. A. Welty. Identity, unity, and individuality: Towards a formal toolkit for ontological analysis. In *ECAI*, pages 219–223, 2000.

[14] P. Haase, J. Broekstra, M. Ehrig, M. Menken, P. Mika, M. Plechawski, P. Pyszlak, B. Schnizler, R. Siebes, S. Staab, and C. Tempich. Bibster - a semantics-based bibliographic peer-to-peer system. In *Proceedings of the 3rd ISWC*, pages 122–136, 2004.

[15] P. Haase, R. Siebes, and F. V. Harmelen. Peer selection in peer-to-peer networks with semantic topologies. In *International Conference on Semantics of a Networked World: Semantics for Grid Databases*, 2004.

[16] G. Hu, J. Liu, H. Li, Y. Cao, J.-Y. Nie, and J. Gao. A supervised learning approach to entity search. In *Information Retrieval Technology*, volume 4182 of *LNCS*, pages 54–66. Springer Berlin / Heidelberg, 2006.

[17] D. Janakiram, F. Giunchiglia, H. Haridas, and U. Kharkevich. Twolayered architecture for peertopeer concept search. Technical report, University of Trento, 2010.

[18] S. Joseph. Neurogrid: Semantically routing queries in peer-to-peer networks. In *Proc. Intl. Workshop on Peer-to-Peer Computing*, pages 202–214, 2002.

[19] J. Li, B. Thau, L. Joseph, M. Hellerstein, and M. F. Kaashoek. On the feasibility of peer-to-peer web indexing and search. In *2nd International Workshop on Peer-to-Peer Systems (IPTPS 2003)*, 2003.

[20] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, 7:72–93, 2005.

[21] W. Ma, W. Fang, G. Wang, and J. Liu. Concept index for document retrieval with peer-to-peer network. In *Proc. SNPD '07*, 2007.

[22] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. Edutella: A p2p networking infrastructure based on rdf. In *Proceedings of WWW'02*, 2002.

[23] M. Paşca. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the sixteenth ACM conference on CIKM '07*, pages 683–690, New York, NY, USA, 2007. ACM.

[24] O. Papapetrou, W. Siberski, and W. Nejdl. Pcir: Combining dhts and peer clusters for efficient full-text p2p indexing. *Computer Networks*, 54(12):2019–2040, 2010.

[25] J. Risson and T. Moors. Survey of research towards robust peer-to-peer networks: Search methods. *Computer Networks*, 50:3485–3521, 2006.

[26] K. Spripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *Proceedings of IEEE INFOCOM*, volume 3, pages 2166–2176, 2003.

[27] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of WWW'07*, pages 697–706, New York, NY, USA, 2007. ACM.

[28] C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *Proceedings of ACM SIGCOMM*, pages 175–186, 2003.

[29] H. Xiao and I. F. Cruz. Ontology-based query rewriting in peer-to-peer networks. In *Proceedings of the 2nd Int. Conf. on Knowledge Engineering and Decision Support*, pages 11–18, 2006.

[30] H. Zhuge, J. Liu, L. Feng, X. Sun, and C. He. Query routing in a peer-to-peer semantic link network. *Computational Intelligence*, 21:197–216, 2005.