SECCO: On Building *Semantic Links* in Peer-to-Peer Networks

Giuseppe Pirrò[1], Massimo Ruffolo[2] and Domenico Talia[1]

[1]D.E.I.S, University of Calabria
87036 Rende, Italy
{gpirro, talia}@deis.unical.it
[2]Exeura
87036 Rende, Italy
ruffolo@exeura.it

Abstract. Ontology Mapping is a mandatory requirement for enabling semantic interoperability among different agents and services relying on different ontologies. This aspect becomes more critical in Peer-to-Peer (P2P) networks for several reasons: (i) the number of different ontologies can dramatically increase; (ii) mappings among peer ontologies have to be discovered on the fly and only on the parts of ontologies "contextual" to a specific interaction in which peers are involved; (iii) complex mapping strategies (e.g., structural mapping based on graph matching) cannot be exploited since peers are not aware of one another's ontologies. In order to address these issues, we developed a new ontology mapping algorithm called Semantic Coordinator (*SECCO*). *SECCO* is composed by three individual matchers: syntactic, lexical and contextual. The *syntactic matcher*, in order to discover mappings, exploits different kinds of linguistic information (e.g., comments, labels) encoded in ontology entities. The *lexical matcher* enables discovering mappings in a semantic way since it "interprets" the semantic meaning of concepts to be compared. The *contextual matcher* relies on a "how it fits" strategy, inspired by the contextual theory of meaning, and by taking into account the contexts in which the concepts to be compared are used refines similarity values. We show through experimental results that *SECCO* fulfills two important requirements: fastness and accuracy (i.e., quality of mappings). *SECCO*, differently from other semantic P2P applications (e.g., Piazza, GridVine) that assume the preexistence of mappings for achieving semantic interoperability, focuses on the problem of finding mappings. Therefore, if coupled with a P2P platform, it paves the way towards a comprehensive semantic P2P solution for content sharing and retrieval, semantic query answering and query routing. We report on the advantages of integrating *SECCO* in the *K-link+* system.

Keywords: Ontology mapping in Peer-to-Peer networks, semantic mapping, semantic P2P applications, semantic web.

1 Introduction

Most of the information available today is in an unstructured and non-standardized form, therefore processing and exchanging it with people via computers is actually

very difficult. This is because "machines" are not able to recognize the meaning of information they deal with. Solving this challenge is one of the main goals of Semantic Web technologies. The Semantic Web vision [3] aims at providing Web resources (e.g., web pages, documents) with supplementary meaningful information (i.e., metadata) in order to improve and facilitate their retrieval, enable their automatic processing by machines and make it possible the interoperability among different systems. Ontologies are key enablers towards this "new" Web of semantically rich resources. Ontologies can be exploited to give *shared conceptualizations* of knowledge domains and make *explicit* and machine understandable the meaning of the terminology adopted [24]. They aim at capturing knowledge typically shared by a group. The reference to a domain of interest indicates their usage not for modeling the whole world but rather those parts relevant to a particular task. Many ontology languages used today are based on XML (e.g., RDF(S) [28], OWL [39]) which make ontologies exploitable as semantic support in different classes of distributed applications such as Semantic Peer-to-Peer [48] and Semantic Grid [22].

In a recent interview [4], Tim Berners-Lee stated that: "*The Semantic Web is designed to smoothly interconnect personal information management, enterprise application integration, and the global sharing of commercial, scientific and cultural data...*". From this interview emerges that semantic-based data sharing is expected to begin in controlled environments smaller than the World Wide Web as for instance: enterprise networks and small-medium Peer-to-Peer (P2P) networks. Moreover, the Semantic Web is expected to follow the same path of the Internet, which started in bounded environments.

In distributed environments, it is not feasible to have a single (and universally accepted) ontology describing a knowledge domain, but there will be several (possibly overlapping) ontologies created w.r.t "the point of view" of their designers. In fact, as people see the world differently these viewpoints inevitably will be encoded in ontologies. For instance, for a computer company a computer is a product, for an economist, it is a household appliance, while for a student it is just a computer. In order to promote interoperability among these different perspectives about the world, it is necessary to ensure "reciprocal understanding". This problem has been a core issue of recent ontology research activities and in the literature is referred to as the Ontology Mapping (or Matching) Problem (OMP) [21].

OMP concerns discovering correspondences (*aka* mappings) among entities belonging to different ontologies (i.e., a *source* and a *target* ontology). The problem becomes more challenging in P2P networks for several reasons: (i) the number of overlapping ontologies can dramatically increases, in theory each peer will have its own ontology that reflects peer's needs and interests; (ii) mappings among peer ontologies must be quickly discovered and only on the parts of ontologies "contextual" to a specific interaction in which peers are involved; (iii) complex mapping strategies (e.g., structural mapping based on graph matching) cannot be exploited since peers are not aware of one another's ontologies. Thus, ontology mapping algorithms for P2P networks should ensure a trade-off between fastness (not achievable by adopting complex mapping strategies) and accuracy (i.e., quality of results).

To date, several approaches to solve the OMP have been proposed [11]. These are based on techniques borrowed from various research areas such as Bayesian Decision

theory (see OMEN [36] and [38]) Graph Similarity (see GMO [51]) Information Retrieval (see LOM [41] and V-doc [42]) just to name a few of them. However, these approaches underestimate the following aspects:

- They do not adequately consider the OMP in open environments such as P2P networks. A recent survey on ontology mapping [11] contains only a bibliographic reference to mapping systems designed for P2P networks.
- They do not take into account the need for "on the fly" mappings crucial in P2P networks. In such networks, a complete mapping between peer ontologies is not a requirement for interactions among peers; they only need to quickly map the parts of their ontologies related to the specific interaction in which they are involved. Moreover, since peers are often unaware of one another's ontologies the amount of ontological information exploitable to discover mappings is quite limited.
- They do not adequately interpret the semantic meaning of ontology concepts to be compared. In addition, the context in which concepts appear is not carefully scrutinized from a semantic point of view. Even if there are some approaches addressing these issues, they often are not designed on the basis of well-founded experimental results.

In this paper, we address the OMP in P2P networks by defining a new ontology mapping algorithm called SEmantiC COordinator (*SECCO*). We especially focused on the OMP in P2P environments since P2P applications seem to be a class of applications that will take advantage of Semantic Web technologies in a near future. *SECCO* is composed by three individual matchers: *syntactic*, *lexical* and *contextual*, each of which tackles the OMP from a different perspective. In particular, the *syntactic matcher* aims at discovering mappings by an Information Retrieval approach called LOM [41] that exploits linguistic information (e.g., comments, labels) encoded in ontology entities. The *lexical matcher* assesses semantic relatedness, even among syntactically unrelated concepts (e.g., *car* and *automobile*), by combining two approaches exploiting WordNet [35] as background knowledge. The *contextual matcher* implements a new similarity strategy called "how it fits". This strategy complies with the contextual theory of meaning [34] and is founded on the idea that two concepts are related if they fit well in each other's context. This approach allows comparing the structures of two concepts both in terms of their position in the ontological taxonomy and constituent properties. This is achieved at an affordable computational cost since it is not required to take into account the whole structure of ontologies.

Specifically, the main contributions of the paper are:

- We exploit the idea of *concept mapping* with the aim to gather similarity information among concepts belonging to different peer ontologies. *Concept mappings* allow building *semantic links* among peers that can be exploited in several classes of semantic P2P applications (e.g., semantic search, semantic query routing, and community formation).
- We designed and extensively evaluated *SECCO*, which is endowed with three new mapping strategies facing the OMP from a different perspective but that share accuracy and fastness. In particular, *concept mappings* are derived by combining the results of these three mapping strategies.

- Differently from other semantic P2P applications (e.g., Piazza, GridVine) that assume the preexistence of mappings for achieving semantic interoperability, we focus on the problem of finding mappings.

Extensive experimental results, aimed at comparing *SECCO* w.r.t the state of the art, show that the combination of the proposed mapping strategies provides an adequate trade-off between accuracy (in terms of quality of mappings) and fastness (in terms of elapsed time for discovering mappings).
Moreover, we want to emphasize that *SECCO*, if coupled with a P2P platform, paves the way towards a comprehensive semantic P2P solution for content sharing and retrieval, semantic query answering and semantic routing. We report on the advantages of integrating *SECCO* within the *K-link+* system [31].

The remainder of this paper is organized as follows. Section 2, after introducing the terminology adopted in the rest of the paper, presents the *SECCO* ontology mapping algorithm. Section 3 describes and evaluates the individual matchers of *SECCO*. In Section 3.4 we motivate the design of *SECCO*. Section 4 presents a detailed evaluation of the system in two different settings. In particular, Section 4.1 compares *SECCO* with H-Match [8, 9] and performs a sensitivity analysis of the different parameters of the algorithm (Section 4.1.1). Then, Section 4.2 evaluates the algorithm on four real-life ontologies of the OAEI 2006 benchmark test suite. Here *SECCO* is also compared with other mapping algorithms not explicitly designed for facing the OMP in P2P networks. Section 5 reviews related work. Section 6 draws some conclusions. Finally, Section 7 sketches future work.

2 The *SECCO* ontology mapping algorithm

We designed the *SECCO* ontology mapping algorithm for addressing the OMP in P2P networks, since semantic P2P applications, built by interconnecting knowledge managed at personal level, seem to be the applications taking advantage of Semantic Web technologies in a near future [4]. We argue that most of the existing mapping algorithms are not suitable for P2P networks since they, to work properly, need to deal with the whole two ontologies to be mapped. For instance, top ontology mapping algorithms, i.e., Falcon [25], and RiMOM [54] have structural mapping strategies built upon graph matching techniques. These techniques are well suited to work "offline" while they are not applicable in P2P environments where the OMP has to be faced "online" and peers are not aware of one another's ontologies.
In this section, after introducing the terminology adopted in the rest of the paper (Section 2.1) we describe the ontology model exploited by *SECCO* to discover mappings (Section 2.2). Section 2.3 presents a scenario of usage of *SECCO* and provides the pseudo-code of the algorithm.

2.1 Preliminary definitions

We consider a P2P network in which each peer owns an *ontology* (i.e., peer ontology) that represents the point of view of the peer on a particular knowledge domain. Each

(*seeker*) peer can request to other (*providers*) peers a *concept mapping* whose aim is to provide information of similarity among a concept belonging to the *seeker* peer ontology and concepts belonging to ontologies of *provider* peers. The aim of the request depends on the application class, as will be discussed in Section 3.4. In this scenario, we define both *seeker* and *provider* peers as *semantic* peers since they manage, share and exchange knowledge by exploiting ontologies.

An ontology is basically composed of two parts: (i) the *intensional model*, represented by means of an ontology schema and (ii) the *extensional part*, implemented by a knowledge base. In this paper, we adopt the following simplified ontology model that is inspired by the formal ontology definition proposed in [12].

Definition 1 (***SECCO* Ontology model**). The *SECCO* ontology model is a six-tuple of the form:

$$O=\langle C,R,L,\leq,\varphi_R,\varphi_L\rangle$$

consisting of a set of concepts names C, a set of relation names R and a set of strings L that contains ontology metadata like comment(s) and label(s). Concepts names are arranged in a hierarchy by means of the partial order $\leq$ (which intrinsically defines the ISA relation). The signature φ_R: $R \rightarrow C\times C$ associates each relation name $r\in R$ with a couple of concepts. Given a relation name $r\in R$, the first attribute of the tuple defines the relation domain $dom(r)=\pi_1(\varphi_R(r))$ and the second attribute the range $range(r)=\pi_2(\varphi_R(r))$. The signature φ_R: $C\cup R \rightarrow 2^L$ associates each concept and relation with a subset of strings representing its metadata.

This simplified ontology model is built up starting from OWL ontologies as described in Section 2.2.

Definition 2 (Seeker peer). A *seeker* peer is a semantic peer that sends a semantic *request* over the P2P network to *provider* peers and receives a set of *concept mappings*.

Definition 3 (Provider peer). A *provider* peer is a semantic peer that receives a *request* from a *seeker* peer and returns a *concept mapping* obtained by exploiting *SECCO*.

Definition 4 (Request). Let O be an ontology, a *request* is a two-tuple of the form $RQ=\langle c,ctx(c)\rangle$ where $c\in C$ is a concept belonging to the *seeker* peer ontology and ctx(c) is the *context* of c.

Definition 5 (Concept Context). Let O be an ontology, the set of strings ctx(c) is the *context* of the *concept* $c\in C$. This set contains names of concepts related to c by relations in R that correspond to OWL *objectype* properties [39] and the names of relations in R that correspond to OWL *datatype* properties [39]. More formally, $ctx(c)=C_{range}\cup C_{dom}\cup R_{dt}$ where: (i) C_{range} and C_{dom} are the sets of concepts names for which respectively hold the following conditions: $c\in dom(r)\wedge c_{range}\in range(r)$ and $c_{dom}\in dom(r)\wedge c\in range(r)$ with $r\in R$ and *range*(r) and *dom*(r) both corresponding to user defined classes; (ii) R_{dt} is the set of relation names for which either *range*(r) or *dom*(r) is defined on a data type [39].

Datatype property names, present in the original OWL ontology, are included in ctx(c) as described in Section 2.2.

The *concept mapping* between a *seeker* peer concept and the set of concepts belonging to a *provider* peer ontology is defined as follows:

Definition 6 (Concept Mapping). Given the *seeker* peer request $RQ=\langle s,ctx(s)\rangle$ and the ontology O belonging to a *provider* peer, a *concept mapping* M between each *provider* concept $p \in C$ and the *seeker* peer concept s is a set of 3-tuples of the form $M=\langle s,p,\sigma\rangle$ where $\sigma \in [0,1]$ is the similarity value between the couple of concepts s and $p \in C_p$.

Similarity values between couples of concepts are obtained by adopting the similarity measure defined as follows:

Definition 7 (Similarity Measure). Given two ontologies O_s and O_p belonging to a *seeker* and a *provider* peer respectively, a request $RQ_s=\langle c_s,ctx(c_s)\rangle$ and the set CTX_p composed by two-tuples of the form $\langle c_j,ctx(c_j)\rangle$ where $\forall c_j \in C_p$ $ctx(c_j)$ is the context of c_j; the similarity between the couples of concepts $c_s \in C_s$ and $c_p \in C_p$ is computed by the following function:

$$sim(c_s, c_p) : \{sim_{syn}(c_s, c_p), sim_{lex}(c_s, c_p), sim_{con}(c_s, c_p)\} \rightarrow [0.1]$$

where $sim_{syn}(c_s,c_p)$: $C_s \times C_p \rightarrow [0,1]$ is the *syntactic similarity*, $sim_{lex}(c_s,c_p)$: $C_s \times C_p \rightarrow [0,1]$ is the *lexical similarity*, $sim_{con}(c_s,c_p)$: $RQ_s \times CTX_p \rightarrow [0,1]$ is the *contextual similarity*. These three similarity measures are symmetric and reflexive i.e., $\forall c_s \in C_s$ and $\forall c_p \in C_p$,

$sim(c_s,c_s)=1$ (reflexivity)
$sim(c_s,c_p)=sim(c_p,c_s)$ (symmetry)

How to represent mappings in SECCO

Even if the OMP has received a lot of attention from the scientific community, a standardized format for storing ontology mappings does not exist. In order to overcome this problem, there are two possible ways:

1. Exploiting features of ontology languages. For instance, OWL provides built-in constructs for representing equivalence between concepts (i.e., *owl:equivalentClass*), relations (i.e., *owl:equivalentProperty*) and instances (i.e., *owl:sameAs*). This approach allows OWL inference engines to automatically interpret the semantics of mappings and perform reasoning across different ontologies. However, by adopting this approach, a confidence value cannot be interpreted.
2. Adopting the approach described in [20]. This mapping representation exploits RDF/XML to formalize ontology mappings. Each individual mapping is represented in cells and each cell has the following attributes: *entity 1* (i.e., the concept in the *source* ontology), *entity 2* (i.e., the concept in the *target* ontology), *measure* (i.e., the confidence value), *type of mapping* (usually equivalence). Due to its different parameters, this representation can easily be exploited by several kinds of applications.

In *SECCO*, we adapt the second type of representation to the context of a P2P ontology mapping system. The adopted mapping representation is depicted in Fig.1. This representation allows a *seeker* peer (i.e., the *seeker_peer* tag), for a given *seeker* concept (i.e., the *seeker_concept* tag), to maintain both the URIs of *provider* concepts (i.e., *provider_concept* tag) and values of similarity (i.e., the *similarity* tag) grouped on the basis of *provider* peers (i.e., the *provider_peer* tag) that answered to the *seeker* request.

```
<mapping>
   <seeker_peer name= ID>
   <seeker_concept=URI>
   <provider_peer name = ID>
            <provider_concept ID=URI>
                 <similarity>σ</similarity>
            </provider_concept>
            <provider_concept ID=URI>
                 <similarity> σ</similarity>
            </provider_concept>
             …
   </provider_peer name>
     ...
</mapping>
```

Fig. 1. Representation of mappings in *SECCO*.

2.2 The *SECCO* ontology model construction

The *SECCO* ontology model (see Definition 1) is built by exploring ontology class definitions contained in peer ontologies. To explain how the *SECCO* ontology model is constructed, let us consider the fragment of the *Ka* ontology (available at http:// www.cs.man.ac.uk/~horrocks/OWL/Ontologies/ka.owl) depicted in Fig. 2.

```
<owl:Class rdf:about="file:F:/Projects/OIL/DAMLOilEd/ontologies/ka.daml#Publication">
 <rdfs:subClassOf>
     <owl:Restriction>
         <owl:onProperty rdf:resource="file:F:/Projects/OIL/DAMLOilEd/ontologies/ka.daml#title"/>
         <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
     </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
   <owl:Restriction>
   <owl:onProperty rdf:resource="file:F:/Projects/OIL/DAMLOilEd/ontologies/ka.daml#describesProject"/>
         <owl:allValuesFrom>
             <owl:Class rdf:about="file:F:/Projects/OIL/DAMLOilEd/ontologies/ka.daml#Project"/>
         </owl:allValuesFrom>
   </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
     <owl:Restriction>
         <owl:onProperty rdf:resource="file:F:/Projects/OIL/DAMLOilEd/ontologies/ka.daml#abstract"/>
         <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
     </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
     <owl:Restriction>
         <owl:onProperty rdf:resource="file:F:/Projects/OIL/DAMLOilEd/ontologies/ka.daml#year"/>
         <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
     </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="file:F:/Projects/OIL/DAMLOilEd/ontologies/ka.daml#keyword"/>
        <owl:allValuesFrom rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
      </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="file:F:/Projects/OIL/DAMLOilEd/ontologies/ka.daml#Book">
        <rdfs:subClassOf>
            <owl:Class rdf:about="file:F:/Projects/OIL/DAMLOilEd/ontologies/ka.daml#Publication"/>
        </rdfs:subClassOf>   OMISSIS
</owl:Class>
```

Fig. 2. A fragment of the *Ka* ontology

Given an input ontology *SECCO* executes the following four main steps to construct its ontology model:

1. *Class name extraction*: for each class a concept name is created in the *SECCO* ontology model.
2. *Subclass properties* (i.e., ISA) *analysis*: for each class definition, *SECCO* scrutinizes its sub classes defined by the construct *rdfs:subClassOf* and generates the taxonomic structure.
3. *Datatype properties analysis*: values of these properties are data literals. For each datatype property *SECCO* considers the linguistic information encoded in the property name (e.g., *year* in Ka) and includes in its ontology model, a new concept name representing the property (i.e., *year*) and a relation (i.e., *has_year* as shown in Fig. 3) to relate this new concept with the original class.
4. *Object properties analysis*: these are properties for which the value is an individual. In this case, for each property, *SECCO* exploits the original OWL encoding and generates a relation, in its ontology model, that has the same name, domain and range of the original one.

The ontology fragment depicted in Fig. 2, contains the definition of a class *Publication* along with some object and datatype properties, and related classes. By running *SECCO,* we obtain the ontology model representation depicted in Fig. 3. Notice that the construction of this representation also exploits the definitions of classes (e.g., *Project*, *Event*) that are not represented in the excerpt shown in Fig.2.

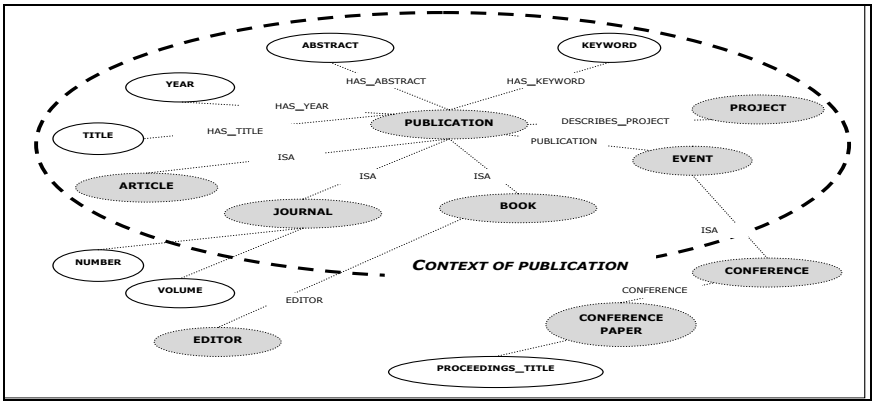

Fig. 3. The *SECCO* representation of the *Ka* ontology related to the concept *Publication*.

In Fig. 3 filled oval represent ontology concepts (i.e., classes) as defined in the original OWL ontology whereas empty ovals are the concepts introduced in the *SECCO* ontology model to exploit information encoded in *Datatype properties*.
The context of the *Publication* concept, as defined in Definition 5, is the dashed area in Fig. 3. In more detail, *ctx(Publication)={title, year, abstract, keyword, Project, Event, Book, Journal, Article}*.

2.3 The *SECCO* ontology mapping algorithm

SECCO aims at discovering a *concept mapping* between a *seeker* peer concept and ontology concepts belonging to ontologies of *provider* peers. Each peer in the network plays a twofold role: (i) *seeker* peer, when it sends a request to the network; (ii) *provider* peer, when it executes locally the *SECCO* algorithm. Whenever a *provider* peer receives a request, it runs *SECCO* with an input of the following form:

$$I=<c_s, ctx(c_s), O_p, T_h, w_s, w_L, w_c>$$

where: c_s is a concept belonging to a *seeker* peer ontology; $ctx(c_s)$is the context of c_s; O_p is the *provider* ontology, $T_h \in [0,1]$ is a threshold value that can be used for filtering results. Moreover, w_s, w_L, and w_c are the weights assigned to the values of *syntactic*, *lexical* and *contextual* similarity respectively. The overall similarity value is computed by the *Combiner* module that weights the similarity values provided by the individual matchers (see Fig. 4) and discards values that do not exceed a given threshold (i.e., the T_h parameter in Fig.4). Once *SECCO* has terminated, it returns a *concept mapping* as defined in Definition 6.

The overall approach is described in Fig. 4. A *seeker* peer issues an information request by picking a concept along with the related context from its ontology. This request reaches *provider* peers that run the *SECCO* algorithm on their ontologies and return to the *seeker* peer *concept mappings* that will be stored in the *mapping store* for future reuse. Fig. 5 describes the *SECCO* algorithm in a pseudo-code.

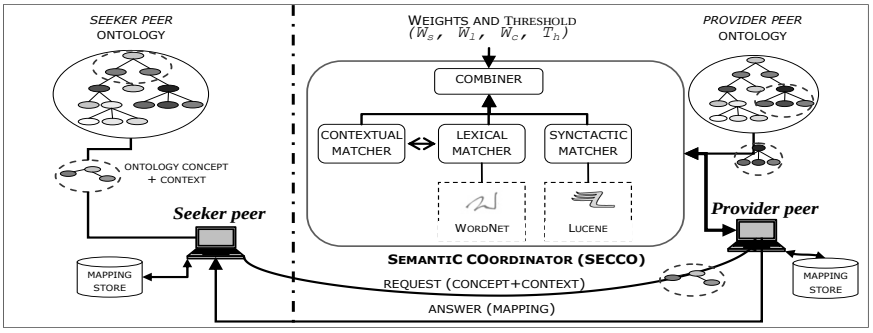

Fig. 4. The *SECCO* architecture and usage scenario.

The function *evaluate_syntactic_similarity* is implemented by the *syntactic matcher* (see Section 3.1) while the function *evaluate_lexical_similarity*, is implemented by the *lexical matcher* (see Section 3.2). The function *evaluate_contextual_similarity* (see Fig. 7), implemented by the *contextual matcher* (see Section 3.3), relies on the function *evaluate_how_it_fits* (see Fig. 8) that adopts a "see how it fits" strategy that is founded on the idea that two concepts are related if they *fit well* in each other's context (see Section 3.3). The *contextual matcher* takes as input the context obtained by the function *extract_context* (see Fig. 6).

The *SECCO* algorithm

Input: An input $I=<c_s, ctx(c_s), O, T_h, w_s, w_L, w_c>$ where $O=\langle C,R\rangle$ *is the SECCO ontology model*
Output: The concept mapping *M*
Method:

```
1.  M=∅;
2.  for each  c∈C do
3.    sim_syn=evaluate_syntactic_similarity(c_s,c);
4.    sim_lex=evaluate_lexical_similarity(c_s,c);
5.     ctx(c)=extract_context(c,O); /*Fig.6*/
6.     sim_con=evaluate_contextual_similarity(c_s,Ctx(c_s),c,Ctx(c));
7.     sim=(w_s*sim_syn+w_L*sim_lex+w_c*sim_con); /* overall similarity value */
8.      if sim>T_h then
9.         m.s=c_s
10.        m.p=c;
11.        m.σ=sim
12.        M=M∪m;
13.     end-if
14. end-for
15. return M;
```

Fig. 5. The *SECCO* algorithm in pseudo-code.

Function extract_context

Input: An ontology $O=\langle C,R\rangle$ *and* a concept $c\in C$
Output: The context *ctx(c)*
Method:

```
1.     ctx_c=∅; ctx_r=∅;
2.     for each c_c∈C do
3.         for each r_c∈R do
4.             if ∃r_c(c,c_c)|∃r_c(c_c,c) then
5.                ctx_c=ctx_c∪{c_c}
6.                ctx_r=ctx_r∪{r_c}
7.             end-if
8.         end-for
9.     end-for
return ctx(c)=⟨ctx_c,ctx_r⟩;
```

Fig. 6. The *extract_context* function.

Function evaluate_contextual_similarity

Input: Two concepts c_1 and c_2 and their contexts $ctx(c_1)$ and $ctx(c_2)$
Output: A numerical value $sim_{con}\in[0,1]$ representing the contextual similarity between the concepts c_1 and c_2
Method:

```
1.     s2s=evaluate_how_it_fits(c_1, ctx(c_1)); /* see Figure 8 */
2.     s2t=evaluate_how_it_fits(c_1, ctx(c_2));
3.     t2s=evaluate_how_it_fits(c_2, ctx(c_1));
4.     t2t=evaluate_how_it_fits(c_2, ctx(c_2));
5.     sim_con=((1-||s2s-t2t|-|s2t+t2s||));
6.     return sim_con
```

Fig. 7. The *evaluate_contextual_similarity* function.

```
                    Function evaluate_how_it_fits
Input: A concept c and a context ctx(x)= ⟨C_x,R_x⟩
Output: A numerical value m∈[0,1] representing the fitness between the
concept c and the context ctx(x)
Method:
    1.    T=0;
    2.    for each c_e∈C_x do
    3.          T+=evaluate_lexical_similarity(c,c_e);
    4.    end-for
    5.    return m=T/|ctx(x)|;
```

Fig. 8. The *evaluate_how_it_fits* function.

In the next section, the individual matchers of *SECCO* are described and evaluated.

3 Individual matchers: the building blocks of *SECCO*

The idea of combining different heuristics, each of which implemented by an individual matcher, for the assessment of an overall similarity value between two ontology entities is not new (see [16,17]). The main motivation of adopting such a strategy is that from some ontology mapping initiatives (e.g., [19]) emerged that a combination of mapping strategies, in general, allows to obtain better results. Moreover, it is not arguable that a single heuristic is able to exploit all the types of information (e.g., lexical, structural) encoded in ontology entities.

With these motivations in mind, we decided to endow *SECCO* with three different matchers. Each matcher respectively exploits syntactic/linguistic (*syntactic matcher*), lexical (*lexical matcher*) and contextual (*contextual matcher*) information contained in ontology entities. We adopt the *syntactic matcher*, since in our previous work on ontology mapping [41] we noticed that a merely syntactic approach can be effective and fast in discovering mappings. The *lexical matcher* is successful in discovering mappings in a semantic way, that is, by considering the semantic meaning of the compared terms and not treating them just as strings. Through this approach, it is possible, for instance, to discover that the *automobile* concept used in a *seeker* peer ontology is similar to the concept of *car* used in a *provider* peer ontology. Finally, the *contextual matcher* that relies on the *lexical matcher* allows refining similarity between concepts by considering the contexts in which they appear. The *contextual matcher* rationale complies with the contextual theory of meaning [34] according to which the relatedness between concepts can be defined in terms of their interchangeability within the contexts in which they appear. The *contextual matcher* allows the assessment of similarity between two concepts in terms of their structure/properties, but on a local basis, that is, by only considering the properties and neighbors of the two concepts and not the whole ontology structures in which they appear. Notice that for the scenario in which *SECCO* has to work (i.e., a P2P network) a structural matching strategy could affect the requirement of time accuracy given that it requires to compare entire ontologies (e.g., [51]). Indeed, in *SECCO*, a *provider* peer only receives a request (i.e., a concept along with its context) from a *seeker* peer and not its entire ontology. Through experimental evaluations (see

Section 4), we prove that the lack of a structural matcher will not significantly affect mapping results. Furthermore, it is worthwhile noting that the modular architecture of *SECCO* allows easily designing and adding new matchers to be included into the algorithm. In the Sections 3.1-3.3, we provide both a description and an evaluation of the three individual matchers. Section 3.4 motivates the designing of *SECCO* by reporting on the advantages of integrating it in *K-link+* [31], a P2P system for collaborative work and content sharing and retrieval.

3.1 The Syntactic Matcher

This matcher that implements the function *evaluate_syntactic_similarity* (see Fig. 5, line 3), mainly relies on the Lucene Ontology Matcher (LOM) described in our previous work on ontology mapping [41]. Here we provide an overall description of LOM, further details along with complete experimental results can be found in [41].
Given a *source* ontology O, in order to discover mappings, LOM aims at exploiting metadata of ontology entities (e.g., comments, and labels) contained in L (i.e., linguistic information). In particular, for each entity e in C∪R a *virtual document* that contains its metadata in L is encoded by exploiting the concept of Lucene Document [33]. Virtual documents are stored into an index maintained in main memory. Ontology mappings are derived by using values of entities of a *target* ontology as search arguments against the index created from the source ontology. Similarity values are computed by exploiting the scoring schema implemented in Lucene, which relies on Vector Space techniques [45].

In order to show the suitability of LOM in terms of both speed and accuracy, here we report on its evaluation on the OAEI 2006 benchmark test suite [37] as compared to three string matching techniques (i.e., I-Sub [49], Jaro Winkler [52] and Edit Distance [32]) that are typically exploited to perform syntactic matching of ontology entities. Ontologies in the OAEI benchmark test suite are based on one particular ontology defined in the bibliography domain and a number of variations of such ontology for which alignments are provided. There are different categories of alteration related to both linguistic aspects (variation in names and comments of entities) and structural aspects (variation in relations among entities). The benchmark is composed of five groups of tests that are constructed on the basis of the above mentioned types of alterations. Fig. 9 shows the average results obtained by LOM in terms of Precision (i.e., the number of correct mapping among all the mapping found), Recall (i.e., the number of correct mapping among all the existing mapping) and F-measure (i.e., the harmonic mean of Precision and Recall) [15].

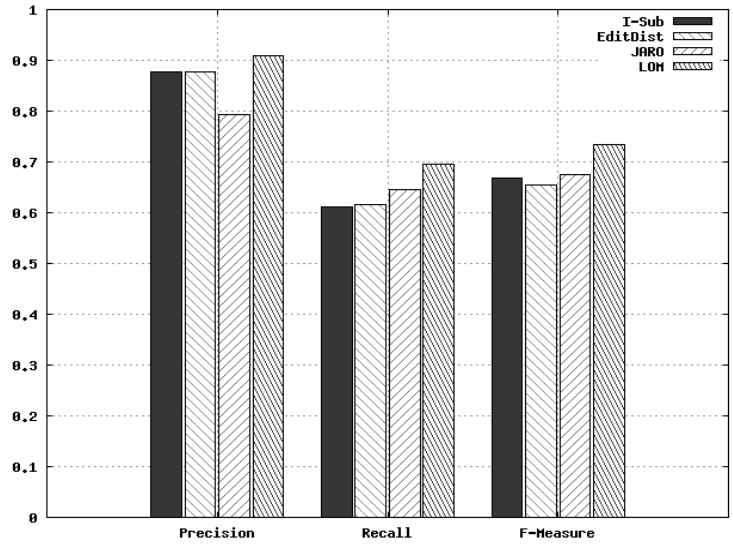

Fig. 9. Evaluation of LOM on the OEAI 2006 benchmark test suite.

As can be noticed, LOM outperforms the competitors. This is because it can profitably exploit all the linguistic information included in ontologies. In fact, even in test cases where entity names are altered (e.g., randomized, expressed in another language) LOM, by exploiting other linguistic information (e.g., labels, comments), can correctly assess similarity values. Moreover as discussed in [41] the average time for computing a mapping between two ontologies of the OAEI tests is 1.47s.
In the light of these considerations, LOM can be exploited as an individual matcher of *SECCO*, instead of a classical string-based approach, since it is more effective in terms of accuracy and is adequately fast.
In particular, in order to adopt LOM in *SECCO* we made the following adaptations:

- Ontologies of both *seeker* and *provider* peers are indexed. Therefore, each peer exploits its index to search for similar entities for requests coming from *seeker* peers (i.e., acts as a *provider*).
- Since in *SECCO* we do not want to compare whole ontologies, but a *seeker concept* along with its context with *provider* ontology concepts, we construct a new type of *virtual document* that contains linguistic information of the concept along with linguistic information of its context. This way, the linguistic information of a concept is augmented with linguistic information of entities in its context. Therefore, also the *syntactic matcher* takes into account a certain degree of structural information.

3.2 The Lexical Matcher

The *lexical matcher*, that implements the function *evaluate_lexical_similarity* (see Fig. 5, line 4), is the central component of the whole system. It allows implementing

the semantic mapping by "interpreting" the semantic meaning of concepts to be compared. The *lexical matcher* exploits WordNet [35] as a source of knowledge about the world. WordNet is a lexical ontology organized in synsets (or senses) that encompass terms with synonymous meaning. Each synset has a gloss, which is a description in natural language of the concepts it represents. Synsets are connected to one another by a predefined set of semantic relations, some of which are reported in Table 1.

Table 1. Semantic relations between synsets in the WordNet 3.0 noun taxonomy.

Relation	Description	Example
Hypernymy	*is a generalization of*	*Plant* is an hypernym of *Flower*
Hyponymy	*is a kind of*	*Tulip* is hyponym to *Flower*
Meronymy	*is a part of*	*Finger* is a meronym of *Hand*
Holonymy	*contains part*	*Tree* is a holonym of *Bark*
Antonymy	*opposite of*	*Man* is an antonym of *Woman*
Instance of	*is an instance of*	*California* is an instance of *American state*
Has instance	*has instance*	*American state* has instance *California*

Some of these relations define inheritance relations (Hypernymy and Hyponymy), other part-of relations (Holonymy and Meronymy). The Antonymy relation is used to state that a noun is the opposite of another. The relations *instance of* and *has instance* have been introduced in WordNet 3.0 and represent instantiation relations. Fig.10 shows an excerpt of the WordNet noun taxonomy.

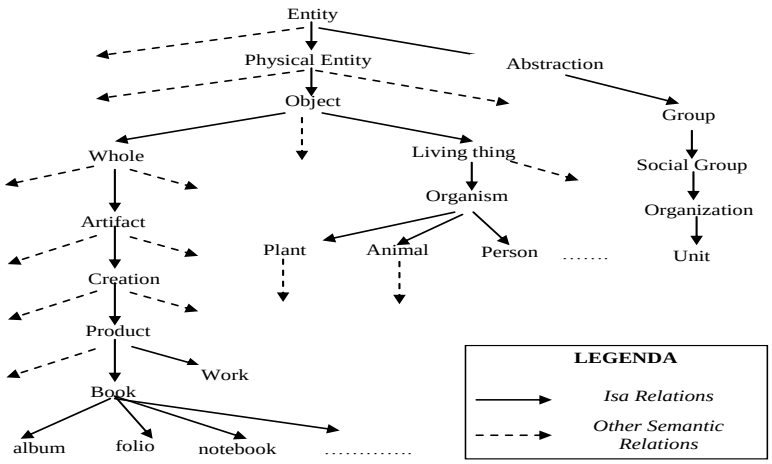

Fig. 10. An excerpt of the WordNet 3.0 noun taxonomy.

Through the *lexical matcher* we aim at assessing the relatedness between ontology entities by exploiting their definitions within the WordNet database and position in the taxonomy. Semantic relatedness is the question of how related two concepts are by considering different kinds of relations connecting them. On the other hand, semantic similarity only considers the hypernymy/hyponymy relations among

concepts. For instance, *Car* and *Gasoline* may be closely related to each other, e.g. because gasoline is the fuel most often used by cars. *Car* and *Bicycle* are semantically similar, not because they both have wheels and means of steering and propulsion, but because they are both instances of *Vehicle*. The relation between semantically similar and semantically related is asymmetric: if two concepts are similar, they are also related, but they are not necessarily similar just because they are related.
In literature (see [54]), there are several metrics for assessing similarity and relatedness among concepts in WordNet. In the context of ontology mapping, several approaches (e.g., [29]) compute semantic similarity between concepts by exploiting semantic similarity metrics. However, these approaches only consider the hypernymy / hyponymy relations linking synsets.
In order to take into account a wide range of semantic relations connecting synsets we included two components in the *lexical matcher*. A *similarity assessor* aimed at assessing semantic similarity and a *relatedness assessor* aimed at assessing semantic relatedness. The final lexical similarity value is obtained by combining the contribution of the two assessors.

3.2.1 The semantic similarity assessor

The semantic similarity assessor aims at exploiting the structure of WordNet, which contains *per se* a certain degree of semantic information encoded in synsets. In the literature several approaches to compute semantic similarity are presented. In order to choose the most appropriate one, we evaluated results of several approaches and correlated them w.r.t human judgments of similarity. A detailed description of the dataset and evaluation methodology along with complete experimental results can be found at http://grid.deis.unical.it/similarity.

Among the evaluated metrics, the most performant are these based on the notion of Information Content (IC). IC can be considered a measure that quantifies the amount of information a concept expresses and is computed as log the negative likelihood of the occurrences of a concept in a large corpus. Resnik in [43] exploited the notion of IC for assessing semantic similarity between terms in a taxonomy. The basic intuition behind the use of the negative likelihood is that the more probable a concept is of appearing then the less information it conveys, in other words, infrequent words are more informative than frequent ones. Knowing the IC values for each concept, we may then calculate the similarity between two given concepts.
In the *lexical matcher* we adapt the Jiang and Conrath distance metric (J&C) [27]. This metric computes the semantic similarity between two concepts c_s and c_p as follows:

$$sim(c_s, c_p) = 1 - \frac{IC(c_s) + IC(c_p) - 2 * IC(sub(c_s, c_p))}{2} \quad . \qquad (\mathbf{1})$$

We consider the opposite of the semantic distance metric defined by J&C, as a similarity measure. Moreover, in order to quantify IC of concepts we exploit the function *IC* defined as follows [46]:

$$IC(c) = 1 - \frac{\log(hypo(c) + 1)}{\log(\max_{wn})} \quad . \qquad (\mathbf{2})$$

where the function *hypo* returns the number of hyponyms of a given concept *c*. Notice that concepts that represent leaves in the taxonomy will have an IC equals to one. Moreover, max_{wn} is a constant that indicates the total number of concepts in the WordNet noun taxonomy (i.e., 82115 in WordNet 3.0).
The function $sub(c_s,c_p)$ in equation 1 returns the concept (the lowest in the taxonomy) that subsumes both c_s and c_p.

3.2.2 The semantic relatedness assessor

In order to select the most appropriate relatedness assessor we evaluated several approaches. A complete description of the dataset and evaluation methodology along with complete experimental results is available at the similarity experiment website: http://grid.deis.unical.it/similarity.
In our evaluation, we found that the gloss vector relatedness metric described in [40] is the most correlated w.r.t human judgment. This metric is based on the following intuition: the relatedness between two concepts can be assessed by comparing their glosses. In particular, this approach exploits "second order" vectors for glosses, that is, rather than just matching words that occur in glosses, the words in the gloss are replaced with co-occurrence (extracted from a corpus) vectors. Therefore, each gloss is represented by the average of its word vectors. Hence, pairwise comparisons can be made between vectors to measure relatedness between the concepts they represent. In the following, we summarize the step followed to compute the relatedness between two concepts c_s and c_p:

1. Get the gloss of c_s from WordNet. Create a gloss vector by adding the word vectors of all the words in the gloss.
2. Get the gloss of c_p from WordNet. Create a gloss vector by adding the word vectors of all the words in this gloss.
3. Compute the cosine of the gloss-vectors. In addition, this metric use the relations represented in Table 1 to augment the glosses of c_s and c_p, with gloss information of concepts that are directly linked to c_s and c_p. This makes the augmented glosses of c_s and c_p much bigger than the just the glossed of c_s and c_p.

If v_s and v_p are the gloss vectors for c_s and c_p, their relatedness in computed as follows:

$$relat(c_s, c_p) = \frac{\vec{v_s} * \vec{v_p}}{|\vec{v_s}| * |\vec{v_p}|} \quad . \tag{3}$$

Computing the overall lexical similarity score

Overall, the lexical similarity is computed as a weighted sum of the scores provided by the two assessors:

$$\mathrm{sim}_{lex}(c_s, c_p) = w_s * sim(c_s, c_p) + w_r * relat(c_s, c_p) \quad . \tag{4}$$

From experimental evaluation, we found that equally weighting the two contributions (i.e., assigning 0.5 to both w_s and w_r) gives the best accuracy in terms of correlation w.r.t human judgment.

Reducing the elapsed time

Since WordNet is a huge lexical database, some performance issues related to its access can arise. In order to provide a fast access to the database and implement our similarity and relatedness measures we built an ad-hoc Lucene index that maintains the information about synsets. In particular, both values of IC and gloss vectors are stored in the index. The index is built by parsing the Prolog release of WordNet [53].

A running example

In order to see how the *lexical matcher* works, let us compute the lexical similarity between the concepts *Animal* and *Person*. According to eq. (4) we have to compute the semantic similarity and relatedness between the two concepts.

Computing semantic similarity

For the semantic similarity, we have to calculate the following coefficients:

- IC(*Animal*): *Animal* in the WordNet taxonomy has 3998 hyponyms, therefore according to equation (2) we have that IC(*Animal*)=0.2670.
- IC(*Person*): *Person* has 6978 hyponyms, in this case we have: IC (*Person*)=0.2178.
- IC(*Organism*): *Organism*, which subsumes both concepts (see Figure 10) has 16110 hyponyms. Therefore we have IC(*Organism*)=0.1439.

The semantic similarity value according to equation (1) is

$$sim(Animal, Person) = 0.9014$$

Computing semantic relatedness

In order to compute semantic relatedness between *Animal* and *Person* we have to compare their glosses augmented with glosses of neighbors concepts. The neighbors concepts of a given concept are concepts related to it by any of the relations reported in Table 1. In our example, the gloss of the concept Person is" *a human being*...". The gloss of the concept Animal is "*a living organism characterized by voluntary movement ... ".* Each of these concepts has a representative vector which contains for each dimension a number indicating the frequency of the word encoded in that dimension. Here we do not report the vectors of the two concepts since they have a very large dimension (about 12000). The semantic relatedness between *Animal* and *Person* is:

$$relat (Animal, Person) = 0.4667$$

Overall, the lexical similarity between *Animal* and *Person* is:

$$sim_{lex} (Animal, Person) = 0.5*0.9014+0.5*0.4667=0.6840$$

Compound terms

The *lexical matcher* treats compound terms by following the heuristic that in English the last token appearing on the right side of a compound term denotes the central concept, while other concepts encountered from the left side to the right side of denote a qualification of its meaning [30].

Remark

To summarize, our *lexical matcher* is a good candidate for being included in *SECCO* since it respects the requirements of fastness (by exploiting the Lucene index) and accuracy (proven by several experimental evaluations whose results are available at http://grid.deis.unical.it/similarity). The *lexical matcher* is included in the Java WordNet Similarity Library (JWSL) [26], which is a Java-based library that provides access to information about WordNet Synsets and implements a variety of similarity and relatedness metrics.

3.3 The Contextual Matcher

The aim of the *contextual matcher* is to implement the *evaluate_contextual_similarity* function (see Fig. 5, line 6) exploited to refine similarity values assessed by the *syntactic* and/or *lexical matcher*. It advances a contextual approach to semantic relatedness that builds upon Miller et al. definition in terms of the interchangeability of words in contexts [34]. Contexts help to refine the search of correct mappings since they intrinsically contain both information about the domains in which concepts to be compared are used and their structure in terms of properties and neighbors concepts. Contexts represent possible patterns of usage of concepts and the *contextual matcher* is founded on the idea that similar concepts have similar patterns of usage. If two concepts can be used in a similar context then they are related. A concept C_s (i.e., *seeker* concept) in a context $ctx(c_s)$ (i.e., *seeker* context) not similar to a concept C_p (i.e., *provider* concept) in a context $ctx(c_p)$ (i.e., *provider* context) will likely *fit bad* into $ctx(c_p)$ as well as c_p will do in $ctx(c_s)$. Conversely, if the two concepts can be interchangeably used, that is *fit well* in each other's contexts, then they can be considered related. We call this strategy *how it fits* and, in order to quantify how well a concept fits in a context, we calculate the *lexical similarity* between the concept and all the concepts in the considered context and take the average value (see Fig. 8). The overall *contextual similarity* is computed by exploiting the following similarity indicators:

1. *s2s*: indicates how the *seeker* concept fits in the *seeker* context
2. *s2t*: indicated how the *seeker* concept context fits in the *provider* context
3. *t2t*: indicated how the *provider* concept fits in the *provider* context
4. *t2s*: indicates how the *provider* concept fits in the *seeker* context

The overall contextual similarity is calculated according to the following equation.

$$sim_{con}(c_s, c_p) = 1 - (|s2s - t2t| + |s2t - t2s|) \ . \quad (5)$$

It is worthwhile noting that this strategy aims at taking into account structural information about concepts on a local basis, that is, by only considering properties and nearest neighbor concepts in the taxonomy. This is justified by the fact that a complete mapping among peer ontologies is not required; they only need to map their part of ontologies contextual to the interaction in which they are involved. Moreover, in computing a *concept mapping* by *SECCO*, a *provider* peer is not aware of the whole ontology of the *seeker* peer.

Here we provide a detailed evaluation of the *contextual matcher* on the two excerpts of ontologies depicted in Fig.11.

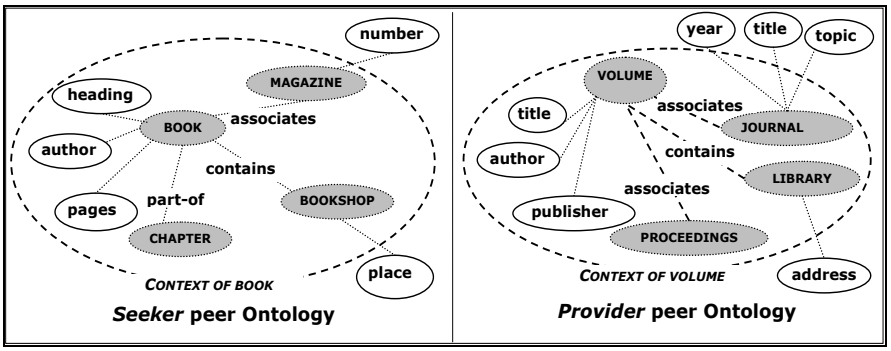

Fig. 11. Excerpts of a *seeker* and *provider* peer ontologies.

We consider *Book* in the ontology of the *seeker* peer, as *seeker* concept, and *Volume* in the ontology of the *provider* peer, as *provider* concept.
In order to assess the contextual similarity between *Book* and *Volume* we start with calculating the coefficients defined in equation (5). In particular, Table 2 and 3 show how the *s2t* and *t2s* coefficients are calculated.

Table 2. Calculation of the *s2t* coefficient.

Seeker concept (*Book*)	**Context of *Volume***	**Lexical Similarity value**
Book	*Journal*	0.8554
	Library	0.5102
	Proceedings	0.3961
	Title	0.5553
	Author	0.4735
	Publisher	0.3105
s2t value	0.5168	
Elapsed time	0.21 s	

Table 3. Calculation of the *t2s* coefficient.

Provider concept (*Volume*)	**Context of *Book***	**Lexical Similarity value**
Volume	*Magazine*	0.7088
	Bookshop	0.2574
	Chapter	0.3179
	Heading	0.3279
	Author	0.3944
	Pages	0.3967
t2s value	0.40	
Elapsed time	0.20 s	

In a similar way, *SECCO* computes values for *s2s* and *t2t*. In the considered example such values are: *s2s*=0.6578 and *t2t*=0.5467. The final contextual similarity between *Book* and *Volume* is 0.7057. Contextual similarity values for other couples of concepts are shown in Table 4.

Discussion of results

Similarity values obtained by the *contextual matcher* underline the fact that the *contextual similarity* between two concepts is affected by concepts and properties included in their contexts. For instance, even if *Book* and *Volume* can be linguistically considered very similar (their *lexical similarity* is 1), the *contextual matcher* correctly decreases their similarity value to 0.7057 (see equation 5) since they respectively appear in a *Bookshop* and *Library* context. Moreover, properties included in the definition of *Book* and *Volume* only share the concept of *author*. The highest

contextual similarity value is obtained by the couple *Bookshop* and *Library*. Even being the two concepts linguistically not so much similar, their lexical similarity is 0.3467, if we only consider the contexts to which they belong, we can observe that the *seeker* context defines a *Bookshop* with *Place* as a property while the *provider* context defines a *Library* with *Address* as property. Both contexts refer to places containing books (one in which they are sold and another in which they are stored) that are characterized by an attribute indicating their location. In this case, the high similarity value obtained by the *contextual matcher* will be refined by the lexical similarity value (which is lower) when weighing their individual contributions.

Table 4. Results obtained by the *contextual matcher* for some couples of concepts in Fig. 11.

Seeker concept	Provider concept	Contextual Similarity	Elapsed time (s)*
Book	*Journal*	0.60567	0.26
Book	*Library*	0.3278	0.25
Book	*Proceedings*	0.1789	0.28
Bookshop	*Journal*	0.3878	0.24
Bookshop	*Library*	0.8067	0.25
Chapter	*Proceedings*	0.60879	0.16
Chapter	*Volume*	0.22345	0.28

By continuing to evaluate further results, the couple *Book* and *Proceedings* receives a low contextual similarity value. They are not lexically very similar (their lexical similarity is 0.3987) and their respective contexts represent different things. The *seeker* context defines a set of properties of a *Book* (e.g., *author*, *pages*) and provides relations with its constituent parts (i.e., *chapter*), with the place where it can be sold (i.e., *Bookshop*) and so forth. Conversely, the *provider* context just provides information about the fact that a *Proceedings* is related to a *Volume*. Similar consideration can be done for the other couples of concepts.
In the light of these considerations, we can conclude that the *contextual matcher* is a suitable approach to interpret the use of concepts in different contexts. In fact, it can correctly interpret similarity between contexts, as in *Library* and *Bookshop*, while it is also able to interpret their dissimilarity, as in the case of the couple *Book* and *Proceedings*. However, it is worthwhile noting that it becomes more effective when combined with the *lexical matcher*, as in the case of the couple *Book* and *Volume*.

Finally, a consideration about elapsed time (see the last column of Table 4). We can see, as one can expect, that the elapsed time depends on the number of concepts/properties contained in the *seeker* and *provider* contexts. However, the elapsed time values, even in the case in which the dimension of the contexts in terms of number of concepts/properties is quite high (both the couples *Book* and *Volume* have 6 entities in total), never reach 0.3 s.

* Times elapsed are computed on a P4 running at 3 GHz with 1Gb of memory.

Since the *contextual matcher* fulfills the requirement of speed and seems to be a reasonable approach for exploiting contextual information of ontology concepts, it can also be included in *SECCO*.

3.4 Why do we need *SECCO?*

This section explains why *SECCO* has been designed and how it can be practically exploited. The main motivation for designing *SECCO* is to provide an ontology mapping algorithm in open environments (e.g., P2P, Grid). As pointed out in Section 1, there are several mapping algorithms, but there is a lack of algorithms especially designed for open environments. In such scenario, time accuracy is a mandatory requirement to perform "online" mapping and the amount of ontological information exploitable to discover mappings is quite limited. *SECCO* has been designed to provide the semantic foundation for the *K-link+* system [31]. *K-link+* is a P2P system for collaborative work based on the concept of workspace. The system allows workers to work concurrently in the same and shared environment (i.e., workspace) by a set of tools for sharing and exchanging knowledge in a semantic way. In such an open architecture, it would be very useful to discover and interact with semantically neighbor peers. The concept of semantic proximity can be represented by exploiting *SECCO*. In fact, mappings discovered by *SECCO*, establish *semantic links* among peers of a *K-link+* network. These links can be exploited in the following ways:

- *Semantic based search*: contents (e.g., web pages, documents) can be annotated to ontology concepts in order to provide them with an *explicit* and *machine understandable* semantic meaning. Therefore, content search can be performed by specifying ontology concepts instead of keywords. Retrieving similar concepts by *SECCO* will result in discovering contents annotated to such concepts.
- *Semantic building of workspaces*: *semantic links* between peers are supposed to reflect common interests shared by the peers involved in these links. Therefore, by following these links peers with common interests can be discovered and grouped together.
- *Semantic query routing*: *semantic links* can be exploited to forward queries. When a query reaches a peer, it can forward this query to other peers with which it has *semantic links*. This way a new semantic path between "unknown" peers can be constructed. Moreover, the amount of network traffic generated by queries (as compared with flooding techniques) can be significantly reduced by adopting a semantic-aware routing strategy.

We want to point out how, in designing a comprehensive semantic P2P solution, the central problem is to find out *semantic links* among peers. Once found, these can be exploited for several purposes. Therefore, differently from other approaches (e.g., [1, 24]) where the preexistence of mapping ensures semantic interoperability among peers, we provide a comprehensive solution that tackles the problem of designing semantic P2P systems from all the perspectives, that is, construction of the semantic overlay (provided by *SECCO*) and underling physical P2P architecture (provided by *K-link+*).

4 *SECCO: a* double evaluation

In this section, we show how the requirements that driven the design of *SECCO* are fulfilled in real case scenarios. In Sections 3.1-3.3 the matchers of *SECCO* have been individually described and how they cope with the requirements of fastness and accuracy has been shown. The *syntactic matcher* has been evaluated on the OAEI2006 real life ontologies (see Fig. 9). The *lexical matcher* has been extensively evaluated through the similarity experiment whose results are available at http://grid.deis.unical.it/similarity. The rationale of the *contextual matcher* has been described through the example depicted in Fig. 11.

In this section, we want to evaluate *SECCO* as a whole. The evaluation has been split in two parts (referred to as *Experiment 1* and *Experiment 2* in the following). In *Experiment 1* (see Section 4.1), we evaluated *SECCO* by comparing it with H-Match [8,9] that actually is the only system designed for mapping ontologies in open environments offering very similar features. In Section 4.1.1 we perform a sensitivity analysis of the assignment of weights to the individual matchers and observe how results provided by *SECCO* in *Experiment 1* and the correlation w.r.t those produced by H-Match vary. In *Experiment 2* (see Section 4.2), we evaluate how *SECCO* performs as a general mapping algorithm. In this experiment, we evaluate it on four real-life ontologies included in the OAEI 2006 benchmark test suite [37], and compare its results with those of other algorithms not explicitly designed for ontology mapping in P2P networks. We evaluate *SECCO* only on ontologies 301-304 of the OAEI 2006 in order to have an indicator of how it performs in real case scenarios.

4.1 Experiment 1*:* comparing *SECCO* with H-Match

This section presents the comparison of *SECCO* w.r.t H-Match on two excerpts of (online available) ontologies. The first ontology (*Ka*) describes research projects while the second one (*Portal*) describes content of a Web portal. We suppose that *Ka* belongs to a *seeker* peer while *Portal* to a *provider* peer. These ontologies have also been adopted to evaluate the H-Match system as described in [8]. We have chosen to adopt the same two ontologies in order to have an objective comparison between the two approaches. Fig. 12 shows two excerpts of *Ka* and *Portal* describing the concept of *Publication* are shown.

In this evaluation, we aim at constructing, by exploiting *SECCO*, a mapping (see Definition 6) between the concept *Publication* in *Ka* and some concepts belonging to *Portal*. In particular, we want to emphasize how *SECCO* can profitably discover similarities even among terms apparently not related and how it behaves w.r.t H-Match.

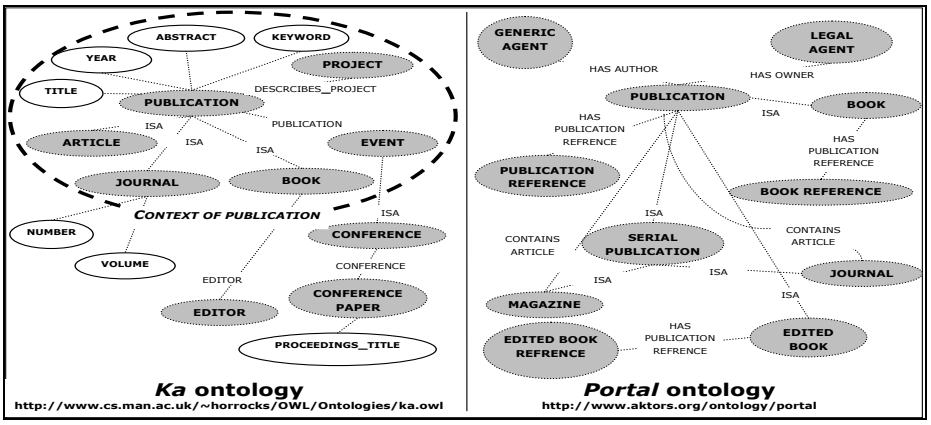

Fig. 12. Excerpts of the *Portal* and *Ka* ontologies defining the concept *Publication*. The context of *Publication* (dashed area) is also shown.

Configuration of SECCO for Experiment 1

In this experiment, the input *I* of *SECCO* (see Section 2.3) takes the values shown in Table 5. We do not set a threshold value (the T_h parameter) since we want to create one-to-many mappings.

Table 5. The input *I* of *SECCO* for *Experiment 1*.

Parameter		Value
C_s	Seeker Concept	Publication
ctx(C_s)	Seeker Context	ctx(Publication)
O	Provider Ontology	Portal (the excerpt shown in Fig.12)
T_h	Threshold	0
w_s	Syntactic similarity weight	0.1
w_L	Lexical similarity weight	0.6
w_c	Contextual similarity weight	0.3

Since we want to give more emphasis to the semantic component of the algorithm, we consider *lexical similarity* more reliable than *syntactic similarity* or *contextual similarity* (i.e., we assign a higher value to w_L). A detailed analysis on how the assignment of weights can affect results will be provided in Section 4.1.1.

Results obtained by SECCO for Experiment 1

Table 6 shows the results obtained by *SECCO* with the input *I* (see Table 5) along with overall elapsed times.

Table 6. Results obtained by *SECCO* by considering *Publication* as *seeker* concept.

		Similarity Values				
***Ka* concept**	***Portal* concept**	**Syntactic**	**Lexical**	**Contextual**	**Overall**	**Elapsed time (s)**[†]
Publication	*Publication*	1	1	0.697	**0.909**	0.49
Publication	*Book*	0	0.823	0.199	**0.553**	0.29
Publication	*Journal*	0	0.767	0.221	**0.526**	0.31
Publication	*Magazine*	0	0.737	0.088	**0.468**	0.29
Publication	*Edited Book*	0	0.823	0.674	**0.696**	0.29
Publication	*Publication Reference*	0.3	0.549	0.118	**0.395**	0.27
Publication	*Book Reference*	0	0.549	0.118	**0.365**	0.28
Publication	*Edited Book Reference*	0	0.549	0.118	**0.365**	0.31

These examples show the suitability of the *lexical matcher* which allows to discover mappings in a *semantic* way. In fact, by considering the analyzed couples of concepts only from a syntactic point of view we would obtain similarity values equal to 0 apart from the couples *Publication (Ka)* and *Publication* (*Portal*) and *Publication (Ka)* and *Publication Reference* (*Portal*). In the following, we compare these results with those obtained by H-Match.

Discussion of results and comparison with H-Match

Comparing ontology mapping algorithms is a hard task, especially when an objective and reliable reference alignment is not provided. Moreover in a P2P scenario, since a mapping algorithm usually aims at finding one-to-many mappings (it provides a similarity ranking between concepts) it is very difficult to interpret ranking values. In literature, there exist very few algorithms that address the ontology mapping problem in P2P environments. The approach closer to *SECCO* is H-Match. In order to make an objective comparison between them, we considered the results obtained by H-Match for the same couples of concepts on which *SECCO* has been evaluated.
In the example depicted in Fig. 11 authors in [9] provide only a similarity value (related to the couple *Book* and *Volume*). For this couple, H-Match obtained the results shown in Table 7.

Table 7. Comparison between *SECCO* and H-Match on the example of Fig. 11.

Couple	***SECCO***	**H-Match**			
		Shallow	*Intermediate*	*Deep*	*Average*
Book-Volume	0.8117	1	0.78	0.70	0.8266

The overall similarity value between the couple *Book* and *Volume* obtained by *SECCO* is computed as follows:

$$sim(Book, Volume) = w_s * sim_{syn} + w_L * sim_{lex} + w_C * sim_{con}$$

[†] Elapsed times are computed on a P4 running at 3 GHz with 1Gb of memory.

Therefore, we obtain:

sim(*Book*, *Volume*)=0.1*0+0.6*1+0.3*0.7057=0.8117

The *lexical matcher* correctly interprets the linguistic similarity between the *Book* and *Volume* concepts; in fact, it gives 1 as output. The high value of lexical similarity is because the *Book* and *Volume* concepts belong to the same WordNet synset and therefore are synonyms. Same things are valid for H-Match, whose *shallow* matching model has similar features to the *lexical matcher* of *SECCO*. The *contextual matcher* of *SECCO*, since *Book* and *Volume* respectively appear in a *Bookstore* context and a *Library* context, correctly decreases the overall similarity value (this aspect has been discussed is Section 3.3).

H-Match obtains a similarity score of 0.78 with the *intermediate* matching model, which takes into account concept names and properties. Through the *deep* matching model, which considers the whole context of concepts (i.e., all the properties) H-match obtains 0.70. This matching model is the most similar to that implemented by *SECCO*. The average value given by H-Match, obtained by averaging results of the three matching models, is 0.8266, which is very close to the result obtained by *SECCO*. Therefore, in this case, we can conclude that the similarity value between *Book* and *Volume* obtained by *SECCO* is comparable with that obtained by H-Match.

A more detailed comparison between the two approaches can be done by considering the two excerpts of the *Ka* and *Portal* ontologies depicted in Fig. 12. Similarity values obtained by both *SECCO* and H-Match [8] are shown in Table 8. For the sake of comparing only semantic features of the two approaches we do not considered the contribution of the *syntactic* similarity of *SECCO* for the couples *Publication* (*Ka*) and *Publication* (*Portal*) and *Publication* (*Ka*) *Publication Reference* (*Portal*).

Table 8. Comparison between *SECCO* and H-Match on the example of Fig. 12.

***Ka* concept**	***Portal* concept**	***SECCO***	**H-Match**				
			Surface	*Shallow*	*Deep*	*Intensive*	*Average*
Publication	*Publication*	**0.909**	1	0.7384	0.8047	0.7814	0.8318
Publication	*Book*	**0.553**	0.8	0.6184	0.66	0.6394	0.6795
Publication	*Journal*	**0.526**	0.64	0.5224	0.5538	0.5381	0.5636
Publication	*Magazine*	**0.468**	0.8	0.6184	0.6498	0.6341	0.6756
Publication	*Edited Book*	**0.696**	0.64	0.5224	0.5641	0.5434	0.5675
Publication	*Publication Reference*	**0.395**	0.64	0.5531	0.5741	0.5503	0.5794
Publication	*Book Reference*	**0.365**	0.64	0.5531	0.5733	0.5497	0.5790
Publication	*Edited Book Reference*	**0.365**	0.64	0.5531	0.5637	0.5420	0.5747

In this experiment, it is interesting noting that the higher similarity values obtained by *SECCO* and H-Match are related to the couple *Publication* (*Ka*) and *Publication* (*Portal*). These two values are very close. Same considerations are valid for the couple *Publication (Ka)* and *Book (Portal)*. An interesting consideration can be done for the last three rows of Table 8. While *SECCO* obtains low similarity values, H-Match obtains values that always exceed 0.5. For instance, for the couple *Publication*

and *Publication Reference*, *SECCO* obtains 0.395 while H-Match obtains 0.5794 as average result. However, by objectively analyzing the concepts, one can assess that these concepts are not much similar. In fact, the first describes the concept of *Publication* while the second defines a reference to a *Publication*.
It is very difficult to comparing results between the two strategies with very few matching of couples, as in the case of *Book* and *Volume* (see Table 7). Moreover, comparing mapping results, without a reference alignment, implicitly includes a certain degree of subjective interpretation.

In order to obtain an overall indicator of how the two approaches are (un)related, we computed the Pearson correlation coefficient [13] between their results. This coefficient represents an agreement between the values of two data sets (in our case between similarity results) by expressing the degree of association between them (see Table 9).

Table 9. Correlation between results of *SECCO* and H-Match shown in Table 8.

	H-Match					
	Surface	*Shallow*	*Deep*	*Intensive*		*Average*
SECCO	0.898	0.8559	0.9051	0.908		0.8919

As can be noticed the higher value of correlation is 0.908 meaning that results obtained by *SECCO* are closer to these obtained by H-Match through the *intensive* matching model. Through this model of matching, H-Match considers both linguistic feature of ontology concepts and whole context of concepts (in terms of properties and semantic relations) in which they appear. In addition, also the correlation w.r.t the *deep* model is high. The average correlation value is 0.8919, which underlines how he two approaches are very close. In fact, a value of correlation higher that 0.7 can be interpreted as an indicator of high similarity [44]. It is very important notice that *SECCO* performs very close to those of H-Match even if *SECCO* does not adopt complex matching strategies.

Since both approaches heavily rely on linguistic features of ontologies, we also computed the correlation (see Table 10) between results of our *lexical matcher* that relies on WordNet and the *surface matching* model of H-Match that relies on an ad-hoc thesaurus built by exploiting WordNet.

Table 10. Correlation between the *lexical matcher* of *SECCO* and the *surface* matching model of H-Match.

	H-Match
	Surface
SECCO	0.7123
Lexical matcher	

As can be noticed, the value of correlation is high even if it is very difficult to estimate which approach is more accurate. However, the *lexical matcher* of *SECCO* is not an ad-hoc thesaurus but it is able to exploit the whole structure of WordNet by including in the similarity computation a wide set of semantic relations between

concepts. Moreover, the metrics included in the *lexical matcher* have been extensively evaluated by the similarity experiment [26].

4.1.1 Discussion on similarity aggregation and assignment of weights

SECCO, in order to perform similarity aggregation, adopts a weighted sum of similarity values given by the individual matchers. Do and Rahm [14] address some aspects of weights assignment and similarity aggregation for database structures. A similarity aggregation function is a function that takes results from several matchers, weights these results, and gives as output an overall similarity indicator. The weights are assigned manually or learned, e.g., using machine learning on a training set. Berkovsky et al. [5] have thoroughly investigated the effects of different weights on the alignment results.

We chose to adopt a strategy based on multiple matchers since experimental results have shown that a combination of similarity measures (provided by different matchers) leads to better alignment results than using only one matcher at a time. We realize that this technique needs a certain degree of expertise from the *SECCO* user. In facts, if the different weights are not correctly assigned, mapping results can be affected. However, notice that in a P2P scenario it is not possible to a priori analyze the structure of ontologies to be compared, in order to find the best mapping strategy (as done in [25]), since peers are not aware of the ontologies of other peers.

In the experiments, we manually settled values of the different weights (i.e., the w_S, w_L, w_c parameters). However, it would be interesting to see how the correlation coefficient w.r.t H-Match (*Experiment 1*) changes when assigning different weights. Table 11 shows correlation values between the two approaches by assigning different values of w_s, w_L and w_c. For sake of space, we do not report, for each variation of the weights, the similarity values obtained by *SECCO*.

Table 11. Correlation between results of *SECCO* and H-Match by varying the weights of the individual matchers on the couples of concepts listed in Table 8.

SECCO			***H-Match***				
Syntactic similarity w_S	*Lexical similarity* w_L	*Contextual similarity* w_c	*Correlation w.r.t the different matching models*				*Average correlation*
			surface	*shallow*	*deep*	*intensive*	
0.1	***0.6***	***0.3***	0.898	0.8559	0.9051	0.908	0.8917
0.3333	***0.3333***	***0.3333***	0.9023	0.8657	0.9102	0.9117	0.8974
0.3	***0.4***	***0.3***	0.8415	0.8367	0.8567	0.8645	0.8498
0.3	***0.3***	***0.4***	0.8218	0.8123	0.8657	0.8756	0.8438
0.1	***0.3***	***0.6***	0.7656	0.7567	0.8123	0.8198	0.7886
0.1	***0.1***	***0.8***	0.4978	0.4478	0.5218	0.5123	0.4949

An interesting consideration arises from results shown in Table 11. As it can be noticed, if we assign equal weights to the matchers (row 2) the correlation raise up to 0.9117 with the *intensive* model of H-Match and to 0.8974 in the average. Moreover, it is interesting to point out that if we assign a little higher value to the *contextual matcher* (row 4), the correlation remains high (0.8756 for the intensive model and

0.8438 in the average). Even in the case in which contextual similarity has a higher value (row 5), the average correlation value remains quite high. Conversely, if we give much more emphasis to the *contextual matcher* (row 6), the average correlation drastically decrease to 0.4949 in the average. As final remark, we can conclude that assigning equal weight to the matchers can increase the correlation value w.r.t H-Match, that does not necessarily mean better results since in the considered example alignments are not provided. However, in the light of these considerations in *Experiment 2* we assign equal weights to the different matchers.

4.2 Experiment 2*:* comparing *SECCO* with other ontology mapping algorithms not designed for ontology mapping in P2P networks

This section provides an extensive evaluation of *SECCO* on four real-life ontologies contained in the OAEI 2006 [37] test suite. We compared *SECCO* with other mapping algorithms not explicitly designed to tackle the OMP in P2P networks. This way we want to show how much the designing strategy of *SECCO*, which has to ensure fastness and cannot exploit the whole structures of ontologies to be mapped, affects accuracy (i.e., quality of results).
In particular, we focused on the group of tests that contain four real-life ontologies (i.e. tests from 301 to 304) in order to investigate how *SECCO* performs in mapping real ontologies. For each of these ontologies the OAEI organizers provided a reference alignment. We computed measures of Precision (i.e., the number of correct mapping among all the mapping found), Recall (i.e., the number of correct mapping among all the existing mapping) and F-measure [15] (i.e., the harmonic mean of Precision and Recall). In particular, we compared results obtained by *SECCO* with those provided by the OAEI organizers.
Notice that *SECCO*, even being designed for P2P networks and therefore to work "online", can be exploited to compare entire ontologies by reiterating the process described in Section 2 for each concept in the source ontology (i.e., the reference ontology 101 contained in the OAEI tests).

Configuration of SECCO for Experiment 2

Table 12 shows the values of the input of *SECCO* for this experiment. Here we are interested in obtaining one-to-one mappings.

Table 12. *SECCO* configuration for *Experiment 2*.

Parameter		Value
C_s	Seeker Concept	Each concept C_i contained in the reference ontology (i.e., 101)
$ctx(C_s)$	Seeker Context	$ctx(C_i)$
O	Provider Ontologies	301-302-303-304
T_h	Threshold	0.51
w_S	Syntactic similarity weight	0.333
w_L	Lexical similarity weight	0.333
w_c	Contextual similarity weight	0.333

Results obtained by SECCO for Experiment 2

Fig. 13 shows values of Precision, Recall and F-Measure obtained by *SECCO*. As can be noticed, *SECCO* performs well. It always obtains a Precision around 0.9. The Recall, reaches the highest value (i.e., 0.9387) for ontology 304 while the lowest value (i.e., 0.6211) for ontology 302. However, it always remains higher than 0.5. The F-Measure values are 0.8269 for ontology 301, 0.7375 for ontology 302, 0.8012 for ontology 303 and 0.949 for ontology 304. Values of F-Measure that represent an overall indicator of the performance of a mapping algorithm are in all the cases high.

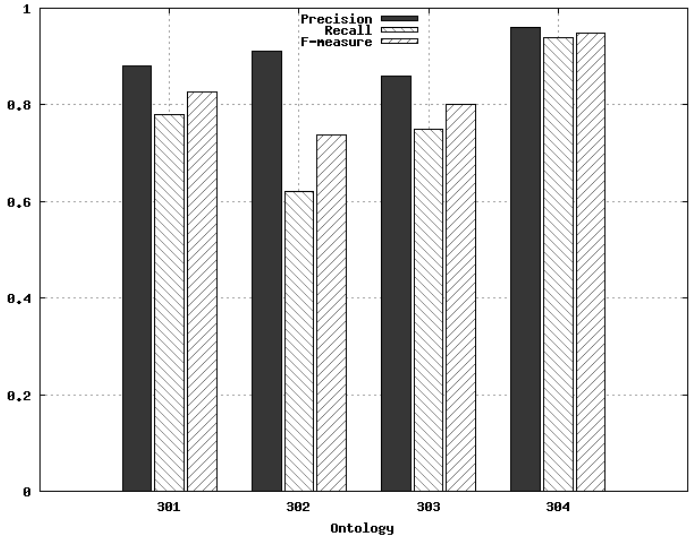

Fig. 13. Results of *SECCO* on the OAEI 2006 real life ontologies.

Discussion of results and comparison with other ontology mapping algorithms

In order to have an objective evaluation of *SECCO*, we decided to compare its average results with those of other ontology mapping approaches. The results are shown in Table 13. *SECCO* obtained an average Precision of 0.81, an average Recall of 0.81 and an average F-measure of 0.81. As can be noticed, *SECCO* is one of the most precise algorithms. It is only slightly outmatched by Automs and Falcon.
In terms of Recall *SECCO* is outperformed only by RiMOM. In terms of F-Measure, *SECCO* is only dominated by Falcon and RiMOM.
An important consideration emerges from these results. *SECCO* is an ontology mapping algorithm that in its current implementation cannot exploit the whole structural information encoded in ontologies. Conversely, most of the presented approaches have a solid structural matching strategy. For instance, Falcon relies on the GMO approach [51] that exploits a graph-matching algorithm for discovering mappings while RiMOM exploits an adaptation of the Similarity Flooding algorithm. Such strategies require a complex analysis of the ontologies that is not conceivable in a P2P environment for two reasons: (i) peers are not aware of the whole ontologies of other peers; (ii) the fundamental requirement of fastness in P2P networks can be

affected. It is worthwhile noting that *SECCO* obtains very good results without using that strategy.

Table 13. Average results obtained by some ontology mapping algorithms on the OAEI 2006 real-life ontologies, as reported in [19].

	SECCO‡	*Jhu/apl [6]*	*Automs [29]*	*Falcon [25]*	*RiMOM [55]*	*H-Match [10]*
Precision	**0.81**	**0.18**	**0.91**	**0.89**	**0.83**	**0.78**
Recall	**0.81**	**0.50**	**0.70**	**0.78**	**0.82**	**0.57**
F-Measure	**0.81**	**0.26**	**0.79**	**0.83**	**0.82**	**0.65**
Average Elapsed Time (s)	**3.05 s**	**Na**	**70.25 s**	**7.22 s**	**3.14 s**	**Na**

Notice that the elapsed time by *SECCO* is the lowest. In particular, it is 25 times lower that that obtained by *Automs* that also exploits WordNet and about 3 times lower than that of Falcon, which adopts a structural matching strategy. Moreover, the comparison with H-Match, the system actually very similar to *SECCO*, shows how *SECCO* is better in terms of Precision, Recall and F-Measure. It would be also interesting to compare the approaches in terms of elapsed time but unfortunately, authors in [10] do not provide information about execution times.
On the one side, these results show how a structural mapping strategy can improve mapping results as in the case of Falcon. On the other side, they show that *SECCO* obtains results comparable with those of the most performant ontology mapping algorithms without adopting complex structural analysis of ontologies. Finally, we can conclude *SECCO* is faster than other mapping algorithms and the cost paid, in terms of accuracy, is not so high.

5 Related work

Recently several ontology mapping algorithms have been proposed. A detailed survey is given in [11]. In that survey, only a bibliographic reference to ontology mapping in P2P systems is listed. This underlines the fact that the OMP has not adequately been tackled in open environments. In literature, there are few approaches similar to *SECCO* explicitly designed for mapping ontologies in open environments.

The CtxMatch algorithm [7] aims at discovering mappings between Hierarchical Categories (HCs). It relies on WordNet for interpreting the correct sense of concepts in the context in which they appear. Therefore, it performs a transformation of the concepts to be compared in Description Logics axioms that are exploited to reduce the problem of discovery mappings to a SAT problem. CtxMatch similarly to *SECCO* implements a semantic based approach since it relies on WordNet. However, the main difference between these systems is that CtxMatch focuses on matching HCs and provides as output a semantic relation between terms while *SECCO* can also deal with ontologies and provides a confidence value.

‡ Elapsed times are computed on a P4 running at 3 GHz with 1Gb of memory.

H-Match [8, 9] is an algorithm for dynamically matching concepts in distributed ontologies. H-Match allows for different kinds of matching depending on the level of accuracy needed. The system aims at supporting knowledge sharing and ontology-addressable content retrieval in peer-based systems. It is actually the system closer to *SECCO*. Indeed, there are at least two main differences between these approaches:

- The lexical matcher of H-Match is based on a ad-hoc thesaurus, while that of *SECCO* is based on WordNet. H-Match defines an ad hoc similarity metric between concepts in the thesaurus. Conversely, *SECCO* can benefit from a similarity metric evaluated by the similarity experiment [26]. The metric adopted in *SECCO* is highly correlated w.r.t similarity judgments given by human[§].
- H-Match in performing contextual affinity exploits predefined weights assigned to the different types of relations among concepts. It introduces five types of relations (i.e., same-as, part-of, kind-of, contains, associates) among terms of a peer ontology. These relations are assessed by exploiting relations that concepts have in WordNet. Conversely, *SECCO* adopts the "how it fits" strategy from which the contextual similarity indicator can emerge by combining measures of semantic similarity, to take into account hypernymy/hyponymy relations among synsets, and relatedness to take into account a broader range of semantic relations (e.g., part of).

We deeply compared *SECCO* with H-Match concluding that results obtained by the two approaches are, for several aspects, comparable. However, *SECCO* performs a little better on real-life ontologies included in the OAEI 2006 tests. Since the two approaches are both designed to work in open environments, it would be interesting also to compare them in terms of performance (i.e., execution time for computing mappings).

Falcon-AO [25] is an automatic tool for aligning ontologies based on three alignment strategies: the I-Sub [48] metric is exploited to compare strings, the V-Doc [42] is a linguistic matcher based on Information Retrieval, while the GMO [51] is a matcher based on graph matching.

The RiMOM system [55] combines different strategies to assess ontology mappings. In particular, it includes an edit distance metric and an adaption of the similarity flooding algorithm to the context of ontology mapping.

iMapper [50] is an ontology mapping tool based on the idea of semantic enrichment. It makes use of ontology instances to calculate the similarity between concepts. The mapping process is split in two phases. In the first one (i.e., enrichment phase) documents (i.e., instances) associated to ontology concepts are analyzed thus building the enriched ontology. The association of documents to concepts can be done automatically, but user refinement it is also allowed. The output of this phase are representative vectors (one for each concept) built from the textual content of their associated documents. In the second phase (i.e., mapping phase) similarities between ontology elements are computed as the cosine between their representative vectors. Further refinements are employed to re-rank the results via the use of WordNet.

The abovementioned systems discover ontology mappings by exploiting both structural and linguistic information encoded in ontology entities. However, in order

[§] For preliminary experimental results refer to: http://grid.deis.unical.it/similarity

to work properly they need to scrutinize the two ontologies to be mapped. For instance, the structural matcher of Falcon is based on a graph matching algorithm (i.e., the GMO matcher) which requires to construct the adjacency matrix of the two ontologies. In addition, the lexical matcher of Falcon (i.e., the V-Doc matcher [42]) requires analyzing both the ontologies to be mapped. Similar things hold for RiMOM, which adopts as a structural matcher a variant of the Similarity Flooding algorithm. iMapper needs to access the whole two ontologies and requires ontology concepts to be associated with instances. As can be notice, a common denominator among these approaches is that they "need to know" the whole two ontologies. Conversely, *SECCO* does not impose this requirement since as usually happens in P2P networks peers are not aware of one another's ontologies. Therefore, it would be interesting to see how the abovementioned approaches perform without completely knowing the two ontologies to be mapped.

In the literature, there are some semantic P2P applications sharing common characteristics with *SECCO*.

SWAP (Semantic Web and Peer to Peer) [18] aims at combining ontologies and P2P for knowledge management purposes. SWAP allows local knowledge management through a component called LR (Local node repository), which gathers knowledge from several sources and represents it in RDF-Schema. In SWAP, each node is responsible for a single ontology: ontologies might represent different views of a same domain, multiple domains with overlapping concepts, or might be obtained by partitioning an upper level ontology. Knowledge sharing is obtained through ontology mapping and alignment.

GridVine [1] is a semantic P2P system whose aim is to build a semantic overlay network based on two layers: *logical layer* and *physical layer*. The logical layer provides a set of functionalities such as: attribute-based search, schema management and schema mapping. The physical layer is used as support to the logical layer in constructing the overlay and forwarding queries. In GridVine, semantic interoperability is achieved by *semantic gossiping* [2]. Semantic gossiping assumes the *preexistence* of local agreements provided as mappings between different schemas. Peers introduce their own schemas and exchanging translations between them can incrementally come up with an implicit "consensus schema".

Piazza [24] is a P2P Data Management system whose main aim is to enable efficient query processing. Piazza takes into account the structure of the knowledge domain and documents in order to achieve interoperability between different information sources. Similarly, to GridVine it assumes the preexistence of mappings between data sources. Therefore, these mappings are *chained* together and exploited for query rewriting/answering.

In the SWAP system, mappings between peer ontologies are dynamically obtained by exploiting techniques based on lexical features, structure and instances of ontologies. Conversely, neither the GridVine nor the Piazza approach tackle the problem of discovering mappings among the different representations (i.e., schema, ontologies) belonging to different peers since they assume the preexistence of mappings.

6 Concluding remarks

This paper described *SECCO*, an ontology mapping algorithm aimed at discovering *concept mappings* in P2P networks. A *concept mapping* has been defined as a similarity ranking between a *request* (composed by a concept along with its context) performed by a *seeker* peer and concepts belonging to *provider* peer ontologies. Since we assume that peers are not aware of one another's ontologies, in order to discover mappings, we designed an ad-hoc mapping strategy. This strategy aims at fulfilling two important requirements (i.e., fastness and accuracy) through three individual matchers. The main problem we faced is related to the fact that we cannot adopt sophisticated and time-consuming structural matching strategies that require to know the whole two (peer) ontologies to be compared. Hence, we adopted the notion of *context*, defined as a concept along with its properties (obtained as described in Section 2.2) and nearest neighbor concepts. Through contexts, we aim at encoding the amount of structural information needed in a particular request. We compare the contextual information of different concepts by the "how it fits" strategy that is founded on the idea that two concepts are related if they fit well in each other's context. This strategy is supported by the *lexical matcher* whose aim is to exploit an accurate (proven by the similarity experiment [26]) similarity metric in WordNet. This metric allows assessing similarity even among syntactically unrelated concepts. Moreover, in order to exploit all the linguistic information of ontology entities (i.e. ontology metadata) we adopt the *syntactic matcher*. This matcher encodes linguistic information in *virtual documents* that are created and compared by an information retrieval approach. All these matching strategies have been extensively evaluated.
Along the paper, we discussed the exploiting of *SECCO* in the context of P2P networks and proven through experimental evaluation the suitability of the algorithm. In particular, *SECCO* has been compared (*Experiment 1*) with the H-Match algorithm, designed for ontology mapping in open environments, with very promising results. Furthermore, *SECCO* has been compared (*Experiment 2*) with other mapping algorithms not explicitly designed for mapping in P2P networks and even in this case results are satisfactory. We also performed a sensitivity analysis from which emerged an interesting aspect related to weight assignments to the different matchers.

7 Future work

Here we briefly describe possible improvements of the algorithm. First in a future version of *SECCO* we aim at distinguishing relations between concepts from relations that describe properties of concepts. This way, the definition of *context* exploited by *SECCO* will give much emphasis to the relations that have *per se* a semantic meaning as for instance the ISA relations. We are also performing further improvements of *SECCO* along two directions.

On the one side, we are investigating a strategy for automatically tuning the weights of the different matchers and aggregating results. In particular, we are evaluating the following possibilities:

- The use of sophisticated techniques such as the Dempster Shafer theory for combining results of the different matchers. The Dempster-Shafer theory [47] is a mathematical based on *belief functions* and *plausible reasoning*, which is used to combine separate pieces of information (evidence) to calculate the probability of an event. In our case, we aim at exploiting this strategy for combining uncertain results given by the different matchers for obtaining a more reliable overall similarity value.
- The use of a linear aggregation formula. According to this strategy, a weight of 1 is given to results provided by each matcher. The overall similarity is obtained as the average similarity value given by the different matchers.

On the other side, we aim at exploiting the World Wide Web for refining similarity values among concepts. In fact, we argue that the Web could be a valuable source of knowledge. Our aim is to design a similar strategy based on the analysis of relations between terms extracted from the snippets (related to concepts to be compared) given by a search engine.

Finally, since we included *SECCO* as semantic support in our *K-link+* [31] system, we also would like to evaluate its performance within *K-link+*. In this case, we are interested in evaluating *SECCO* in a complete semantic P2P solution for cooperation and contents sharing and retrieval.

References

1. Aberer, K., Cudré-Mauroux, P., Hauswirth, M., Van Pelt, T.: GridVine: Building Internet-Scale Semantic Overlay Networks. In Proc. of ISWC 2004, Hiroshima, Japan (2004) 107-121
2. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: The Chatty Web: Emergent Semantics Through Gossiping. In Proc. of WWW 2003, Budapest, Hungary (2003) 197-206
3. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, 2001
4. Berners-Lee, T.: The Semantic Web: An interview with Tim Berners-Lee. Consortium Standards Bulletin, 4(6), June 2005
5. Berkovsky, S., Eytani, Y., Gal, A.: Measuring the relative performance of schema matchers. In Proc. of WI 2005, Compeigne, France, (2005) 366–371
6. Bethea, W. L., Fink, C.R., Beecher-Deighan, J.S.: JHU/APL Onto-Mapology Results for OM 2006. In Proc. of OAEI 2006, Athens, Georgia, USA (2006)
7. Bouquet, P., Serafini, L., Zanobini S.: Semantic coordination: a new approach and an application. In Proc. of ISWC 2003. Florida, USA, (2003) 20-23
8. Castano, S., Ferrara A., Montanelli S, Racca, G.: From Surface to Intensive Matching of Semantic web Ontologies. In proc. of WEBS 2004, Zaragoza, Spain (2004) 140-144
9. Castano, S., Ferrara A., Montanelli S.: H-MATCH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems. In Proc. of SWDB, Berlin, Germany (2003) 231-250
10. Castano, S., Ferrara, A., Messa, G.: Results of the HMatch Ontology Matchmaker in OAEI 2006. In Proc. of OM 2006, Athens, Georgia, USA (2006) 134-143
11. Choi, N., Song, I., and Han, H. A survey on Ontology Mapping. SIGMOD Record 35(3) 2006 34-41
12. Davies J., Studer R., and P. Warren (eds.): Semantic Web Technologies - trends and research in ontology-based systems, Wiley 2006
13. Devore, J.L.: Probability and Statistics for Engineering and the Sciences. International Thomson Publishing Company

14. Do H., Rahm E.: COMA – a system for flexible combination of schema matching approaches. In Proc. of VLDB-2002, Hong Kong, China (2002) 610-621.
15. Do, H., Melnik, S., Rahm E.: Comparison of schema matching evaluations. In Proc. GI-Workshop Web and Databases, Erfurt, Germany, (2002) 221-237
16. Ehrig, M., Staab, S.: Qom - fast ontology mapping. In Proc. of ISWC 2004, Hiroshima, Japan (2004) 289-303
17. Ehrig, M., Sure, Y.: Ontology Mapping – an integrated approach. In Proc. of ESWS 2004, Heraklion, Greece (2004) 76-91
18. Ehrig, M., Tempich, C., Broekstra, J., Van Harmelen, F., Sabou, M., Siebes, R., Staab, S., Stuckenschmidt, H.: SWAP: Ontology-based Knowledge Management with Peer-to-Peer Technology. In Proc. of WOW, Luzerne, Switzerland (2003)
19. Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Šváb, O., Svátek, V., van Hage, W. R., Yatskevich, M.: Results of the Ontology Alignment Evaluation Initiative 2006. In Proc. of OM 2006, Athens, Georgia, USA (2006)
20. Euzenat, J.: An API for ontology alignment. In Proc. of ISWC 2004, Hiroshima, Japan (2004) 698-712
21. Euzenat J., Shvaiko P.: "Ontology Matching", Springer 2007
22. Goble, C.A,. De Roure, D.: The Semantic Grid: Myth busting and bridge building. In Proc. of ECAI 2004, Valencia, Spain, (2004) 1129-1135
23. Gruber, T. R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2) (1993)199-220
24. Halevy, A.Y., Ives, Z.G., Jayant Madhavan Mork, P., Suciu, D., Tatarinov. I.: Piazza: Data Management Infrastructure for Semantic Web Applications. In Proc. of WWW2003, Budapest, Hungary (2003) 556-567
25. Hu, W., Cheng, G., Zheng, D., Zhong, X., Qu, Y.: T Results of Falcon- OM in the OAEI 2006 Campaign, In Proc. of .OM-2006, Athens, Georgia, USA, (2006) 124-133
26. Java WordNet Similarity Library (JWSL) and the Similarity Experiment. http://grid.deis.unical.it/similarity
27. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In Proc. of ROCLING X, Taiwan (1997)
28. Klyne, G., Caroll, J.J.: Resource Description Framework (RDF): Concepts and abstract Syntax. W3C Recommendation 10 February 2004. Latest version available at http:///www.w3.org/TR/rdf-concepts/ visited October 2007
29. Kotis, K., Valarakos, A., Vouros, G.: AUTOMS: Automated Ontology Mapping through Synthesis of methods. In Proc. of OM 2006, Athens, Georgia, USA (2006)
30. Lauer, M.: Designing Statistical Language Learners: Experiments on Noun Compounds. In: Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL 1995), Cambridge, Massachusetts, USA (1995) 47-54
31. Le Coche, E., Mastroianni, C., Pirrò, G., Ruffolo, M., Talia, D.: A P2P Virtual Office for Organizational Knowledge Management. In Proc. of PAKM, Wien, Austria (2006) 166-177
32. Levenshtein, I.V. Binary Codes Capable of Correcting Deletions, Insertion and Reversals. Soviet Physics-Doklady 10(8) (1966), 707-710
33. Lucene- The Apache Lucene project. http://lucene.apache.org visited October 2007
34. Miller, G.A., Charles, W.G.: Contextual Correlates of Semantic Similarity. Language and Cognitive Processes, 6 (1), 1991 1-28.
35. Miller, G.: WordNet An On-line Lexical Database. International Journal of Lexicography, 3 (4) (1990) 235-312
36. Mitra, P., Noy, N. F., Jaiswal, A. R.: OMEN: A Probabilistic Ontology Mapping Tool. In proc. of MCN-04, Hiroshima, Japan (2004) 71-83
37. Ontology Alignment Evaluation Initiative. http://oaei. ontologymatching.org visited October 2007

38. Pan, R., Ding Z., Yu, Y., Peng., Y.: A Bayesian Network Approach to Ontology Mapping. In Proc. of ISWC 2005, Galway, Ireland (2005) 563-577
39. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL Web Ontology Language Semantic and Abstract Syntax. W3C Recommendation 10 February 2004. Latest version available at http://www.w3.org/TR/owl-semantics/ visited October 2007
40. Patwardhan S, Pedersen T. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In proc. of EACL 2006 workshop, (2006)1-8
41. Pirrò, G., Talia, D.: An approach to Ontology Mapping based on the Lucene search engine library. In proc. of SWAE '07, Regensburg, Germany (2007) 407-412
42. Qu Y., Hu W., and Cheng G. Constructing Virtual Documents for Ontology Matching. In Proc. of WWW2006, Edinburgh, Scotland (2006) 23-31
43. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In Proc. of IJCAI 1995, Montréal, Québec, Canada (1995) 448–453
44. Rodgers J.L., Nicewander W.A.: Thirteen ways to look at the correlation coefficient. The Amer. Statistician 42 (1988) 59–65
45. Salton, G., Wong, A., Yang C. S.: A Vector Space Model for Automatic Indexing, Communications of the ACM 18 (1) (1975) 613-620
46. Seco, N., Veale, T., Hayes, J.: An intrinsic information content metric for semantic similarity in WordNet. In Proc. of ECAI, Valencia, Spain (2004) 1089-1090
47. Shafer G. : A mathematical theory of evidence, (1976) Princeton University Press
48. Staab, S., Stuckenschmidt, H.: Semantic Web and Peer-to-Peer , Decentralized Management and Exchange of Knowledge and Information, Springer 2006
49. Stoilos, G., Stamou, G., and Kollias, S. A String Metric for Ontology Alignment. . In Proc. of ISWC 2005,Galway, Ireland, (2005) 623-637
50. Su, X., Gulla J. A.: An information retrieval approach to ontology mapping, Data & Knowledge Engineering, (58) 1 (2006), 47-69.
51. Wei, H., Ningsheng, J., Yuzhong, Q., Yanbing, W.: GMO: A Graph Matching for Ontologies. In Proc. of K-Cap 2005, Banff, Canada (2005) 43-50
52. Winkler, W. E. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication (4) (1999)
53. WordNet: a lexical database for the English language. http://wordnet.princeton.edu/obtain visited October 2007
54. WordNet-Similarity bibliography. http://www.d.umn.edu/~tpederse/wnsim-bib/ visited October 2007
55. Yi, L., Juanzi, L., Duo, Z., Jie, T.: Result of Ontology Alignment with RiMOM at OAEI'06. In Proc. of OM-2006, Athens, Georgia, USA (2006) 181-191