

# Integrating Ontologies in Mobile Agents

F. Corradini, R. Culmone, M.R. Di Berardini and E. Merelli

Dipartimento di Matematica e Informatica

Università di Camerino

via Madonna delle Carceri

62032 Camerino, Italy

Email: {flavio.corradini, rosario.culmone, mariarita.diberardini, emanuela.merelli}@unicam.it

**Abstract**—The process of information extraction and data integration in a global information system demands automatic techniques for quickly determining semantic similarity among concepts across different ontologies. This paper presents a graph based approach for computing, on-the-fly, semantic similarities among ontologies of a specific domain. The approach consists of integrating mobile agents and ontologies to support a variety of applications in distributed environments. The resulting technique is illustrated on Hermes, agent-based middleware for mobile computing, by an example in molecular biology domain.

## I. INTRODUCTION

In recent years, ontologies [13], [14], [8], have played an important role in many research areas such as information retrieval and data integration; ontologies are useful for semantic interoperability among heterogeneous information systems [20]. In the information and computer science, an ontology is a type of knowledge-base that describes the concepts, through definitions, that are sufficiently detailed to capture the semantics of a specific domain [13], [15]. An ontology captures a certain view of the world, it provides a vocabulary of terms and relations to model the domain [8]; it supports intentional queries regarding the content of one or more data repositories, and it reflects the relevance of data by providing a description of semantic information independent of the data representation.

In a global environment, the interoperability of information systems, is based on the possibility to offer a query environment in which users may enter a request without knowing how and where the requested information are stored. Thus, due to the heterogeneity of distributed information sources, the use of ontologies become essential to support the semantic interoperability; as well as, the availability of automatic techniques for quickly determining semantic similarity among concepts to describe queries and information sources to be queried.

In addition, the new Web applications, as described by T.B. Lee et al. in [2], aim at guaranteeing the almost completely automatic execution of complex distributed processes, where autonomy, adaptability and cooperation are essential requirements. Agents technology [19] is an appealing approach to build automatic applications. Agents being an autonomous entity, able to react and adapt in a proactive way, in a dynamic execution context, can encapsulate the execution of several independent activities. The agent ability to cooperate with other agents allows to have a useful interaction within an

heterogeneous environment [31]. The integration of agents and ontologies as discussed in Hendler [17], provides a powerful approach to automate distributed computation, to support semantic interoperability and to allow meaningful agents interaction. Furthermore, an agent can move from one environment to another. In some specific domains as computational biology and bioinformatics, the quantity of data to be processed is often prohibitively large to be retrieved in an acceptable time, thus the possibility to move the computation is a promising approach. In an environment with multiple information systems, such those visited by mobile agents, different domain ontologies can coexist [16]. Although the use of single shared ontology would ensure the complete integration across information systems, it is quite impractical because it forces information systems to commit to this single ontology by making difficult the input of new concepts. Thus, a mobile agent has to face two problems: the ability to measure, on-the-fly, the *similarity* among concepts of different ontologies, (its own, e.g. its knowledge base and those used in the visited sites, e.g. conceptual schema of local data repository), and the ability to enrich his own knowledge with new concepts.

In this work, we propose integrating mobile agents with suitable tools for managing ontologies during their migration across distributed heterogeneous information systems. To that purpose, we have defined an abstract data model, the *ontological graph*, derived from the graph-based conceptual model proposed in Mitra et al. [24], and we have defined a minimum set of operators essential to manage *ontological graphs* to determine the similarity. We define three algebraic operators, to isolate a concept in an ontology (*projection*), to measure similarity between two concepts (*similarity*) and to enrich an ontology with a new concept (*enriching*).

The proposed approach shows some advantages: any domain ontology, being represented by RDF or OWL or DAML+OIL, can be mapped into the *ontological graph*; every information system can use a local conceptualization of the domain without to commit to a single global one; every agent can enrich its own knowledge by generating a collection of synonyms; and it can choose the most suitable similarity function [22], [27], [10], [11], to relate domain-specific ontologies. Last but not least, the algebraic operators can be considered the ground for designing a declarative language to specify the agent behaviour.

The three operators has been implemented in Hermes, mid-

middleware for mobile computing, to support the description of the mobile agents behaviour in a distributed environment. In particular, the similarity operator has been implemented over three algorithms: the semantic similarity algorithm proposed by Maedche et al in [22], that proposed by Rodriguez and Egenhofer in [27] and over a new algorithm based on structural similarity, proposed in this work as an extension of our previous work [7].

The paper is structured as follows: Section II describes a motivating scenario with an example in Bioinformatics. Section III defines the *ontological graph* and the algebraic operators. Section IV proposes a new similarity function. Section V, discusses the behaviour of a mobile agent by an example in Bioinformatics. Finally, Section VI remarks conclusions and future work.

## II. MOTIVATING SCENARIO

Nowadays, the widespread interconnection of distributed systems, with the global distribution of information sources and computational tools, offers a scenario where to build distributed applications in every domain of social and life science (Medicine, BioMedicine, Computational and Systems Biology, Health Care ...). In a wide view, we can think to design a scenario where a user (human or application) describes his goal (e.g complex queries, workflows of activities) by using a vocabulary of terms and relations close as much as possible to his application domain. He will not worry about where information are stored, what data formats have been used, how tools can be integrated and services coordinated. The achievement of the user goal is delegated to one or more software entities or agents that are responsible for a correct achievement of the user goal.

In particular, in Bioinformatics, a biologist would be able to specify his experiment like a workflow of activities, ranging from researching and integrating information, to coordinating computational tools executed over specific data. In this domain, where the amount of interrelated information exponentially increases, it is very difficult for a human to exploit all available data, to identify, select, clean and use all relevant data, also because of different data formats with different semantics.

If on the one hand the integration of heterogeneous data can be achieved in different ways, on the other to manage all suitable data in an acceptable time asks for remote computation. In fact, data can be extracted and integrated either in a unique datawarehouse to which users can submit a query using a global schema, or instantaneously in native data sources. In the first approach data are centralized, there is no instant schema translation, but it is difficult to add new data repositories, to maintain data updated and to modify any schema. In the second approach, data remain in native repositories where they are constantly updated and free to be represented in any format, but an instant (on-the-fly) schema mapping must be done by the data collector; in fact different data sources may use different names and formats to refer the same object, or the same name to refer objects with different meaning.

The latter is a typical scenario where a mobile agent works on; therefore, integrating mobile agents with suitable tools for managing ontologies would enrich its capabilities. It is worth pointing out that an act of communication between two agents is feasible only if a common ontology is shared. Even if this restriction guarantees an agreement on the semantics of exchanged data, not all data are a priori shared, thus agents must be able to reach an on-the-fly agreement by measuring semantic similarity of different ontologies. Whenever an agent acquires new information, it can enrich its personal knowledge base.

### A. Running Example

Suppose a biologist has prepared his experiment within an interactive virtual laboratory for Bioinformatics. The experiment consists of the set of concurrent and coordinated activities each of which is described by using the specific terminology taken from an ontology of the biological domain. The execution of the experiment is delegated to the runtime support of the virtual laboratory. Suppose to have a computational environment based on middleware for mobile computation, where every experiment is compiled in a pool of mobile agents activated to support the execution of the whole experiment. Also, suppose that one of the agents is involved in the execution of a query, which implies its migration across several places to query different data repositories. Let

*“Find all Complementary DNA transcribed from Messenger RNA whose DNA is ... ”*

be a meaningful query for a biologist. We can observe that *“Complementary DNA”*, *“Messenger RNA”* and *“DNA”* are terms that identifies domain specific concepts while *“Transcribed from”* is a relation between two concepts.

As the mobile agent reaches a destination, it will interact with a local stationary service agent, passing on to it the query and the reference of the domain ontology. If the service agent shared the domain ontology it will just translate the query in the local format. If not, it will offer its local ontology to the mobile agent, which in turn will decide whether to come to an agreement or to move to next place. The final decision could be taken over the result given by the similarity function measured between the two ontologies. Once the mobile agent has decided which are the most similar concepts to those describing its query, it will rewrite the query in terms of new concepts, and submit the query to the local service agent. Afterward, the service agent will be able to convert the incoming query to a corresponding local one. In some cases, the mobile agent, could decide to enrich his knowledge with new learned concepts, which could be also used by the biologist to interpret the extracted data. Then, the mobile agent moves to the next place.

What are suitable tools to support the interaction of mobile agents with local service agent? To provide an answer, we define the *ontological graph* an abstract data model more flexible and light than an ontology; suitable to map every ontology a mobile agent will manage and analyze during its migration. In the next section, before the definition of

*ontological graph* the general concept of ontology will be introduced.

### III. ONTOLOGICAL GRAPH MODEL

Gruber in [13] defines an ontology as an explicit specification of a conceptualization. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. We can note that every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly. Usually, agents share a common specification (common ontology) which supports the communication with each other and they commit “on-the-fly” specifications (local ontology) to operate on a specific domain. Guarino ([14]) underlines that interoperating systems need two types of ontologies: a top-level and a domain-level ontology. The top-level ontology describes very general common concepts (e.g primitives of a communication protocol [9] or concepts as time, space, event, etc.) which are independent of a particular problem or domain, by which to verify the consensus on sharing a generic domain ontology. The domain ontology describes the vocabulary related to a specific domain (like Biomedicine, Molecular Biology, etc.) or a specific task or activity (like Protein, Enzyme, etc.) by specializing the terms introduced in a top-level ontology. The evolution of ontology research in computer science recently shifted from theoretical to practical issues. Noy and Klein report in [26] that issues like what a formal ontology is - what requirements an ontology must satisfy - what representation language is suitable to define and exchange ontologies, shifted to issues associated with the use of ontologies in real-world, for large-scale applications, like how to use multiple distributed heterogeneous ontologies - how to maintain updated an ontology - how to integrate similar ontologies, etc.

An ontology, denoted by  $O$ , is a formal specification of a conceptualization, that is the knowledge structure that describes, using a lexicon, the semantics of a given domain. A lexicon is defined in [28], as a “knowledge-base about some subset of words in the vocabulary of a natural language denoting concepts of the domain and relations among concepts”.

#### Notations

For ease of notation and retention of all definitions, in the sequel of this paper, we will use the following notation.

Let:

$\mathcal{L}$  be the set of lexicons, ranged over by  $L_1, L_2 \dots L, L', L''$ ;

$\mathcal{O}$  be a set of ontologies, ranged over by  $O_1, O_2, \dots O, \bar{O}, \bar{O}', \bar{O}''$ ;

$C$  be a set of ontological concepts, ranged over by  $c_1, c_2, \dots c, c', c''$ ;

$R$  be a set of ontological relations, ranged over by  $r_1, r_2, \dots r, r', r''$ ;

$N, N_1, N_2 \dots H, K$  be a set of nodes, ranged over by  $n_1, n_2, \dots, n, m$ ;

$A, A_1, A_2 \dots$  be a set of arcs, ranged over by  $a, a_1, a_2, \dots$

Since, a lexicon contains terms to identify both concepts and relations which semantically describe the domain, in the sequel of this paper, we will separately use concepts and relations as terms of a given lexicon. Therefore, a lexicon  $L$  can be represented as a finite, not empty set  $C$  of concepts and a not empty set  $R$  of relations among concepts.  $L$  can be represented as  $L = \langle C, R \rangle$ . We assume that  $R$  contains a special relation *similar* which will be used to denote a similarity relation between concepts.

Formally, an ontology  $O$  is a, node and arc labelled, graph, where the labelling functions are expressed over a lexicon  $L \in \mathcal{L}$ ; the set of graph nodes represents concepts and the set of arcs represents relations between concepts. The association between a node and a concept, so as between an arc and a relation, is unique. Any concept can be described by its lexical name and the set of relations it has with other concepts. A concept is represented by a rooted subgraph.

An *ontological graph* is formalized by the following definition:

*Definition 3.1 (Ontological Graph):* An *ontological graph*  $O = (N, A, n)$  is a directed, rooted, node and arc labelled over a lexicon  $L = \langle C, R \rangle$ , graph. Where  $N$  is the finite set of ontology concepts,  $A$  is the finite set of relations among concepts and  $n$  is the root. The node labelling function,  $\lambda : N \rightarrow C$  uniquely associates a node to a concept in the lexicon. The arc labelling function  $\delta : A \rightarrow R$  uniquely associates an arc to a relation in the lexicon.

The functions,  $\lambda$  and  $\delta$ , are neither injective nor surjective mapping function; this property allows the existence of concepts and relations in the ontology, that are not expressed in the *ontological graph*.

Each ontology  $O \in \mathcal{O}$  is associated to the corresponding *ontological graph*  $O$ . Each concept  $c \in C$  is associated to the corresponding subgraph. Each subgraph is, in turn, an *ontological graph*. Each node  $n$  of an *ontological graph* is associated to the name of the concept described by the *ontological graph* rooted on the node  $n$ .

In the sequel of this paper, both ontology and *ontological graph*, so as concept name and node, concept and *ontological graph*, will be used interchangeably.

Figure 1 shows the *ontological graph* corresponding to a small set of concepts in Bioinformatics, whose lexicon is  $L_1 = \{DNA, RNA, Ribozyme, Nucleotide, Ribonucleotide, Deoxinucleotide, Nucleic-acid, Protein, Macromolecule, Complementary-DNA, Messenger-RNA; Polymer-of, Subclass-of, Transcribe-from, Translate-to\}$ . In the figure, the concept of *Ribozyme* is described in the subgraph rooted at the node labelled by *Ribozyme*. Thus, the *Ribozyme* is *Subclass-of* a *Macromolecule* and a *Polymer-of* the *Ribonucleodite*; this latter, in turn is a *Subclass-of Nucleotide*. A *Nucleotide*, in this conceptualization, is a primitive concept of the domain, a leaf node of the graph. The Molecular Biology ontology used to derived the *ontological graph* has been taken from TAMBIS

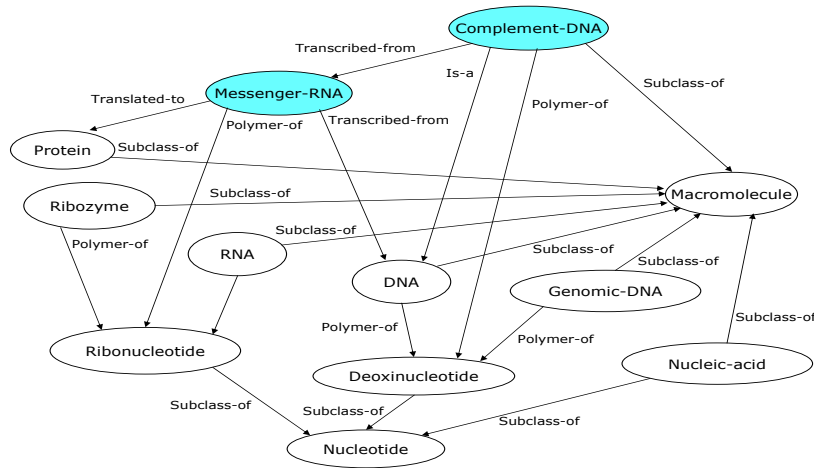


Fig. 1. An ontological graph for the lexicon  $L_1$

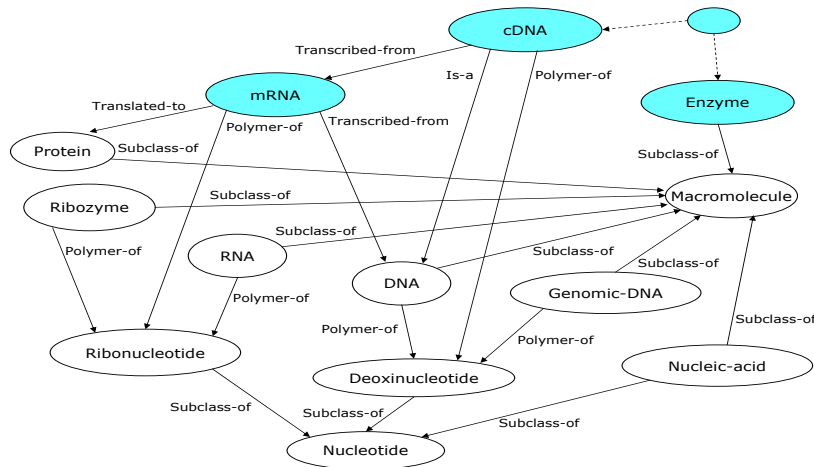


Fig. 2. ontological graph for the lexicon  $L_2$

project [12] in OIL [18], [29] description.

In Figure 2, we consider an *ontological graph* corresponding to a different lexicon  $L_2 = \{DNA, RNA, Ribozyme, Nucleotide, Ribonucleotide, Deoxynucleotide, Nucleic-acid, Protein, Macromolecule, cDNA, mRNA, Enzyme; Polymer-of, Is-a, Subclass-of, Transcribe-from, Translate-to\}$ . Figure 2 shows that the *Ribozyme* concept is described by the same subgraph of Figure 1. Whereas, the concept of *Complementary-DNA* from the lexicon  $L_1$  and *cDNA* from the lexicon  $L_2$  are described by two different subgraphs whose degree of similarity will be later discussed.

A. Algebraic operators

To allow the manipulation of ontologies by *ontological graph*, we concentrate on a minimum set of operators necessary to measure on-the-fly the similarity among concepts of different ontologies. The three main operators are: *projection*, *similarity* and *enriching* (see Table I).

<i>projection</i>	$\pi : \mathcal{O} \times \mathcal{N} \rightarrow \mathcal{O}$
<i>similarity</i>	$\sigma : \mathcal{O} \times \mathcal{O} \rightarrow [0, 1]$
<i>enriching</i>	$\epsilon : \mathcal{O} \times \mathcal{O} \rightarrow \mathcal{O}$

TABLE I  
ALGEBRAIC OPERATORS

The *projection* allows to reduce the *ontological graph* into a subgraph whose root node corresponds to a given concept. The *similarity* operator is a function which measures the similarity of two concepts and returns a coefficient that ranges over by  $[0,1]$ ; the coefficient is 1, if the two concepts are equal; it is

0 if they completely mismatch. The *enriching* operator allows to enrich the *ontological graph* with new concepts.

The small set of operators could be easily extended with other operators, for example those proposed by Mitra et al. in [25] for ontology composition: *Select*, *Intersection*, *Union* and *Difference*;

In the following, the description of behaviour of the proposed operators is given. We have omitted the formal definitions which can be found in Appendix I.

1) *Projection*  $\pi$ : The projection of an *ontological graph* over a given concept, reduces the *ontological graph* by isolating the subgraph consisting of those concepts and relations – except for the relation *similar* – that describe the given concept. The root node of the projected graph is represented by the concept itself.

The *projection* is a binary operator defined over an *ontological graph* and a concept name.

$$\pi : \mathcal{O} \times N \rightarrow \mathcal{O}$$

Given an *ontological graph*  $O=(N,A,n)$  and a node  $m \in N$ , the *projection* of  $O$  over  $m$  returns the subgraph  $O'$ , rooted in  $m$  corresponding to the concept associated to  $m$ . Suppose we wish to isolate the concept *Ribozyme* in the *ontological graph*  $O$  given in Figure 1, the *projection* operator can be used as follows

$$\pi(O, \text{Ribozyme}) = O'$$

Figure 3 shows the graphical behaviour of the *projection* operator over the example.

2) *Similarity*  $\sigma$ : The measure of the similarity between two concepts determines how much the two corresponding *ontological graphs* are similar. There are several ways to measure the similarity among two concepts. Giunchiglia in [11] proposes to classify the process of discovering the graphs mapping in syntactic and semantic matching. The syntactic similarity (matching) [21] is based on searching the semantic correspondence among node labels, the resulting coefficient, that ranges over  $[0,1]$ , measures the similarity between the labels of the given nodes by performing linguistic analysis. The semantic similarity is based on analyzing the position that a node has in a graph, that can be done either analysis the position of a given node in terms of neighbours nodes [11], or by following a path in the graph [3] or by analyzing both the semantic and the syntactic concepts matching, as Maedche et al. propose in [22]. In the above cases, the similarity algorithm returns a coefficient that ranges over  $[0,1]$ , except for the Giunchiglia algorithm that returns a set of values that range over  $\{=, \subseteq, \supseteq, \perp\}$  (equality, more specific, more general and mismatch respectively). In Section IV, we propose a new approach which determines semantic similarity by clustering concepts satisfying common relations. The algorithm allows to measure on-the fly semantic similarity without sharing a domain ontology.

The *similarity* operator allows to measure the similarity between two concepts of different *ontological graphs*. The

*similarity* returns a coefficient that ranges over  $[0,1]$ . How the similarity is measured, it depends on the algorithm chosen to implement the operator. In any case, two concepts are equal if the *similarity* returns 1, two concepts mismatch (no affinity) if the *similarity* returns 0.

The *similarity* is a function defined over two *ontological graphs*.

$$\sigma : \mathcal{O} \times \mathcal{O} \rightarrow [0, 1]$$

Given two *ontological graphs*  $O=(N_1, A_1, n)$  and  $O'=(N_2, A_2, m)$  respectively, the *similarity* of  $O$  and  $O'$  over the two root nodes  $n$  and  $m$  returns a real number  $\alpha \in [0, 1]$  that quantitatively estimates the similarity degree of the concepts described by the *ontological graphs*.

Suppose we wish to measure the similarity between the concept *Ribozyme* in the *ontological graph*  $O=(N_1, A_1, \text{Ribozyme})$  given in Figure 1, and *Ribozyme* in the *ontological graph*  $O'=(N_2, A_2, \text{Ribozyme})$  given in Figure 2, the *similarity* operator most likely will return the value 1.

$$\sigma(O, O') = 1$$

If we measure the similarity between *Complementary-DNA* and *cDNA* described in the two graphs respectively, the *similarity* operator will return the value  $\alpha$  depending on the algorithm that implements the operator.

$$\sigma(\text{Complementary-DNA}, \text{cDNA}) = \alpha$$

3) *Enrichment*  $\epsilon$ : When an agent discovers a new concept, he can decide to enrich his knowledge by storing the new knowledge. This can be done in several ways, by creating a new data structure or by adding the projection of the new concept to its *ontological graph*. The *enriching* is an operator defined over two concepts, i.e. two *ontological graphs*:

$$\epsilon : \mathcal{O} \times \mathcal{O} \rightarrow \mathcal{O}$$

Given two *ontological graphs*  $O=(N_1, A_1, n)$  and  $O'=(N_2, A_2, m)$ , representing two similar concepts, the enrichment of  $O$  with  $O'$  is obtained by adding a new arc from the root  $n$  of  $O$  to the root  $m$  of  $O'$  labelled by “*similar*”; *similar* is a special relation meaningful only for the agent purpose. Suppose we wish to store the knowledge that *cDNA* is similar to *Complementary-DNA*, we can use the *enriching* operator between  $O=(N_1, A_1, \text{Complementary-DNA})$ ,  $O'=(N_2, A_2, \text{cDNA})$  as follows

$$\epsilon(O, O') = \tilde{O}$$

The resulting graph  $\tilde{O}$  is depicted in Figure 4.

#### IV. A STRUCTURAL SEMANTIC SIMILARITY FUNCTION

In this section, a new function to assess the semantic similarity between concepts is proposed. We only compare the structural (topological) similarity among sets of concepts without considering syntactic matching between node labels.

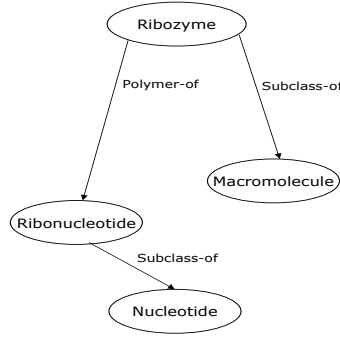


Fig. 3. The projected *ontological graph* in Figure 1 over the node “Ribozyme”

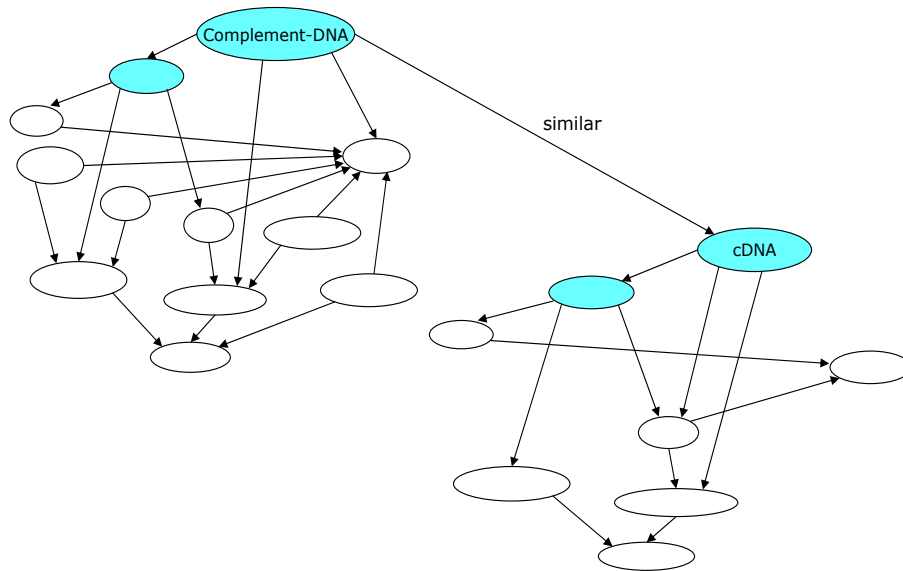


Fig. 4. Enriching operator over the running example

Given two *ontological graph*  $O_1=(N_1, A_1, n)$ ,  $O_2=(N_2, A_2, m)$  and two sets of nodes  $H \subseteq N_1$  and  $K \subseteq N_2$ , with  $H \neq \emptyset \neq K$ , the similarity between  $H$  and  $K$  is measured by the function in Table II, where, for any given set of nodes  $N$ ,  $\tau(N)$  is the number of outgoing arcs from (nodes in)  $N$ ,  $R_N$  is the set of relations – different from *similar* – associated to the arcs outgoing from  $N$ , and  $sons_r(N)$  is the set of nodes reachable from any node in  $N$  through the relation  $r \in R$ .

Intuitively, two set of concepts  $H$  and  $K$  are equal (mismatch) if both of them (one but not the other) contain only primitive concepts, that is, have no outgoing arcs. Otherwise, if both  $H$  and  $K$  contain no primitive concepts (the number of outgoing arcs from  $H$  and  $K$  is greater than zero), we consider the set  $R_H \cap R_K$  of the relations they have in common. For

each  $r \in R_H \cap R_K$ , we recursively apply  $f$  to the sets of nodes reachable from  $H$  and  $K$  through the relation  $r$ .

It is worth noting, that the proposed similarity function is relevant when the relation are meaningful for the application domain.

## V. ON-THE-FLY CONCEPTS COMPARISON ACROSS ONTOLOGIES

As we mention in the introduction, in the context of multiple information systems, the semantic interoperability must allow users to enter a request without knowing where and how data are stored.

We have implemented the biological example described in Section II, in the framework of Hermes, middleware for mobile computation proposed by Corradini et al. in [5], [6].

$$f(H, K) = \begin{cases} 1 & \text{iff } \tau(H) = \tau(K) = 0 \\ 0 & \text{iff } \tau(H) = 0 \text{ xor } \tau(K) = 0 \\ \frac{\sum_{r \in R_H \cap R_K} f(\text{sons}_r(H), \text{sons}_r(K))}{\text{card}(R_H \cup R_K)} & \text{otherwise} \end{cases}$$

TABLE II  
THE SIMILARITY FUNCTION

As Figure 5 shows, Hermes is structured as a component-based system with 3-layer software architecture: *user layer*, *system layer* and *run-time layer*. At the user layer, it allows users to specify their application as a workflow of activities using the graphical notation provided by DroFlo (OpenWFE, 2005) and JaWE editor (Enhydra, 2003). At the system layer, it provides a context-aware compiler to generate a pool of user mobile agents from the workflow specification. At the run-time layer, it supports the activation of a set of specialized service agents, and it provides all necessary components to support agent mobility and communication. Hermes can be configured for specific application domains by adding domain-specific components.

Biological workflow	User Layer
Workflow Management	
Bio-agents	System Layer
Context-aware compiler	
Bio-Service Agent	Run-Time Layer
Core	

Fig. 5. Hermes Software Architecture

Suppose to have Hermes as middleware that allows the interoperability among different information systems and suppose to have a service agent, *ontology service agent* which interfaces local repositories and interacts with mobile agents to allow semantic querying. Let us consider the running example described in Section II, and suppose that the definition of a workflow in the biological domain consists of a single task: retrieval of information about *Complement-DNA*. Also assume that the ontological graphs in Figures 1 and 2 (denoted in the following by  $O_1$  and  $O_2$ ) are used to describe the query and the remote data schema repositories, respectively.

In the Hermes context, a mobile agent (called bio-agent) is created, by using  $L_1$  related to  $O_1$  on platform 1, to support the execution of the workflow activity. The bio-agent has the goal to move to platform 2 and to search the concept *Complement-DNA*. The bio-agent starts to extract the subgraph of *Complement-DNA* from the derived *ontological*

*graph*  $O_1$  (Figure 1) and moves to platform 2. There, it will interact with a local stationary service agent, passing on to it the query and the reference of the domain ontology. If the service agent shared the domain ontology it will just translate the query in the local format. If not, it will offer its local ontology to the bio-agent. The bio-agent will search for the most similar *Complement-DNA* concept on  $O_2$ . Note that nodes and arcs of the subgraphs are indicated by corresponding labels in the lexicon. The *Complement-DNA* concept is compared to all the subgraphs, thus the agent will compute  $f(\text{Complement-DNA}, \text{mRNA}) = 0.3333^1$  and  $f(\text{Complement-DNA}, \text{Enzyme}) = 0.333$ . To evaluate  $f(\text{Complement-DNA}, \text{cDNA})$ , the agent needs to evaluate the following values:  $f(\text{Messenger-RNA}, \text{mRNA}) = \frac{1+1+1}{4} = \frac{3}{4} = 0.75$  and  $f(\text{Deoxinucleotide}, \text{Deoxinucleotide}) = 1$ .

The agent can now complete the comparison between *Complement-DNA* and *cDNA*, computing  $f(\text{Complement-DNA}, \text{cDNA}) = \frac{0.75+1}{4} = \frac{1.75}{4} = 0.4375$ , which yields *cDNA* as the concept most similar to *Complement-DNA*, with a similarity degree of 0.4375.

Having obtained the degree of similarity between *Complement-DNA* on platform 1 and *cDNA* on platform 2, the mobile agent can ask service agent to extract information (projection) about *cDNA* from platform 2, having learned that *cDNA* is “sufficient” similar to *Complement-DNA*, it will enrich its knowledge. Then, it will rewrite the query and proceed in the task execution.

#### A. Related Work

A general approach to data integration has been to map the local terms of distinct ontologies onto a single shared ontology, as described in [27]. In this work the semantic similarity is typically determined as a function of the path distance between terms in the hierarchical structure underlining this ontology [4]. Another strategies for ontology integration are based on the mapping of a local ontology onto a more generic ontology [1], [30]. The ONIONS methodology [1] integrates local ontologies by inheriting from shared generic ontologies, but does not automatically compare concepts (as proposed herein). The OBSERVER [23] ontology-based system combines intensional and extensional analysis to calculate lower and upper bounds for the precision and recall of queries that are translated across ontologies.

<sup>1</sup>Given two nodes  $n$  and  $m$ , we write  $f(n, m)$  to denote  $f(\{n\}, \{m\})$

Weinstein et al. [30] propose differentiated ontologies as support to communications in distributed systems subject to semantic heterogeneity. Ontologies are described by using Description Logic. Concepts are formally defined in relation to other concepts, so that concepts in local ontologies inherit definitional structure from concepts in shared ontologies.

Recently, Rodriguez in [27], has suggested that in the area of information retrieval and data integration, the use of ontologies and semantic similarity functions have recently been emphasized as a mechanism for comparing objects that can be retrieved or integrated across heterogeneous repositories [16]. The authors proposed a model for semantic similarity among Entity Classes from different ontologies. Ontologies are described as objects by using BNF. The similarity model provides a systematic way to detect similar entity classes across ontologies based on the matching process of each of the specification components in the entity class representation (i.e., synonym sets, distinguishing features, and semantic neighborhoods).

The approach presented in this paper measures the structural similarity by only considering the relations among concepts.

## VI. CONCLUSIONS AND FUTURE WORK

The present work aims at integrating ontologies in mobile agents. This approach allows the information retrieval and data integration in a scenario where a pool of mobile agents can migrate across different data repositories where updated information can be instantaneously retrieved.

The integration ontologies and mobile agents allows to discover new knowledge by combining information extracted from different data repositories and to move computational tools over data, by delegating a software entity. This approach supports the decentralization of the execution of local activities, to avoid the warehousing of highly dynamic data, to reduce network traffic and to free the users from network faults and from the need to be continuously connected to a laptop.

Future work will be geared towards reducing complexity through hypergraphs in place of graphs, generalizing the similarity function to cyclic graphs, exploiting the similar relation in the definition of similarity function and validating the proposed approach on a real application.

## Acknowledgements

This research was partially supported by the MIUR strategic project Oncology over Internet and FIRB project LITBIO.

## REFERENCES

- [1] A. Gangemi, M. Pisanelli, and G. Steve. Some requirements and experiences in integrating terminological ontologies in medicine. In *Eleventh Workshop on Knowledge Acquisition, Modelling and Management*, 1998.
- [2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
- [3] P. Bouquet, G. Kuper, M. Scoz, and S. Zonobini. Asking and answering semantic queries. In *Proceedings of Meaning Coordination and Negotiation Workshop (MCNW-04) in conjunction with International Semantic Web Conference (ISWC-04)*, 2004.
- [4] M. Bright, A. Hurson, and S. Pakzad. Automated resolution of semantic heterogeneity. *ACM Transaction on Distributed Systems*, 19(2):212–253, 1997.
- [5] F. Corradini, L. Mariani, and E. Merelli. An agent-based approach to tool integration. *Journal of Software Tools Technology Transfer*, 6:231–244, 2004.
- [6] F. Corradini and E. Merelli. Hermes: agent-based middleware for mobile computing. In *Tutorial Book of SFM-05*. Springer-Verlag, 2005. LNCS 3465.
- [7] R. Culmone, Gloria Rossi, and Emanuela Merelli. An ontology similarity algorithm for bioagent. In *NETTAB Workshop on Agents in Bioinformatics*, July 12–14 2002.
- [8] D. Fensel. *Ontologies: a silver bullet for Knowledge Management and Electronic Commerce*. Springer, 2001.
- [9] The Foundation for Intelligent Physical Agents. <http://www.fipa.org/>.
- [10] Gustavo A. Gimnez-Lugo, Analia Amandi, Jaime Simo Sichman, and Daniela Godoy. Enriching information agents' knowledge by ontology comparison: A case study. In M. Toro F.J. Garijo, J.C. Riquelme, editor, *Advances in Artificial Intelligence - IBERAMIA 2002: 8th Ibero-American Conference on AI, Seville, Spain, November 12–15, 2002. Proceedings*, volume 2527 of *Lecture Notes in Artificial Intelligence*, pages 546–555. Springer-Verlag, 2003.
- [11] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-match: an algorithm and an implementation of semantic matching. In Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab, and M. Uschold, editors, *Semantic Interoperability and Integration*, number 04391 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005.
- [12] C. A. Goble, R. Stevens, G. Ng, S. Bechhofer, N. W. Paton, P. G. Baker, M. Peim, and A. Brass. Transparent access to multiple bioinformatics information sources. *IBM Systems Journal*, 40(2):532–551, 2001.
- [13] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Technical Report 93-04, Knowledge Systems Laboratory, Stanford University, 1993.
- [14] N. Guarino. Formal ontology and information systems. In *Proceedings of FOIS - Formal Ontology in Informatin Systems*, 1998.
- [15] N. Guarino and P. Giaretta. Ontologies and knowledge bases: Towards a terminological clarification. In N.J.I. Mars, editor, *Towards Very Large Knowledge Bases*. IOS Press, 1995.
- [16] N. Guarino, C. Masolo, and G. Verete. Ontoseek: Content-based access to the web. *IEEE Tran. on Information Systems*, 14(3):70–80, 1999.
- [17] J. Hendler. Agents and the semantic Web. *IEEE Intelligent Systems*, 16(1):30–37, January/ February 2001.
- [18] I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, R. Studer, and E. Motta. The ontology inference layer OIL. Technical report, Free University of Amsterdam, 2000.
- [19] Nicholas R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117:277–296, 2000.
- [20] Yannis Kalfoglou, Marco Schorlemmer, Michael Uschold, Amit Sheth, and Steffen Staab. 04391 – semantic interoperability and integration: Executive summary. In Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab, and M. Uschold, editors, *Semantic Interoperability and Integration*, number 04391 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005. <<http://drops.dagstuhl.de/opus/volltexte/2005/50>> [date of citation: 2005-01-01].
- [21] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 49–58, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [22] Alexander Maedche and Steffen Staab. Measuring similarity between ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW2002)*, pages 251–263, Madrid, Spain, October 1–4 2002.
- [23] E. Mena, A. Illarramendi, V. Kashyap, and A. Sheth. Observer: an approach for query processing in global information system based on interoperation across preexisting ontologies. *Distributed and parallel Databases*, 8(2):223–271, 2000.
- [24] P. Mitra, G. Wiederhold, and M. L. Kersten. A graph-oriented model for articulation of ontology interdependencies. In *Extending Database Technology*, pages 86–100, 2000.
- [25] Prasenjit Mitra and Gio Wiederhold. An algebra for semantic interoperability of information sources. In IEEE, editor, : *2nd IEEE International Symposium on Bioinformatics and Bioengineering (BIBE'01)*, 2001.
- [26] F. N. Noy and M. Klein. Ontology evolution: not the same as schema evolution. *Knowledge and Information Systems*, (6):428–440, march 2004.



- [27] A. Rodriguez and M. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE Tran. on Knowledge and Data Eng.*, 16(2):442–456, 2003.
- [28] J.F Sowa. *Knowledge representation, logical, philosophical and Computational Foundations*. Brooks/Cole, 2000.
- [29] R. Stevens. An ontology of molecular biology and the questions that can be performed on the resources containing data about molecular biology. <http://img.cs.man.ac.uk/stevens/tambis-oil.html>.
- [30] P. C. Weinstein and W. P. Birmingham. Comparing concepts in differentiated ontologies. In *Workshop on Knowledge Acquisition, Modeling and Management (KAW)*, 1999.
- [31] Michael Wooldridge and Nicholas R. Jennings. Agent theories, architectures and languages: a survey. In Michael Wooldridge and Nicholas R. Jennings, editors, *Intelligent Agents*, Lecture Notes in Computer Science, pages 1–22. Springer-Verlag, Berlin, 1995.

$$\sigma(\mathbf{O}, \mathbf{O}') = f(n, n')$$

### Properties of the similarity

- $\sigma(\mathbf{O}, \mathbf{O}) = 1$
- $\sigma(\mathbf{O}, \emptyset) = \sigma(\emptyset, \mathbf{O}) = \sigma(\emptyset, \emptyset) = 0$
- $\sigma(\mathbf{O}, \mathbf{O}') = \sigma(\mathbf{O}', \mathbf{O})$

Several similarity functions can be defined over two ontologies, as long as, they return a real positive number that ranges over by  $[0, 1]$ ; 0 means no affinity and 1 overlapping (equivalence) of the two ontological concepts.

As an example, the similarity function has been discussed in details in Section IV.

3) *Enrichment*  $\epsilon$ : The *enriching* is a binary operator defined over *ontological graphs*. Given two *ontological graphs*  $\mathbf{O}$ ,  $\mathbf{O}'$  the *enriching* of  $\mathbf{O}$  with  $\mathbf{O}'$  returns  $\mathbf{O}$  enriched by adding a new arc from the root of  $\mathbf{O}$  to the root of  $\mathbf{O}'$  labelled by *similar*.

*Definition 1.3 (Enrichment)*: Let  $\mathbf{O}=(N, A, n)$ ,  $\mathbf{O}'=(N', A', n')$  be two *ontological graphs*. The enrichment of  $\mathbf{O}$  over  $\mathbf{O}'$  is defined as:

$$\epsilon(\mathbf{O}, \mathbf{O}') = \bar{\mathbf{O}}$$

where  $\bar{\mathbf{O}}=(\{N \cup N'\}, \{A \cup A' \cup \{(n, n')\}, n\})$  and  $\delta(n, n') = \text{similar}$ .

$$\pi(\mathbf{O}, m) = \bar{\mathbf{O}}$$

where  $\bar{\mathbf{O}} = (N_1, A_1, m)$  is the subgraph of  $\mathbf{O}$  such that:

- 1)  $N_1$  is the set of nodes  $n_j \in N$  such that either  $n_j = m$  or  $\exists n_0, n_1, \dots, n_j \in N$ , with  $j \geq 1$ , such that  $n_0 = m$  and  $(n_0, n_1), \dots, (n_{j-1}, n_j) \in A$
- 2)  $A_1 = \{(n_1, n_2) \in A \mid n_1, n_2 \in N_1\}$

### Properties of the projection

- $\pi(\mathbf{O}, \text{null}) = (\emptyset, \emptyset, \text{null})$
- $\pi(\mathbf{O}, n) = (\emptyset, \emptyset, \text{null})$  se  $n \notin N$
- $\pi(\emptyset, n) = (\emptyset, \emptyset, \text{null})$

2) *Similarity*  $\sigma$ : The *similarity* is a function defined over two *ontological graphs*. Given two *ontological graphs*  $\mathbf{O}$ ,  $\mathbf{O}'$  and two node  $n$ ,  $m$ , the *similarity* of  $\mathbf{O}$  and  $\mathbf{O}'$  over  $n$  and  $m$  returns a real number  $\alpha \in [0, 1]$ ;  $\alpha$ , quantitatively estimates the similarity degree of the two concepts.

*Definition 1.2 (Similarity)*: Let  $\mathbf{O}=(N, A, n)$  and  $\mathbf{O}'=(N', A', n')$  be two *ontological graphs*. The *similarity* of  $\mathbf{O}$  and  $\mathbf{O}'$  is defined as:

$$\sigma(\mathbf{O}, \mathbf{O}') = \alpha$$

where  $\alpha \in [0, 1]$ .

The similarity operator can be implemented with one of the algorithms available in literature [7], [22], [27] or the one introduced in Section IV, in that case the similarity will be as follows