Albertoni R., De Martino M., Asymmetric and context-dependent semantic similarity among ontology instances**,**

Journal on Data Semantics X

Series: Lecture Notes in Computer Science , Vol.4900
Sublibrary: Information Systems and Applications, incl. Internet/Web, and HCI
Spaccapietra, Stefano (Ed.)
2008, XIII, 265 p., Softcover
ISBN: 978-3-540-77687-1
Doi: 10.1007/978-3-540-77688-8_1

Asymmetric and context-dependent semantic similarity among ontology instances

Riccardo Albertoni, Monica De Martino

CNR-IMATI,
Via De Marini, 6 – Torre di Francia - 16149 Genova, Italy
{albertoni, demartino}@ge.imati.cnr.it

Abstract. In this paper we propose an asymmetric semantic similarity among instances within an ontology. We aim to define a measurement of semantic similarity that exploit as much as possible the knowledge stored in the ontology taking into account different hints hidden in the ontology definition. The proposed similarity measurement considers different existing similarities, which we have combined and extended. Moreover, the similarity assessment is explicitly parameterised according to the criteria induced by the context. The parameterisation aims to assist the user in the decision making pertaining to similarity evaluation, as the criteria can be refined according to user needs. Experiments and an evaluation of the similarity assessment are presented showing the efficiency of the method.

1 Introduction

Semantic similarity plays an important role in information systems as it supports the identification of objects that are conceptually close but not identical. Similarity assessment is particularly significant in different areas of knowledge management (such as data retrieval, information integration, and data mining) because it facilitates the comparison of the information resources in different types of domain knowledge [1,2].

Nowadays domain knowledge is often available in the form of an ontology, which reflects the understanding of a domain that a community has agreed upon. An ontology consists of different parts, including a set of concepts and their mutual relations and instances. In particular, ontologies have recently been imposed as means of organizing the *metadata* (called ontology-driven metadata) of complex information resources. According to Sheth et al. [3] ontology-driven metadata provide syntactic and semantic information about complex information resources. *Syntactic metadata* describe non-contextual information about the content (e.g. language, a bit rate, format). This offers no insight into the meaning of a document. In contrast, *semantic metadata* describe domain specific information about the content and contextual information, such as which entities take part in the production and usage of the information resource. The metadata of the resources are encoded as instances in the

ontology. Therefore, the definition of a method for assessing the semantic similarity among ontology instances becomes essential in order to compare all these complex resources.

The concept of similarity among information resources is not univocal as it is affected by the human way of thinking as well as by the application domain [4]. Its evaluation cannot ignore some cognitive properties related to the human way of perceiving the similarity. In particular, we underline three main aspects. Firstly, considering that, in the naïve view of the word, similarities defined in terms of a conceptual distance are frequently asymmetric, the formulation of similarity should for many applications provide an asymmetric evaluation [5]. Secondly, it should be flexible and adaptable to different application contexts, which affect the similarity criteria. Moreover, considering that part of the domain knowledge as it is perceived by the domain expert is already formalized in the ontology and the ontologies are artefacts whose definitions require time consuming and costly processes, the similarity evaluation should be able to exploit as much as possible all the hints that have already been expressed in the ontology.

So far, most of the research activity pertaining to similarity and ontologies has been carried out within the field of ontology alignment or in order to assess the similarity among concepts. Unfortunately, these methods produce results that are inappropriate for the similarity among instances. On the one hand, similarities for ontology alignment strongly focus on the comparison of the structural parts of distinct ontologies, and their application for assessing the similarity among instances might give misleading results. On the other hand, similarities among concepts mainly deal with the lexicographic database, ignoring the comparison of the values of the instances. Apart from these, few methods for assessing similarities among instances have been proposed. Unfortunately, these methods rarely take into account the different hints hidden in the ontology, and they do not consider that the ontology entities concur differently in the similarity assessment according to the application.

To overcome the limitations mentioned above, our ongoing research is aimed at defining a framework for assessing the semantic similarity among instances. This paper proposes an asymmetric similarity assessment, where the asymmetry is explicitly adopted to stress the principle of “containment” between the two sets of characteristics of two instances representative of two information resources. The similarity between two instances tends to be greater for instances that have a higher level of containment.
The measurement of the asymmetric semantic similarity is defined by an amalgamation function. The amalgamation function combines and extends different similarities already defined in literature: it takes into account both the structural comparison between two instances, in terms of the classes that the instances belong to, and the comparison between the attributes and relations of the instances. Moreover, the framework provides a parametric evaluation of the similarity with respect to different applications. The application induces the criteria of similarity, which are explicitly formalized in the application context. An application context models the importance of the entities, which concur in the assessment of similarity,

and the operations used to compare the instances. The parametric evaluation allows us to tailor the similarity assessment to specific application contexts, but also allows us to obtain different similarity assessments employing the same ontology.
The main framework contributions are:

- To exploit as much as possible the implicit knowledge stored in the ontology: the similarity assessment is set up by considering different kinds of hints in the ontology.
- To tailor the similarity assessment according to the needs arising from the specific application contexts: different similarity assessments can be defined for the same ontology, according to the criteria arising from different applications.
- To improve the decision making of the user in the similarity evaluation: as the similarity assessment is completely parameterized on context criteria, the criteria can be refined according to user needs.

This paper is an extension of an ongoing research programme whose first result has been presented previously [6]. Here, we aim to provide more information useful for exploiting our similarity evaluation: detailed illustrations of the motivations that are behind the principle of our approach are discussed and some scenarios are illustrated. In addition, the asymmetric property in the assessment is stressed and argued more deeply with each equation. The paper is organized as follows. In the first section, we illustrate the motivation and the scenario that drove us to the similarity definition. Then, after providing some useful assumptions (section 3), we discuss the main principle of the approach (section 4). The approach description is characterised by three main parts: context, ontology, data and knowledge layers according with the framework proposed by Ehrig et al. [7]. A formalization of the similarity criteria induced by the context is proposed as context layer (section 5). The ontology layer (section 6) and data and knowledge layer (section 7) are devoted to the definition of the similarity functions that characterize our approach, followed by two experiments and an evaluation of the results (section 8). At the end, we evaluate related works (section 9), underlining how they have been useful as a starting point for our research but how, contrary to the proposed framework, they do not fulfil the requirements and goal we address by our contributions.

2 Motivations and scenarios

This section discusses the motivations that are behind the design of our approach as well as the reference scenario that has been developed with respect to this work.

2.1 Motivations

Here we provide the motivations behind our approach underlying the need of a similarity evaluation among ontology instances that takes into account the hints hidden in the ontology as well as the dependence on the context. In particular, we aim to answer the following questions:

- Why define a semantic similarity among ontology instances?
- What is the role of the implicit knowledge expressed by the ontology in setting up a similarity assessment?
- What is the role of the application context in the similarity evaluation?

Why define a semantic similarity among ontology instances?
Defining a semantic similarity among ontology instances represents a challenging priority in future research as it will pave the way for the next wave of knowledge intensive methods that will facilitate intelligent browsing as well as information analysis.
Here we do not refer to similarity as a tool for identifying possible mapping or alignment among different ontologies. Rather, we address a different problem related to the comparison of the ontology instances. We realize the importance of solving this problem from our direct research experience working in the European founded Network of Excellence AIM@SHAPE [8].
Within the NoE AIM@SHAPE, ontology has been adopted to organize the metadata of complex information resources. Different ontologies are integrated to describe 3D / 2D models (i.e. models of mechanical objects, digital terrains or artefacts from cultural heritage) as well as the tools for processing the models [9,10,11]. From our experience, we realize that the ontology driven metadata definition turns out to be outrageously expensive in terms of man-month efforts needed, especially whenever the domain that is expected to be formalized is complex and compound. The "standard ontology technology" provides reasoning facilities that are very useful in supporting querying activity as well as in checking ontology consistency, but the current technology lacks an effective tool for comparing the resources (instances). In addition to efforts to formalize the ontology, domain experts are often quite willing to provide the domain knowledge required to characterize their resources. However, they are disappointed when their efforts do not result in any measure of similarity among the resources.
Aware of this shortcoming, we address our research efforts towards investigating how to better employ the information encoded in the ontology and to provide tools that exploit as much as possible the result of the aforementioned efforts [6,12].

What is the role of the knowledge expressed by the ontology in setting up a similarity assessment?
An ontology reflects the understanding of a domain, which a community has agreed upon. Gruber defines an ontology as "the specification of conceptualizations, used to help programs and humans share knowledge" [13].
There is a strong dependence between the knowledge provided by the domain expert in order to define the ontology and his expectation of the results of the semantic similarity. Actually, the domain expert will perceive a similarity that is based on the knowledge he has provided.
The main ontology components (concepts, relations, instances) as well as its structure are representative of the domain knowledge conceptualized in the ontology. Therefore, they provide the base on which to set up the different hints to define the similarity. Classes provide knowledge about the set of entities within the domain. Properties, namely relations and attributes, provide information about the interactions

between classes as well as further knowledge about the characteristics of concepts. Moreover, the class structure within the ontology is also relevant as the attributes and relations shared by the classes, as well as their depth in the ontology graph, are representative of the level of similarity among their instances. In our proposal, the similarity assessment takes advantage of all of these ontology entities, which are usually available in the most popular ontology languages. Other entities could be considered as long as more specific ontology languages are adopted.

What is the role of the application context in the similarity evaluation?
The definition of a similarity explicitly parameterized according to the context is essential because the similarity criteria depend on the application context. Two instances may be more closely related to each other in one context than in another since humans compare the instances according to their characteristics but the characteristics adopted vary with the context.
In particular, as a consequence of the explicit parameterization of the similarity with respect to the application context, it is possible to:

- Use the same ontology for different application contexts. The ontology design usually ignores the need to tailor the semantic similarity according to specific application contexts. In that case, to assess the similarity between two different applications, two distinct ontologies need to be defined instead of simply defining two contexts.
- Provide a tool for context tuning that supports the decision-making process of the ontology user. The user often has not clearly defined in his mind the set of characteristics relevant for the comparison of the instances, or his specification does not match the result induced by the information system. A parameterization of the semantic similarity measurement supports a refinement process of the similarity criteria. The parameterization provides a flexible and adaptable way to refine the assessment toward the expected results and, therefore, it reduces the gap between user-expected and system results.

2.2 Framework scenario

We have identified two main scenarios where the proposed similarity framework is relevant: scenario 1 refers to a similarity evaluation in different application contexts exploiting the same ontology; scenario 2 refers to the iterative criteria refinement process used to properly assess the similarity in accord with the expectations of the domain expert.
In both scenarios we assume that we have an ontology describing the metadata of the resources in a complex domain and that the different resources are already annotated according to this ontology driven metadata.
Two actors play important roles in the two scenarios:

- The user who is the domain expert and who is looking for the semantic similarity. He has the proper knowledge to formulate the similarity criteria in the domain.
- The ontology engineer who is in charge of defining the similarity assessment on the basis of the ontology design and the information provided by the domain expert. He plays the role of communication channel for the requests of the domain

expert, with the system defining the application context to properly parameterize the similarity assessment.

2.2.1 Scenario 1: two different application contexts

Fig. 1 illustrates the first scenario, which highlights the dependence of the similarity result on the similarity criteria induced by the application. The domain expert user formulates different similarity criteria in two different application contexts. The two sets of criteria are formalized by the ontology engineer according to the system formalization, and the evaluation is performed. Two different results of the similarity evaluation are provided by the system and represented by similarity matrices. It is evident in this scenario how two application contexts induce two different similarity matrices just by exploiting the same ontology.

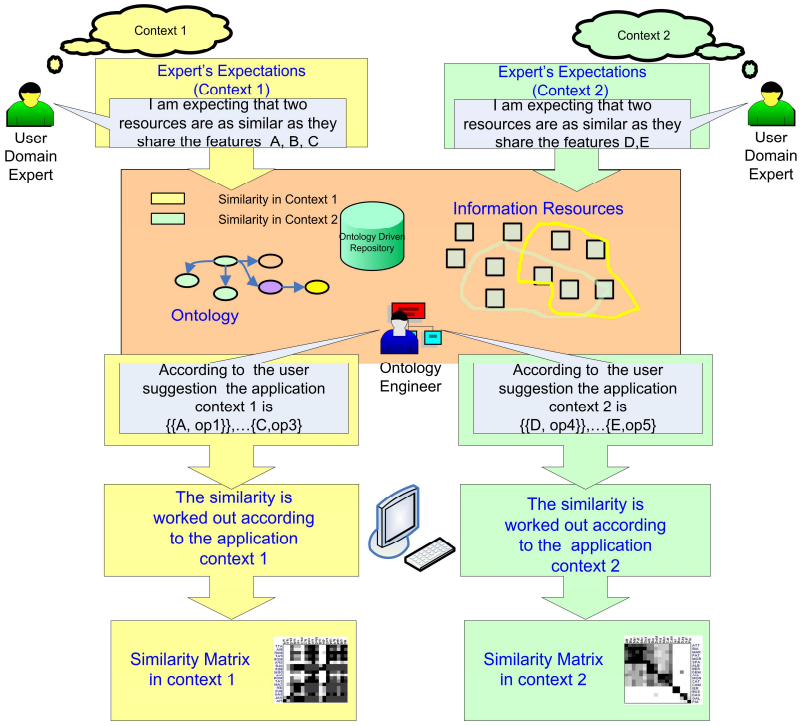

Fig. 1. Scenario 1: similarity evaluation according to different application contexts.

2.2.2 Scenario 2: similarity criteria refinement

This scenario is characterized by an interactive exchange of information between the two actors. The domain expert browses the repository looking for similar resources. He relies on his domain of knowledge to compare the resources, perceives the similarities among resources (which are not provided directly from the standard ontology reasoning technology), and provides some informal similarity criteria to be

adopted in the similarity evaluation. The ontology engineer translates the user requests to the system: he figures out which ontology entities are relevant and how to use them during the similarity assessment. The ontology engineer runs the similarity evaluation proposed in this paper and he shows the result to the domain expert.

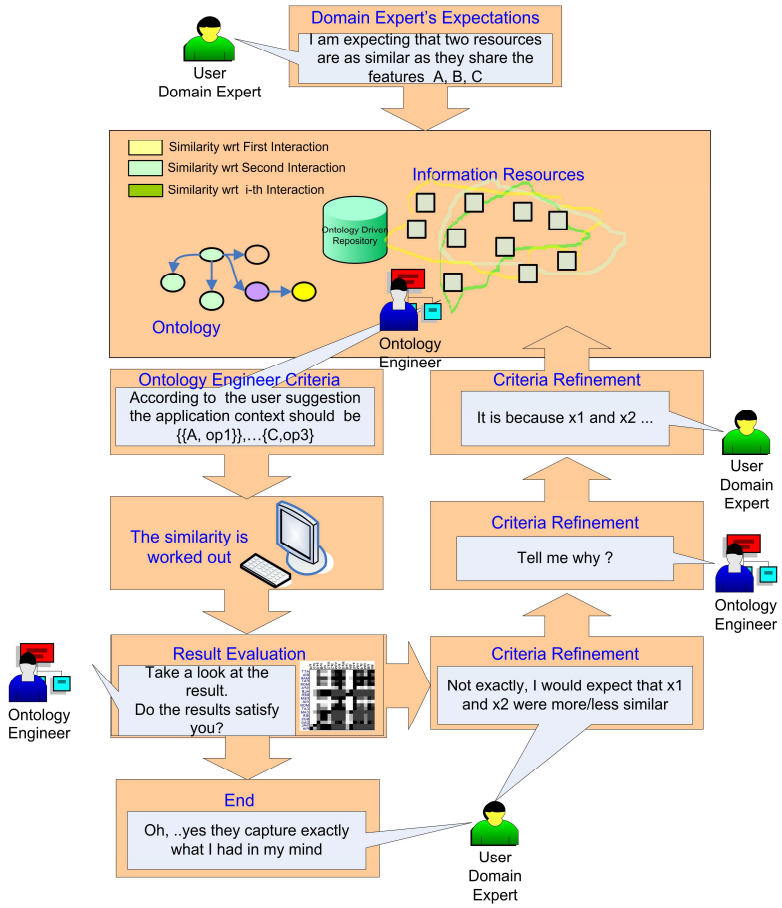

Fig. 2 Scenario 2: similarity criteria refinement.

Analysing the result, the domain expert might point out some unexpected result to the ontology engineer. Then the ontology engineer refines the similarity criteria, interacting with the domain expert, until the results are correct.

We assume that usually the user expert is so familiar with the domain conceptualized in the ontology that his expectations about similarities are often implicit. Thus, he does not provide to the ontology engineer a complete set of information concerning the criteria of similarity to be used. With this assumption the criteria definition process requires further iterative refinement.

In this scenario the framework supports the iterative criteria refinement process to precisely adapt the similarity assessments to the user expectations.

3 Preliminary assumptions

This paper proposes a semantic similarity among instances taking into account the different hints hidden in the ontology. As the hints that can be considered largely depend on the level of formality of the ontology model adopted, it is important to state clearly to which ontology model a similarity method is referring. In this paper, the ontology model with data type defined by Ehrig et al. [7] is considered.

Definition 1: Ontology with data type *An Ontology with data type is a structure* $O := (C,T,\leq_c,R,A,\sigma_R,\sigma_A,\leq_R,\leq_A,I,V,l_C,l_T,l_R,l_A)$ *where* C,T,R,A,I,V *are disjoint sets, respectively, of classes, data types, binary relations, attributes, instances and data values, and the relations and functions are defined as follows:*

$\leq_C$	the partial order on C, which defines the classes hierarchy,
$\leq_R$	the partial order on R which defines the relation hierarchy,
$\leq_A$	the partial order on A which defines the attribute hierarchy,
$\sigma_R : R \to C \times C$	the function that provides the signature for each relation,
$\sigma_A : A \to C \times T$	the function that provides the signature for each attribute,
$l_C : C \to 2^I$	the function called class instantiation,
$l_T : T \to 2^V$	the function called data type instantiation,
$l_R : R \to 2^{I \times I}$	the function called relation instantiation,
$l_A : A \to 2^{I \times V}$	the function called attribute instantiation.

A symmetric normalized similarity is a function $S : IxI \to [0,1]$, which satisfies the following axioms:

$\forall x, y \in I \quad S(x,y) \geq 0$ *Positiveness*

$\forall x \in I, \forall y,z \in I, S(x,x) \geq S(y,z)$ *Maximality*

$\forall x, y \in I \quad S(x,y) = S(y,x)$ *Symmetry*

An asymmetric normalized similarity is a function $\overline{S} : IxI \to [0,1]$ that does not satisfy the symmetric axioms. The preference between symmetric and asymmetric similarity mainly depends on the application scenario; in general, there is no a-priori reason to formulate this choice. A complete framework for assessing the semantic similarity should be provided by both of them.

The preference between symmetric and asymmetric similarity mainly depends on the application scenario; often the symmetric similarity is preferred because it is mathematically closer to the inverse of distance measure than the asymmetric one. However, according to the assumption of Tvesky, often a non-prominent item is more similar to a prominent item than vice versa [14]. In this paper we chose to propose an asymmetric similarity because we think it is more informative. This informativeness is useful for example in application such as the browsing of information resources.
During the browsing, we need to identify similar resources that are representative of a searched resource and that can be used to replace it. For instance if we consider as

information resources the members of a research staff, and we suppose to search for a member with a specific scientific expertise, usually a *PhD student* can be replaced by his *PhD advisor*, because the experience of a *PhD stude*nt is usually contained in the expertise of his *PhD advisor* but the vice versa is not true. As a consequence the similarity between the *PhD student* and his *PhD advisor* is greater than the similarity between the *PhD advisor* and his *PhD student*. The symmetric similarity is not suitable to support this characteristic of containment.
Then a representative resource is the resource that includes others. A similar approach has been proposed in [15] for the retrieval of documents. We stress the relation of containment between the sets of characteristics of two information resources. The information resources are characterized by ontology driven metadata; therefore, each resource is assumed to be an instance and the similarity is defined among pairs of instances.

Definition 2: Containment between two information resources/instances. *Given two information resources x, y (represented as instances in the ontology) and their sets of characteristics (coded as instance attributes and relation values), x is contained in y if the set of characteristics of x is contained*[1] *in the set of characteristics of y.*

We assume that instance similarity behaves coherently with the concept of containment. Given two instances x, y, their similarity is sim(x,y)=1 if and only if the set of characteristics of x is contained in the set of characteristics of y. On the contrary, unless y is contained in x, the similarity between y and x is sim(y,x)<1. The similarity value between x and y tends to decrease as long as the level of containment of their sets of properties decreases. Of course, the containment has to consider also the inheritance between the classes: if x belongs to a sub-class of the class of y, the asymmetric evaluation is performed relying on the idea that humans perceive similarity between a sub-concept and its super-concept as greater than the similarity between the super-concept and the sub-concept [16].

4 Semantic similarity approach

The proposed approach adopts the schematization of the similarity framework defined by Ehrig et.al. [7]: the similarity is structured in terms of *data*, *ontology* and *context* layers plus the *domain knowledge* layer, which spans all the others. The *data layer* measures the similarity of entities by considering the data values of simple or complex data types such as integers and strings. The *ontology layer* considers the similarities induced by the ontology entities and the way they are related to each other. The *context layer* assesses the similarity according to how the entities of the ontology are used in some external contexts. The framework defined by Ehrig et al. is suitable for supporting the ontology similarity as well as instances similarity.

[1] The containment is not meant as proper containment. In other words each set A is considered as an A subset.

Our contribution with respect to the framework defined by Ehrig et al. is mainly in the definition of a *context layer* including an accurate formalization of the criteria in order to tailor the similarity with respect to a context and in the definition of an *ontology layer* explicitly parameterized according to these criteria. Concerning the data and domain knowledge layers, this paper adopts a replica of what is illustrated in [7]. The formalization of the criteria of similarity induced by the context is employed to parameterize the computation of the similarity in the *ontology layer*, forcing it to adhere to the application criteria.

The overall similarity is defined by the following amalgamation function ($\overline{Sim}$), which aggregates two similarity functions defined in the ontology layer named *external similarity* ($\overline{ExternSim}$) and *extensional similarity* ($\overline{ExtensSim}$). The external similarity performs a structural comparison between two instances $i_1 \in l_c(c_1)$, $i_2 \in l_c(c_2)$ in terms of the classes c_1, c_2 that the instances belong to, whereas the extensional similarity performs a comparison of the instances in terms of their attributes and relations.

$$\overline{Sim}(i_1, i_2) = \frac{w_{ExternSim} * \overline{ExternSim}(i_1, i_2) + w_{ExtensSim} * \overline{ExtensSim}(i_1, i_2)}{w_{ExternSim} + w_{ExtensSim}} \quad \textbf{(1)}$$

$w_{ExternSim}$ and $w_{ExtensSim}$ are the weights used to balance the importance of the functions. By default they are equal to 1/2.

In the section, we have illustrated a full description of the approach. In the next, the approach is detailed in three sections. In particular our definition of context layer is described in detail as well as the ontology layer where the two similarities $\overline{ExternSim}$ and $\overline{ExtensSim}$ are designed, while the description of data and knowledge layer is shortly provided.

5 Context layer

The context layer, according to Ehrig at al. [7], describes how the ontology entities concur in different contexts. Here we adopt the same point of view. However, we aim to formalize the application context in the sense of modelling the criteria of similarity induced by the context. This design choice does not hamper the eventual definition of a generic description of context followed by an automatic determination of which criteria would have been suitable for a given context. Rather, it allows us to calculate directly the similarity acting on the criteria, especially when it is necessary to refine them. In the following we underscore the importance of this formalization.

5.1 Motivation behind the application context formalization

The application context provides the knowledge for formalizing the criteria of similarity induced by the application. The criteria are context-dependent as the

context influences the choice of classes, attributes and relations that are considered in the similarity assessment and the operations used to compare them.

We describe the motivation behind the proposed formalization through an example based on the domain of academic research, considering as resources to be compared the researchers of a research institution. We chose this domain instead of a more specific area related to our research experience in the AIM@SHAPE project (such as solid modelling, 3D model reconstruction, virtual humans, etc.) as it is without doubt a more familiar field to the readers of this paper. Let us consider a simplified version of the ontology KA[2] that defines concepts from academic research (Fig. 3) and focus on the two applications "comparison of the members of the research staff according to their working experience" and "comparison of the members of the research staff with respect to their research interest".

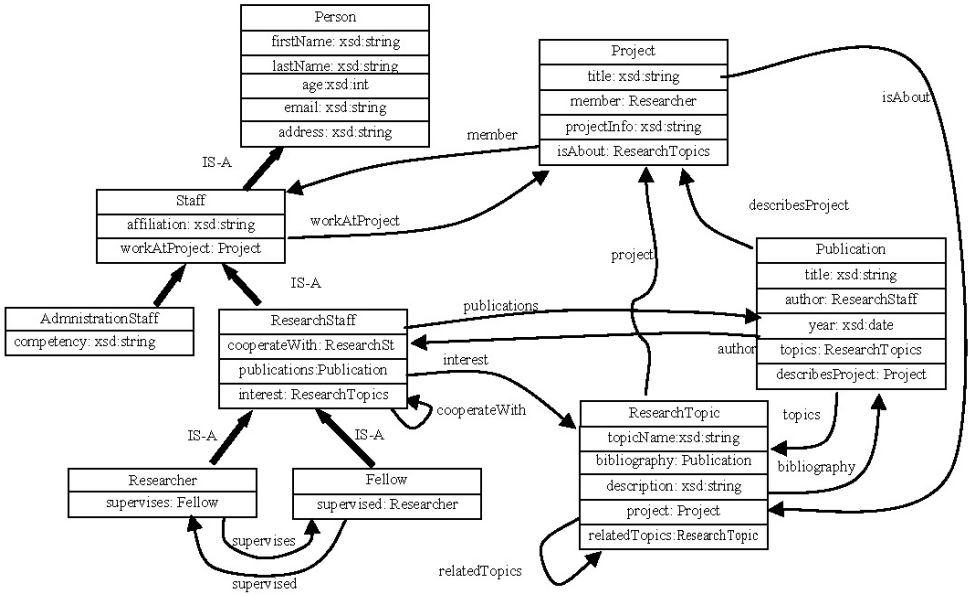

Fig. 3 Ontology defining concepts related to academic research.

Two distinct application contexts may be induced according the applications:

- "Exp" induced by the comparison of the members of the research staff according to their working experience. The similarity among the members of the research staff (instances of the class *ResearchStaff*[3]) is roughly assessed by considering the member's age (the attribute *age* inherited by the class *Person*) and the number of projects and publications a researcher has worked on (the number of instances reachable through the relation *publications* and the relation *workAtProject* inherited by *Staff*).

[2] http://protege.stanford.edu/plugins/owl/owl-library/ka.owl

[3] The italics is used to explicit the reference to the entities (attributes, relations, classes) of the ontology in Fig 1.

- "Int" induced by the comparison of the members of the research staff with respect to their research interests. The researchers can be compared with respect to their interests (instances reachable through the relation *interest*) and, again, their publications (instances reachable through the relation *publications* and the relation *workAtProject*).

The following points need to be considered when analysing these examples:

1. The similarity between two instances can depend on the comparison of their related instances: the researchers are compared with respect to the instances of the class *Publication* connected through the relation *publications.*
2. The attributes and relations of the instances can contribute differently to the evaluation according to the context: the attribute *age* of the researchers is functional in the first application but it might not be interesting in the second; the relations *publication* and *workAtProject* are included in both application contexts but using different operators of comparison—in the first case just the number of instances is important whereas in the latter case the related instances have to be compared.
3. The ontology entities can be considered recursively in the similarity evaluation: in the context "Int" the members' research topic (instances of *ResearchTopic* reachable navigating through the relation *ResearchStaff->interest*[4]*)* are considered and their related topics (instances of *ResearchTopic* reachable via *ResearchStaff->interest->relatedTopic*) are recursively compared to assess the similarity of distinct topics.
4. The classes' attributes and relations can contribute differently to the evaluation according to the recursion level of the assessment: in the second application the attribute *topicName* and the relation *relatedTopic* can be considered at the first level of recursion to assess the similarity between *researchTopic*. By navigating the relation *relatedTopic* it is possible to apply another step of recursion, and here the similarity criteria can be different from the previous ones. For example, in order to limit the computational cost and stop the recursion, only the *topicName* or the instances identifier could be used to compare the *relatedTopic*.

As pointed out in the second remark, different operations can be used to compare the ontology entities:

- Operation based on the "cardinality" of the attributes or relations: the similarity is assessed according to the number of instances the relations have or the number of values that an attribute assumes. For example, in the first context **"**Exp", two researchers are similar if they have a similar "number" of publications.
- Operation based on the "intersection" between sets of attributes or relations: the similarity is assessed according to the number of elements they have in common. For example, in the context "Int", the more papers two researchers share, the more their interests are similar.

Operation based on the "similarity" of attributes and relations: the similarity is assessed in terms of the similarity of the attribute values and related instances. For

[4] The arrow is used to indicate the navigation through a relation, for example *A->B->C* means that starting from the class *A* we navigate through the relations B and C.

example, in the context **"**Int**",** two researchers are similar if they have "similar" research topics.
The example shows that an accurate formalism is needed to properly express the criteria that might arise from different application contexts. The formalization has to model the attributes and relations as well as the operations to compare their values. Moreover, as stated in the fourth remark, the level of recursion of the similarity assessment also has to be considered.

5.2 Application context formalization

The formalization provided here represents the restrictions that the application context must adhere to. An ontology engineer is expected to provide the application context according to specific application needs. The formalization relies on the concepts of a "sequence of elements belonging to a set X", which formalizes generic sequences of elements, and a "path of recursion of length i" to track the recursion during the similarity assessment. In particular, a "path of recursion" represents the recursion in terms of the sequence of relations used to navigate the ontology.

The application context function (*AC*) is defined inductively according to the length of the path of recursion. It yields the set of attributes and relations as well as the operations to be used in the similarity assessment. The operations considered are those described in the previous section and named, respectively, *Count* to evaluate the cardinality, *Inter* to evaluate the intersection, and *Simil* to evaluate the similarity.

Definition 3: Sequences of a set X *Given a set X, a sequence s of elements of X with length n is defined by the function* $s:[1,..,n]\to X, n\in N^+$ *and represented in a simple way by the list [s(1),..,s(n)].*

Let $S_X^n=\{s \mid s:[1,n]\to X\}$ be the set of sequences of *X* having length *n*.
Let $\cdot: S_X^n x S_Y^m \to S_{X\cup Y}^{n+m}$ be the operator "concat" between two sequences.

Table **1** defines the polymorphism functions, which identify specific sets of entities in the ontology model.

Table 1. List of functions defining specific sets of elements in the ontology model.

$\delta_a: C\to 2^A; \delta_a(c)=\{a: A \mid \exists\, t\in T, \sigma_A(a)=(c,t)\}$	set of attributes of $c\in$ C,
$\delta_a: R\to 2^A,$ $\delta_a(r)=\{a: A \mid \exists c,c'\in C\ \exists\, t\in T\ \sigma_R(r)=(c,c')\wedge\sigma_A(a)=(c',t)\}$	set of attributes of the classes which are reachable through the relation $r\in$ R,
$\delta_r: C\to 2^R; \delta_r(c)=\{r: R \mid \exists\, c'\in C, \sigma_R(r)=(c,c')\}$	set of relations of $c\in$ C,
$\delta_c: R\to 2^C; \delta_c(r)=\{c': C \mid \exists\, c\in C\ \ \sigma_R(r)=(c,c')\}$	set of concepts reachable through $r\in$ R,
$\delta_r: R\to 2^R;$ $\delta_r(r)=\{r': R \mid \exists\, c\in C, \exists\, c'\in\delta_C(r);\ \wedge\sigma_R(r')=(c',c)\}$	set of relations of the concepts reachable through r,
$\delta_c: C\to 2^C; \delta_c(c)=\{c': C \mid \exists\, r\in\delta_r(c); \sigma_R(r)=(c,c')\}$	set of concepts related to $c\in$ C through a relation.

Definition 4: Path of recursion *A path of recursion p with length i is a sequence whose first element is a class and whose other elements are relations recursively reachable from the class:* $p \in S^{i}_{C \cup R} \mid p(1) \in C \wedge \forall j \in [2,i]\ \ p(j) \in R \wedge p(j) \in \delta_r(p(j-1))$.

For example, a path of recursion with length longer than three is a path that starts from a class p(1) and continues to one of its relations as the second element p(2) and then to one of the relations of the class reachable from p(2) as the third element p(3), and so on. In general, a path of recursion p represents a path that is followed to assess the similarity recursively. The recursion expressed in the previous section in the context "Int" as *ResearchStaff->interest->relatedTopic* is formalized with the path of recursion [ResearchStaff, interest, relatedTopic].

Let P^i be the set of all paths of recursion with length i and P be the set of all paths of recursion $P= \cup_{i \in N} P^i$.

Definition 5: Application context AC *Given the set P of paths of recursion,* $L = \{Count, Inter, Simil\}$ *, the set of operations adopted as an application context is defined by a partial function AC having the signature* $AC: P \rightarrow (2^{A \times L}) \times (2^{R \times L})$ *, yielding the attributes and relations as well as the operations to perform their comparison.*

In particular, each application context AC is characterized by two operators $AC_A : P \rightarrow 2^{A \times L}$ and $AC_R : P \rightarrow 2^{R \times L}$, which yield, respectively, the parts of the context AC related to the attributes and the relations. Formally $\forall p \in P\ AC(p) = (AC_A(p),\ AC_R(p))$ and $AC_A(p)$ and $AC_R(p)$ are set of pairs $\{(e_1,o_1), (e_2,o_2),\ldots, (e_i,o_i),\ldots,(e_n,o_n)\}$ $n \in N$ where e_i is, respectively, the attribute or the relation relevant to define the similarity criteria and $o_i \in L$ is the operation to be used in the comparison.
We provide two examples of AC formalization referring to the two application contexts "Exp" and "Int" mentioned in the previous section.

Example 1. Let us formalize the application context "Exp" with AC_{Exp} to assess the similarity among the members of a research staff according to their experience. We consider the set of paths of recursion {[ResearchStaff], [Research], [Fellow]} and we compare them according to age similarity and the numbers of publications and projects. Thus AC_{Exp} is defined by:

$$[\text{ResearchStaff}] \xrightarrow{AC_{Exp}} \{\{(\text{age},\text{Simil})\},\{(\text{publications},\text{Count}),(\text{workAtProject},\text{Count})\}\} \quad \textbf{(2)}$$

$$[\text{Researcher}] \xrightarrow{AC_{Exp}} \{\{(\text{age},\text{Simil})\},\{(\text{publications},\text{Count}),(\text{workAtProject},\text{Count})\}\}$$

$$[\text{Fellow}] \xrightarrow{AC_{Exp}} \{\{(\text{age},\text{Simil})\},\{(\text{publications},\text{Count}),(\text{workAtProject},\text{Count})\}\}$$

An example of AC_R is {(publication,Count),(workAtProject,Count)} while an example of AC_A is {(age,Simil)}.

Note that [Researcher] and [Fellow] belong to the set of paths of recursion considered in AC_{Exp} because their instances are also instances of *ResearchStaff*. The application

context can be expressed in a more compact way assuming that, whenever a context is not defined for a class but is defined for its super class, the comparison criteria defined for a super class are by default inherited by the subclasses. According to this assumption AC_{Exp} can be expressed by:

$$[\text{ResearchStaff}] \xrightarrow{AC_{Exp}} \{\{\text{age}, \text{Simil}\}, \{(\text{publications}, \text{Count}), (\text{workAtProject}, \text{Count})\}\} \quad \textbf{(3)}$$

Example 2. Let us formalize the application context "Int" to assess the similarity among the members of a research staff according to their research interest. The similarity is computed considering the set of paths of recursion {[ResearchStaff],[ResearchStaff, interest]}. The researchers are compared considering common publications, common projects or similar interests. A compact formalization for "Int" is defined by AC_{Int}:

$$[\text{ResearchStaff}] \xrightarrow{AC_{Int}} \{\{\phi\}, \{(\text{publications}, \text{Inter}), (\text{workAtProject}, \text{Inter}), (\text{interest}, \text{Simil})\}\} \quad \textbf{(4)}$$
$$[\text{ResearchStaff}, \text{interest}] \xrightarrow{AC_{Int}} \{\{\text{topicName}, \text{Inter}\}, \{(\text{relatedTopics}, \text{Inter})\}\}$$

In general, the operator *Count* applied to attributes or relations means that the number of attribute values or related instances is considered in the similarity assessment. For example, according to the context formalized in equation 2 (second row), two researchers, who are represented as instances of *Researcher*, are similar if they have a similar numbers of instances of *Publication* reachable through the relation *publications*.

The operator *Inter* applied to attributes or relations means that common attribute values or related instances are considered in the similarity assessment. For example, according to the context formalized in equation 4 (first row) two researchers are considered as similar if they have common project instances.

When applied to an attribute, the operator *Simil* determines that the attribute values of two instances will be compared according to a datatype similarity provided by the data layer (see the example in equation 2, first row, attribute age). When it is applied to a relation, it determines a step of recursion, in the sense that the instances related through the relation have to be considered during the similarity assessment. How these related instances have to be compared is specified by the value provided by the context function for the corresponding recursion path. Note that the researchers are compared recursively in the context expressed by equation 4. In fact the relation *interest* is included with the operator *Simil* in the first row of equation 4. This means that the instances of *ResearchTopic* associated with the researcher via *interest* have to be accessed and compared recursively when the researchers' similarity is worked out. Actually, [ResearchStaff,interest] is the path of recursion to navigating the ontology from *ResearchStaff* to *ResearchTopic* via the relation *interest.* Once the assessment has accessed the related instances, it compares them as indicated by the second row of equation 4. The interests are compared with respect to both their *topicName* and their *relatedTopic*; thus, two *ResearchTopics* having distinct *topicNames* but some *relatedTopics* in common are not considered completely dissimilar.

The image of an AC function can be further characterized by the following.

1. For a path of recursion p, AC has to yield only the attributes and relations belonging to the classes reached through p. For example, considering the ontology

in Fig. 3 and the path of recursion [ResearchStaff,interest], it is expected that only the attributes and relations belonging to the class *ResearchTopic* reachable via [ResearchStaff,interest] can be identified by AC([ResearchStaff,interest]). Attributes or relations (such as *age, publications, etc),* which do not belong to *ResearchTopic,* define an incorrect application context.

2. Given a path of recursion p, an attribute or a relation can appear in the context image at most one time. In other words, given a path of recursion it is not possible to associate two distinct operations with the same relation or attribute. For example, the following application context definition is not correct as *interest* is specified twice

$$[\text{ResearchStaff}] \longrightarrow \{\{\phi\}, \{(\text{publications}, \text{Inter}), (\text{interest}, \text{Simil}), (\text{interest}, \text{Inter})\} \quad \textbf{(5)}$$

6 Ontology layer

The ontology layer defines the asymmetric similarity functions $\overline{ExternSim}$ and $\overline{ExtensSim}$ that constitute the amalgamation function (equation 1). The "external similarity" $\overline{ExternSim}$ measures the similarity at the level of the ontology schema computing a structural comparison of the instances. Given two instances, it compares the classes they belong to, considering the attributes and relations shared by the classes and their position within the class hierarchy. The "extensional similarity" $\overline{ExtensSim}$ compares the extension of the ontology entities. The similarity is assessed by computing the comparison of the attributes and relations of the instances.

At the ontology layer additional hypotheses are assumed:

- All classes defined in the ontology have the fake class *Thing* as a super-class.
- Given $i_1 \in l_c(c_1)$, $i_2 \in l_c(c_2)$, if c_1, c_2 do not have any common super-class different from *Thing,* their similarity is equal to 0.
- The least upper bound *(lub)* between c_1 and c_2, is unique and it is c_2 if c_1 IS-A c_2, or c_1 if c_2 IS-A c_1, or the immediate super-class of c_1 and c_2 that subsumes both classes.

The aim is to force the *lub* to be a sort of "template class" that can be adopted to perform the comparison of the instances whenever the instances belong to distinct classes. Referring to the ontology in Fig. 3, it can be appropriate to compare two instances belonging, respectively, to *AdministratorStaff* and *ResearchStaff* as they are both a kind of *Staff* and *Staff* is their *lub*. However, it does not make sense to evaluate the similarity between two instances belonging to *Publication* and to *Staff,* because they are intimately different; in fact, there is not any *lub* available for them. Whenever a *lub* x between two classes exists, the path of recursion [x] is the starting path in the recursive evaluation of the similarity.

6.1 External similarity

The external similarity ($\overline{ExternSim}$) performs the structural comparison between two instances i_1, i_2 in terms of the classes c_1, c_2 that the instances belong to: more formally $\overline{ExternSim}(i_1, i_2) = \overline{ExternSim}(c_1, c_2)$ where $i_1 \in l_c(c_1), i_2 \in l_c(c_2)$.

In this paper the external similarity function is defined starting from the similarities proposed by Maedche and Zacharias [17] and Rodriguez and Egenhofer [16]. The structural comparison is performed by two similarity evaluations:

- **Class Matching**, which is based on the distance between the classes c_1, c_2 and their depth with respect to the hierarchy induced by $\leq_C$.
- **Slot Matching,** which is based on the number of attributes and relations shared by the classes c_1, c_2 and the overall number of their attributes and relations. Then two classes having many attributes/relations, some of which are in common, are less similar than two classes having fewer attributes but the same number of common attributes/relations.

Both similarities are needed to successfully evaluate the similarity with respect to the ontology structure. For example, let us consider the ontology schema in Fig. 3 and let us compare an instance of the class *ResearchStaff* with an instance of the class *AdministrationStaff*.
They are quite similar with respect to Class Matching but less similar with respect to Slot Matching. In fact, the sets of IS-A relations joining the classes *ResearchStaff* and *AdministrationStaff* to *Thing* are largely shared. However, from the point of view of the slots, *ResearchStaff* and *AdministrationStaff* share only the attribute inherited and they differ with respect to the others. Likewise, it would be easy to show an example of two classes that are similar with respect to Slot Matching and less similar according to Class Matching.

Definition 6: ExternSim similarity *The similarity between two classes according to the external comparison is defined by:*

$$\overline{ExternSim}(c_1, c_2) = \begin{cases} 1 & if \ \ c_1 = c_2 \\ \dfrac{w_{SM} * \overline{SM}(c_1, c_2) + w_{CM} * \overline{CM}(c_1, c_2)}{w_{SM} + w_{CM}} & Otherwise \end{cases} \qquad \textbf{(6)}$$

where ($\overline{SM}$) is Slots Matching, ($\overline{CM}$) is Classes Matching and w_{SM}, w_{CM} are weights in the range [0,1].

For the purpose of this paper, w_{SM} and w_{CM} are defined as equal to 1/2.

6.1.1 Class Matching

Classes Matching is evaluated in terms of the distance of the classes with respect to the IS-A hierarchy. The distance is based on the concept of Upwards Cotopy (*UC*) [17]. We define an asymmetric similarity adapting the symmetric definition of CM in [17] .

Definition 7: Upward Cotopy (UC) *The Upward Cotopy of a set of classes C with the associated partial order $\leq_C$ is:*

$$UC_{\leq_C}(c_i) := \{c_j \in C \mid (c_i \leq_C c_j) \vee c_i = c_j\} \quad \textbf{(7)}$$

It is the set of classes composing the path that reaches from c_i to the furthest super-class (*Thing*) of the IS-A hierarchy: for example, considering the class Researcher in Fig. 3 $UC_{\leq_C}$(Researcher) = {Researcher, ResearchStaff, Staff, Person, Thing[5]}

Definition 8: Asymmetric Class Matching *Given two classes c_1, c_2 and the Upward Cotopy $UC_{\leq_C}(c_i)$, the asymmetric Class Matching is defined by:*

$$\overline{CM}(c_1, c_2) := \frac{\left|UC_{\leq_C}(c_1) \cap UC_{\leq_C}(c_2)\right|}{\left|UC_{\leq_C}(c_1)\right|} \quad \textbf{(8)}$$

$\overline{CM}$ between two classes depends on the number of classes they have in common in the hierarchy. Let us note that the Class Matching is asymmetric: for example, referring to Fig. 3, $\overline{CM}(\text{AdministrationStaff}, \text{Researcher}) = 3/4$ but $\overline{CM}(\text{Researcher}, \text{AdministrationStaff}) = 3/5$. Moreover it is important to note that $\overline{CM}(\text{Staff}, \text{Researcher}) = 1$. The rationale behind this choice of design pertains to the property of containment between instances: the instances of *Researcher* fit with the instances of *Staff*, and they can replace the instances of *Staff* at the class level.

6.1.2 Slot Matching

Slot Matching is defined by the slots (attributes and relations) shared by the two classes. We refer to the similarity proposed by Rodriguez and Egenhofer [16], based on the concept of distinguishing features employed to differentiate subclasses from their super-class. In their proposal, different kinds of distinguishing features are considered (i.e. functionalities and parts) but none coincides immediately with the native entities in our ontology model. Of course it would be possible to manually annotate the classes, adding the distinguishing features, but we prefer to focus on what is already available in the adopted ontology model. Therefore only attributes and relations are mapped as two kinds of distinguishing features.

Definition 9: Slot Matching *Given two classes c_1,c_2, two kinds of distinguishing features (attributes and relations), and w_a, w_r, the weights of the features, the similarity function $\overline{SM}$ between c_1 and c_2 is defined in terms of the weighted sum of the similarities $\overline{S}_a$ and $\overline{S}_r$, where $\overline{S}_a$ is the Slot Matching according to the attributes and $\overline{S}_r$ in the Slot Matching according to the relations.*

$$\overline{SM}(c_1, c_2) = \omega_a \cdot \overline{S}_a(c_1, c_2) + \omega_r \cdot \overline{S}_r(c_1, c_2) \quad \textbf{(9)}$$

[5] The class Thing is not explicitly included in the **Fig. 3** but it is expected to be the super class of all the other classes, so it can be seen as superclass of Person, Project, Publication, ResearchTopic.

The sum of the weights is expected to be equal to 1, and by default we assume $w_a=w_r=1/2$. The two Slot Matching similarities $\overline{S}_a$ and $\overline{S}_r$ rely on the definitions of slot importance as defined in the following.

Definition 10: Function of "slot importance" α *Let c_1, c_2, be two distinct classes and d be the class distance $d(c_1,c_2)$ in terms of the number of edges in an IS-A hierarchy, then α is the function that evaluates the importance of the difference between the two classes.*

$$\alpha(c_1,c_2)=\begin{cases}\dfrac{d(c_1,\text{lub}(c_1,c_2))}{d(c_1,c_2)} & d(c_1,\text{lub}(c_1,c_2))\le d(c_2,\text{lub}(c_1,c_2))\\ 1-\dfrac{d(c_1,\text{lub}(c_1,c_2))}{d(c_1,c_2)} & d(c_1,\text{lub}(c_1,c_2))> d(c_2,\text{lub}(c_1,c_2))\end{cases} \quad \textbf{(10)}$$

where $d(c_1,c_2)=d(c_1,\text{lub}(c_1,c_2))+d(c_2,\text{lub}(c_1,c_2))$.

$\alpha(c_1, c_2)$ is a value in the range [0,0.5]. Referring to the image in Fig. 3, α(Researcher,ResearchStaff) is equal to zero because the *lub* between *Researcher* and *Researcher* is *Researcher* itself, d(ResearchStaff,Researcher)=1 and d(Researcher,Researcher)=0. Whereas α(Researcher,Fellow) is equal to 0.5 because the *lub* is still *Researcher*, and d(Researcher,Fellow)=2.

Definition 11: Slot Matching according to the kind of distinguishing feature t *Given two classes c_1 (target) and c_2 (base) and t, a kind of distinguishing feature (t=a for attributes or t=r for relations), let C_1^t and C_2^t be the sets of distinguishing features of type t, respectively, of c_1 and c_2; then Slot Matching $\overline{S}_t(c_1,c_2)$ is defined by:*[6]

$$\overline{S}_t(c_1,c_2)=\frac{\left|C_1^t\cap C_2^t\right|}{\left|C_1^t\cap C_2^t\right|+(1-\alpha(c_1,c_2))\left|C_1^t\setminus C_2^t\right|+\alpha(c_1,c_2)\left|C_2^t\setminus C_1^t\right|} \quad \textbf{(11)}$$

According to the ontology in Fig. 3, considering the classes *Researcher* and *Fellow*, their sets of distinguishing features of type relation are Researcher^r ={workAtProject, cooperateWith, pubblications, interest, supervises} and Fellow^r={workAtProject, cooperateWith, pubblications, interest, supervised} and α(Fellow,Researcher)=0.5; then $\overline{S}_r(\text{Fellow},\text{Researcher})$ = 4/5. Furthermore, this formulation of Class Matching is coherent to the containment property: considering the classes *Staff* and *Fellow*, their sets of distinguishing features of type relation are respectively Staff^r={workAtProject}, Fellow^r={workAtProject, cooperateWith, publications, interest, supervised} and α(Staff,Fellow)=0, so that $\overline{S}_r(\text{Staff},\text{Fellow})$ =1. This means that the instances of *Fellow* can replace the instances of *Staff* because they have some

[6] This formulation is slightly different from that provided by Egenhofer and Rodriguez: the parameters of the similarity have been reversed to be coherent with the relation between instances containment and the similarity value equal to 1.

quality more rather than less similar. The contrary is not true; in fact α(Fellow,Staff)=0 and $\overline{S}_r(\text{Fellow},\text{Staff})$ =1/5. In general, whenever α=0.5 the differences between features of both classes are equally important for the matching: for example, this happens when the classes are sisters, as for *Researcher* and *Fellow*. In the case of α=0, only the features that are in c_1 and not in c_2 are important for the matching.

6.2 Extensional similarity

The extension of entities plays a fundamental role in the assessment of the similarity among the instances: it is needed to perform a comparison of the attribute and relation values.
The extensional comparison is characterized by two similarities functions: a function based on the comparison of the attributes of the instances and a function based on the comparison of the relations of the instances.

Definition 12: Extensional asymmetric similarity *Given two instances $i_1 \in l_c(c_1)$, $i_2 \in l_c(c_2)$, $c=lub(c_1,c_2)$ and $p=[c]$, a path of recursion defined in the application context AC,*[7] *let $\overline{Sim}_a^p(i_1,i_2)$ and $\overline{Sim}_r^p(i_1,i_2)$ be the similarity measurements between instances considering, respectively, their attributes and their relations. The extensional similarity with asymmetric property is defined by*

$$\overline{ExtensSim}(i_1,i_2)=\begin{cases}1 & i_1=i_2\\ \overline{Sim}_I^p(i_1,i_2) & Otherwise\end{cases} \qquad \textbf{(12)}$$

where $\overline{Sim}_I^p(i_1,i_2)$ is defined by

$$\overline{Sim}_I^p(i_1,i_2)=\frac{\sum_{a\in\delta_a(c)}\overline{Sim}_a^p(i_1,i_2)+\sum_{r\in\delta_r(c)}\overline{Sim}_r^p(i_1,i_2)}{|AC_A(p)|+|AC_R(p)|} \qquad \textbf{(13)}$$

A first principle of the proposed extensional similarity between two instances is to consider the *lub* x of their classes as the common base for comparing them when the instances belong to different classes. Note that the index p, is a kind of stack of recursion adopted to track the navigation of relations whenever the similarity among instances is recursively defined in terms of the related instances. [x] is adopted to initialize p at the beginning of the assessment.
$\overline{Sim}_a^p(i_1,i_2)$ and $\overline{Sim}_r^p(i_1,i_2)$ are defined by a unique equation as follows.

Definition 13: Similarity on attributes and relations *Given two instances $i_1 \in l_c(c_1)$, $i_2 \in l_c(c_2)$, $c=lub(c_1,c_2)$, $p=[c]$ (a path of recursion), X (a placeholder for the "A" or*

[7] Note that $|AC_A(p)|+|AC_R(p)|\neq 0$ each time the context AC specifies at least a relevant attribute or relation for the recursion path p.

"R", $x \in A \cup R$), *then let*

- $i_A(i) = \{v \in V \mid (i,v) \in l_A(a), \exists y \in C \; s.t. \; \sigma_A(a) = (y,T) \wedge l_T(T) = 2^V\}$, *the set of values assumed by the instance i for the attribute a,*
- $i_R(i) = \{i' \in l_c(c') \mid \exists c \; i \in l_c(c) \exists c' \; s.t. \; \sigma_R(r) \in (c,c') \wedge (i,i') \in l_R(r)\}$, *the set of instances related to the instance i by the relation r,*
- *AC be the application context defined according to the restrictions defined in paragraph 5.2*
- $F_X = \{g : i_X(i_1) \to i_X(i_2) \mid g \text{ is partial and bijective}\}$.

The similarity between instances according to their attributes or relations is:

$$\overline{\mathrm{Sim}}_x^p(i_1,i_2) = \begin{cases} 1 & \text{if } (i_X(i_1) \text{ are empty sets}) \\ 0 & \text{if } (i_X(i_1) \neq \phi \wedge i_X(i_2) = \phi) \\ \dfrac{|i_X(i_2)|}{\max(|i_X(i_1)|,|i_X(i_2)|)} & \text{if } (x, Count) \in AC_X(p) \\ \dfrac{|i_X(i_1) \cap i_X(i_2)|}{|i_X(i_1)|} & \text{if } (x, Inter) \in AC_X(p) \\ \dfrac{\max\limits_{f \in F_A} \sum\limits_{v \in i_A(i_1)} \overline{Sim}_T^a(v, f(v))}{\min(|i_A(i_1)|,|i_A(i_2)|)} * (1 - \max(0, \dfrac{|i_A(i_1)| - |i_A(i_2)|}{|i_A(i_1)|})) & \text{if } (x = a) \wedge (a, Simil) \in AC_A(p) \\ \dfrac{\max\limits_{f \in F_R} \sum\limits_{i \in i_R(i_1);} \overline{Sim}_I^{pNew}(i, f(i))}{\min(|i_R(i_1)|,|i_R(i_2)|)} * (1 - \max(0, \dfrac{|i_R(i_1)| - |i_R(i_2)|}{|i_R(i_1)|})) & \text{if } (x = r) \wedge (r, Simil) \in AC_R(p) \\ & pNew = p \cdot s, \; s \in S_R^1, s(1) = r \end{cases}$$

These equations are designed to be asymmetric and to respect the properties of containment among instances: if an instance i_2 has at least the same attribute and relation values as i_1, then the extensional similarity between i_1 and i_2 is equal to one.

The approach computes $\overline{\mathrm{Sim}}_x^p$, selecting one of the above equations according to the definition of *AC:*

- In the first case, the similarity is 1 if the set of the property values of the first instance is empty, because an instance having no characteristics is contained in all the other instances.
- In the second case, the similarity is 0 if the first instances having at least a property value are compared with an instance that does not have any value.
- The third expression is adopted if AC yields a relation or attribute associated with the operation *Count*.
- The fourth expression is adopted if AC yields a relation or attribute associated with the operation *Inter*.

- The fifth expression is adopted if AC yields an attribute with the operation *Simil*.
- The last expression is adopted if AC yields a relation with the operation *Simil*. It is important to note that each time the similarity is assessed in terms of related instances (whenever $(r,Simil) \in AC_R(p)$), the relation r followed to reach the related instances is added to the path of recursion. Thus, during the recursive assessment, the AC is always worked out on the most updated path of recursion.

In the last two expressions, the comparison of the attribute values relies on the function $\overline{Sim}_T^a$, which defines the similarity for the values of the attribute a having data type T. $\overline{Sim}_T^a$ is provided by the data layer as suggested by [7] and briefly discussed in the next paragraph.

Example of extensional similarity according with the definition 12.
We refer to the ontology in Fig. 3. We consider two instances illustrated in Table 2: AB and RA respectively of the classes *Researcher* and *Fellow* and their instances related to the classes *Publication, Project, ResearchTopic.* We adopt the application context AC_{int} (equation 4). We evaluate their similarity applying the equation 13.

Table 2 Example of instances of the academic research ontology.

Instance ID	Instance class	Publication Instance	Project Instance	ResearchTopic Instance
AB	Researcher	P2	Pr1, Pr2	T1, T2
RA	Fellow	P2, P1	Pr1	T3

Table 3: Details of ResearchTopic instances.

Instances ID	Instance class	topicName attribute	RelatedTopic instance
T1	ResearchTopic	Topic 1	
T2	ResearchTopic	Topic 2	T4
T3	ResearchTopic	Topic 3	T4
T4	ResearchTopic	Topic 4	

Their *lub* is the class *ResearchStaff* then p=[ResearchStaff] and according to the context defined in equation 4 the similarity assessment is performed considering the relations *publication*, *workAtProject* and *interest*, respectively using the operations *Inter*, *Inter* and *Simil*. Therefore, the equation 13 is an average among the three addends calculated with the formula in definition 13:

$$\overline{Sim}^{[\text{ResearchStaff}]}_{\text{publication}}(\text{AB},\text{RA}) = 1,\ \overline{Sim}^{[\text{ResearchStaff}]}_{\text{workAtProject}}(\text{AB},\text{RA}) = 1/2,\ \overline{Sim}^{[\text{ResearchStaff}]}_{\text{Interest}}(\text{AB},\text{RA}) = 1/4$$

The first two is calculated applying the fourth expression.
The last is calculated with the sixth expression in definition 13. It requires a more detailed argumentation.
The set of partial functions in F_X in definition 13 is employed to represent the possible matching among the set of values when the instances have relations or attributes with multiple values. In the example depicted in Table 2, the instances AB and RA are respectively related via the relation *interest* to T1, T2 and T3, then x is equal to "interest" and $i_R(AB) = \{T1,T2\}$ and $i_R(RA) = \{T3\}$. When AB and RA are compared, two possible partial and bijective functions f_1 and f_2 can be considered between the instances related to AB and RA: f_1:T1→T3 and f_2:T2→T3. The max operator selects the function which provides the matching with the highest

contribution: in the example, it is f_2. Thus the sum has only one addend: $\overline{Sim}_I^{pNew}(\text{T2}, f_2(\text{T2}))$ which leads to the recursive call of the similarity assessment.

The difference in number of attributes values or related instances affects the similarity evaluation as modelled in the multiplying factors in the fifth and sixth expression of definition 13:

$$(1-\max(0,\frac{|i_A(i_1)|-|i_A(i_2)|}{|i_A(i_1)|}) \text{ and } (1-\max(0,\frac{|i_R(i_1)|-|i_R(i_2)|}{|i_R(i_1)|}) .$$

These factors yield 1 if i_1 is contained in i_2; otherwise they yield the ratio between the number of properties of i_1 and the number of properties of i_2. In the example of AB and RA, looking at the Table 3, T1 and T2 are the instances of *ResearchTopic* related to AB, T3 is the instance related to RA. In this case the second factor induces a multiplying factor equal to 1/2 because half of the instances related to AB are leaved out from the matching.

The functions $\overline{Sim}_I^{pNew}(\text{T2}, f_2(\text{T2}))$ is applied to assess the similarity between AB and RA recursively with respect to the class *ResearchTopic* which are their interest. During the recursion the sixth expression in definition 13 is applied: [ResearchStaff,interest] is a new path of recursion and assigned to pNew.
Applying the application context to the new path of recursion, new criteria are listed. In particular, according to equation 4 the instances of *ResearchTopic* related to AB and RA are compared according to the values assumed by their attribute *topicName* and relation *relatedTopics*.
The similarity between T2 and T3 with respect to *topicName* is equal to 0, whereas with respect to *relatedTopic* is 1, then $\overline{Sim}_I^{pNew}(\text{T2}, f_2(\text{T2}))=1/2$. It is multiplied for the aforementioned multiplying factor thus $\overline{Sim}_{\text{Interest}}^{[\text{ResearchStaff}]}(\text{AB,RA})=1/4$.
The overall similarity is $\overline{Sim}_I^{p}(\text{AB}, RA)= 7/12$.

7 Data layer and knowledge layer

Data layer assesses the similarity of entities by considering the data values of simple types such as integers and strings or more complex data types such as geographical reference and shapes descriptors. The knowledge layer represents special shared ontology domains, which have their own additional vocabulary. As it can be placed at any level of the ontological complexity, it spans all the other layers.
In this paper, we adopt the data layer proposed by [7]. It relies on the distance measure proposed in [18] to assess the similarity between misspelled terms (e.g. Alignment and Allignment). Moreover, in real world data values are often affected by inconsistencies: for example there are data values that differ in representation of entity abbreviation (e.g. Genova, GE, GOA are terms referring to the same city, or IMATI-CNR-GE, IMATI-GE, GE-IMATI are terms referring to the same research institute). Contrary to the similarity assessment among misspelled terms, the management of inconsistence of data values requires a full-matching among the terms

in order to obtain a satisfactory evaluation of their similarity. The aspect of different representations of abbreviation can be addressed relying on both the data layer and the knowledge layer. The knowledge layer contains explicitly information about the relation of equivalence among terms used in a specific knowledge domain. The data layer can exploit such information to evaluate the similarity among terms. The lexical similarity introduced by [18] is applied only if the terms are defined not equivalent in the knowledge layer.

8 Experiments and evaluations

We evaluated our approach for the similarity assessment among the members of the research staff working at the Institute (CNR-IMATI-GE). An experiment was performed to demonstrate both the need for the content-dependent similarity and the importance of defining an asymmetric similarity based on the containment to select similar resources.

8.1 Experiments

Two experiments were performed considering the contexts "Exp"and "Int**"** mentioned in section 4.1. Eighteen members of the research staff were considered. The information related to their projects, journal publications and research interests was inserted as instances in the ontology depicted in Fig. 3 according to what was published at the IMATI web site.[8] The ontology was expressed in OWL ensuring that only the language constructs consistent with the ontology model considered in definition 1 were adopted. The resulting ontology is available at the web site [19].
Our method was implemented in JAVA and tested on this ontology.

Using the formalization of the two application contexts AC_{Int} and AC_{Exp} previously defined [equations (3), (4)], we have computed the similarity through the proposed framework. The results are represented by the similarity matrices in Fig. 4: (a) is the result related to the context "Exp**"** and (b) is the result related to the context "Int". Each column j and each row i of the matrix represents a member of the research staff (identified by the first three letters of his name). The grey level of the pixel (i,j) represents the similarity value (Sim(i,j)) between the two members located at row i and column j: the darker the colour, the more similar are the two researchers.
Analysing the similarity matrices we can make the following statements.

- It is easy to see that they are asymmetric: for example sim(Dag,Bia)=1 while sim(Bia, Dag)<1. This confirms that the proposed model assesses an asymmetric similarity. The asymmetry result is particularly useful for comparing researchers because it behaves according to the property of containment defined in Definition 2**.** For example, the two results sim(Dag,Bia)=1 and sim(Bia, Dag)<1 in Fig. 4.a mean that if Bia has at least the experience of Dag, then Dag can replace Bia. The inverse is not true, and if the domain expert decides to choose Dag instead of Bia,

[8] http://www.ge.imati.cnr.it, accessed the 12/05/2006

the similarity value provides a hint about the loss inherent in this choice [for example, if sim(Bia, Dag)=0.85, then the loss is 15%].

- The comparison of the two matrices shows how they are different; it is evident that the two contexts induce completely different similarity values. For example, "Dag" results are very similar to "Bia" with respect to their experience (black pixel in Fig. 4.a), but they are not similar with respect to their research interests (white pixel in Fig. 4.b).
- During the test process we realized that the approach provides a sort of tool for context tuning, supporting us in the decision-making process to formulate the similarity criteria. From the similarity results we were able to learn and refine our criteria to obtain the expected results.

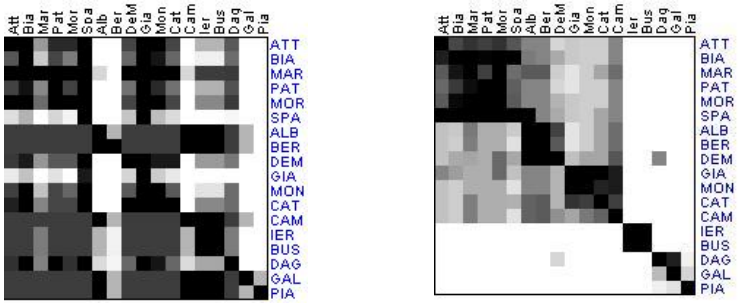

Fig. 4. (a) Similarity matrix for context "Exp"; (b) Similarity matrix for context "Int".

8.2 Evaluations

Two kinds of evaluations of the results concerning the similarity obtained with respect to research interests (Fig. 4.b) were performed.

The first evaluation was based on the concept of recall and precision, calculated considering the same adaptation of recall and precision made by [20]. More precisely, considering an entity x, the recall and precision were defined, respectively, as $(A \cap B)/A$ and $(A \cap B)/B$, where A is the set of entities expected to be similar to x and B is the set of similar entities calculated by a model. A critical issue in the similarity evaluation is to have a ground truth with respect to comparing the results obtained. We faced this problem in referring to the research staff of our institute when considering as "similar" two members of the same research group. In fact at IMATI researchers and fellows are grouped into three main research groups, and one of those is composed of three further sub-groups. Therefore, we considered the research staff as split into five groups. For each member i, A is the set of members of his research group while B is composed of the first n members retrieved by the model. We have calculated recall and precision for each group considering "n" as the smallest number of members needed to obtain a recall of 100%, and then we have evaluated the

precision. The average recall was estimated to be equal to 100% with a precision of 95%. These results are quite encouraging: a recall equal to 100% demonstrates that, for each research group, the similarity is able to rank all the expected members, while a precision equal to 95% means that the average number of outsiders that need to be included to rank all group members is equal to 5%.

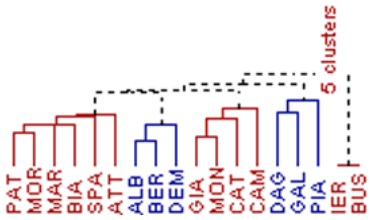

Fig. 5. The dendrogram obtained through hierarchical gene clustering.

We have performed a second evaluation according to the context "Int" using a data mining application. For each researcher and fellow we have computed his similarity with respect to the other members applying our method. In this way, we associated with each research staff member a string of values, which correspond to his relative distances from the other members. The strings correspond to the rows of the similarity matrix (Fig. 4.b). Then we have applied a tool to perform hierarchical clustering among the genetic microarray [21] to the set of strings, considering each string as a kind of researcher genetic code. The dendrogram obtained is shown in Fig. 5. It recognizes the five clusters that resemble the research group structure of our institute.

9 Related work

Semantic similarity is employed differently according to the application domain where it is adopted. Currently it is relevant in ontology alignment [22,23] and conceptual retrieval [24] as well as in semantic web service discovery and matching [25,26]. It is expected to increase in relevance in the framework for metadata analysis [27]. We discuss here related works according to their purpose and the ontology model they adopt.

Similarities in the ontology alignment. There are many methods for aligning ontology, as pointed out by Euzenat et al. [23]. Semantic similarity is adopted in this context to figure out relations among the entities in the ontology schemas. It is used to compare the names of classes, attributes and relations, determining reasonable mapping between two distinct ontologies. However, the method proposed in this paper is specifically designed to assess similarity among instances belonging to the same ontology. Some similarities adopted for ontology alignment consider quite expressive ontology language (e.g., reference [22] focuses on a subset of OWL Lite), but they mainly focus on the comparison of the structural aspects of ontology. Due to the different purposes of these methods, they turn out to be unsuitable for properly solving the similarity among instances.

Concept similarity in lexicographic databases. Different approaches to assessing semantic similarity among concepts represented by words within lexicographic databases are available. They mainly rely on edge counting-based [28] or information theory-based methods [29]. The edge counting-based method assigns terms that are subjects of the similarity assessment as edges of a tree-like taxonomy and defines the similarity in terms as the distance between the edges [28]. The information theory-based method defines the similarity of two concepts in terms of the maximum information content of the concept that subsumes them [30,31]. Recently, new hybrid approaches have been proposed: Rodriguez and Egenhofer [16] take advantage of the above methods and add the idea of features matching introduced by Tversky [14]. Schwering [24] proposes a hybrid approach to assess similarity among concepts belonging to a semantic net. The similarity in this case is assessed by comparing properties of the concept as features [14] or as geometric space [32]. With respect to the method presented in this paper, Rada et al. [28], Resnik [30] and Lin [31] work on lexicographic databases where the instances are not considered. If they are adopted, as they were originally defined, to evaluate the similarity of the instances, they are doomed to fail since they ignore important information provided by the instances, attributes and relations. Moreover, Rodriguez and Egenhofer [16] and Schwering [24] use the features or even conceptual spaces, information that is not native in the ontology design and would have to be manually added. Instead our approach aims at addressing the similarity, as much as possible, by taking advantage of the information that has already been disseminated in the ontology. Additional information is considered only to tune the similarity with respect to different application contexts.

Similarities that rely on ontology models with instances. Other works define similarity relying on ontology models closer to those adopted in the semantic web standards. D'Amato et al. [33] present a dissimilarity measure for description logics considering the expressivity of $\mathcal{ALC}$, and comparing concept descriptions and individuals/instances. Hau et al. [26] identify similar services measuring the similarity between their descriptions. To define a similarity measure on semantic services explicitly refers to the ontology model of OWL Lite and defines the similarity among OWL objects (classes as well as instances) in terms of the number of common RDF statements that characterize the objects. Maedche and Zacharias [17] adopt a semantic similarity measure to cluster ontology-based metadata. The ontology model adopted in this similarity refers also to IS-A hierarchy, attributes, relations and instances. Even if these three methods consider ontology models, which are more evolved than the taxonomy or terminological ontology, their design ignores the need to tailor the semantic similarity according to specific application contexts. Thus, to assess the similarity investigated in this paper, two distinct ontologies need to be defined instead of simply defining two contexts as we do.

Contextual-dependent similarity. Some studies combine the context and the similarity. Kashyap and Sheth [34] use the concept of semantic proximity and context to achieve interoperability among different databases. The context represents the information useful for determining the semantic relationships between entities belonging to different databases. However they do not define a semantic similarity in the sense we are addressing, and the similarity is classified as some discrete value

(semantic equivalence, semantic relevance, semantic resemblance, etc). Rodriguez and Egenhofer [16] integrate the contextual information into the similarity model. They define as the application domain the set of classes that are subject to the user's interest. Janowicz [35] proposes a context-aware similarity theory for concepts specified in expressive description logics such as $\mathcal{ALCNR}$. As in our proposal, the last two works aim to make the similarity assessment parametric with respect to the considered context. Moreover, in contrast with our methods, they formalize the context ignoring the similarity criteria induced by the context (e.g. they ignore the need of operations) and they do not directly address the similarity among instances.

This discussion of related works shows that, apart from the different definitions of semantic similarity proposed by different parties, these definitions are far from providing a complete framework as intended in our work. They often have different purposes, they consider a simpler ontology model, or they completely ignore the need to tailor the similarity assessment with respect to a specific application context. Of course, some of the works mentioned have been particularly important in the definition of our proposal. As already mentioned, both Maedche and Zacharias [17] and Rodriguez and Egenhofer [16] have strongly inspired the part related to structural similarity. However, to successfully support our purposes, the class slots have been considered as distinguishing features. Furthermore, the methods proposed by Maedche and Zacharias [17] for Class Matching define a similarity that is symmetric, thus we have adapted the original in order to make it asymmetric.

The similarity framework proposed in this paper contributes, along with related work, toward paving the way to a tool that each ontology engineer can adopt

- to define different similarities among instances on the same ontology according to different application contexts;
- to refine the similarity criteria as long as new instances are inserted or the obtained result does not satisfy the user domain expert.

The explicit parameterization of the similarity assessment with respect to the application contexts yields a precise definition of the hints to be considered in similarity assessment as well as complete control of the recursive comparison needed to work out the similarity.

10 Conclusions and future work

This paper proposes a framework for assessing semantic similarity among instances within an ontology. It combines and extends different existing similarity methods, taking into account, as much as possible, the hints encoded in the ontology and considering the application context. A formalization of the criteria induced by the application is provided as a means of parameterizing the similarity assessment and to formulate a measurement more sensitive to the specific application needs.
The framework is expected to bring great benefit in the analysis of the ontology driven metadata repository. It provides a flexible solution for tailoring the similarity assessments according to the different applications: the same ontology can be

employed in different similarity assessments simply by defining distinct criteria, and it is not necessary to build a different ontology for each similarity assessment. The formalization of the application contexts in terms of explicit similarity criteria paves the way to an iterative and interactive process where the ontology engineer and the domain experts can perform fine-tuning of the resulting similarity.
Nevertheless, some research and development issues are still open, such as human subject testing. Moreover, in the proposed approach the formalization of the application context affects only the similarity defined by the extensional comparison. It would be interesting to determine if the context results also in external comparison similarity. It would also be worthwhile to extend the similarity to ontology models towards OWL and to test it in more complex use cases.

Acknowledgements

This research started within the EU founded INVISIP project and partially performed within the Network of Excellence AIM@SHAPE.

References

1. Schwering, A. and Raubal, M.: Measuring Semantic Similarity Between Geospatial Conceptual Regions. *Proc. of the First International Conference on GeoSpatial Semantics*. LNCS Vol. 3799 Springer-Verlag Berlin Heidelber (2005) 90-106
2. Wang, H., Wang, W., Yang, J., and Yu, P. S.: Clustering by pattern similarity in large data sets. *ACM SIGMOD Conference* (2002)
3. Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., and Warke, Y.: Managing semantic content for the Web. *IEEE Internet Comput.* Vol. 6 Issue 4 (2002) 80-87
4. Medin, D. L., Goldstone, R. L., and Gentner, D.: Respects for similarity. *Psychological Review*. Vol. 100 (1993) 254-278
5. Egenhofer, M. J. and Mark, D. M.: Naive Geography. *COSIT*.LNCS Vol. 998 (1995) 1-15
6. Albertoni, R. and De Martino, M.: Semantic Similarity of Ontology Instances Tailored on the Application Context. Meersman, R., Tari, Z., and et al. *ODBASE-OTM 2006*. LNCS Vol. 4275 Springer-Verlag (2006) 1020-1038
7. Ehrig, M., Haase, P., Stojanovic, N., and Hefke, M.: Similarity for Ontologies - A Comprehensive Framework. *ECIS 2005*. Regensburg, Germany. (2005)
8. AIM@SHAPE IST NoE No 506766, http://www.aimatshape.net
9. Albertoni, R., Papaleo, L., Pitikakis, M., Robbiano, F., Spagnuolo, M., and Vasilakis, G.: Ontology-Based Searching Framework for Digital Shapes. *SWWS-OTM Workshop 2005*. LNCS Vol. 3762 Springer-Verlag (2005) 896-905
10. Papaleo, L., Albertoni, R., Marini, S., and Robbiano, F.: An ontology-based Approach to Acquisition and Reconstruction. *Workshop towards Semantic Virtual Environment. Villars*, Switzerland. (2005)
11. Falcidieno, B., Spagnuolo, M., Alliez, P., Quak, E., Vavalis, E., and Houstis, C.: Towards the Semantics of Digital Shapes: The AIM@SHAPE Approach. *Proceedings of the European Workshop for the Integration of Knowledge, Semantics and Digital Media Technology*. London, UK. QMUL (2004)
12. Albertoni, R., Camossi, E., De Martino, M., Giannini, F., and Monti, M.: Semantic Granularity for the Semantic Web. Meersman, R., Tari, Z., Herrero, P., and et al. *SWWS-OTM Workshops 2006*. LNCS Vol. 4278 Springer-Verlag (2006) 1863-1872

13. Gruber, T. R.: Toward principles for the design of ontologies used for knowledge sharing? *Int. J. Hum.-Comput. Stud.* Vol. 43 (1995) 907-928
14. Tversky, A.: Features of similarity. *Psychological Review.* Vol. 84 Issue 4 (1977) 327-352
15. Yoshida, H., Shida, T., and Kindo, T.: Asymmetric similarity with modified overlap coefficient among documents. *IEEE Pacific Rim Conference on Communications, Computers and signal Processing*, Vol. 1 (2001)
16. Rodriguez, M. A. and Egenhofer, M. J.: Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure. *Int. J. Geogr. Inf. Sci.* Vol. 18 Issue 3 (2004) 229-256
17. Maedche, A. and Zacharias, V.: Clustering Ontology Based Metadata in the Semantic Web. *PKDD 2002*. LNAI Vol. 2431 Springer-Verlag (2002) 348-360
18. Maedche, A. and Staab, S.: Measuring Similarity between Ontologies. *EKAW*. LNCS Vol. 2473 Springer-Verlag (2002) 251-263
19. Sicilia, M. A.: Metadata and semantics research. *Online Information Review*. Vol. 30 Issue 3 (2006) 213-216
20. Rodriguez, M. A. and Egenhofer, M. J.: Determining semantic similarity among entity classes from different ontologies. *IEEE Trans. Knowl. Data Eng.* Vol. 15 Issue 2 (2003) 442-456
21. Hierarchical Clustering Explorer, 3.0, http://www.cs.umd.edu/hcil/multi-cluster/
22. Euzenat, J. and Valtchev, P.: Similarity-Based Ontology Alignment in OWL-Lite. *ECAI*. Valencia, Spain. IOS Press (2004) 333-337
23. Euzenat, J., Le Bach, T., and et al.: State of the Art on Ontology Alignment. (2004), http://www.starlab.vub.ac.be/research/projects/knowledgeweb/kweb-223.pdf
24. Schwering, A.: Hybrid Model for Semantic Similarity Measurement. *ODBASE-OTM Conferences*. LNCS Vol. 3761 Springer-Verlag (2005) 1449-1465
25. Usanavasin, S., Takada, S., and Doi, N.: Semantic Web Services Discovery in Multi-ontology Environment. *OTM Workshop 2005*. LNCS Vol. 3762 Springer (2005) 59-68
26. Hau, J., Lee, W., and Darlington, J.: A Semantic Similarity Measure for Semantic Web Services. Web Service Semantics: *Towards Dynamic Business Integration, workshop at WWW 05*. (2005)
27. Albertoni, R., Bertone, A., and De Martino, M.: Semantic Analysis of Categorical Metadata to Search for Geographic Information. *Proceedings 16th International Workshop on Database and Expert Systems Applications, 2005*. IEEE (2005) 453-457
28. Rada, R., Mili, H., Bicknell, E., and Blettner, M.: Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*. Vol. 19 Issue 1 (1989) 17-30
29. Li, Y., Bandar, Z., and McLean, D.: An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. *IEEE Trans. Knowl. Data Eng.* Vol. 15 (2003) 871-882
30. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *Proc. of the Fourteenth Int. Joint Conference on Artificial Intelligence*. (1995) 448-453
31. Lin, D.: An Information-Theoretic Definition of Similarity. *Proc. of the Fifteenth Int. Conference on Machine Learning*. Morgan Kaufmann (1998) 296-304
32. Gädenfors, P.: How to make the semantic web more semantic. *FOIS*. IOS Press (2004) 17-34
33. d'Amato, C., Fanizzi, N., and Esposito, F.: A dissimilarity measure for ALC concept descriptions. *ACM Symposium of Applied Computing*, ACM (2006) 1695-1699
34. Kashyap, V. and Sheth, A.: Semantic and schematic similarities between database objects: a context-based approach. *VLDB J.* Vol. 5 Issue 4 (1996) 276-304
35. Janowicz, K.: Sim-DL: Towards a Semantic Similarity Measurement Theory for the Description Logic ALCNR in Geographic Information Retrieval. *OTM Workshops 2006.* LNCS Vol. 4278. Springer-Verlag (2006) 1681-1692