Bootstrapping Semantics on the Web: Meaning Elicitation from Schemas

Paolo Bouquet
University of Trento
Via Sommarive, 14
38050 Trento, Italy
bouquet@dit.unitn.it

Luciano Serafini
Istituto per la Ricerca Scientifica e Tecnologica
Via Sommarive, 10
38050 Trento, Italy
serafini@itc.it

Stefano Zanobini
University of Trento
Via Sommarive, 14
38050 Trento, Italy
zanobini@dit.unitn.it

ABSTRACT

In most web sites, web-based applications (such as web portals, e-marketplaces, search engines), and in the file systems of personal computers, a wide variety of schemas (such as taxonomies, directory trees, thesauri, Entity-Relationship schemas, RDF Schemas) are published which (i) convey a clear meaning to humans (e.g. help in the navigation of large collections of documents), but (ii) convey only a small fraction (if any) of their meaning to machines, as their intended meaning is not formally/explicitly represented. In this paper we present a general methodology for automatically eliciting and representing the intended meaning of these structures, and for making this meaning available in domains like information integration and interoperability, web service discovery and composition, peer-to-peer knowledge management, and semantic browsers. We also present an implementation (called CTXMATCH2) of how such a method can be used for semantic interoperability.

Categories and Subject Descriptors

I.2.4 [**Computing Methodologies**]: Artificial IntelligenceKnowledge Representation Formalisms and Methods; I.2.11 [**Computing Methodologies**]: Artificial IntelligenceDistributed Artificial Intelligence[Coherence and coordination]

General Terms

Languages

Keywords

Semantic Web, Meaning elicitation, Schema matching

1. INTRODUCTION

There is a general agreement that "encoding some of the semantics of web resources in a machine processable form" would allow designers to implement much smarter applications for final users, including information integration and interoperability, web service discovery and composition, semantic browsing, and so on. In a nutshell, this is what the Semantic Web is about. However, it is less obvious how such a result can be achieved in practice, possibly starting from the current web. Indeed, providing explicit semantic to already existing data and information sources can be extremely time and resource consuming, and may require skills that users (including web professionals) may not have.

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.

Our work starts from the observation that in many Web sites, web-based applications (such as web portals, e-marketplaces, search engines), and in the file system of personal computers, a wide variety of schemas (such as taxonomies, directory trees, Entity Relationship schemas, RDF Schemas) are published which (i) convey a clear meaning to humans (e.g. help in the navigation of large collections of documents), but (ii) convey only a small fraction (if any) of their meaning to machines, as their intended meaning is not formally/explicitly represented. Well-known examples are: classification schemas (or directories) used for organizing and navigating large collections of documents, database schemas (e.g. Entity-Relationship), used for describing the domain about which data are provided; RDF schemas, used for defining the terminology used in a collection of RDF statements. As an example, imagine that a multimedia repository uses a taxonomy like the one depicted in Figure 1 to classify pictures. For humans, it is straightforward to understand that any resource classified at the end of the path:

`PICTURES/TRENTINO/COLOR/LAKES`

is likely to be a color picture of some lake in Trentino. However, this path is of little use for a standard search engine (perhaps the labels on the path can be matched with keywords, but this would not solve the usual problems of keyword-based search), or for a more semantic-aware application, as the meaning of the path is very partially encoded in the path itself, and in the labels used to name the elements of the path. Indeed, our understanding of the path heavily depends on a large amount of contextual and domain knowledge (e.g. that pictures can be in colors or black-and-white, that lakes have some geographical location, that Trentino is a geographical location, that pictures typically have a subject, and so on); and it is only the use of this knowledge which allows humans to infer that a file `garda-panorama.jpg` appended at the end of the path above is very likely to be a color picture containing a view of a lake called Garda located in Trentino, a region in the Italian side of the Alps. Similar arguments could be used for other structures, like the ER schema in Figure 2 or the RDF Schema in Figure 3.

In the paper, we present a general methodology and an implementation to make this rich meaning available and usable by computer programs. This is a contribution to bootstrapping semantics on the Web, which can be used to automatically elicit knowledge from very common web objects. The paper has two main parts. In the first part we argue that, in making *explicit* the meaning of a schema, most approaches tend to focus on what we call *structural* meaning, but almost completely disregard (i) the *linguistic* meaning

of components (typically encoded in the labels), and (ii) its composition with structural meaning; our thesis is that this approach misses the most important aspects of how meaning is encoded in schemas. The second part of the paper describes our method for eliciting meaning from schemas, and presents an implementation called CTXMATCH2. In conclusion, as an example of an application, we show how the results of this elicitation process can be used for schema and ontology matching and alignment.

2. MEANINGFUL (?) SCHEMAS

Consider three very common types of schemas: hierarchical classifications (HCs), ER Schemas and RDF Schemas. Examples are depicted in Figures 2, 3 and 1.

Hierarchical Classifications. HCs are labeled tree-like structure whose purpose is to organize/classify data (e.g. documents[1], web pages[2], multimedia assets, goods[3], activities, services[4]). An example of a HC is depicted in Figure 1;

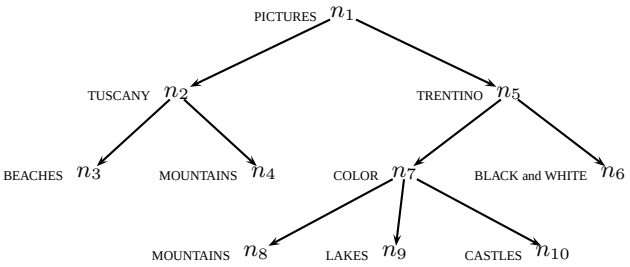

Figure 1: An example of directory structure

Entity-Relation Schemas. Entity-Relation schemas are a widely used specification language for the conceptualization of the domain of data stored in a database. An example of ER is provided in Figure 2;

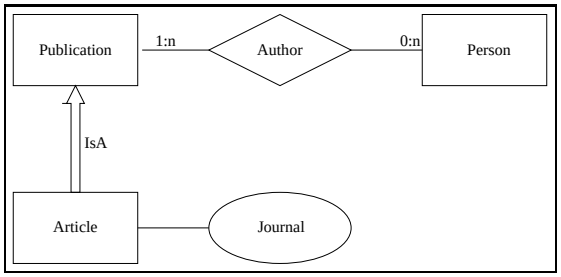

Figure 2: An example of ER Schema

[1]See for example content management tools and web portals
[2]See for example the Google™ Directory or the Yahoo!™ Directory.
[3]See e.g. standards like UNSPSC or Eclss.
[4]Web services are typically classified in a hierarchical form in a service registry, e.g. in UDDI.

RDF Schemas. An RDF schema is a specification of a vocabulary that can be used for expressing RDF statements. A tiny example of an RDF Schema can be found in Figure 3.

```
<rdfs:Class rdf:ID="Staff">
    <rdfs:comment="A Staff member at ISI"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Researcher">
    <rdfs:comment="A Researcher at ISI"/>
    <rdfs:subClassOf rdf:resource="#Staff"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Paper">
    <rdfs:comment="A Published paper"/>
</rdfs:Class>

<rdf:Property rdf:ID="Author">
    <rdfs:comment="Authors of the paper"/>
    <rdfs:domain rdf:resource="#Researcher"/>
    <rdfs:range rdf:resource="#Paper"/>
</rdf:Property>
```

Figure 3: An example of RDF document

They are used in different domains (document management, database design, vocabulary specification) to provide a structure which can be used to organize information sources. However, there is a second purpose which is typically overlooked, namely to provide humans with an easier access to those data. This is achieved mainly by labelling the elements of a schema with meaningful labels, typiclaly from some natural language. This is why, in our opinion, it is very uncommon to find a taxonomy (or an ER schema, or an RDF Schema specification) whose labels are meaningless for humans. Imagine, for example, how odd (and maybe hopeless) it would be for a human to navigate a classification schema whose labels are meaningless strings; or to read a ER schema whose nodes are labeled with random strings. Of course, humans would still be able to identify and use some formal properties of such schemas (for example, in a classification schema, we can always infer that a child node is more specific than its parent node, because this belongs to the structural understanding of a classification), but we would have no clues about what the two nodes are about. Similar observations can be made for the two other types of schemas. So, our research interest can be stated as follows: can we define a method which can be used to automatically elicit and represent the meaning of a schema in a form that makes available to machines the same kind of rich meaning which is available to humans when going through a schema?

3. STRUCTURAL AND LEXICAL ANALYSIS

We said that each node (e.g. in a HC) has an intuitive meaning for humans. For example, the node n_4 in Figure 1 can be easily interpreted as "pictures of mountains in Tuscany", whereas n_8 can be interpreted as "color pictures of mountains in Trentino". However, this meaning is mostly implicit, and its elicitation may require a lot of knowledge which is either encoded in the structure of the schema, or must be extracted from external resources. In [5], we identified at least three distinct levels of knowledge which are used to elicit a schema's meaning:

Structural knowledge: knowledge deriving from the arrangement of nodes in the schema;

Lexical knowledge: knowledge about the meaning of words used to label nodes;

Domain knowledge: real world knowledge about meanings and their relations.

Most past attempts focused only on the first level. A recent example is [19], in which the authors present a methodology for converting thesauri into RDF/OWL; the proposed method is very rich from a structural point of view, but labels are disregarded, and no background domain knowledge is used. As to ER schemas, a formal semantics is defined for example in [4], using Description Logics; again, the proposed semantics is completely independent from the intuitive meaning of expressions used to label single components. For RDF Schemas, the situation is slightly different. Indeed, the common understanding is that RDFS schemas are used to define the meaning of terms, and thus their meaning is completely explicit; however, we observe that even for RDFS the associated semantics (see `http://www.w3.org/TR/rdf-mt/`) is purely structural, which means that there is no special interpretation provided for the labels used to name classes or other resources.

However, as we argued above through a few examples, labels (together with their organization in a schema) appear to be one of the main sources of meaning for humans. So we think that considering only structural semantics is not enough, and may lead to at least two serious problems:

- we may be unable to discriminate between schemas that are structurally, but not semantically, isomorphic;
- we may be unable to make any conjecture on the meaning of edges connecting nodes (elements) of a schema.

The first issue can be explained through a simple example. Suppose we have some method ϵ for making explicit the meaning of paths in HCs, and that ϵ does not take the meaning of labels into account. Now imagine we apply ϵ to the path n_1–n_3 in Figure 1, and compare to a path like `ANIMALS`/`MAMMALS`/`DOGS` in another schema (notice that typical HCs do not provide any explicit information about edges in the path). Whatever representation ϵ is capable of producing, the outcome for the two paths will be structurally isomorphic, as the two paths are structurally isomorphic. However, our intuition is that the two paths have a very different semantic structure: the first should result is a term where a class ("pictures") is modified/restricted by two attributes ("pictures **of** beaches **located in** Tuscany"); the second is a standard Is-A hierarchy, where the relation between the three classes is subsumption. The only way we can imagine to explain this semantic (but not structural) difference is by appealing to the meaning of labels. We grasp the meaning of the first path because we know that pictures have a subject (e.g. beaches), that beaches have a geographical location, and that Tuscany is a geographical location. All this is part of what we called lexical and domain knowledge. Without it, we would not have any reason to consider "pictures" as a class and "Tuscany" and "beaches" as values for attributes of pictures. Analogously, we know that (a sense of the word) "dog" in English refers to a subclass of the class denoted by (a sense of the word) "mammals" in English, and similarly for "animals".

The second issue is closely related to the first one. How do we understand (intuitively) that `PICTURES`/`TUSCANY`/`BEACHES` refers to pictures **of** beaches **located in** Tuscany, and not e.g. to pictures **working for** Tuscany **teaching** beaches? After all, the edges between nodes are not qualified, and therefore any structurally possible relation is in principle admissible. The answer is trivial: because, among other things, *we know* that pictures do not work for anybody (but they may have a subject), that Tuscany can't be the teacher of a beach (but can be the geographical location of a beach). It is only this body of background knowledge which allows humans to conjecture the correct relation between the meanings of node labels. If we disregard it, there is no special reason to prefer one interpretation to the other.

The examples above should be sufficient to support the conclusion that any attempt to design a methodology for eliciting the meaning of schemas (basically, for reconstructing the intuitive meaning of any schema element into an explicit and formal representation of such a meaning) cannot be based exclusively on structural semantics, but must seriously take into account at least lexical and domain knowledge about the labels used in the schema[5]. The methodology we propose in the next section is an attempt to do this.

4. MEANING REPRESENTATION

Intuitively, the problem of semantic elicitation can be viewed as the problem of computing and representing the (otherwise implicit) meaning of a schema in a machine understandable way. Clearly, meaning for human beings has very complex aspects, directly related to human cognitive and social abilities. Trying to reconstruct the entire and precise meaning of a term would probably be a hopeless goal, so our intuitive characterization must be read as referring to a reasonable approximation of meaning.

In our method, meanings are represented in a formal language (called WDL, for Wordnet Description Logic), which is the result of combining two main ingredients: a *logical language* (in this paper, use the logical language $\mathcal{ALCIO}$ which belongs to the family of *Description Logics* [2]), and IDs of lexical entries in a dictionary (more specifically, from WORDNET [8], a well-known electronic lexical database). Description logics are a family logical languages that are defined starting from a set of primitive concepts, relations and individuals, with a set of logical constructors, and has been proved to provide a good compromise between expressivity and computability. It is supported with efficient reasoning services (see for instance [14]); WORDNET is the largest and most shared online lexical resource, whose design is inspired by psycholinguistic theories of human lexical memory. WORDNET associates with any word "`word`" a list of senses (equivalent to entries in a dictionary), denoted as `word#1`, . . . , `word#1`, each of which denotes a possible meaning of "`word`".

The core idea of WDL is to use a DL language for representing structural meaning, and any additional constraints (axioms) we might have from domain knowledge; and to use WORDNET to anchor the meaning of labels in a schema to lexical meanings, which are listed and uniquely identified as WORDNET senses. Indeed, the primitives of any DL language do not have an "intended" meaning; this is evident from the fact that, as in standard model-theoretic semantics, the primitive components of DL languages (i.e. concepts, roles, individuals) are interpreted, respectively, as generic sets, relations or individuals from some domain. What we need to do is to "ground" their interpretation to the WordNet sense that best represents their intended meaning in the label. So, for example, a label like `LAKES` can be interpreted as a generic class in a standard DL semantics, but can be also assigned an intended meaning by attaching it to the the first sense in WORDNET (which in version 2.0 is defined as "a body of (usually fresh) water surrounded by land").

[5]We say "at least", as there are other obvious types of knowledge which one may think of using, from an analysis of data associated with an element to more general contextual factors, like the application or the processes in which the schema is used. Here we ignore these other factors for the sake of simplicity, but of course they are relevant.

The advantage of WDL w.r.t. a standard DL encoding is that assigning an intended meaning to a label allows us to import automatically a body of (lexical) knowledge which is associated with a given meaning of a word used in a label. For example, from WORDNET we know that there is a relation between the class "lakes" and the class "bodies of water", which in turn is a subclass of physical entities. In addition, if an ontology is available where classes and roles are also lexicalized (an issue that here we do not address directly, but details can be found in [17]), then we can also import and use additional domain knowledge about a given (sense of) a word, for example that lakes can be holiday destinations, that Trentino has plenty of lakes, even that a lake called "Lake Garda" is partially located in Trentino, and so on and so forth.

Technically, the idea described above is implemented by using WORDNET senses as primitives for a DL language. A WDL language is therefore defined as follows:

- the sets $\mathcal{C}$, $\mathcal{R}$ and $\mathcal{O}$ of (names for) primitive concepts, roles and individuals of WDL are subsets of WORDNET senses;
- complex concepts can be defined with the following production rule

$$\begin{array}{rcl} C & := & \mathsf{c\#k} \mid C \sqcap C \mid C \sqcup C \mid \neg C \mid \forall R.C \mid \exists R.C \\ & & \mid \{\mathsf{o\#k}, \ldots, \mathsf{o\#k}\} \\ R & := & \mathsf{r\#k} \mid \mathsf{r\#k}^{-} \end{array}$$

  where $\mathsf{c\#k} \in \mathcal{C}$, $\mathsf{r\#k} \in \mathcal{R}$, and $\mathsf{o\#k} \in \mathcal{O}$;
- An axiom in WDL is an expression of the form $C \sqsubseteq D$, where C and D are complex concepts, and $C(a)$, where C is a concept and a is an individual.

Some remarks are necessary.

- Unlike standard DL, in a WDL language, concepts, roles, and individuals, are not disjoint sets. This is required for modeling the fact that a word sense like location#3 (as "a determination of the place where something is") can be both a concept and a role. Formally, this is not a problem, as the context where a primitive object occurs makes it possible to determine whether it must be considered a concept, a role, or an individual.
- WDL has two semantics: a *formal semantics* and an *intended semantics*. The formal semantics is a mathematical function $\cdot^{\mathcal{I}}$ that associates with each primitive concept C a set $C^{\mathcal{I}}$ of objects, to each primitive role R a binary relation $R^{\mathcal{I}}$, and to each individual o, an object $o^{\mathcal{I}}$. The formal semantics of complex concepts, and axioms can be defined inductively (see [2] for details).
- the *intended semantics* is a new (derived) sense, which might not be in WORDNET, and that can be associated with a gloss obtained by combining the glosses of the components. So, the intended semantics of $\mathsf{Car\#1} \sqcap \exists \mathsf{Color\#1}.\{\mathsf{Red\#1}\}$ is "a motor vehicle with four wheels; usually propelled by an internal combustion engine", which has "a visual attribute ... that results from the light it emits or transmit or reflect", which is "... the chromatic color resembling the hue of blood". In short, a red car.

Despite the fact that the intended semantics cannot be formally represented or easily determined by a computer, one should accept its existence and consider it at the same level as a "potential" WORDNET sense. Under this hypothesis we can assume that expressions in WDL convey meanings, and can be used to represents meaning in a machine. Put it differently, since the WDL primitives represent common-sense concepts, then the complex concepts of WDL will also represent common-sense concepts, since common-sense concepts are closed under boolean operations and universal and existential role restriction.

EXAMPLE 1. *Let us give some examples of the use of WDL descriptions to represents the meanings of the nodes of the schemas introduced in the previous section.*

- *The meaning of the node labeled with "Publication" considered in the context of the ER schema of Figure 2*[6] *is*

  $$\mathit{Publication\#1} \sqcap \exists \mathit{Author\#1}^{-}.\mathit{Person\#1}$$

  and the intuitive semantics is "a copy of a printed work offered for distribution" that "a human being", "writes ... professionally ... ".
- *The meaning of the node labeled with* **paper** *is*

  $$\mathit{paper\#2}$$

  and the intuitive semantics is "an essay (especially one written as an assignment"
- *Finally, the meaning of the node n_3 of the hierarchical classification of Figure 1 is*

  $$\mathit{image\#2} \sqcap \exists \mathit{subject\#4}.(\mathit{beaches\#1} \sqcap \exists \mathit{Location\#1}.\{\mathit{Tuscany\#1}\})$$

  and the intuitive semantics is "a visual representation produced on a surface of" "areas of sand sloping down to the water of a sea or lake" "situated in a particular spot or position" which is "a region in central Italy"

From this perspective, **the problem of semantic elicitation** can be thought of as the problem of finding a WDL expression $\mu(n)$ for each element n of a schema, so that the intuitive semantics of μn is a good enough approximation of the intended meaning of the node.

5. SEMANTIC ELICITATION IN PRACTICE

This section is devoted to the description of a practical semantic elicitation algorithm. This algorithm has been implemented as basic functionality of the CTXMATCH2 matching platform [17], and has been extensively tested in the 2nd Ontology Alignment Evaluation Initiative[7].

In the following we will adopt the notation $\mu(m)$ to denote the meaning of a node n. $l(n)$ to denote the label of the node, and $\mu(l(n))$ or simply $\lambda(n)$ to denote the meaning of a label associated with the node n considered out of its context. $\lambda(n)$ is also called the *local meaning*.

The algorithm for semantic elicitation is composed of three main steps. In the first step we use the structural knowledge on a schema to build a *meaning skeleton*. A meaning skeleton describes only the structure of a WDL complex concepts that constitutes the meaning of a node. In the second step, we fill nodes of with the appropriate concepts and individuals, using linguistic knowledge, and in the final step, we provide the roles, by exploiting domain knowledge.

[6]Notice that the same node considered in a different context could have a different meaning.

[7]Results are described at `http://oaei.inrialpes.fr/2005/results`, and discussed in [7].

5.1 Meaning Skeletons

Given a schema, the structural knowledge (structural semantics) associated with this schema provides the skeleton for the meaning of each node. Therefore our procedure will start from this skeleton, and will try to fill the gaps with the extra, implicit semantics, obtained from lexical and domain knowledge. In this section we will describe the structural knowledge which can be associated with each of the three types of schemas presented above, and how it can be used to produce a meaning skeleton.

Meaning skeletons are DL descriptions together with a set of axioms. The basic components of a meaning skeleton (i.e. the primitive concepts and roles) are the meanings of the single labels associated with nodes, denoted by $\lambda(n)$), and the semantic relations between different nodes (denoted by R_{ij}). Intuitively R_{ij} represents a semantic relation between the node n_i and the node n_j. In the rest of this section we show how the meaning skeletons of the types of schema considered in this paper are computed.

A number of alternative formalizations for HCs have been proposed (e.g., [15, 20, 9]). Despite their differences, they share the idea that, in a HC, the meaning of a node is a specification of the meaning of its father node. E.g., the meaning of a node labeled with "clubs", with a father node which means "documents about Ferrari cars" is "Ferrari fan clubs". In DL, this is encoded as $\mu(n) = \lambda(n) \sqcap \exists R_{nm}.\mu(m)$, where R_{nm} is some node that connets the meaning of n with that of m. If the label of n is for instance "F40" (a Ferrari model) then the meaning of n is "documents about Ferrari F40 car", then it is the meaning of the label of n that acts as modifier of the meaning of m. In description logics this is formalized as $\mu(n) = \mu(m) \sqcup \exists R_{mn}.\lambda(n)$. The choice between the first of the second case essentially depends both on lexical knowledge, which provides the meaning of the labels, and domain knowledge, which provides candidate relations between $\mu(m)$ and $\lambda(n)$. The following table summarizes some meaning skeletons associated with the HC provided above:

node	meaning skeleton
n_1	$\lambda(n_1)$
n_2	$\lambda(n_1) \sqcap \exists R_{12}.\lambda(n_2)$ or $\lambda(n_2) \sqcap \exists R_{21}.\lambda(n_1)$
n_3	$\lambda(n_1) \sqcap \exists R_{12}.\lambda(n_2) \sqcap \exists R_{13}.\lambda(n_3)$ or $\lambda(n_2) \sqcap \exists R_{21}.\lambda(n_1) \sqcap \exists R_{13}.\lambda(n_3)$ or $\lambda(n_3) \sqcap \exists R_{31}.(\lambda(n_1) \sqcap \exists R_{12}.\lambda(n_2))$ or $\lambda(n_3) \sqcap \exists R_{31}.(\lambda(n_2) \sqcap \exists R_{21}.\lambda(n_1))$

Notice that, since at this level we do not have knowledge to distinguish which node is the modifier of the other, we have to consider all the alternative meaning skeletons.

Unlike HCs, the formal semantics for ER schemata is widely shared. In [4], one can find a comprehensive survey of this area. Roughly speaking, any ER schema can be converted in an *equivalent* set of DL axioms, which express the formal semantics of such a schema. This formal semantics is defined independently from the meaning of the single nodes (labels of nodes). Every node is considered as an atom. To stress this fact in writing meaning skeletons for ER, we will assign to each node an anonymous identifier. For instance we use $n_1, \ldots, n_5$ to denote the 5 nodes of the schema of Figure 2.

If we apply the formal semantics described in [4] to the example of ER given above, we obtain the following meaning skeletons.

node	label	meaning skeleton
n_1	Publication	$\lambda(n_1) \sqcap \exists\lambda(n_2).\lambda(n_3)$
n_2	Author	$\lambda(n_2)$ plus the axioms $\top \sqsubseteq \forall\lambda(n_2).\lambda(n_3)$, $\lambda(n_1) \sqsubseteq \exists\lambda(n_2).\lambda(n_3)$
n_3	Person	$\lambda(n_3)$
n_4	Article	$\lambda(n_4) \sqcap \lambda(n_1) \sqcap \exists R_{45}.\lambda(n_5)$
n_5	Journal	$\lambda(n_5)$.

This table can be read as follows: The meaning skeleton of node n_1 labeled with "Publication" is a DL concept description $\lambda(n_1) \sqcap \exists\lambda(n_2).\lambda(n_3)$, denoting any set of objects which are related to at least another object of some other set. The node n_2 labeled with "author", is associated with a binary relation, that satisfies the associated domain and range axioms. Similar interpretation can be given to the other nodes. It is important to notice that *the meaning skeleton is independent from the labels, and ER schemas which are structurally the same will produce meaning skeletons which are equal*.

The meaning skeleton of the RDF Schema described in Figure 3 is provided by the formal semantics for RDF schema described for instance in [11]. Most commonly used RDFS constructs can be rephrased in terms of description logics, as discussed in [13]. As we did above, we report the meaning skeletons for some of the nodes of the RDF Schema of Figure 3 in a table, in which we "anonymize" the nodes, by giving them meaningless names.

node	label	meaning skeleton
n_1	Staff	$\lambda(n_1)$
n_2	Researcher	$\lambda(n_2)$ with the axiom $\lambda(n_2) \sqsubseteq \lambda(n_1)$
n_3	Paper	$\lambda(n_3)$
n_4	Author	$\lambda(n_4)$ with the axioms $\exists\lambda(n_4).\top \sqsubseteq \lambda(n_2)$ $\top \sqsubseteq \forall\lambda(n_4).\lambda(n_3)$

The observations about ER schemas mostly hold also for the meaning skeletons of RDF Schemas. Moreover, it is worth observing that the comments of the RDF Schema are not considered in the formal semantics, and therefore they are not reported in the meaning skeletons. However, we all know that comments are very useful to understand the real meaning of a concept, especially in large schema. As we will see later, they are indeed very important to select and add the right domain knowledge to the meaning skeleton.

5.2 Local meaning ($\lambda(n)$)

The local meaning of a node in a schema, denoted by $\lambda(n)$, is a DL description approximating all possible meanings of the label associated with a node. To compute $\lambda(n)$, we make an essential use of linguistic resources. Following the WORDNET approach, we define a *linguistic resource* as a function which, to any word, associates a set of *senses*, each representing an acceptable meaning of that word. Examples of linguistic resouces are WORDNET itself, thesauri, databases for acronyms, even lists of names, etc.

If the label of a node n is a simple word like "Image", or "Florence", then $\lambda(n)$ represents all senses that this word can have in any possible context. For example, WORDNET provides seven senses for the word "Images" and two for "Florence". If m and n are nodes labeled with these two words, then $\lambda(m) = \mathsf{Image\#1} \sqcup \mathsf{Image\#2} \sqcup \cdots \sqcup \mathsf{Image\#7}$ and $\lambda(n) = \mathsf{Florence\#1} \sqcup \mathsf{Florence\#2}$.

When labels are more complex than a single word, as for instance "University of Trento", or "Component of Gastrointestinal Tract"

T-Box

$\mathsf{Image\#1} \sqsubseteq \exists \mathsf{Subject\#4}.\mathsf{PhysicalEntity\#1}$
$\mathsf{Image\#1} \sqsubseteq \exists \mathsf{Size\#1}.\mathsf{Size\#2}$
$\mathsf{Town\#1} \sqsubseteq \mathsf{PhysicalEntity\#1} \sqcap \exists \mathsf{Located\#2}.\mathsf{State\#4}$
$\mathsf{Image\#1} \sqcap \exists \mathsf{Subject\#4}.(\mathsf{Town\#1} \sqcap \mathsf{Located\#2}.\{\mathsf{Italy\#2}\}) \sqsubseteq \mathsf{Image\#2} \sqcap \exists \mathsf{Subject\#4}.\{\mathsf{Italy\#1}\}$
$\mathsf{Article\#3} \sqsubseteq \exists \mathsf{publish\#2}^{-}.(\mathsf{Journal\#3} \sqcup \mathsf{Proceedings\#1})$

A-Box

$\mathsf{State\#4}(\mathsf{Italy\#1})$
$\mathsf{State\#4}(\mathsf{SouthCarolina\#1})$
$\mathsf{FootbalTeam\#1}(\mathsf{Italy\#2})$
$\mathsf{Located\#2}(\mathsf{Florence\#1}, \mathsf{Italy\#1})$
$\mathsf{Located\#2}(\mathsf{Florence\#2}, \mathsf{SouthCarolina\#1})$

Figure 4: An example of Knowledge Base

(occurring in Galen Ontology [16]) then $\lambda(n)$ is a more complex DL description computable with advanced natural language techniques. The description of these techniques is beyond the scope of this paper and we refer the reader to [12]. For the sake of explanation we therefore concentrate our attention to single word labels.

5.3 Relations between local meanings (R_{mn})

With respect to our methodology, a body of domain knowledge (called a knowledge base) can be viewed as a set of facts describing the properties and the relations between the objects of a domain. For instance, a geographical knowledge base may contain the fact that Florence is a town located in Italy, and that Florence is also a town located in South Carolina. Clearly, the knowledge base will use two different constants to denote the two Florences. From this simple example, one can see how knowledge base relations are defined between meanings rather than between linguistic entities.

More formally, we define a knowledge base to be a pair $\langle T, A\rangle$ where T is a T-box (terminological box) and A is an A-box (assertional box) of some descriptive language. Moreover, to address the fact that knowledge is about meanings, we require that the atomic concepts, roles, and individuals that appear in the KB be taken from a set of senses provided by one (or more) linguistic resources. An fragment of knowledge base relevant to the examples given above is shown in Figure 4.

Domain knowledge is used to discover semantic relations holding between local meanings. Intuitively, given two primitive concepts C and D, we search for a role R that possibly connect a C-object with a D-object. As an example, suppose we need to find a role that connects the concept Image#2 and the nominal concept $\{\mathsf{Florence\#1}\}$; in the knowledge base of Figure 4, a candidate relation is Subject#4. This is because Florence#1 is a *possible value* of the attributed Subject#4 of an Image#1.

More formally, R is a semantic relation between the concept C and D w.r.t., the knowledge base KB if and only if

i KB $\models C \sqsubseteq \exists R.E$ for some primitive concept E,

ii KB $\models D \sqsubseteq E$, and

iii for all primitive concepts F, KB $\models C \sqsubseteq \exists R.F$ implies that KB $\models E \sqsubseteq F$.

Conditions *i–iii* intuitively state that R is a role that connects C with D if, every C has an R which is F (condition i), and F is the smaller super-concept of D (conditions ii and iii) that has this property. By including R_{id} (the Identity Role $= \{x, x | x$ is any element of the domain$\}$) as a possible semantic relation between two concepts, the above definition captures the is-a relationship. Indeed, $C \sqsubseteq \exists R_{id} D$ is equivalent to $C \sqsubseteq D$ (C is a D). [17] contains the detailed description of an algorithm for computing semantic relations between concepts.

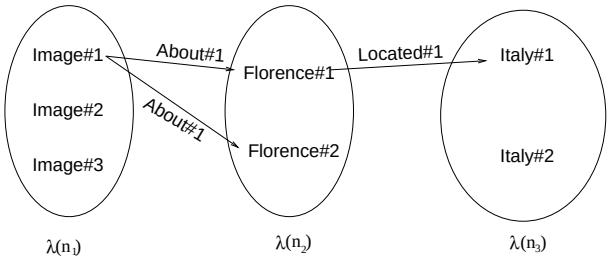

Figure 5: Semantic relation between senses

According to this definition one can verify that Subject#4 is a semantic relation between Image#1 and the nominal concept $\{\mathsf{Florence\#1}\}$. Indeed KB $\models \mathsf{Image\#1} \sqsubseteq \mathsf{PhysicalEntity\#1}$ (condition i), KB $\models \{\mathsf{Florence\#1}\} \sqsubseteq \mathsf{PhysicalEntity\#1}$ (condition ii) and for no other primitive concepts F different from PhysicalEntity#1 we have that KB $\models \mathsf{Image\#1} \sqsubseteq \exists \mathsf{Subject\#4}.F$ (condition iii). Similarly Located#2 is a semantic relation between the nominal concepts Florence#1 and Italy#1, but it is not a semantic relation between Florence#2 and Italy#1.

The relations computed via conditions *i–iii* can be used also for disambiguation of local meanings. Namely, the existence of a semantic relation between two senses of two local meanings, constitutes an evidence that those senses are the right one. This allows us to discard all the others. For instance in the situation depicted in Figure 5, it to keep the sense Image#1 and eliminate the other two senses from the local meaning $\lambda(n_1)$. Similarly we prefer Florence#1 on Florence#2 since the former has more semantic relations that the latter.

6. AN APPLICATION: MATCHING HCS

As we said in the introduction, the idea and method we proposed can be applied to several fields, including semantic interoperability, information integration, peer-to-peer databases, and so on. Here, as an illustration, we briefly present an application which we developed, where semantic elicitation is used to implement a semantic method for matching hierarchical classifications (HCs).

Matching HCs is an especially interesting case for the Web. Indeed, classifying documents is one of the main techniques people use to improve navigation across large collections of documents. Probably the most blatant example is that of web directories, which most major search engines (e.g. Google, Yahoo!, Looksmart) use to classify web pages and web accessible resources. Suppose that a Web user is navigating Google's directory, and finds an interesting category of documents (for example, the category named 'Baroque' on the left hand side of Figure 6 along the path `Arts > Music > History > Baroque`) She might want to find semantically related categories in other web directories. One way of achieving this result is by "comparing" the meaning of the selected category with the meaning of other categories in different directories. In what follows, we will describe a P2P-like approach to this application, which was developed as part of a tool for supporting distributed knowledge management called KEx [3]. The example discussed in this section is adapted from [6]. The entire matching process is run by CTXMATCH2.

Imagine that both Google and Yahoo had enabled their web di-

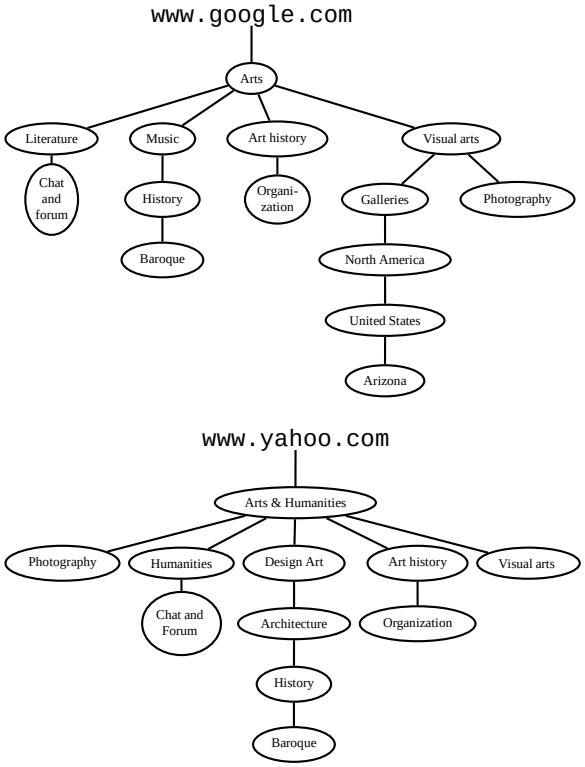

Figure 6: Two HCs on the Web

rectories with some semantic elicitation system[8]. This means that each node in the two web directories is equipped with a WDL formula which represents its meaning. In addition, we can imagine that each node contains also a body of domain knowledge which has been extracted from some ontology; this knowledge is basically what it is locally known about the content of the node (for example, given a node labeled `TUSCANY`, we can imagine that it can contain also the information that Tuscany is a region in Central Italy, whose capital is Florence, and so on).

Let us go back to our Google user interested in Baroque music. When she selects this category, we can imagine that the following process is started

1. the WDL formula representing the meaning of the node labeled `BAROQUE` in the Google's directory is extracted;

2. this formula is sent to other web directories (e.g. to Yahoo), together with a request to find semantically related nodes;

3. the semantic application at Yahoo will try to logically deduce (using any DL reasoner, like Racer, Pellet[9], or Fact[10]) whether any of the formulae attached to the local directory is in a relevant relation (e.g. equivalence, or subsumption) with the WDL formula attached to the request. Notice that, in computing potential relations, other background knowledge can be extracted from a local ontology to maximize the chances of discovering a relation;

4. if a Yahoo's WDL formula can be proved to be semantically related to the received formula, then the corresponding node in the web directory is returned as a search result. Otherwise, nothing is returned.

In the following table we present some results obtained through CTXMATCH2 for finding relations between the nodes of the portion of Google and Yahoo classifications depicted in Figure 6.

Google node	Yahoo node	semantic relation
`Baroque`	`Baroque`	Disjoint ($\bot$)
`Visual Arts`	`Visual Arts`	More general than ($\supseteq$)
`Photography`	`Photography`	Equivalent ($\equiv$)
`Chat and Forum`	`Chat and Forum`	Less general than ($\subseteq$)

In the first example, CTXMATCH2 returns a 'disjoint' relation between the two nodes `Baroque`: the presence of two different ancestors (`Music` and `Architecture`) and the related world knowledge 'Music is disjoint with Architecture' allow us to derive the right semantic relation.

In the second example, CTXMATCH2 returns the 'more general than' relation between the nodes `Visual Arts`. This is a rather sophisticated result: indeed, world knowledge provides the information that 'photography IsA visual art' (`photography#1` $\rightarrow$ `visual art#1`). From structural knowledge, we can deduce that, while in the left structure the node `Visual Arts` denotes the whole concept (in fact `photography` is one of its children), in the right structure the node `Visual Arts` denotes the concept 'visual arts except photography' (in fact `photography` is one of its siblings). Given this information, it easy to deduce that, although despite the two nodes lie on the same path, they have different meanings.

The third example shows how the correct relation holding between nodes `Photography` is returned ('equivalence'), despite the presence of different paths, as world knowledge tells us that `photography#1` $\rightarrow$ `visual art#1`.

Finally, between the nodes `Chat and Forum` a 'less general than' relation is found as world knowledge gives us the axiom 'literature is a humanities'.

7. RELATED WORK

This work has been inspired from the approach described in [12] in which the technique of semantic elicitation has been applied to the special case of hierarchical classification. The approach described in this paper extends this initial approach in three main directions. First, the logic in which the meaning is expressed in some description logic, while in [12] meaning was encoded in propositional logic. Second, [12] adopts only WORDNET as both linguistic and domain knowledge repository, while in this approach we allow the use of multiple linguistic resources, and knowledge bases. Third, in [12] no particular attention was paid to structural knowledge, while here we introduced the concept of a meaning skeleton, which captures exactly this notion.

The paper [1] describes an approach which enrich xml schema with the semantic encoded in an ontology. This approach is similar in the spirit of the idea of semantic elicitation of schemas, but it does not make an extensive use of explicit structural knowledge,

[8]Here, for the sake of simplicity, we will assume that both Google and Yahoo use the same elicitation system, which basically means that they use the same WDL language. This makes everything easier, as two WDL formulae can be directly compared. However, in [6], this assumption is relaxed, and it is discussed how peers with different lexical resources and ontologies can still try to coordinate their local HCs.

[9]See `http://www.mindswap.org/2003/pellet/`.

[10]See `http://www.cs.man.ac.uk/~horrocks/FaCT`.

and of linguistic knowledge, which are two of the three knowledge sources used in our approach.

The approach described in [18] describes a possible application of the linguistic enrichment of an ontology in the area of keyword based document retrieval. This approach is quite similar in the spirit on what we have proposed here, with the limitation of considering only hierarchical classifications. Moreover, in the process of enriching a concept hierarchy, no domain knowledge is used.

Finally, most of the approaches of schema matching uses linguistic knowledge (WORDNET) and domain knowledge to find correspondences between elements of heterogeneous schemata. Among all the approaches CTXMATCH [5] and [10] is based on the idea of matching meaning, rather than matching syntax. Both approaches implement a two step algorithm, and the first phase computes the meaning of a node by using linguistic and domain knowledge. However both approaches are based on propositional logic.

8. CONCLUSIONS

Semantic elicitation mat be an important method for bootstrapping semantics on the web. Our method does not address the issue of extracting knowledge from documents, which of course will be the main source of semantic information. But knowledge extraction from documents is still an expensive and error prone task, as it must address a lot of well-known problems related to natural language analysis. Instead, semantic elicitation can be applied to objects which have a simpler structure (labels are typically quite simple from a linguistic point of view), and thus is less demanding from a computational point of view and more precise (needless to say, a lot of errors may occur, see [5] for a few tests). But schemas, as we said, are very common on the web, and have a very high informative power. Moreover, in many applications in the area integration of semantic web services the only available information is based on schemas and no data are present. Therefore, we assume that, in the short-mid term, this would be one of the main ways to add semantics to data on the web on a large scale.

9. ACKNOWLEDGMENTS

The work described in this paper has been partly funded by the European Commission through grant to the project VIKEF (Virtual Information and Knowledge Environment Framework) under the number IST-507173.

10. ADDITIONAL AUTHORS

Additional authors: Simone Sceffer (Istituto per la Ricerca Scientificica e Tecnologica, email: sceffer@itc.it).

11. REFERENCES

[1] Y. An, A. Borgida, and J. Mylopoulos. Constructing complex semantic mappings between xml data and ontologies. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *4th International Semantic Web Conference (ISWC'05)*, LNCS 3729, pages 6–20. Springer, 2005.

[2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[3] M. Bonifacio, P. Bouquet, G. Mameli, and M. Nori. Kex: a peer-to-peer solution for distributed knowledge management. In D. Karagiannis and U. Reimer, editors, *Fourth International Conference on Practical Aspects of Knowledge Management (PAKM-2002)*, Vienna (Austria), 2002.

[4] A. Borgida, M. Lenzerini, and R. Rosati. Description logics for databases. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The Description Logics Handbook*, chapter 16, pages 472–494. Cambridge University Press, 2002.

[5] P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: A new approach and an application. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *The Semantic Web - ISWC 2003*, LNCS 2870, pages 130–145, Sanibel Island (FL, USA), Springer, 2003.

[6] P. Bouquet, L. Serafini, and S. Zanobini. Peer-to-peer semantic coordination. *Journal of Web Semantics*, 2(1), 2005.

[7] P. Bouquet, M. Yatskevich, and S. Zanobini. Critical analysis of mapping languages and mapping techniques. Technical Report DIT-05-052, DIT-University of Trento, 2005.

[8] C. Fellbaum, editor. WORDNET*: An Electronic Lexical Database*. MIT Press. ISBN 0-262-06197-X.

[9] F. Giunchiglia, M. Marchese, and I. Zaihrayeu. Towards a theory of formal classification. Technical Report DIT-05-048, DIT University of Trento, 2005.

[10] F. Giunchiglia, P. Shvaiko, and M. 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*, Dagstuhl Seminar Proceedings 04391, 2005.

[11] P. Hayes. RDF Semantics - W3C Recommendation 10 Feb 2004. http://www.w3.org/TR/rdf-mt/.

[12] B. Magnini, L. Serafini, and M. Speranza. Making explicit the semantics hidden in schema models. In A. Cappelli and F. Turini, editors, *8th Congress of the Italian Association for Artificial Intelligence*, Pisa, Italy, LNAI 2829, pages 436–448. Springer, 2003.

[13] L. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview - W3C Recommendation 10 Feb 2004. http://www.w3.org/TR/owl-features/.

[14] R. Möller and V. Haarslev. Description logic systems. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The Description Logic Handbook*, chapter 8, pages 282–305. Cambridge University Press, 2003.

[15] A. H. P. Cimiano and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24:305–339, 2005.

[16] A. L. Rector, J. E. Rogers, and P. A. Pole. The Galen high level ontology. In *Proceedings MIE 96*, pages 174–178. IOS Press, 1996.

[17] L. Serafini, S. Sceffer, and S. Zanobini. Semantic coordination of hierarchical classifications with attributes. Technical Report T04-12-04, ITC-IRST, 2004.

[18] S. Tiun, R. Abdullah, and T. Enyakong. Automatic topic identification using ontology hierarchy. In *2nd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2001)*, 2001.

[19] M. van Assem, M. R. Menken, G. Schreiber, J. Wielemaker, and B. Wielinga. A method for converting thesauri to RDF/OWL. In *Proceedings of the Third International Semantic Web Conference (ISWC'04)*, LNCS 3298, pages 17–31, Hiroshima (Japan), Springer, 2004

[20] L. L. Whyte, A. G. Wilson, and D. Wilson, editors. *Hierarchical structures*. American Elsevier, 1969.