Manageable Approaches to the Semantic Web

Philippe Martin & Peter Eklund

Distributed System Technology Centre &
School of Information Technology and Electrical Engineering,
The University of Queensland, St Lucia QLD 4072, Australia
philippe.martin@gu.edu.au & peklund@itee.uq.edu.au

Abstract. The Semantic Web is usually envisaged as a collection of Web accessible RDF documents that re-use RDF schemas. These schemas are expected to be most often independently designed and hence not sharing many categories. We are unconvinced that this approach is viable because the lack of semantic relationships between the categories will most often make it impossible for future Web search engines to semantically compare RDF statements, and hence use them for logical inferencing or even permit their retrieval.

We believe a first requirement for a viable Semantic Web is to permit knowledge providers to use common vocabulary and representation means. This implies: (i) lexical, structural and ontological conventions; (ii) a high-level expressive notation guiding the knowledge representation process and restricting the ways things can be expressed (rather that what can be expressed); (iii) a rich ontology of knowledge represention primitives (or library of complementary ontologies); and (iv) a large ontology for natural language that knowledge providers can use and specialize to describe their domains. We have collected, complemented and integrated such conventions, notations and ontologies, and introduce them in this article. Such a set (not necessarily ours) needs to be recommended by the W3C (who else?) in order to be used and thus permit knowledge sharing.

Another step (not involving W3C intervention) is the development of large-scale knowledge base (KB) servers allowing users to retrieve, re-use, complement, annotate and be guided by other users' knowledge. An implementation of such a server, WebKB-2 (www.webkb.org), is described in this article. From an external viewpoint, WebKB-2 can be exploited as a large virtual document in RDF or other export formats. Mirroring techniques between KB servers can also be used; in this architecture no unique server is relied upon and the server where a Web user publishes information would be of no concern. Thus, this more centralized approach to the Semantic Web maintains the advantages of the expected "highly decentralized" approach while solving its problems.

Keywords: Semantic Web, Ontology, Ontology Server, Cooperation, Knowledge Representation/Retrieval/Engineering/Sharing/Re-use.

Approximate word count: 7950.

1 Introduction

The success of the Web is often attributed to its open decentralized architecture. But centralizing factors are also important: Domain Name servers, Web document indexes ("search engines"), standard protocols and languages.

The principles of the "decentralized" approach of the Web are to allow anyone to publish information, refer to any piece of information, and add new terms/categories to the languages. Hence, for the "Semantic Web"[1], RDF and RDFS[2] were designed to permit anyone to declare new terms in RDF schemas, describe certain semantic relations between them, invent new semantic relations, and write statements within RDF documents using terms from various schemas. A common expectation is that many small, specialized (and possibly competing) schemas will be developed, and that to make statements, individuals or businesses will select some schemas, import them, and create new schemas to define terms they have not found [4]. Then, according to Tim Berners-Lee[3], future Web search engines may be able to find various statements related to certain queries and manage to logically combine a few of them to answer the query; "while nothing will make the combinatorial explosion go away, many real life problems can be solved using just a few (say two) steps of inference out on the web".

We believe these expectations are unrealistic and also undesirable because, as this article is intended to show, a more centralized approach can offer much more without loosing any of the advantages of the "highly decentralized" approach.

Such expectations are *unrealistic* because it is unlikely that different people will create statements that can be *logically* matched or combined when they use unconnected or loosely connected schemas (ontologies), when only a few ontological primitives are standardized (those in RDFS and DAML+OIL[4]), and when no lexical/ontological/structural/semantic conventions are adopted. Like today, Web search engines would mostly have to rely on term *lexical* matching, and applications would have to write a special wrapper for each knowledge source they want to utilize (furthermore, even wrappers cannot compensate for badly

[1] http://www.w3.org/2001/sw/
[2] http://www.w3.org/RDF/
[3] http://www.w3.org/DesignIssues/Semantic.html
[4] http://www.daml.org/2001/03/daml+oil-index

structured or impoverished knowledge). Matching or combining statements[5], and hence finding knowledge relevant to a query (or even only "related" to a particular object) is a problem even in a single large knowledge base (KB) such as CYC[6] where knowledge providers are trained knowledge engineers following conventions, using a unique large ontology and an expressive knowledge representation language. Yet, even in this ideal case, some choices in the ontological and structural conventions have led to knowledge which is not explicit enough to be exploited by many applications[7].

The expectations are also *undesirable* because large-scale knowledge servers such as WebKB-2[8] [7] (designed by the authors and described in this article) can (i) permit a large number of users (people or agents) to cooperatively build large KBs with explicit, expressive, normalized, highly inter-connected statements and categories, and hence permit knowledge retrieval by simple hypertext navigation or provide reasoning services at selected levels of effectiveness and completeness, and (ii) exploit the KB to ease, guide and cross-check the insertion of new knowledge by each user and her re-use/annotation/correction of other users' knowledge. These features are permitted by the incremental insertion of knowledge into centralized repositories *when* it is developed, instead of *afterwards* by Web search engines (knowledge in isolation is not knowledge but merely data; hence, loosely connected schemas/RDF documents cannot be *logically* combined and their re-use requires the development of an ad-hoc wrapper for each one). To keep the advantages of the decentralized approach of the Web, categories and statements in a knowledge server must be referable via URLs (and then exported in a standard language such as RDF), and be allowed to refer to other objects on the Web via URLs. These are easy-to-achieve constraints.

---

5 For example, the statement "John is owner of a duplex in Southport" can easily be identified as a specialization of the query statement "a person is owner of an apartment in a city part of the Gold Coast" provided that "John" has been declared as a "person", "duplex" as a specialization of "apartment" and "Southport" as being a "city" which is "part" of the "Gold Coast".

6 http://www.cyc.com/tech.html#cycl

7 For example, actions/processes are represented as n-ary relations instead of concept nodes with explicit thematic relations to the related objects. As shown in Section 4, this decreases the possibility of matching or combining statements about processes.

8 WebKB-2 can be tested and used at www.webkb.org

For distribution reasons, all Web-users would not use the same knowledge server but rather a few general knowledge servers (e.g. managed by portal companies) and more specialized knowledge servers dedicated to specific domains. By (partly) mirroring one another's content, general/specialized servers would share a similar general ontology like WordNet[9] or CYC's ontology, and competing specialized knowledge servers would also share some similar content[10]. Thus, it would not matter where a Web user publishes information first, no unique server would have to be relied upon, and this "more centralized" approach maintains the advantages of the current decentralized approach, without its problems. (A similar architecture for distributed KBs and a small-scale implementation of it is discussed in CYC[11]).

In this paper we first show that *within a KB as well as across the Web*, knowledge sharing and exchange implies that knowledge providers use of a unique set of ontological primitives, follow lexical/structural/ontological recommendations, and use (directly or via interfaces) high-level expressive knowledge representation languages that ease the adoption of the recommendations and lead to comparable knowledge representations. Thus, we argue that the Semantic Web implies the standardization of such elements and show the elements adopted in WebKB-2. We then summarize the protocols used in WebKB-2 to permit the asynchronous cooperative building of the KB by the users and present some interfaces for its use. Finally, we compare our approach to others.

2 Need for a Standard Library of Ontological Primitives

RDF is not a particularly expressive language even with the semantic augmentations provided by the "standard" schemas RDFS and DAML+OIL. For example, we have not found any (non ad-hoc) way to represent simple sentences like "5 persons dance together"[12] or "51% of people are women" in RDF.

---

[9] http://www.cogsci.princeton.edu/~wn/

[10] The similarity of the KBs also permits the processes of mirroring and answering queries involving several KBs.

[11] http://www.cyc.com/applications.html#dai

[12] There is no "set" class nor "size" property/relation in RDF, RDFS or DAML+OIL. There is a "cardinality" property in DAML+OIL but it is about the number of relations that instances of a certain class can have. Representing "together" is also a problem since there

The lack of expressiveness of RDF and the absence of standard ontological primitives force knowledge providers to represent information in a biased or impoverished way or invent their own (mutually incompatible) extensions. Both cases make knowledge exploitation, sharing and re-use difficult.

Many formal specification languages such as Z[13] come with a mathematical toolkit, i.e. functions and relations related to the building blocks for knowledge representation: sets, relations, functions, numbers, sequences and bags. KIF[14], the best accepted knowledge exchange format, also comes with a similar toolkit. More precisely, KIF is a Lisp notation and a set of ontologies (e.g. one about sets, one about relations) that defines the language, its semantics and its use. Each ontology uses elements from other ontologies to define itself. The Ontolingua server[15] permits access to a repository of ontologies intended to permit knowledge sharing and translation between knowledge representation languages. This repository includes the ontologies about KIF plus many others, e.g. about physical dimensions and quantities, chemical elements, documents, languages (OKBC, CML), applications. Web users are allowed to add new ontologies.

A similar mathematical toolkit needs to be *standardized* in schemas such as RDFS to permit knowledge representation, sharing and exploitation. For example, RDF engines cannot provide an implementation for handling sets, general negation or universal quantification if a vocabulary is not fixed. (We recognize the DAML+OIL schema is a first step in that direction).

Usual in the XML/RDF/KIF worlds is that an inference engine is not obliged to take account of all the features of the language (i.e. all the categories in the "standard" schemas/ontologies) and perform all the logical deductions. What inferencing is done is an application-dependant choice[16]. Hence, the issues of completeness and decidability are *not* related to notations but to inference en-

is neither a way to represent that an universally quantified variable is within the scope of an existentially quantified variable, nor a special keyword to specify a "collective" interpretation for a collection (RDF only proposes the "distributive" and "cumulative" interpretations).

[13] http://spivey.oriel.ox.ac.uk/~mike/zrm/

[14] http://logic.stanford.edu/kif/dpans.html

[15] http://WWW-KSL-SVC.stanford.edu:5915/

[16] Not simply for efficiency reasons: the kinds of rules to apply (e.g. to handle modalities) can also sometimes *only* be chosen according to the application.

gines. Some knowledge exchange languages and APIs, such as KIF and OKBC[17], lists various levels of conformance. Alternatively, inferences engines may list the categories (concept types, relation types and individuals) to which they accord a special interpretation. They do not even have to exploit the definitions associated to these categories: they may implement some efficient ad-hoc exploitation of them (while the formal definitions permit the semantics of the categories to be specified and permit the programmer to know and delimit the kinds of deduction the implementation performs). Path retrieval techniques (based on structural matching and exploiting specialization links between categories) can be efficient *and* provide satisfying results for knowledge retrieval. For example, by treating a relation/property "not" as if it had no special meaning, WebKB-2 can retrieve the representations of "there is no duplex for rent in Southport" and "Southport is part of the Gold Coast" in answer to the (formalization of the) query "Is there an apartment for rent on the Gold Coast?". These are not "logic specializations" of the query but nonetheless relevant answers.

3 Need for High-level Expressive Notations

A problem for automatic knowledge retrieval and inferencing is that a same piece of information can be expressed in many different incomparable ways. This problem is particularly acute when a low-level general syntax such as KIF (LISP) or RDF (XML) is employed or when standard schemas offer partially redundant ontological primitives[18].

Some ways to represent information are more explicit, re-usable, comparable and easier-to-handle than other ones. Hence, to improve knowledge use and re-use possibilities: (i) *knowledge representation conventions (or "recommendations") should be standardized*; (ii) *high-level languages (or graphical interfaces) should guide the user and lead her to use the adopted conventions*. We have proposed a minimal set of lexical/structural/ontological recommendations in [6]. We give a summary of these in the next section. These recommendations are also usefully

[17] http://www.ai.sri.com/~okbc/

[18] For example, to represent an "xor" between two statements, one could think of using an RDF "alt" container, a DAML+OIL "disjointWith" relation (by creating an anonymous class for each of the statements) or a classic "xor" relation (e.g. KIF "xor" relation).

observed within a KB server. WebKB-2 users are asked to follow them and the high-level notations that we have designed – Frame-CG and Formalized English – encourage their adoption.

Frame-CG (FCG)[19] is a notation that we have derived from CGLF, the Conceptual Graph[20] linear form, to improve on its readability and expressivity (which was already one of the main reasons of the success of Conceptual Graphs). The three main improvements were: (i) the introduction of many kinds of quantifiers in the form of English articles or expressions (e.g. "many", "between 2 and 5", "at least 6.5%"); (ii) a notation inspired from frame languages to express relations between objects in a shorter and more natural way; and (iii) the convention that the scope and precedence of quantifiers in a graph (seen as a logic formula) are related to the graph structure and node order (as in predicate logic)[21].

Formalized English (FE) is identical to FCG apart from some syntactic sugar used for grouping and connecting objects[22].

To illustrate FCG and FE and compare them to the other cited languages, here is the representation of an English sentence in CGLF, FCG, FE, KIF, predicate logic (PL) and RDF/XML (the XML format for the RDF data model)[23]. Namespaces are omitted. "Ned" is assumed to be a declared identifier for an instance of the type "Person". The 's' at the end of "cars" and "sells" in the FCG and FE representations are automatically removed by WebKB-2 (since a universal-like quantifier is used with these categories).

```
E[24]:  Ned sold (the same) 3 cars twice on the 21/1/2001.
CGLF: [Person: Ned]←(agent)←[Sell: {*}@2]-
                                { →(object)→[Car: {*}@3 @certain];
                                  →(time)→[Date: #21/1/2001];  }
```

[19] Grammar in http://www.webkb.org/doc/F_languages.html#FCG

[20] http://www.cs.uah.edu/~delugach/CG/

[21] In CGLF, only contexts are important to determine the scope of quantifiers; otherwise, universal quantifiers are assumed to have wider scope than existential quantifiers except when the keyword "@certain" is associated to them; this convention leaves room to ambiguities.

[22] The model used by WebKB-2 for storing the graphs (statements) is an extension of the Conceptual Graph model. Like the RDF model and terminological logics, it is a logic-based semantic network and can be used to store logical statements.

[23] This RDF representation is only a tentative.

[24] This sentence does not specify whether the cars have been sold individually, 2 by 2, or 3 by 3. This ambiguity is kept in the representations.

```
FCG:  [3 cars, object of: (2 sells, agent: Ned, time: 21/1/2001)]
FE:   3 cars are object of 2 sells with agent Ned and time 21/1/2001.
KIF[25]:(forAllN 3 ?c car (forAllN 2 ?s sell
        (and (agent ?s Ned) (object ?s ?c) (time ?s '21/1/2001))))
```

PL:
$$\exists cars\ set(cars) \wedge size(cars, 3) \wedge \forall c \in cars$$
$$\exists sells\ set(sells) \wedge size(sells, 2) \wedge \forall s \in sells$$
$$agent(s, Ned) \wedge object(s, c) \wedge time(s, 21/1/2001)$$

```
RDF: <kif:Set ID="cars"><size>3</size></kif:Set>
     <rdf:Description aboutEach="#cars">
       <rdf:type resource="Car"/>
       <object><rdf:Description>
                 <kif:Set ID="sells"><size>2</size></kif:Set>
                 <rdf:Description aboutEach="#sell">
                    <agent resource="Ned"/> <time>21/1/2001</time>
                 </rdf:Description>
               </rdf:Description></object>
     </rdf:Description>
```

The need for higher-level (and more expressive) notations than RDF/XML is well recognized[26]. As "an academic excercise", Tim Berners-Lee has begun the design of Notation3[27], another notation for RDF which has some points in common with CGLF, FCG, FE and frame languages. (However, Notation3 does not (yet) have have any special syntax for extended quantifiers, collections, functions and definitions). Although Berners-Lee has not designed Notation3 "as an alternative to RDF's XML syntax which has the fundamental advantage that it is in XML", one may wonder what this advantage is supposed to be since he also acknowledges that most notations may be "web-ized"[28] by using URIs for category identifiers. Even if knowledge can be represented in XML, it is unlikely that XML objects are

[25] Here is our KIF definition for the "forAllN" quantifier:

```
(defrelation forAllN (?num ?var ?type ?predicate) :=
  (exists ((?s set)) (and (size ?s ?num)
    (truth ^(forall (,?var) (⇒ (member ,?var ,?s) (and (,?type ,?var) ,?predicate)))))))
```

[26] http://www.w3.org/DesignIssues/Logic.html

[27] http://www.w3.org/DesignIssues/Notation3.html

[28] http://www.w3.org/DesignIssues/RDFnot.html

directly used by advanced inference engines, and that knowledge providers read or write XML-based languages. Hence, translations to and from the XML world are necessary. From a purely syntactical viewpoint, the use of a Lisp-like notation (such as KIF) as a general low-level interlingua makes more sense because Lisp is concise and has adequate quotation (contextualization) features.

From any viewpoint we can think of, the use (and ideally, the standardization) of a high-level expressive notation would make even more sense since then knowledge is easier to write, read, compare, exchange and exploit[29]. Being readable and not XML-based, knowledge representations can also be mixed and hyperlinked with text and images within HTML/XML documents (our tools WebKB-1[5] and WebKB-2 exploit such documents)[30].

4 Need for Lexical/Structural/Ontological Conventions

Consider the statements "a person is doing something" and "Ned is selling a car" and their FCG representations `[a person, agent of: an activity]` and `[Ned, agent of: (a sell, object: a car)]`. The second graph is a "specialization" of the first, i.e. it has more information in its structure (one more relation) and in its components ("Ned" is an instance of the type "person" and "sell" is a subtype of "activity"). Therefore, since only existential quantifiers are involved in those graphs, the second logically entails the first[31]. In other words, if the first is used as a query graph, the second is a logical answer.

Similarly, the second graph can also be seen as a specialization of the FCG graph given in the previous example but, since it involves universal quantifiers, there is no logical entailment relation between the two graphs. Hence, we simply say the graphs are "comparable" (in the same way that two categories are comparable if they are linked by an subtype link or an instance link).

[29] Let us stress again that a high-level expressive language such as FCG or FE is not intended to limit what the knowledge provider can express but how she express it, and furthermore its expressiveness does not impose constraints on what inference engines must do.

[30] WebKB-1 is a knowledge-based private-only annotation tool, as opposed to WebKB-2, a knowledge-based shared annotation tool that can also exploit private knowledge annotations, i.e. knowledge representations within Web documents and their links to other documents elements on the Web.

[31] For more details and a mathematical proof, see [1].

Now, suppose that a user declares a relation type ("property" in RDF) "sell" to represent the information "A person sells a car" via 2 nodes linked by a relation; in FCG: `[a person, sell: a car]`. This graph leaves the "agent" and "object" relations implicit and is not comparable to any of the previous graphs. The user could associate a definition to the relation type "sell" to permit the expansion of the previous graph to: `[a person, agent of: (a sell, object: a car)]` but such an expansion can be a complex process and few inference engines perform it. The relation type "sell" cannot be re-used when other relationships (such as "time" or "purpose") have to be represented, and would be incomparable with other relation types "sell2" and "sell3" used to represent these relationships. Furthermore, relations cannot be quantified. In summary, the use of relations other than basic binary relations should be avoided because it leads to representations that are less explicit and comparable. Even if a Web-based knowledge-oriented information retrieval engine does some lexical matching on category names to complement structural/semantic matching, concept types "sell" are more likely to be used in unrelated KBs (if basic binary relations are used) than relation types such as "sell2" or "sellSomethingAtSomeTime" (these kinds of identifiers are quite typical when relational/functional syntaxes such as Lisp are used).

As opposed to concept types, there is not a great number of basic binary relation types needed to represent natural language. For example, WebKB-2 has about 74,500 concept types derived from the WordNet lexical database about nouns, but it has a stable ontology of only 140 relation types and 50 of these types appeared sufficient to us for representing most usual natural languages sentences. Basic binary relation types are an efficient way to guide and normalize the knowledge representation task. Thanks to the signatures associated with these relation types, an inference engine can easily perform some elementary semantic checking and propose corrections when signatures are violated.

Because of its Lisp-like syntax, KIF does not encourage the use of basic binary relations only. Like most frame-based or graph-based languages, RDF only accepts binary relations but its cumbersome syntax discourages knowledge providers to be precise. For the same reasons, KIF and RDF discourage the use of adequate quantification, and do not prevent the use of verbs, adverbs, and adjectives as category identifiers/names even though such categories cannot be quantified (e.g. "any qualify" and "3 qualified" are meaningless), can rarely be compared to other categories, and leave information implicit. Thus, to permit knowledge sharing, lexical/structural/ontological conventions are required, and their observance needs to be encouraged by high-level notations.

RDF/RDFS and the "Meta Content Framework Using XML"[32] have some "naming conventions" for category identifiers: words used should be singular, with a lowercase first letter for relation types and an uppercase first letter for other kinds of categories, and the intercap style should be adopted when the identifier is composed of several words. Using names in the singular is a sound convention because categories can then be quantified in various ways (whereas for example a category "cars" cannot be used in a universally quantified node and is not comparable to "car"). However, because of the last two conventions, the correct cases in the words may be lost and, at least in English, there is no way to recover that information. Readable and correctly spelled category identifiers are needed when using the identifiers in menus or presenting information with languages such as Formalized English (FE). In RDF, category names can be specified in addition to the category identifier. This is a cumbersome, rarely used feature, and most other knowledge representation languages do not have it. From the developer's viewpoint, using category names (instead of identifiers) introduces possible ambiguities and requires additional handling.

Hence, a summary of a minimal set of conventions that we advocate is:

- *lexical conventions*. Whenever possible, use a correctly written English singular noun for a category identifier. Separation between words is to be done with underscores (dashes and quotes may be used when part of the usual spelling of words, e.g. "Niemann-Pick_disease" and "Fallot's_tetralogy").
- *structural/ontological conventions*. Only use binary basic relations and respect reading conventions[33]. Whenever possible, use or specialize categories from standard ontologies and use the least expressive ontological primitives: try to

[32] http://www.w3.org/TR/NOTE-MCF-XML/#secA.

[33] Most semantic networks models (including RDF) have adopted the convention that a relation "R" from a node "A" to a node "B" should be read "the R of A is B" or "A has for R B". In models where relations can be of any arity (e.g. KIF) no such convention is generally advocated, resulting to various usages and interpretation problems. For example, in the ontologies of KIF, the relations "subset" and "member" counter-intuitively have the source set as a second argument instead of as the first.

avoid general negation, disjunctions, second-order statements, collections, etc. In the RDF context, this amounts to using RDF and DAML+OIL ontological primitives whenever possible. Within WebKB-2, this amounts to selecting categories of the shared ontology and then following the menus or using the "For Ontology" (FO) notation for links between categories and FCG/FE for the other statements. Via these notations and the ontology (relation types, general schemas/templates, etc.), the WebKB-2 user is guided to represent most things in a normalized way: states, processes, descriptions, indexations, characteristics, measures, numbers, collections, temporal/spatial/logical entities/relations, etc.
- *semantic conventions.* Be as precise as possible: give adequate quantifiers, contextualize statements in time, space, authorship, etc. Re-use and complement existing knowledge. To enforce this in WebKB-2, (i) a category can only be declared by connecting it with another and it must have at least one generalization or specialization, and (ii) a statement cannot be entered if it contradicts or is directly comparable to an existing statement, unless the author asserts the relationships between the two statements.

5 Need for flexible ways to refer to a category

In RDF, a category is uniquely identified by a URI, e.g. `http://www.foo.com` and `http://www.bar.com/doc.html#car`. Within a multi-user KB server, *it makes more sense to use user identifiers than document URIs as knowledge source identifiers.* Thus, in WebKB-2, a category identifier can be a URI (or an e-mail address) but also the concatenation of the knowledge provider's identifier and a key name, e.g. `wn#dog`, `wn#time`, `pm#IR_system` ("wn" refers to WordNet 1.7 and "pm" is the login name of the user represented by the category `philippe.martin@gu.edu.au`). In this third case, the category may still be referenced from outside the KB by prefixing the identifier with the URL of the KB, e.g. `http://www.webkb.org/kb/wn#time`. This method is used when knowledge is exported in RDF/XML.

In addition to an identifier, a category may have various names (which may also be names of other categories). In FE, FCG and FO, a category identifier may show several names, e.g. `wn#dog__domestic_dog__Canis_familiaris` (at least 2 underscores must be used for separating the names). Given 95% of current categories in WebKB-2 come from WordNet, the “wn” prefix may be left implicit, e.g. `#time` means `wn#time`. More precisely, “wn” is the default creator. An ordered list of default creators can be specified, e.g. “`default creators: pm wn;`”.

Below is the way the FO notation can be used in WebKB-2 to store that the uppermost concept type has been created on the 29/11/1999, given two names by its creator “pm”, that the user “oc” has added a French name and an “instanceOf” link to the RDF “class” category, that “pm” has added a disjointWith link to the uppermost relation type (the link creator is left implicit since it is the same as creator of the source category) and given 3 subtypes, 2 of which forming a close partition (or “disjoint union” in DAML terminology).

```
pm#thing__top_concept_type (^thing that is not a relation^) 29/11/1999
  _  chose (oc fr),   ^ rdfs#class (oc),  !  pm#relation,
  >  {(pm#situation pm#entity)} pm#thing_playing_some_role;
```

Here is a partial translation in RDF/XML. The creators of the links could not be represented in a standard/simple way.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
         xmlns:daml="http://www.daml.org/2000/10/daml-ont#"
         xmlns:pm="http://www.webkb.org/kb/theKB_terms.rdf/pm#">
<rdfs:Class rdf:about="http://www.webkb.org/kb/theKB_terms.rdf/pm#Thing">
  <rdfs:label xml:lang="en">thing</rdfs:label>
  <rdfs:label xml:lang="en">top_concept_type</rdfs:label>
  <rdfs:label xml:lang="fr">chose</rdfs:label>
  <dc:Creator>philippe.martin@gu.edu.au</dc:Creator>
  <rdfs:comment>thing that is not a relation</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <daml:disjointWith rdf:resource="http://www.webkb.org/kb/theKB_terms.rdf/pm#relation"/>
</rdfs:Class> </rdf:RDF>
```

Here is how the FO notation was used by “pm” to declare a category for the “instanceOf” relation, specify the equivalent RDF category and an inverse relation.

```
pm#kind__type__class (pm#thing,rdfs#class)
  = rdf#type,
  < dc#type,
  - pm#instance;
```

Here is a partial translation in RDF/XML using the previous namespaces.

```
<rdf:Property rdf:about="http://www.webkb.org/kb/theKB_terms.rdf/pm#kind">
  <rdfs:label xml:lang="en">kind</rdfs:label>
  <rdfs:label xml:lang="en">type</rdfs:label>
  <rdfs:label xml:lang="en">class</rdfs:label>
  <dc:Creator>philippe.martin@gu.edu.au</dc:Creator>
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <daml:samePropertyAs rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Type"/>
  <rdfs:subPropertyOf rdf:resource="http://purl.org/metadata/dublin_core#type"/>
  <daml:inverseOf rdf:resource="http://www.webkb.org/kb/theKB_terms.rdf/pm#instance"/>
</rdf:Property>
```

More details on our top-level ontology and how it integrates other top-level ontologies can be found at the WebKB-2 site (www.webkb.org).

WebKB-2 maintains links between the links between each category and its creator and names, and conversely. *This permits the use of names instead of identifiers within statements as long as there is no ambiguity.* Relation signatures are exploited to eliminate candidate categories[34]. If there is more than one candidate for a category, the parsing stops and the list of candidates is printed to help the user refine the statement. For a query graph, there is no harm in making this choice automatically and allowing the user to refine the query when an incorrect category has been selected. For improved readability, we often use names instead of category identifiers in the example graphs of this article.

A problem that prevents this facility to be adopted within RDF documents on the Web is that the RDF schemas they import may change (new names may be added to categories) and hence ambiguities may appear.

Within a KB that integrates a natural language ontology, this facility is particularly useful to accelerate the writing of knowledge.

[34] For example, "flight" is a name currently shared by 9 categories: 4 representing processes, 3 representing collections, 1 representing a psychological feature, and 1 representing a physical entity ("flight of stairs"). If a concept node is about a "flight" and is the destination of a relation with type pm#on_location, given the signature associated to pm#on_location, only one sense of "flight" is relevant, the one representing the physical entity "flight of stairs".

6 Need for a Shared Natural Language Ontology

Links from a natural language ontology such as WordNet[35] form the backbone of a large shared KB. Such links permit WebKB-2 to relate, compare and retrieve knowledge representations. They also provide the user with various categories (meanings) for a word, and various distinctions for a notion, many of which she may not have considered. This leads the user to enter more precise and comparable representations. The semantic constraints associated with the top level categories of the ontology are inherited by all the categories of the natural language ontology, and this permits some automatic checking on all users' statements and extensions to the ontology.

We initialized the current KB of WebKB-2 with the content of the lexical database WordNet 1.7: 108,000 nouns and 74,500 categories referred by nouns (in accordance with our lexical conventions, we ignored information regarding verbs, adverbs and adjectives). Various kinds of links connect these categories: `specialization`, `exclusion`, `similar`, `member`, `part`, `substance`, and their inverse links. We distinguished the Wordnet `specialization` links into `subtype` links and `instance` links, and made a few other structural corrections.

According to Tim Berners-Lee[36], "many KR systems had a problem merging or interrelating two separate knowledge bases, as the model was that any concept had one and only one place in a tree of knowledge ... The RDF world, by contrast is designed for this in mind, ...". Although RDF schemas may import other RDF schemas and RDF documents may import various RDF schemas, in order to compare two statements from different RDF documents, an RDF engine has to classify the categories used in these statements into a *unique* specialization hierarchy[37]. This is most often impossible (unless the two documents mostly re-use the same schemas) because of the disconnected specialization hierarchies (and hence insufficient information to compare the categories and statements). What are currently called "ontology-merging techniques", are only semi-automatic algorithms heuristically matching categories based on their names, links to other

[35] http://www.cogsci.princeton.edu/~wn/
[36] http://www.w3.org/DesignIssues/RDFnot.html
[37] Not simply a tree since a category may have several parents.

categories[38], and sometimes other properties such as their frequency of occurrence in documents[8].

From the knowledge provider's view, re-using distributed RDF schemas is also a difficult and sub-optimal task. First, she must find schemas on the Web with categories similar to the ones she wants to use, then select some schemas that are not mutually inconsistent and write another schema to define the categories she has not found. Tools exploiting distributed schemas cannot provide guidance nor much cross-checking since they do not have a large ontology to exploit. In WebKB-2, thanks to the initialization of the KB with WordNet, the user enters a word and is presented with the categories that represent its various meanings, generalizations and specializations. She can select one category or find a more appropriate category by navigating along semantic links. When a new category is required, the user can add it by connecting the new category to an existing category via a link of a selected type. Since the new category is added to a large and tightly interconnected ontology, it can be accessed and exploited in many ways. With distributed schemas, to achieve a similar level of connectedness, each schema creator would have to check that there is a relation between each of her categories and all relevant categories in all other existing schemas on the Web.

There is an intermediate way between the highly decentralized approach advocated by the W3C and the approach we have adopted. That is to develop RDF schemas/documents by re-using (importing) ontologies of large KB servers such as WebKB-2. Then, tools could provide some guidance and cross-checking, and do a relatively good job at integrating these schemas/documents even when developed separately since they would at least be based on the same large natural language ontology. WebKB-2 permits its categories or parts of its ontology to be referred and accessed via URLs and can import knowledge from Web documents into its shared KB, permanently or for testing puposes. However, the constraints are the same as when knowledge is entered manually, and an import is rejected if a problem is encountered. Future Web search engines will have to be more permissive.

[38] See Chimaera at http://www.ksl.Stanford.EDU/software/chimaera/

7 Mechanisms for cooperative editing of a shared KB

The WebKB-2 user is asked to be as precise as possible when making statements in order to avoid conflicts in the KB and permit to answer queries more adequately. For instance, a user (say "user1") should not simply represent that "birds fly" (i.e., in FCG: `[user1#birdsFly [any bird, agent of: a flight]]`) since this is not always true. If this happens, other users are encouraged to "correct" this representation using a relation of type `pm#corrective_specialization` (then, depending on display options, the first version may be filtered by WebKB-2 when responding to queries). Similarly, if a user thinks a statement from another user can be generalized, she can use a relation of type `pm#corrective_generalization`. For example, if "user1" stated that "birds fly" and "user2" wants to correct and specialize that by "a study made by Dr Foo found that in 1999, 93% of healthy birds could fly", she can write:

```
[user1#birdsFly, corrective_specialization:
   [user2#93pcHealthyBirdsCanFlyAccordingToFoo
        [ [93% of (bird, experiencer of: a good health),
             agent of #: a flying  //"#:" means "can"
          ], time: 1999], source: (a study, author: Foo@bird.org)]
   ]]
```

(Note: if a graph is not explicitly named, WebKB-2 generates a name for it).

Removal/modification/addition protocols are also required for semantic conflicts to be managed asynchronously and without person-to-person agreement. The following four points describe our approach.

1) A user may **remove a category, link or graph** only if she has created it and unless this removal induces an inconsistency in the user's knowledge. If the category, link or graph being removed is used by other users or is necessary for their knowledge to remain consistent, it is actually not removed from the KB but its ownership changed to one of the users relying on its existence. Inconsistency detection in WebKB-2 currently only exploits relation signatures, exclusion links and specialization links. However, we plan to exploit inconsistencies signaled by users with a relation of type `pm#contradiction` between two graphs.

2) The creator of a category may **modify a link** connected to this category – so that the link uses an alternate category – unless this modification

itself induces an inconsistency. The creator of a relation type may modify its signature unless such change induces an inconsistency (in which case, she must first modify the ontology or related graphs so that the inconsistency disappears). A user may **not modify a graph** that she has not created but she can connect it to another graph via a relation of type `pm#corrective_specialization`, `pm#overriding_specialization`, `pm#corrective_generalization` or, if none of the previous ones apply, `pm#correction`. This last relation type should also only be used if the ontology cannot be modified to correct the first graph. Since graphs can be used for representing links these three relation types may also be used by a user to "correct" links from users in the ontology. Depending on display/filtering options, corrected graphs or links may be displayed/used for inference or not.

3) A user may **add a graph or a link** (even if she is not the creator of the linked categories) unless this addition introduces an inconsistency or redundancy. – for consistency and re-use purposes, WebKB-2 does not accept a graph that already has a specialization or a generalization in the KB; an exception is when the graph is an "instantiation" of an already existing graph (more details can be found in [7]). When this happens, the user must either refine her graph before trying to re-add it, modify the ontology or use one of the four "corrective" relations cited above.

4) In any of these previous cases, when the knowledge of a user is modified by another user, the change should automatically be e-mailed to the first user or presented the next time she logs onto WebKB-2.

8 Guiding Search and Updates via generated interfaces

WebKB-2 is intended to be useful to knowledge engineers (or software agents exploiting knowledge) but *also* to be usable by average Web users. Although the first group requires various options to search, filter and browse the ontology and statements, an average Web user only needs to find the right category for the object she has in mind; she should not have to update the ontology apart from sometimes introducing a new category simply by giving it a type or a supertype. Both novices and experts need guidance when entering statements to ease the task and permit the production of explicit and comparable statements.

Fig. 1 shows the interface for knowledge engineers to search categories or links. It proposes various selection options and format options (recursive exploration, language, hyperlinking). The counterpart of this interface for average users is a simple text field (to enter a word, regular expression or directly a category identifier); it is proposed in the WebKB home page. Fig. 2 and Fig. 3 show the result of the query in Fig. 1, i.e. a search for categories with the name "person".

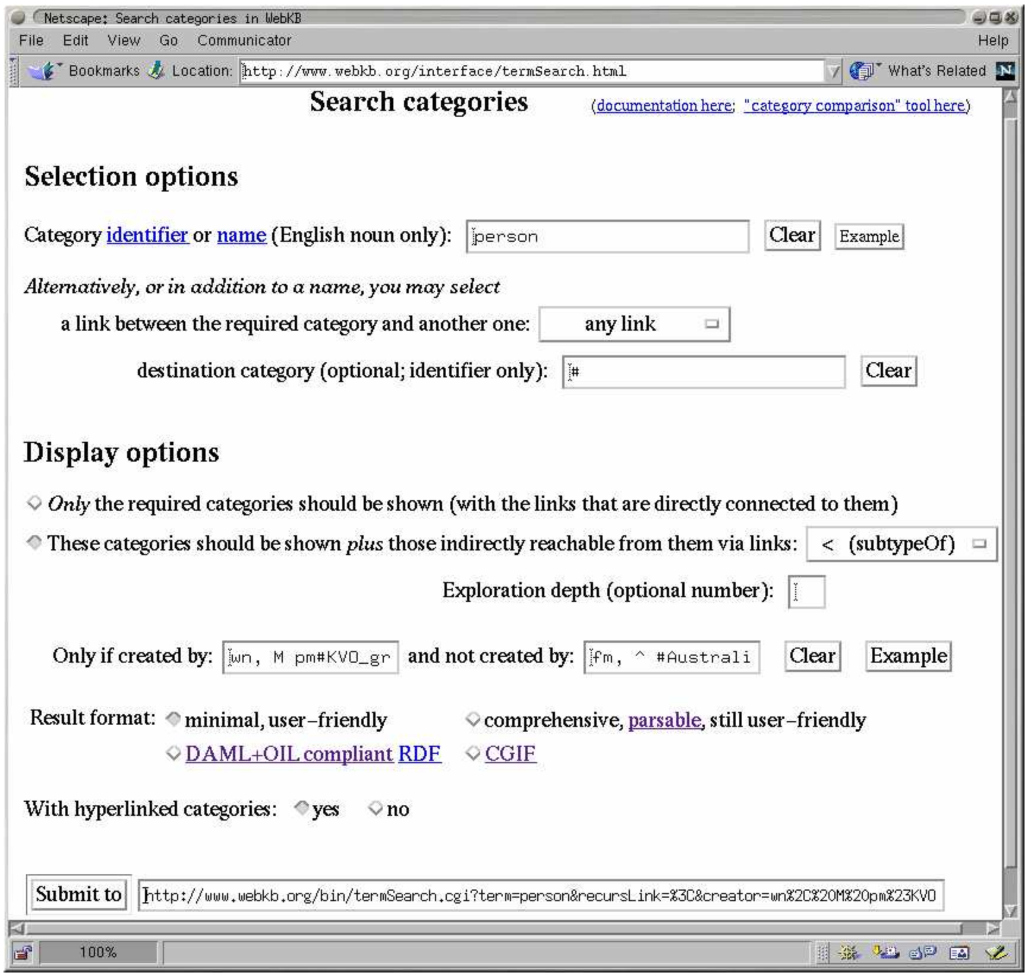

Fig. 1. Query for links and graphs related to #person and created by WordNet (wn) or a member of KVO (M pm#KVO_group) but not by F. Modave (fm) nor an Australian (^ #Australian); subtypeOf links must be recursively explored.

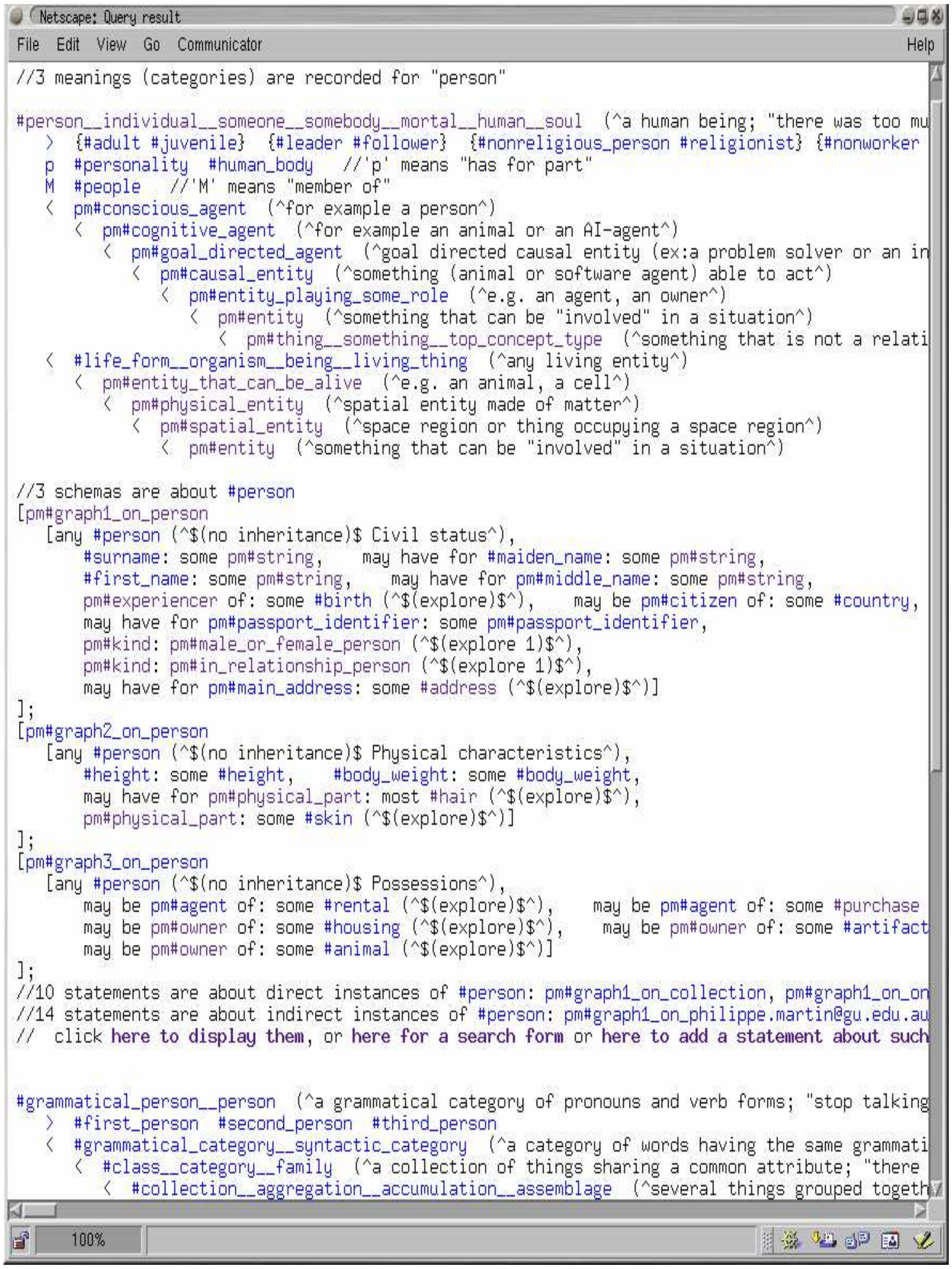

Fig. 2. Result of the previous query (Fig. 1).

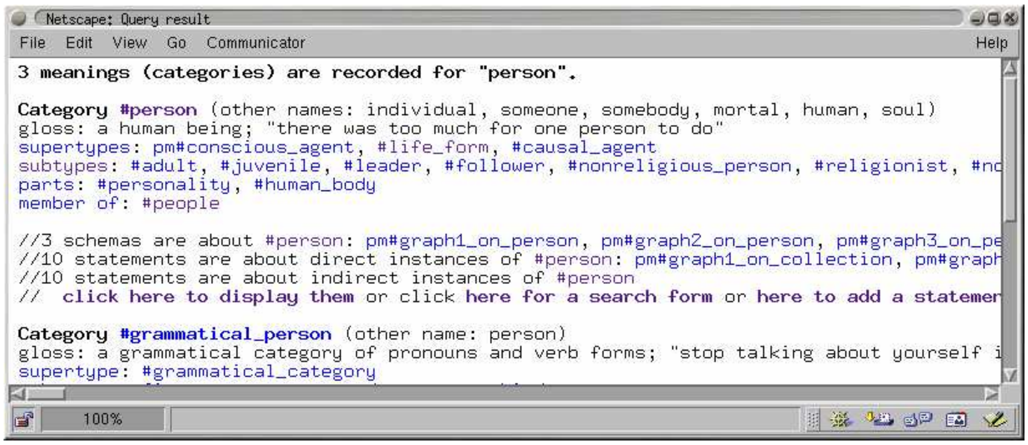

Fig. 3. Result of the previous query (Fig. 1) for "novice users".

Fig. 2 and Fig. 3 show that graphs directly or indirectly using a category are accessible from this category (or a confirmation that no graph uses this category). Each category identifier (even when shown within a graph) is displayed hyperlinked to permit access to its related links and graphs. Most link identifiers are also hyperlinked to ease exploration of the KB. Hyperlinks to search/add forms are also given (e.g. see "click here for a search form" in Fig 3).

These forms are generated based on the schemas (general statements) associated to the category or its supertypes. Fig. 4 shows the form generated to guide the addition of a statement about a new or already registered user. The three schemas exploited for this purpose are shown in Fig. 2. The directives `$(no inheritance)$` and `$(explore)$` stored in the concept node annotations control the generation of the form. The first directive prevents the use of schemas associated to supertypes of the category. The second leads to the generation of an hyperlink to another form to detail a related object. In other words, this second directive permits the re-use of schemas related to related objects to enable form cascading. Fig. 5 illustrates such a cascade. `$(explore)$` is also used to control the depth of menus generated using subtype partitions (e.g. the categories for colors and for days of the week are organised into hierarchies of subtype partitions; such partitions permit WebKB-2 to generate organized menus and filter categories likely to be less relevant).

These forms guide and ease knowledge capture. Since they normalize knowledge capture, they also lead to more comparable statements. At present, schemas in WebKB-2 are mostly associated to top-level concept types (e.g. `pm#situation`, `pm#description` and `pm#physical_entity`). These schemas are inherited by all types in the ontology that have no overriding schemas. They include the most useful relations from a certain object, permitting the user to ignore less precise relation types imported from other ontologies and relation types with structural purpose only (e.g. `pm#relation_from_spatial_entity`). As Fig. 4 shows, each form also has a field to permit the use of relation types not listed in the form.

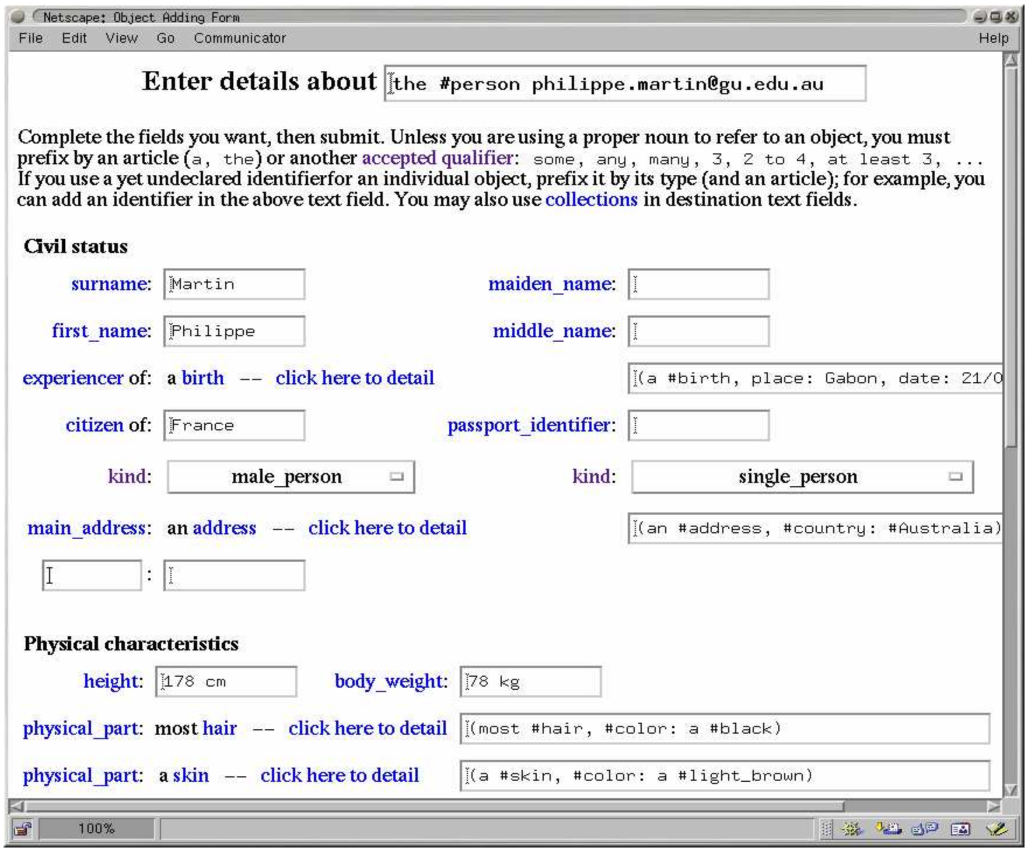

Fig. 4. A generated form to enter a statement about a new/existing person. The schemas shown in Fig. 2 are used to generated it. The knowledge provider must enter its identifier and password at the end of the form.

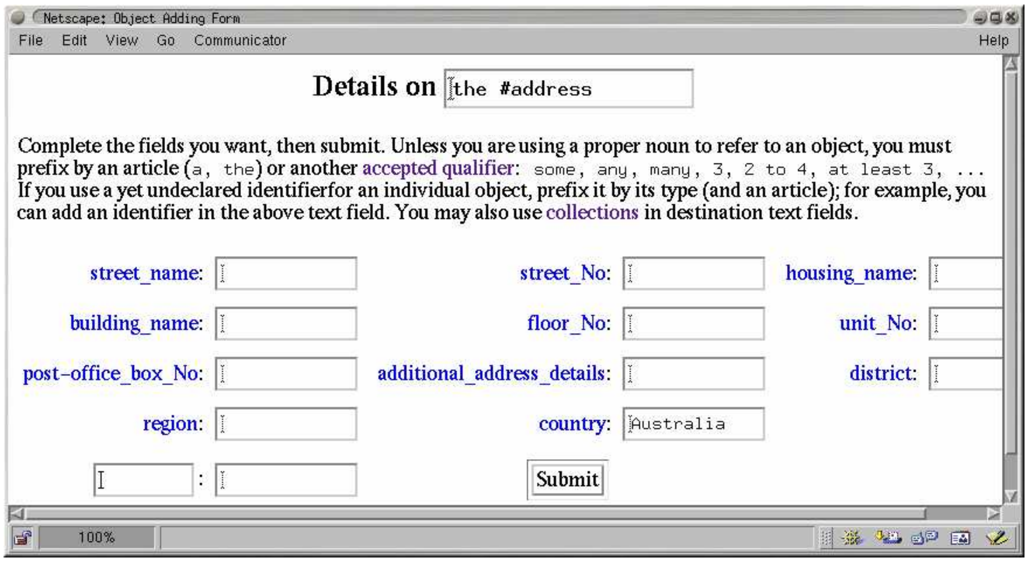

Fig. 5. Form called from the form in Fig. 4 to enter information about an address.

To guide and facilitate the representation of knowledge by average users, many specialized schemas are also required, e.g. for "house", "car", "selling", "renting", etc. Users may also associate schemas to any category: a schema is simply a statement that uses a general quantifier ("any", "most", "20%", ...) in the first concept node.

When a form is submitted, the WebKB-2 server generates a graph with the information (see Fig. 6). If this graph does not violate the syntax/semantic/cooperation rules, and if all the category names it contains can be unambiguously resolved to category identifiers, it is entered into the KB. The creation date and the graph identifier are automatically generated and added to the graph.

Search forms are similar to knowledge capture forms above except that the generated command is not a graph assertion but a query for specializations of the graph.

Graphs may also be searched via more generic interfaces: WebKB-2 can search for graphs that specialize or are comparable (depending on the selected command) to the query graph. If a retrieved graph is contained in a bigger graph which contextualizes it, the entire graph is presented. We plan to extend our graph retrieval

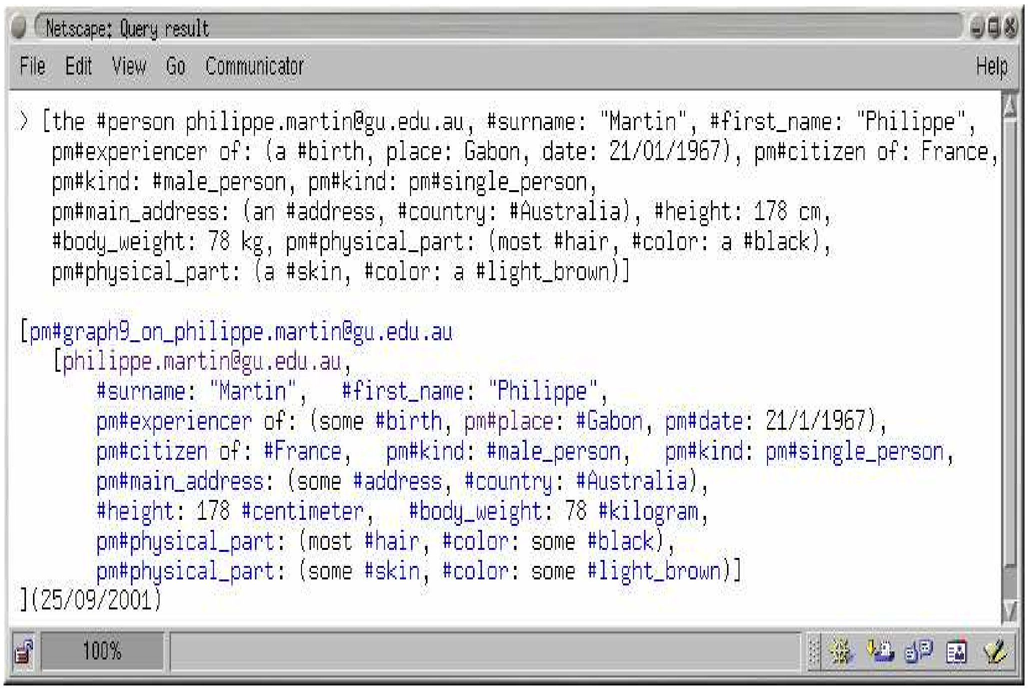

Fig. 6. Command (in FCG) generated when the form of Fig. 4 is submitted.

mechanism and query notation to permit the retrieval of "paths", i.e. graphs that can be traversed to answer a path specification (e.g. "What are the flights from Brisbane to London, departing on a week-end?"). Even if the query is not a path specification, the ability to traverse graphs to answer it is important since the information for an answer may not be stored in a single graph[7].

9 Comparison with other tools

Guarino *et al.* [3] have developed an information retrieval system called Ontoseek that exploits the WordNet lexical database and simple existential conceptual graphs to store the content of Yellow-Pages like catalogs and permits access in a flexible way. It is unclear from [3][39] whether or not users can modify this ontology but they apparently can enter simple existential conceptual graphs via the inter-

[39] We have not received any confirmation from the authors and Ontoseek is not Web-accessible.

face or ask/tell communication protocols. Classic "queries for specializations" may be performed and a query may use names instead of categories.

Both WebKB-1 and WebKB-2 can be called "ontology servers", i.e. Web servers that permit users to build and publish ontologies. Most ontology servers also permit the construction of existential graphs and therefore could also be called "knowledge base servers" but the possibility of modifying the ontology is a rarer feature. WebKB-1 and WebKB-2 are two opposite extremes in the handling of cooperation between users: while most other ontology servers (e.g. the Ontolingua ontology server[40], Ontosaurus[41], Ikarus[42], Tadzebao and WebOnto[43]) store users' knowledge in loosely related modules/files on the server disk, WebKB-1 uses Web-accessible files stored by users on their own local disks and WebKB-2 stores users' knowledge in a single KB on the server disk. Some ontology servers, e.g. the Ontolingua server or Ontosaurus, permit either any user or a restricted group of users to edit the module but, apart from locking/session mechanisms, no particular support for asynchronous cooperation[44] is generally provided: no record of creators for categories/links/graphs, no conventions, no protocols, etc. An exception is the Co4 system[45] which has protocols modeled on submission procedures for academic journals, i.e. on peer-reviewing, resulting in a hierarchy of KBs, the uppermost containing the most consensual knowledge while the lowermost KBs are the KBs of contributing users. This approach leverages some problems of module-based approaches but would doubtfully scale to large KBs or to a large number of users.

Ontoloom/Powerloom[46] relies on comparison procedures and the pre-existence of a large ontology to guide and check users in their extension of a unique KB. There does not seem to be any particular support for cooperation between users nor the possibility to filter certain users' knowledge. This is also the case for

[40] http://WWW-KSL-SVC.stanford.edu:5915/

[41] http://www.isi.edu/isd/ontosaurus.html

[42] http://www.csi.uottawa.ca/~kavanagh/Ikarus/IkarusInfo.html

[43] http://ksi.cpsc.ucalgary.ca:80/KAW/KAW98/domingue/

[44] Some ontology servers support synchronous cooperation, i.e. enable co-temporal users of a same file/module to send messages to each other or be warned of other users' actions. WebKB-2 only supports asynchronous knowledge-centered cooperation.

[45] http://ksi.cpsc.ucalgary.ca/KAW/KAW96/euzenat/euzenat96b.htm

[46] http://www.isi.edu/isd/OntoLoom/hpkb/OntoLoom.html#RTFToC18

Ontobroker [2], an ontology server which (until 2000) permitted registered Web-users to query its KB and add statements to it directly or by annotating their Web pages. However, since the ontology was very small (a few dozens categories mainly about research domains and researcher/student levels) and could only be updated by the KB managers, the users could actually not *represent* their research or anything else. In other words, Ontobroker (claimed by its authors to be the "1st Semantic Web server") was mainly *used as* a small database server.

Compared to other large scale KBMSs, a notable feature of WebKB-2 is that the ontology is large and can be dynamically/interactively modified by the users (no lengthy re-compilation phase or graph re-indexing is necessary). This feature is shared by the Parka-DB system[47].

10 Conclusion

This paper has presented elements helping the realization of the goals of the Semantic Web, i.e. "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation"[48].

First, the paper listed elements needed for knowledge sharing within a KB as well as across the Web: a library of ontological primitives, an ontology of natural language, lexical/structural/ontological conventions, and high-level expressive notations supporting them. Then, it was shown that knowledge-based servers could further ease the knowledge representation task, improve cooperation between knowledge providers, knowledge retrieval and re-use.

Three paradigms have been stressed: (i) a more centralized approched can be adopted to solve the problems of the "highly distributed" approach without loosing any of its advantages, (ii) as far as knowledge is concerned, "the more the better", be it the number of conventions, the size of standard ontologies, the size of the KB, the expressivity of the languages and the precision of the representations (in all cases except the first, some information can easily be ignored or automatically filtered when not needed), (iii) global approaches (i.e. module/file

[47] http://www.cs.umd.edu/projects/plus/Parka/parka-db.html
[48] http://www.w3.org/2001/sw/

based) are more coarse-grained than local approaches (inter-connections between elements) and hence less precise/explicit and flexible when complexity grows.

Three goals also describe the presented aproaches: ease of representation, scalability and use/re-use possibilities. These goals converge. Knowledge capture is a well recognized bottleneck, and knowledge use/re-use both a goal and a method.

Entering information in WebKB-2 is more difficult than entering sentences in a document, but information from documents cannot be interconnected to respond to precise queries. We believe entering information in WebKB-2 is easier than in most other systems thanks to our ontologies, notations and features (generated menus, the possibility to use everyday words instead of category identifiers, etc.). Some kinds of information remain difficult to represent precisely but we think that WebKB-2, or some evolution of it, can be used by Yellow-Pages-like-services or community servers to allow people to advertize products and services or, more generally, publish information.

Acknowledgments

This work is supported by a research grant from the Distributed Systems Technology Centre.

References

1. M. Chein and M.L. Mugnier, "Positive Nested Conceptual Graphs," *Proc. 5th Int'l Conf. on Conceptual Structures* (ICCS 97), Springer Verlag, LNAI 1257, 1997, pp. 95–109.
2. Fensel, D., Decker, S., Erdmann, M., Studer, R.: Ontobroker: Or How to Enable Intelligent Access to the WWW. In: Proc. 11th Knowledge Acquisition Workshop (KAW98), Banff, Canada, April 1998, 8–23.
ftp://ftp.aifb.uni-karlsruhe.de/pub/mike/dfe/paper/OB.KAW.ps
3. N. Guarino, C., Masolo and G. Vetere, "Ontoseek: Content-based Access to the Web," *IEEE Intelligent Systems*, Vol. 14, No. 3, 1999, pp. 70–80.
4. J. Hendler, "Agent and the Semantic Web," *IEEE Intelligent Systems*, Vol. 16, No. 2, 2001, pp. 30–37.
5. P. Martin and P. Eklund, "Embedding Knowledge in Web Documents," *Proc. of the 8th Int'l World Wide Web Conference* (WWW8), Toronto, Canada (1999). http://www.webkb.org/doc/papers/www8/www8.ps

6. P. Martin and P. Eklund, "Conventions for Knowledge Representation via RDF," *Proc. of WebNet 2000*, ACCE press, pp. 378–383. http://www.webkb.org/doc/papers/webnet00/
7. P. Martin and P. Eklund, "Large-scale cooperatively-built heterogeneous KBs," *Proc. 9th Int'l Conf. on Conceptual Structures* (ICCS 01), Springer Verlag, LNAI 2120, 2001, pp. 231–244. http://www.webkb.org/doc/papers/iccs01/iccs01.pdf
8. G. Stumme and A. Maedche, "FCA-Merge: A Bottom-Up Approach for Merging Ontologies," *Proc. 5th Int'l Joint Conference on Artificial Intelligence* (IJCAI 01), Morgen Kaufmann, Seattle, USA, August, 1-6, 2001.