CONVERTING CLASSIFICATIONS INTO OWL ONTOLOGIES

Fausto Giunchiglia, Ilya Zaihrayeu, and Feroz Farazi

May 2008

Technical Report # DISI-08-027

Also: In the proceedings of Artificial Intelligence and Simulation of Behaviour Convention - 2009 Workshop on Matching and Meaning, Edinburgh, UK, 2009

Converting Classifications into OWL Ontologies

Fausto Giunchiglia, Ilya Zaihrayeu, and Feroz Farazi

Department of Information Engineering and Computer Science
University of Trento, Italy
{fausto, ilya, farazi}@disi.unitn.it

Abstract. Classification schemes, such as the DMoZ web directory, provide a convenient and intuitive way for humans to access classified contents. While being easy to be dealt with for humans, classification schemes remain hard to be reasoned about by automated software agents. Among other things, this hardness is conditioned by the ambiguous nature of the natural language used to describe classification categories. In this paper we describe how classification schemes can be converted into OWL ontologies, thus enabling reasoning on them by Semantic Web applications. The proposed solution is based on a two phase approach in which category names are first encoded in a concept language and then, together with the structure of the classification scheme, are converted into an OWL ontology. We demonstrate the practical applicability of our approach by showing how the results of reasoning on these OWL ontologies can help improve the organization and use of web directories.

1 Introduction

A *classification scheme*, or a *classification* for short, is a rooted tree whose nodes are assigned natural language labels and are populated with a (possibly empty) set of documents. Since the invention of classification by Aristotle in the 4th century BC, classifications have been used (and are still used) pervasively to represent various kinds of human knowledge. For example, classifications have been used in libraries (DDC[1], LCC[2] and Colon classification[3]); in Personal Knowledge Management (favorites, personal e-mails and folder hierarchies); and, lately, on the Web (Amazon[4], Google[5], Yahoo[6]).

While classifications are heavily used to categorize web contents, the evolution of the web foresees a more formal structure which can serve this purpose – *ontology*, defined in Computer Science as *a specification of a conceptualization* [10]. Ontologies are core artifacts of Semantic Web, an extension of the current Web, in which information is given formal semantics such that computers

[1] See http://www.tnrdlib.bc.ca/dewey.html.

[2] See http://www.loc.gov/catdir/cpso/lcc.html.

[3] See http://www.iskoi.org/doc/colon.htm.

[4] See http://www.amazon.com.

[5] See http://www.google.com.

[6] See http://www.yahoo.com.

can use inference rules to conduct automated reasoning on pieces of this information [1]. The key factor which makes this possible is the fact that ontologies are expressed in a formal language, suitable for automated reasoning.

In this paper we bridge the gap between informal classifications and formal ontologies by describing an approach to encoding classification labels in a formal language such that, together with the structure of the classification scheme, they can be then converted into OWL [2] ontologies (more precisely, into lightweight ontologies, as described in [9]). In principle, the proposed approach allows for automated reasoning on classifications through reasoning on corresponding OWL ontologies. Moreover, the conversion is fully automated. Web directories can be encoded into OWL ontologies without user intervention. We demonstrate the practical applicability of our approach by showing how the results of reasoning on these OWL ontologies can help improve the organization and use of classification schemes. While encoding classifications into a formal language is not new, the main novelty of this paper consists of converting classifications into OWL ontologies, which demonstrates a proof of concept that classifications can be seamlessly integrated in the Semantic Web infrastructure. The fully automated algorithm described in this paper is also novel, as well the characterization of the expressivity of the formal language (i.e. OWL Lite, OWL DL, OWL Full) needed to encode classifications.

The rest of the paper is structured as follows. In Section 2 we describe a comparison between classification schemes and ontologies. In Section 3 we describe how to convert classification schemes into OWL ontologies and how the generated OWL ontolgies can be enriched with additional axioms. In Section 4 we report the experimental results. Section 5 presents how this work helps in optimizing classifications. In Section 6 we discuss the related work and we conclude the paper in Section 7.

2 Classification Schemes vs Ontologies

In this section we discuss commonalities and differences between classifications and ontologies. In order to ground our discussion on well defined terms, below we give the definitions of these two kinds of artifacts.

A *classification* is a 5-tuple $C = \langle N, E, L, D, cl \rangle$ where N is a finite set of nodes, E is a set of edges on N, such that $\langle N, E \rangle$ is a rooted tree; L is a finite set of labels expressed in natural language, such that for any node $n_i \in N$, there is one and only one label $l_i \in L$; D is a set of documents and cl is a function which maps every $d_i \in D$ to a non-empty set of nodes $\{n_i\} \subseteq N$. In Figure 1 we show an example of a classification. Although classifications have no explicit formal semantics for edges, in this example we labeled each edge with the name of a hypothetical relation that may hold between the linked nodes.

An *ontology* is an *explicit specification of a conceptualization* [10]. They are often thought of as directed graphs whose nodes represent *concepts* and whose edges represent formal *relations* between concepts. The backbone structure of the ontology graph is a taxonomy in which all the relations are `sub-class-of`,

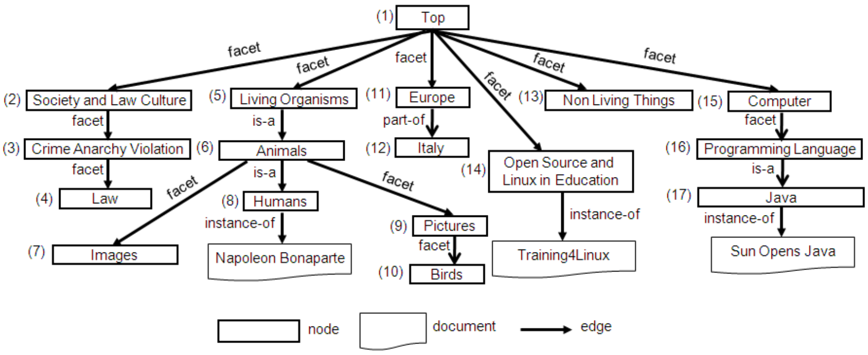

Fig. 1. An example of a classification with link semantics made explicit.

whereas the remaining structure of the graph supplies auxiliary information about the modeled domain and may include relations like `part-of`, `located-in`, `is-parent-of`, and others [11]. Classes can be associated with instances through the `instance-of` relation. In Figure 2 we show an example of a small ontology.

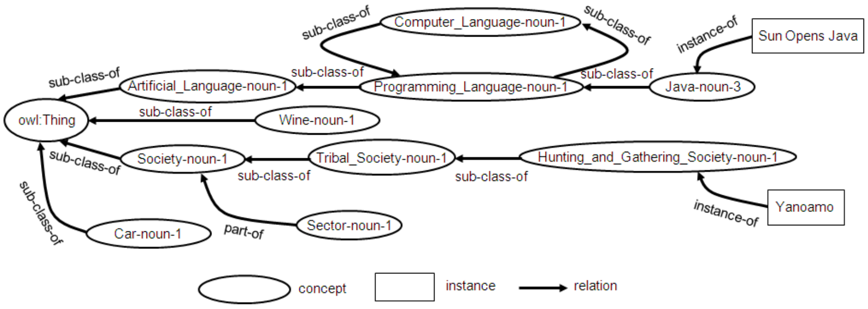

Fig. 2. An example of an OWL ontology.

Even if both ontologies and classifications can often be represented in the form of a graph, ontologies and classifications are quite different in their uses, purpose, language, applications, and in other aspects which we summarize as follows:

- **Users:** a typical user of classifications is a human (e.g., a classifier in a library classification), whereas ontologies are primarily used by machines and, as such, they are the key enablers of the Semantic Web;
- **Purpose:** classifications are primarily used for the organization of (large) document collections into categories and subcategories. Ontologies are primarily used for modeling a particular domain such that the resulting model represents a shared view of a group of individuals [16];

- **Language:** as from the definition, classifications use natural language to describe nodes' categories. Natural language is well understood by humans but, due to its ambiguous nature, it is hard to be "understood" and reasoned about by machines. In contrast, ontologies are codified in a formal language which is unambiguously interpreted by machines, and they are often used for automated reasoning;
- **Nodes:** in an ontology, nodes normally represent atomic concepts (e.g., `car`, `wine`). In a classification, a label can represent a rather complex concept (e.g., "Open Source and Linux in Education") or an individual (e.g., "Napoleon Bonaparte");
- **Edges:** in an ontology graph, edges have a well defined semantics and they usually encode `sub-class-of`, `part-of` and other relations. In a classification, an edge implicitly represents either: (i) a *specification* relation which can be thought of as an `is-a` relation (e.g., an edge from "Animals" to "Humans") or as a `part-of` relation; or, (ii) a *facet* relation which encodes the fact that the label of the child node represents an aspect of meaning of the parent node [3];
- **Instances:** in an ontology, node instances are representatives of the node class and of all its ancestor classes in the `sub-class-of` hierarchy. In a classification, node instances are not necessarily representatives of the class denoted by the node label, and can be documents which are about objects described by the set of labels of the nodes on the path from the given node to the root.

As shown above, classifications and ontologies are quite different and they have their cons and pros with respect to each other. In the next section, we show how we can bridge the gap between them thus combining their pros within a single knowledge representation structure.

3 From Classifications to OWL Ontologies

In this section we show how a classification, as defined in Section 2, can be converted into an OWL ontology. Particularly, we show how classification elements, namely: labels, nodes, edges, documents, and document-node links are encoded into OWL structures. Note that encoding classification labels requires converting from a natural language to a formal language, whereas encoding classification nodes and edges requires only structural manipulation. In Section 3.1 we discuss how we solve the former problem and in Section 3.2 we show how we solve the latter one. In Section 3.3 we show how we encode classification documents and document-node links as class instances. In Section 3.4 we show how the resulting OWL ontology can be enriched with a set of axioms such that it can be better suited for automated reasoning. Finally, in Section 3.5 we discuss which subset of the OWL language is required in order to encode classifications into ontologies.

3.1 From Labels to Concepts of Labels

In the conversion of natural language labels into a formal language we follow the approach presented in [4], which describes how these labels can be converted into a propositional concept language. The underlying idea of this approach is that senses of words, appearing in a label, are converted into atomic concepts, whereas punctuation and syntactic relations between words in the label are converted into logical connectives (such as conjunction $\sqcap$ and disjunction $\sqcup$) and parenthesis. As discussed in [9], the extension of these concepts is the set of documents about the objects or individuals referred to by the (lexically defined) concepts.

In the analysis of natural language labels we use WordNet lexical database [15], and we exploit the natural language processing (NLP) pipeline presented in [19]. The algorithm exploits the structure of the classification, WordNet relations such as hypernymy, and the most frequent sense heuristic to disambiguate the meaning. At this step, we retrieve the senses of each word, we leave only one sense per ambiguous word, and then we convert the disambiguated senses' synsets as well as the words which are not found in WordNet into atomic concepts and encode them as OWL classes.

3.2 From Concepts at Labels to Concepts at Nodes

As discussed in Section 2, edges in a classification represent either a specification or a facet relation, which can be generalized to the following observation: the meaning of a child node consists of what the meaning of its label and the meaning of the parent node have in common. We formalize this observation in the notion of *concept of node* [5, 8, 6, 7], which is defined as follows:

$$C_i = \begin{cases} l_i^F & \text{if } n_i \text{ is the root of } C \\ l_i^F \sqcap C_j & \text{if } n_i \text{ is not the root of } C\text{, where } n_j \text{ is the parent of } n_i \end{cases} \tag{1}$$

where C_i is the concept of node n_i and l_i^F is the concept of label of node n_i. Concepts at nodes are converted into classes in OWL.

Classification edges are implicitly encoded in the definitions of OWL classes representing concepts at nodes. Namely, since these classes are defined as the intersection of the concept at node of the parent and the concept at label of the child node, then the structure of the classification can be reconstructed by analyzing node class definitions.

3.3 From Documents to Class Instances

We convert a document into an instance of the OWL Thing class. Moreover, if a document has a title and a description (as web directory documents normally have), then we encode them in rdfs:label and rdfs:comment properties accordingly. We convert document-node links of a document by defining the rdf:type relation from the instance, representing the document, to the class(es) representing the node(s) in which the document is classified.

3.4 Semantic Enrichment

Since OWL classes, which correspond to word senses, are mapped to synsets in WordNet, we can exploit the relations between synsets and relations between words within synsets in order to enrich the resulting OWL ontologies with additional relations between classes. The enrichment is based on these two rules:

- **Rule 1:** In WordNet, synsets are organized into hierarchies based, for example, on the hypernym (i.e., *is-a* or *is-kind-of*) relation [15]. If two OWL classes (`cl-1` and `cl-2`) correspond to two senses (`sen-1` and `sen-2`) belonging to two synsets (`syn-1` and `syn-2`) among which there is a hypernym relation defined in WordNet (e.g., `syn-2` is a hypernym for `syn-1`), then we define an `rdfs:subClassOf` relation between these two classes (i.e., `cl-1 rdfs:subClassOf cl-2`).
- **Rule 2:** Antonym relations in WordNet are defined among *words* within synsets (and not among synsets). We translate these relations into `owl:disjointWith` relations among classes corresponding to senses of the two antonym words. Classes, associated with these two senses, are declared to be disjoint.

The enrichment of classification OWL ontologies according to the two rules described above allows us to make these ontologies more suitable for reasoning as the underline axiom base grows.

3.5 OWL Sublanguage

OWL ontologies, generated from classifications, fall into the OWL Lite or OWL DL subset of OWL. There are two factors which require OWL DL:

- the logical disjunction that may appear after the conversion of natural language labels and which is converted into the `owl:unionOf` construct;
- disjoint axioms that may appear at the semantic enrichment step and which are converted into the `owl:disjointWith` construct.

Both above mentioned constructs are forbidden in OWL Lite. Note that the conversion to OWL does not require the use of constructs of OWL Full which leaves us within a decidable subset of OWL.

4 Evaluation

To evaluate our approach, we selected four subtrees with the maximum depth of 3 from the DMoz web directory. In Table 1 we report statistical data of the datasets. There are 476 nodes in the selected subtrees, which have 548 tokens in total, out of which, 527 tokens are found in WordNet (i.e., WordNet coverage is 96.17%). Out of the set of words found in WordNet, 223 (i.e., 42.31%) are ambiguous with the average polysemy of 3.36. In our experiments we used WordNet version 2.0.

Table 1. Statistics of the dataset

Dataset	Nodes	Average Branching Factor	Average Subtree Depth	Tokens Per Label	Words with Senses in WordNet	Noun Senses	Adjective Senses
Countries[a]	245	6.26	3	1.07	261	256	5
Europe[b]	75	4.22	3	1.12	86	86	0
Asia[c]	76	4.24	3	1.18	89	88	1
Africa[d]	80	4.31	3	1.15	94	93	1

[a] http://dmoz.org/Regional/Countries/.
[b] http://dmoz.org/Regional/Europe/.
[c] http://dmoz.org/Regional/Asia/.
[d] http://dmoz.org/Regional/Africa/.

4.1 Correctness

We evaluated the most critical step of the NLP pipeline, i.e., the word sense disambiguation (see Section 3.1) algorithm, whose performance results are reported in Table 2. The accuracy of this step largely affects the correctness of the results of reasoning on these OWL ontologies, as we show in Section 5.5.

Table 2. Accuracy of the word sense disambiguation algorithm

Dataset	Ambiguous Tokens	Disambiguation Accuracy(%)
Countries	92	76.54
Europe	38	77.01
Asia	47	80.89
Africa	46	79.13

4.2 OWL Sublanguage

In Table 3 we report statistical data for the generated OWL ontologies.

Table 3. Statistics of the generated OWL ontologies

Ontology	Nodes	Sense Classes	Label Classes	Node Classes	Class Axioms	Individual Axioms	intersectionOf Constructs	unionOf Constructs
Countries	245	261	245	245	873	0	265	4
Europe	75	86	75	75	155	183	76	10
Asia	76	89	76	76	203	125	80	9
Africa	80	94	80	80	212	253	84	9

Noteworthy, most of the constructs in the generated ontologies are valid in OWL Lite. There are only few `owl:unionOf` constructs, which require the use of OWL DL for the representation of these ontologies.

5 Optimizing Classifications

In this section we show some practical examples of reasoning on classification OWL ontologies. For instance, we show how they can be checked for consistency, how their structure can be rationalized, and how nodes with similar contents to a given node can be found.

5.1 Consistency

We used Protégé OWL Plugin [14] and its reasoning capabilities to detect logical inconsistencies within the classification OWL ontologies. We used reasoning capabilities of both Pellet 1.5 and Fact++ OWL reasoners launched with Protégé. None of the reasoners reported that the classification OWL ontologies were inconsistent.

5.2 Rational Forms

Classifications may not be perfect. For this reason we may need to reconstruct a classification based on the "most specific subsumer" relation. Nodes get parents which most specifically describe them, still being more general. The new structure is called, a *rational form* of a classification. The idea behind the rationalization of classifications is to build a classification which better corresponds to a taxonomic structure. The classification given in Figure 3(b) is a rational form of the classification given in Figure 3(a). Note that classification semantics does not change when going from classification to rational form of classification as the set of concepts at nodes remains the same.

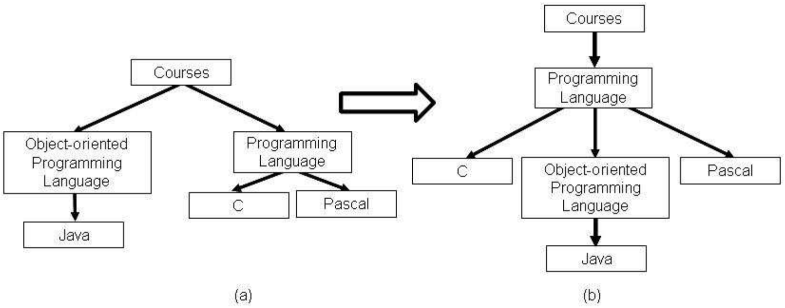

Fig. 3. (a) Classification; (b) Rational form of the classification given in (a)

5.3 Minimizing Effort

The reasoner found an equivalent relation between node class */Regional/Countries/**Italy*** and node class */Regional/Europe/**Italy***. This is an example of how reasoning on classification OWL ontologies can help web directory editors find interrelated parts of the web directory and, thus, improve its organizational structure without manual inspection.

5.4 Computing `See-Also` Links

Apart from the four ontologies, we experimented with another classification OWL ontology and we observed that the individuals asserted to the OWL class which corresponds to the classification node */Games and Activities/Kids and Teens/**Football*** are inferred as the individuals of the OWL class which corresponds to the classification node */Sports Athletics Funs/Youth and High School/**Soccer***, and vice versa. This kind of reasoning can be used for finding similar documents populated in different nodes, which will help in building `see-also` links.

5.5 Errors

Apart from correct relations, we found also some incorrect ones. For example, the reasoner found an erroneous more specific relation between node class */Regional/Europe/**Georgia*** and node class */Regional/Countries/**United States***. As discussed earlier, this problem is caused by the lack of accuracy of the word sense disambiguation algorithm. Evaluating the correctness and completeness characteristics of the computed set of relations between ontology classes is outside the scope of the current paper. Interested readers are referred to [5] for a complete account.

6 Related Work

The current work is a representative of a recent trend in the Semantic Web community towards the use of *lightweight semantics* (as opposed to expressive logic languages) and *lightweight ontologies* [9] (as opposed to full-fledged ontologies), the generation of which can be potentially supported by ordinary users which constitute the long tail of the Semantic Web. The trend has been formed through a number of scientific publications (e.g., see [18, 4, 17, 13]) and is currently supported by a number of R&D projects (e.g., MATURE[7], OpenKnowledge[8]) and systems (e.g., OntoWiki[9]). The current work contributes to this trend by proposing an approach in which classifications, which are often called (informal) lightweight ontologies [9] and whose most representative instantiations on

[7] MATURE, Integrated Project (IP), FP7-216356, see http://mature-ip.eu.

[8] OpenKnowledge, STREP, FP6-27253, see http://www.openk.org/

[9] OntoWiki, see http://ontowiki.net/Projects/OntoWiki.

the web are web directories, can be automatically converted into formal OWL ontologies, ready to be embedded in Semantic Web applications.

There are few lines of work which are close in spirit to our approach. For instance, in [18], the authors propose a method to converting thesauri to OWL ontologies in which they provide a detailed account of how elements of a thesaurus are converted into OWL structures. This approach is based on a manual analysis of thesauri, whereas our approach allows for a fully automatic conversion. Another approach, discussed in [17], comes from the Digital Library community and presents a conceptual structure and transition procedure to support the shift from a traditional knowledge organization system (KOS) and, particularly, a thesaurus, towards a full-fledged and semantically rich KOS. While providing an in-depth analysis of the shortcomings of the traditional KOSs and of the benefits of semantic KOSs as well as providing a set of rules for converting thesaurus elements into ontology constructs, the approach lacks a specification of how a KOS can be converted into an ontology language, such as OWL – the ultimate conversion step discussed in detail in the current paper.

The approach described in [12] allows us to convert a hierarchical classification into an OWL ontology by deriving OWL classes from classification labels and by arranging these classes into a hierarchy (based on the rdfs:subClassOf relation) following the classification structure. The approach is based on some application-dependent assumptions such as that one label represents one atomic concept, and that relations between labels can be defined as `sub-class-of` relations in some particular context (e.g., concept "ice" is more specific than concept "non-alcoholic beverages" when considered in the context of procurement). These assumptions do not hold in a general case and are not made in our approach. Apart from this, our approach differs from [18, 17, 12] in that it is generic and, therefore, suitable for automatic conversion in OWL of any knowledge representation structure whose core can be represented in the form of a classification as defined in this paper.

7 Conclusions

In this paper we have presented a fully automated approach to converting generic classification schemes into OWL ontologies. The proposed approach allows us to leverage on top of classifications, being the interfaces to knowledge for humans, and ontologies, being the interfaces to knowledge for machines on the Semantic Web. Furthermore, as shown above, our approach provides immediate advantage and it allows to help the user in building better classifications more suited for reasoning. Potentially, the approach allows for a cost-free seamless integration of a vast amount of classification structures on the web and in personal repositories into the Semantic Web infrastructure, thus reducing the problem of the lack of semantically rich data. The first experimental results, reported in this paper, show that reasoning on classification OWL ontologies can be used for building practical Semantic Web applications.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, (284(5)):34–43, May 2001.
2. S. Bechhofer et al. OWL Web ontology language reference, W3C recommendation, February 2004.
3. F. Giunchiglia, M. Marchese, and I. Zaihrayeu. Towards a theory of formal classification. In *Proceedings of the AAAI-05 Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&O-2005)*, Pittsburgh, Pennsylvania, USA, 2005.
4. F. Giunchiglia, M. Marchese, and I. Zaihrayeu. Encoding classifications into lightweight ontologies. In *Journal on Data Semantics (JoDS) VIII*, Winter 2006.
5. F. Giunchiglia and P. Shvaiko. Semantic matching. *Knowledge Engineering Review*, 18(3):265–280, 2003.
6. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic schema matching. In *OTM Conferences (1)*, pages 347–365, 2005.
7. F. Giunchiglia and M. Yatskevich. Element level semantic matching. In *Meaning Coordination and Negotiation workshop, ISWC*, 2004.
8. F. Giunchiglia, M. Yatskevich, and E. Giunchiglia. Efficient semantic matching. In *ESWC*, pages 272–289, 2005.
9. F. Giunchiglia and I. Zaihrayeu. Lightweight ontologies. In *The Encyclopedia of Database Systems, to appear*. Springer, 2008.
10. T. R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, 1993.
11. N. Guarino. Some ontological principles for designing upper level lexical resources. In *First International Conference on Lexical Resources and Evaluation*, volume 2830, Granada, Spain, May 1998.
12. M. Hepp. Representing the hierarchy of industrial taxonomies in OWL: The gen/tax approach. In *Proceedings ISWC Workshop on Semantic Web Case Studies and Best Practices for eBusiness (SWCASE05)*, 2005.
13. M. Hepp and J. de Bruijn. Gen tax: A generic methodology for deriving OWL and RDF-S ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies. In *ESWC*, 2007.
14. H. Knublauch, M. A. Musen, and A. L. Rector. Editing description logic ontologies with the Protégé OWL plugin. In *Description Logics*, 2004.
15. G. Miller. *WordNet: An electronic Lexical Database*. MIT Press, 1998.
16. H. S. Pinto, S. Staab, and C. Tempich. Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In *ECAI*, pages 393–397, 2004.
17. Dagobert Soergel, Boris Lauser, Anita Liang, Frehiwot Fisseha, Johannes Keizer, and Stephen Katz. Reengineering thesauri for new applications: The agrovoc example. *J. Digit. Inf.*, 4(4), 2004.
18. M. van Assem, M. R. Menken, G. Schreiber, J. Wielemaker, and B. J. Wielinga. A method for converting thesauri to RDF/OWL. In *International Semantic Web Conference*, pages 17–31, 2004.
19. I. Zaihrayeu, L. Sun, F. Giunchiglia, W. Pan, Q. Ju, M. Chi, and X. Huang. From web directories to ontologies: Natural language processing challenges. In *ISWC/ASWC*, pages 623–636, 2007.