Available online at www.sciencedirect.com
ScienceDirect
Journal of Integrative Agriculture
2012, 11(5): 710-719
May 2012

REVIEW

Dictionary, Thesaurus or Ontology? Disentangling Our Choices in the Semantic Web Jungle

Armando Stellato

Department of Enterprise Engineering, University of Tor Vergata, Rome 00133, Italy

Abstract

The Semantic Web seems finally close to maintaining its promise about a real world-wide graph of interconnected resources. The SPARQL query language and protocols and the Linked Open Data initiative have laid the way for endless data endpoints sparse around the globe. However, for the Semantic Web to really happen, it does not suffice to get billions of triples out there: these must be shareable, interlinked and conform to widely accepted vocabularies. While more and more data are converted from already available large knowledge repositories of companies and organizations, the question whether these should be carefully converted to semantically consistent ontology vocabularies or find other shallow representations for their content naturally arises. The danger is to come up with massive amounts of useless data, a boomerang which could result to be contradictory for the success of the web of data. In this paper, I provide some insights on common problems which may arise when porting huge amount of existing data or conceptual schemes (very common in the agriculture domain) to resource description framwork (RDF), and will address different modeling choices, by discussing in particular the relationship between the two main modeling vocabularies offered by W3C: OWL and SKOS.

Key words: ontologies, thesauri, knowledge modeling, linked open data

INTRODUCTION

When concluding his keynote speech at the 4th International Semantic Web Conference, sir Tim Berners-Lee, the "inventor" of the Semantic Web (and former mind behind the first WWW) explicitly asked the community of practitioners and academics for a slow-down on the theoretic research (mostly tight to logic and knowledge representation) on the Semantic Web and encouraged, in the spirit of a ground approach to the development of the Web of Data, the realization of ontologies about any potential domain of interest. The objective was clear: if we had left scientists discussing too much about what the Semantic Web would have been, the "conditional" would have been the sole tense used to decline verbs when speaking about it.

Later on, Semantic Web evolution has undergone two more focus-switches; the first one has been from *ontologies* to *applications*: looking for the semantic web killer-application has been seen as the objective to pursue for proving the usefulness of the web of data, and thus for reallypushing its realization and success. With SPARQL reaching W3C recommendation in 2008 (Prud'hommeaux and Seaborne 2008), languages for data representation and querying have finally provided

Received 28 June, 2011 Accepted 26 August, 2011
Correspondence Armando Stellato, Tel: +39-06-72597330, E-mail: stellato@info.uniroma2.it

a complete and usable technology stack put at the hands of Semantic Web application developers.

The last move has been towards data publication and sharing: with no data, there would be no applications consuming it and thus no Semantic Web, but with no applications, there is no strong interest in publishing data. This chicken and egg problem has been addressed by the Semantic Web "bootstrapping" initiative *Linked Open Data* (Bizer *et al*. 2009) which promoted and supported the realization of a huge cloud of data. Though some criticism has been expressed towards the fact that the LOD itself – with its registration policy – is not a real representation of what a (Semantic) Web should be (e.g., in the classic Web, no one has to register to any entity to create a website, they just need to be owner of an URL) the LOD is with no doubt a successful attempt to disseminate best practices about linked data publication, to provide software for easing this task (e.g. Pubby[1)]) and to bootstrap the population of the web with linked open data. In the future, the cloud will probably explode and become uncontrollable as the Web, that will be the time for data search engines and links to do their job.

All of the above show a (natural) trend towards "grounding" the Semantic Web; we could say that while up to very few years ago we were in the "Protégé[2)] era", with semantic geeks building ontologies over ontologies, now the shift has drastically moved towards publishing data, more emphasizing the peculiarities (scalability, immediateness) of resource description framwork (RDF) triple stores over traditional relational databases and even reconsidering the role of deep semanticfoundations in the web of data[3)].

However, while skepticism towards ontologies and concept schemes can be motivated in such scenarios where publishing raw data is the main objective (for instance, data which can be described by a couple of concepts and a small bunch of properties, but counting billions of records) still there will be lot of actors in the Semantic Web with precise modeling requirements and need for strong semantic representation of the information they want to publish.

The agriculture domain, which is probably the domain that readers of this journal are mostly interested in, belongs to this second category: from nineties' Machine Readable Dictionaries (Ide *et al*. 1994) to thesauri and later ontologies, the agriculture domain has hosted many knowledge resources, such as GEMET[4)], the GEneral Multilingual Environmental Thesaurus, which has been developed by the European Topic Centre on Catalogue of Data Sources under contract to the European Environment Agency, or AGROVOC[5)], the world's most comprehensive multilingual agricultural vocabulary, developed and maintained by the Agricultural Information Management Standards (AIMS) group of the Food and Agriculture Organization (FAO) of the United Nations.

Representing information about the agriculture domain is a task which may originate from really different exigencies, which we may roughly divide across two kinds of requirements: representing ground data about various aspects of agriculture (e.g., the number of millimeters of rainfall precipitations in South America, per nation and per month across a span of several years), or providing knowledge as for a dictionary of agriculture (e.g., by providing taxonomies of plant species, or by characterizing different sowing techniques). Very often, this categorization is not neat, and the modeling needs seem to lightly fade across the two scenarios above.

The W3C is offering today different instruments for modeling knowledge on the web, which try to fit the above different exigencies and more: taking the appropriate modeling decision is thus an important aspect for publishing knowledge on the web, share it, etc.

Unfortunately, while the web of data seems to be happily spreading around the world, still mastering its technologies and understanding its languages is confined to a (relatively) restricted circle of interested people: spreading a new technology implies divulgation, requires

1) http://www4.wiwiss.fu-berlin.de/pubby/

2) Protégé (Gennari *et al*. 2003) is a Knowledge Management and Acquisition tool developed at the University of Stanford. It has been (and is currently, despite valid products now on the market) the most popular ontology editing tool

3) http://groups.csail.mit.edu/haystack/blog/2009/11/03/does-the-semantic-web-need-ontologies/
http://iswc2009.semanticweb.org/wiki/index.php/ISWC_2009_Research_Track#16:45__.09_Panel:_Does_the_Semantic_Web_Need_Ontologies.3F

4) http://www.eionet.europa.eu/gemet/theme_concepts?th=2&langcode=en

5) www.fao.org/agrovoc/

meeting masses, and building up a generation of professional figures which will be not anymore composed only of PhD students, professors and on-the-edge-of-technology practitioners, but real substitutes for plain old database administrators, knowledgemodelers, knowledge engineers, etc.

No wonder that even domain experts educated to traditional means of conveying information (even those which are familiar with Computer Science) are naturally hampered inhaving their knowledge properly fitting the web of data jigsaw puzzle.

In this paper, I try to sketch some easy-to-grasp insights for traditional domain experts approaching the Semantic Web, trying to provide a simple guide to drive them safely through the Knowledge Representation Jungle. The next section will provide a brief historical introduction on how the requirements for a web of data led to the development of current knowledge representation standards for web: RDF/RDFS/OWL. I'll then address those problematic issues which hampered the development of thesauri or at least made unclear how to realize even their basic constituents, and show how the definition of a dedicated vocabulary (SKOS) for thesauri building broke this impasse.

FROM RDF TO OWL AND SKOS

The idea behind the Semantic Web (Berners-Lee *et al*. 2001) is to establish a parallel version of the WWW where data can be published, accessed and shared on the basis of formal models for knowledge representation and exchange.

The laying principles for the Semantic Web (SW, from now on) have been based on these pillars:

(1) Unambiguous references: URN[6] (Uniform Resource Names) guarantee the univocal reference to a given resource;

(2) Decentralization: the Semantic Web must bear the same decentralized approach of the traditional web, as it will be actually part of it. This naturally implies that we cannot expect the (semantic) web to be a monolithic, globally consistent network of data;

(3) Minimalist design: a simple, shared model, would have provided the fabric for different application, contexts, perspectives, etc, to interoperate over a common ground;

(4) Interoperability and "Evolvability": an incremental, monotonic approach to knowledge evolution, distribution and sharing;

(5) Trust: each agent on the web should clarify what it trust, and so its *beliefs*, so that other agent know how to properly interact with it.

Inspired by the above minimalist design principle, the SW has introduced a foundation technology for representing knowledge: RDF, the Resource Description Framework (W3C 2004).

RDF: resource description framework

The underlying structure of any expression in RDF can be viewed as a directed labeled graph, which consists of nodes and labeled directed arcs that link pairs of nodes. The RDF graph is a set of triples (also called statements) of the form described in Fig. 1.

Every entity in the web may be represented as a node of the graph and may be related to other entities through the use of properties. The relation expressed by the property arc (also called predicate) is directional and thus determines which of the two nodes of the statement is the subject and which is the object. Entities in a RDF graph belong to one of the following three –

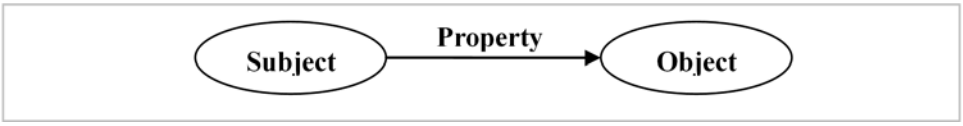

Fig. 1 The RDF triple paradigm.

[6] URNs do not have to provide a physical reference to any resource on the web; however, linked data best practices recommend now to resolve the names of data resources through URL corresponding to their URN name, to provide their RDF description (or access to their corresponding physical resources, in the case of documents or multimedia, through content negotiation).

mutually disjoint – sets:

(1) URIs: these are resources for which a given name (the URI);

(2) BNodes: these represent resources which have no name[7], and which are often used as gluing objects to build complex constructs (e.g., n-ary relationships);

(3) Literals: this set denotes data values, which can be used to describe certain properties of resources. Literals can be either *plainLiterals* (that is, an untyped string, with an optional language tag) or *typed literals*, i.e., data values with an assigned datatype qualifying their nature (e.g., a datatype<xsd:int>) can be used to tell that the given literal must be interpreted as an integer number.

URIs and BNodes are together called resources; the subject of a triple is always a resource, the predicate is always an URI, while the object may be populated by any of the three categories[8].

The meaning of an RDF graph is the conjunction (i.e., logical AND) of all the statements that it contains. The expressive power of RDF corresponds to the existential-conjunctive (EC) subset of first order logic (Sowa 2000). It does not provide means to express negation (NOT) or disjunction (OR).

Serialization syntaxes

A number of different serialization syntaxes exists for representing RDF graphs. The XML syntax, which has already been introduced in the previous section, facilitates integrating Semantic Web documents in the current HTML/XML web. At present time the RDF/XML (the XML-based language which has been developed to represent RDF triples) is currently normatively specified and recommended by W3C for use to exchange information between applications. There are other possibilities, among which we mention N-Triples[9] and Notation 3 (N3[10]) syntax. The former is a plain representation of RDF triples as an explicit list of subject, predicate and object triplets. The latter uses many syntactic tricks to improve human readability and make serialization more compact (especially if compared to the RDF/XML format).

RDFS and OWL

Through RDF it is possible to realize and define simple data models which resemble seventies' Semantic Networks (Sowa 1992). What lacks in these networks is the possibility to specify and layer different levels of abstraction for representing knowledge domains, by specifying classes (types) of resources and by arranging these classes under a taxonomical relation. Another important feature lacking from RDF is the possibility to impose constraints over the application of properties to domain objects (restrictions over domain and range, both on the set of considerable resources and on their quantification).

RDF Schemas (RDFS) provide these features, leveraging RDF to a knowledge representation language with capabilities well over semantic networks. The RDF Schema specification (Brickley and Guha 2004) provides these facilities through dedicated primitives for defining metadata vocabularies, by establishing hierarchies of concepts and properties, through: rdfs:Class (giving new semantics to the rdf:type already define in RDF), rdfs:subClassOf and rdfs:subPropertyOfor to document the resources themselves through appropriate annotations (e.g., annotations properties such asrdfs:comment, rdfs:label, etc.).

RDFS Schemas are arranged in a modular way, which has been inherited from the adoption of an Object Oriented (OO) paradigm to knowledge representation. RDFS approach however differs from typical OO design. Rather than defining classes as templates for objects modeled after them (with an approach typical of OO design and Frame Theory (Minsky 1975)), RDF properties (be them relations or attributive properties) become first-class citizens in the language and maybe associated to any object, eventually allowing language semantics to infer object classification a-posteriori. For instance, given a certain RDFS Schema, having stated

[7] At implementation level, when the RDF graph hosting them is loaded in a triple store, they may be assigned some identifier to assure consistent retrieval and unification. The identifier is however not a persistent name and it is not assured to be maintained across two different accesses to the same RDF store.

[8] There have been various discussion on whether allow subjects of triples to host also literals, see a thorough resume.

[9] http://www.w3.org/2001/sw/RDFCore/ntriples/

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

that a given object "Armando" has brown hair (by stating a triple such as: <Armando, hair, brown>), we could infer that Armando is a Person, because the domain theory stated in that RDFS schema tells us that only instances of class Person may have hairs (formally, the rdfs:domain of property hair is Person). There is no need of specifying that the object belongs to the class Person, as it can be inferred a-posteriori; moreover, the axiom on property restriction and the class Person could have been defined in two different, though interconnected, schemes. This is in line with the nature of the Web, where information is distributed and potentially underspecified.

The web ontology language (OWL) specification (W3C 2004) builds on top of RDFS from the ashes of DAML+OIL (McGuinness *et al*. 2002), which is in turn the result of a EU/US conjoint work (Joint EU/US Committee on Agent Markup Languages) aimed at combining two previous results, the American Darpa Agent Markup Language DAML-ONT (Hendler and McGuinness 2000) and the European Ontology Inference Layer (Fensel *et al*. 2000).

OWL adds to RDFS an improved vocabulary for describing properties and classes: among others, relations between classes (e.g., disjointness), cardinality (e.g., "exactly one"), equality, richer typing of properties, characteristics of properties (e.g., symmetry), and enumerated classes thus enhancing the inferential capabilities of the language.

The OWL vocabulary has been recently improved in a new specification, OWL2 (W3C 2009), offering some syntactic shortcuts as well as wider expressivity, with the introduction of elements such as: keys, property chains, richer datatypes, data ranges, qualified cardinality restrictions, enhanced annotation capabilities, and new property facets (asymmetry, reflexivity and disjointness).

Limitations of OWL in expressing thesauri and concept schemes

OWL allows for a very rich specification of domains for object classification. However, its close binding to 1st order logic (OWL offers different levels of expressivity, known as OWL species, which are in a 1-1 relation with different families of Description Logics, being them in turn fragments of 1st order logic) demands for *classes* (1st order elements) to be used roughly as "folders" where to store (classify) the *objects* (0th order elements) of the world; however, the agriculture domain has been often represented through simple thesauri and concept scheme, more close to semantic encyclopedias than to data catalogues.

Which of the two elements above is thus eligible for being adopted in the role of *concept* in a traditional concept scheme or thesaurus? Before answering to this question, we provide here a very short insight behind 1st order logics and knowledge representation; Fig. 2 represents the predicates/objects dichotomy in 1st order logics: ground objects describe objects of the world (a specific person, a table, etc.) while predicates are

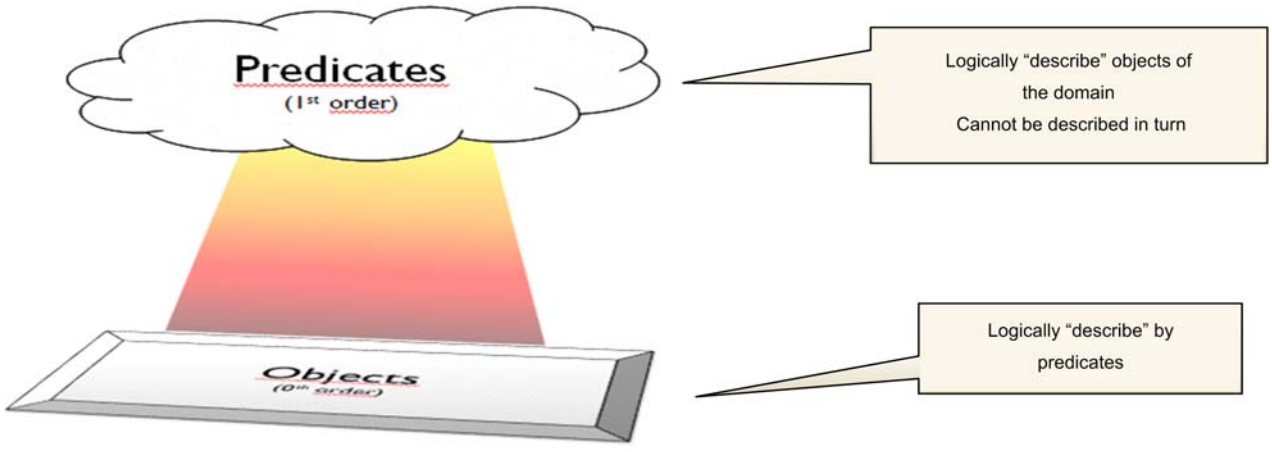

Fig. 2 The predicates/objects dichotomy in first order logics.

used to describe (*predicateover*) them. Predicates are thus the deus-ex-machina through which objects of the domain can be characterized; they "live" in a world which is separated from the domain being represented, and cannot be described in turn: this would require 2nd order predicates, thus bringing into scope one more layer of logic.

In *knowledge representation*, *class*constructs are defined as *monadic predicates* (i.e., with a single argument) used to state the membership of objects to the category represented by them. So, for instance, the expression: **AClass (an_object)** means that the object an_object is an *instance* (is classified as a member) of class AClass. Thus, the ground fact:
Country (China) tells us that China is known to be a country in the knowledge base containing this statement. Description Logics have only such monadic predicates (classes) and dyadic ones (properties) to represent attributes and relationships (in RDF, this is mapped through the subject-*predicate*-object construct previously defined, which is equivalent to the logical predicate (subject, object) notation.

Given the above, it would appear immediate to consider classes as concepts[11], some sort of ideal representation of ground objects: even in the previous example, the object China would be an instance of the general concept Country.

However, instead of easily sticking to what common sense may suggest, we must take into consideration the modeling needs of concept schemes, and then decide how to properly adopt OWL elements for their representation.

Concept schemes usually provide:

(1) A set of *concepts* representing important objects of the domain of discourse. The distinction between *object* and *concept* is not neat, though the main elements are called concepts because they usually represent abstract objects and not specific real-world objects (e.g., the entry "deer" in the previously cited AGROVOC vocabulary does not represent a specific deer, but the general *category* of deers), though important references to singletons are usually present as well (the entry Gobi Desert in the same vocabulary does not represent any generalization at all, and is instead a precise reference to that unique desert in the world).

(2) A *taxonomy*, or *hierarchy* of the above concepts, expressing a generic more_specific/more_genericrelationship between them. Taxonomies are commonly used to provide an intuitive and easy-to-use mean for cataloguing and retrieving information from the scheme.

(3) *Domain properties*: specific properties may be defined to characterize concepts in the scheme.

So, reexamining the requirements, it seems OWL fails in providing a jack-of-all-trades representational object: concepts need to be characterized and related to other concepts, and thus OWL instances, being described through properties, seem to be the ideal candidate to act as concepts, however, the main taxonomical relationship in OWL is held between classes (rdfs: subClassOf), to provide a hierarchical organization of classifiers rather than of the objects. This way, there is not a complete and satisfactory candidate for the role of *concept* in OWL. At the same time, there is no strict exigency of representing the classes/objects dichotomy: most of the thesauri contain no distinction between objects and concepts, and all contents are grouped in generic entries (such as concepts in concept schemes). The example in Fig. 3 shows how this distinction fades away in current concept schemes: the screenshot taken on AGROVOC shows how the same hierarchy (and thus the same broader/narrower relationship) is used to represent the relationship occurring between "Arid zones" and "Deserts" (which is a kind of subclassification) as well as between "Deserts" and "Gobi Desert" (which, in any reasonable interpretation, is a specific instance falling in the "Desert" category). The ones above are not modeling errors but precise choices of simplification in representing information inside semantic dictionaries thesauri and concept schemes.

There is another aspect which makes OWL a difficult choice for representing concept schemes, that is *semantic commitment*: the rdfs:subClassOf relationship – which is used both in RDFS and OWL to represent hierarchy of concepts – actually provides a hierarchy of classifiers (the RDFS/OWL classes) with well-defined semantics, implying that all instances of a given

[11] Note that classes are, in effect, be also referred as concepts in ontology literature.

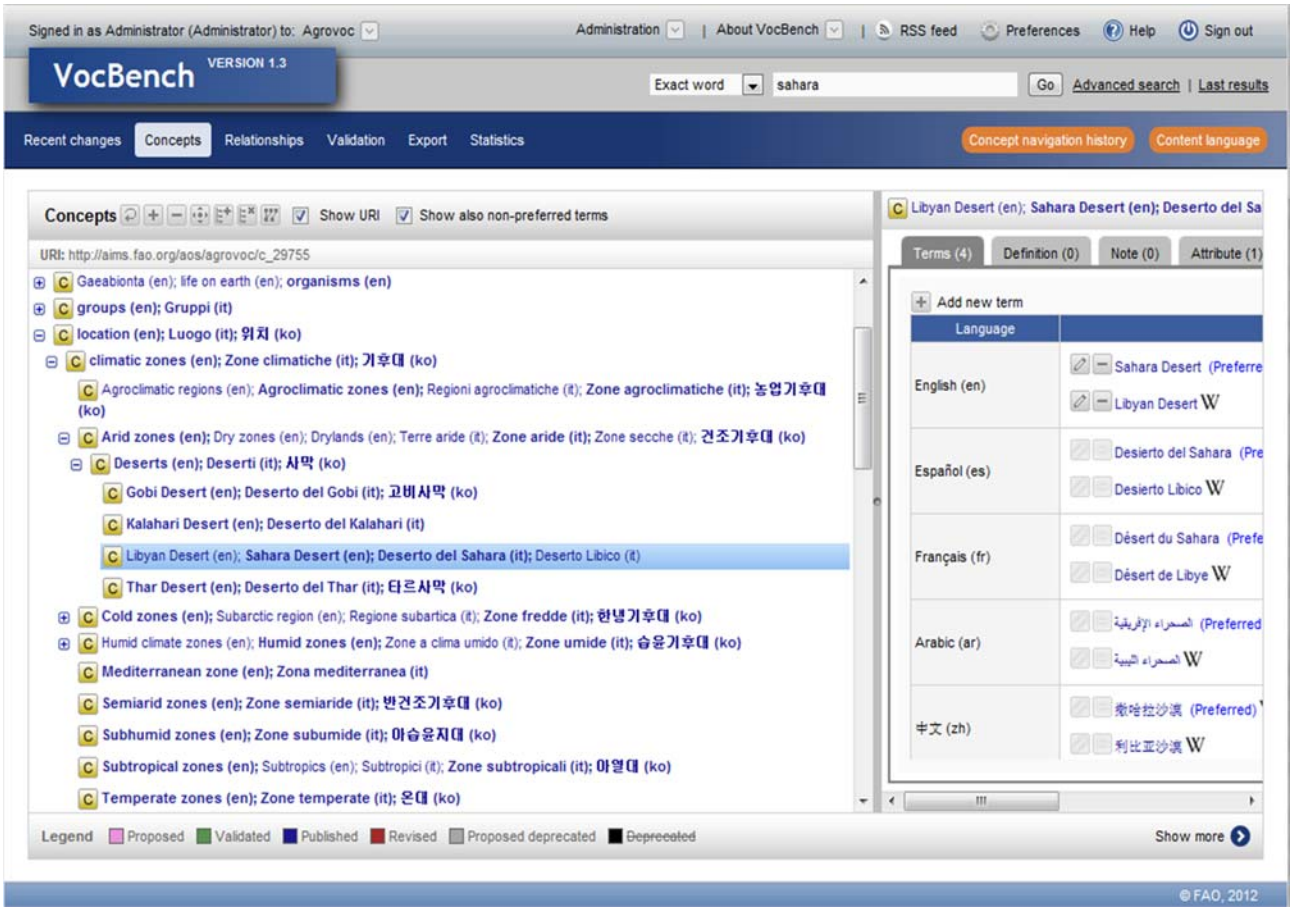

Fig. 3 Subclassification and instantiation blurring in a concept scheme.

class are also instances of all itssuperclasses. This basic yet fundamental axiom is not easy to be checked for semantic validity among potentially huge dictionaries which are being converted to RDF and need to be maintained in this format in years to come: checking the semantic coherency of an ontology is a task which need thorough attention and dedication, and though methodologies (Guarino and Welty 2004) have been proposed and tool support (Welty 2006) has been devised in past years to enable even non-ontology savvy people to cope with this task, still it requires a huge amount of human work when applied to very large scheme and classifications.

SKOS

SKOS (W3C 2009) is a new RDF modeling vocabulary from the W3C developed with the aim of filling the gap left from OWL in representing Simple Organization Systems through shallow semantics.

SKOS has been originally developed in the nineties inside the Language Independent Metadata Browsing of European Resources (LIMBER) EU funded project. Later on, its development has been continued in the Semantic Web Advanced Development for Europe (SWAD-Europe) project and finally moved to incubation inside the W3C until it reached recommendation status in 2009.

The name SKOS originates from *Simple Knowledge Organization* (*System*), referring to the knowledge resources for which the homonym language has been developed: *thesauri, classification schemes, taxonomies, subject-heading systems* and any other type of *structured controlled vocabularies*.

Contribution of SKOS to knowledge modeling with respect to OWL is two-fold, by first providing an alternative vocabulary with shallow semantics for representing and describing concepts and relationships be-

tween them and, secondly, by offering a richer vocabulary for providing a better characterization (a feature which is much demanded in thesauri building) of the linguistic aspects of these schemes and their adherence to classical cataloguing methodologies.

The first contribution can be synthesized in the two following characteristics:

(1) All data down to 0th order of logic: SKOS concepts are, in logical terms, *objects*, and can thus be described as well as organized in catalogues. There is no distinction between classes and instances, just *concepts*.

(2) *Lose* strong semantic assumptions through *loose* semantic relations:

Intra-Scheme: narrower/broader relationship between concepts specify a very generic specialization relationship which has no further entailment

Extra-Scheme: a set of properties (skos:exactMatch, skos:relatedMatch, skos:broadMatch, and skos:closeMatch) is defined to allow for loose semantic matches between concepts from different schemes. Again, no strong impact on inference with respect to use of owl:equivalentClass, owl:equivalentProperty and owl:sameAs, which imply equipotentuse of matched elements in inference rules.

The second contribution of SKOS lies in a dedicated vocabulary specifically thought for modeling Knowledge Organization Systems, through the introduction of properties for language annotation (lexical labeling properties), such as, skos:prefLabel, skos:altLabel, skos:hiddenLabel to better model the linguistic aspects of concepts, the adoption of a dedicated property (skos:notation) for referencing notation identifiers from legacy cataloguing codex, a series of *documentation properties* to document concepts as RDF resources and not as the real-world objects they represent (skos:note, skos:editorialNote, skos:example, etc.), the possibility to draw different parallel schemes over the same content and a few features more.

SKOS and OWL

One important thing to clarify is that, though OWL and SKOS are often depicted as strictly disjoint alternatives (and SKOS is said to be built over RDF and RDFS), they are much more interchangeable than what is expected. By first, *SKOS is an OWL vocabulary*, some of its properties are defined by using OWL (e.g. skos:related is anowl:SymmetricProperty and skos:Concept is an owl:Class) and many of these properties, such as the lexical labeling properties, can be used[12] to describe resources defined through OWL as well.

Also, SKOS benefits of OWL reasoning, as much of the facets of its properties imply that new information can be inferred from explicit assertions: the already cited symmetry in skos:related implies that only one direction may be explicitly stated between related concepts with the other being inferred, or the fact that skos:broader and skos:narrower are owl:inverseOf of each other implies that if <A>skos:narrower<B>, then it holds that: <B>skos:broader<A>.

On the contrary, SKOS features a few constraints, mostly involving property disjointness (i.e., two disjoint properties cannot link the same resource to the same value), which cannot be formalized through OWL axioms[13], and which are left as documented best practices for SKOS developers.

Finally, resources can be freely reused among OWL ontologies and SKOS schemes, though it is advisable to maintain ontologies and knowledge schemes in different repositories. SKOS in fact, is an OWL vocabulary which brings any document modeled upon it into the OWL-Full species (a logically non-decidable fragment of OWL): keeping them in separated semantic repositories linked through common URI references allows for linked data navigation while leaving owl reasoners to work on decidable ontologies.

An interesting and more thorough discussion on this topic can be found in (Jupp *et al*. 2008) though this is partially outdated considering the recent changes in the 2009 release of SKOS (e.g., skos lexical labeling properties are now instances of owl:AnnotationProperty – see again footnote 12 – so these can be used to decorate OWL classes with no danger of incurring in a violation of DL restrictions, and their suggestion to use *punning* for other object and datatype properties

[12] See SKOS reference available at http://www.w3.org/TR/skos-reference/#L1541

[13] OWL2 can however represent most of these axioms; OWL2 was left out from SKOS formalization since it reached recommendation status approximately at the same time as SKOS. It will probably be included in a future version of SKOS.

is now an OWL 2.0 feature) and must thus be reconsidered accordingly.

CONCLUSION AND NOW, WHICH CHOICE?

This article has provided some historical notes and shown the rationale behind the development of the two main modeling W3C vocabularies for knowledge management based on the RDF standard: OWL and SKOS.

SKOS is thus by far the best choice for all cases where one or more of the following conditions occurs: the resource acts more as a (semantic, conceptual) vocabulary providing a thorough description of the domain, a large number of concepts is involved, the purpose of the resource is more close to an index (e. g., to support retrieval of documents tagged after the resource's concepts, or to drive their navigation in browsing systems) or to an encyclopedia, rather than to a database.

OWL is meant for data classification: even not considering its web-oriented nature, we may think of it as an alternative replacement for databases, where potentially billions of records need to be archived according to well-defined data-structures and schema. What we buy with OWL over traditional relational DBs is easy schema scalability (adding more concepts or properties or rearranging a class tree structure is a painless effort with respect to changing a schema in a DB), interoperability (merging two DB schema requires heavy work on data transformation), plus the support for inference allows for direct management of only minimal, non-redundant information, leaving the rest to inference-based classifiers.

The last section has also explained how it is possible to interweave the two vocabularies to make them benefit of each other, and that different perspectives for same entities may also be realized by creating mutual references for same entitiesbetween OWL ontologies and SKOS conceptual schemes. The aim of this paper has thus been to support knowledge management professionals and practitioners (working in domains – such as agriculture – naturally characterized by wide and articulated conceptualizations as well as by the necessity to represent precise data references), in taking modeling decisions suiting their need, whether they are porting previously existing semantic resources into Semantic web standards, or they are creating new ones.

Acknowledgements

The author would like to express his gratitude to Johannes Keizer and all the Agriculture Information Management Standardsteam at the Food and Agriculture Organizationfor introducing him to the world of Agriculture Knowledge Management and finally enabling him to match his abstract graphs with real world objects.

References

Berners-Lee T, Hendler J A, Lassila O. 2001. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, **279**, 34-43.

Bizer C, Heath T, Berners-Lee T. 2009. Linked data – The story so far. *International Journal on Semantic Web and Information Systems (IJSWIS), Special Issue on Linked Data*, **5**, 1-22.

Brickley D, Guha R V. 2004. In: McBride B, ed., *RDF Vocabulary Description Language 1.0: RDF Schema.* [2011-3-22]. Rretrieved from World Wide Web Consortium (W3C): http://www.w3.org/TR/rdf-schema/

Fensel D, Horrocks I, van Harmelen F, Decker S, Erdmann M, Klein M. 2000. OIL in a nutshell. In: Dieng R, ed., *Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000), Lecture Notes in Artificial Intelligence, LNAI.* Springer-Verlag, Berlin.

Gennari J, Musen M, Fergerson R, Grosso W, Crubézy M, Eriksson H. 2003. The evolution of Protégé-2000: An environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, **58**, 89-123.

Guarino N, Welty C. 2004. An overview of ontoClean. In: Staab S, Studer R, eds., *The Handbook on Ontologies.* Springer-Verlag, Berlin. pp. 151-172.

Hendler J, McGuinness D L. 2000. The DARPA Agent Markup Ontology Language. *IEEE Intelligent Systems, Trends and Controversies*. November/December. pp. 6-7.

Ide N, Véronis J, Cedex A. 1994. Machine readable dictionaries: What have we learned, where do we go. In: *Proceedings of the Post-COLING '94 International Workshop on Directions of Lexical Research*. Beijing. pp. 137-146.

Jupp S, Bechhofer S, Stevens R. 2008. SKOS with OWL: Don't be Full-ish! In: Dolbear C, Ruttenberg A, Sattler U, eds., *OWLED.432*. [2009-2-15]. http://CEUR-WS.org

McGuinness D L, Fikes R, Hendler J, Stein L A. 2002.

DAML+OIL: An ontology language for the semantic web. *IEEE Intelligent Systems*, **17**, 72-80.

Minsky M. 1975. A framework for representing knowledge. In: Winston P H, ed., *The Psychology of Computer Vision.* McGraw-Hill.

Prud'hommeaux E, Seaborne A. 2008. *SPARQL Query Language for RDF*. [2011-3-22]. Retrieved from World Wide Web Consortium – Web Standards: http://www.w3.org/TR/rdf-sparql-query/

Sowa J F. 1992. Semantic networks. In: Shapiro S C, ed., *Encyclopedia of Artificial Intelligence.* 2nd ed. John Wiley & Sons, Inc., New York, USA.

Sowa J F. 2000. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, California, United Sates.

W3C. 2004, February 10. *OWL Web Ontology Language.* [2011-3-22]. Retrieved from World Wide Web Consortium (W3C): http://www.w3.org/TR/owl-features/

W3C. 2004. *Resource Description Framework (RDF)*. [2012-4-18]. Retrieved from http://www.w3.org/RDF/

W3C. 2009, October 27. *OWL 2 Web Ontology Language.* Retrieved from World Wide Web Consortium (W3C): http://www.w3.org/TR/2009/REC-owl2-overview-20091027/

W3C. 2009, August 18. *SKOS Simple Knowledge Organization System Reference.* [2011-3-22]. Retrieved from World Wide Web Consortium (W3C): http://www.w3.org/TR/skos-reference/

Welty C. 2006. OntOWLClean: Cleaning OWL Ontologies with OWL. In: Bennet B, Fellbaum C, ed., *Proceedings of FOIS-2006.* IOS Press.

(Managing editor ZHANG Juan)