

# Large scale colour ontology generation with XO

Laurent Lefort and Kerry Taylor

CSIRO ICT Centre  
GPO Box 664 Canberra ACT 2601, Australia  
{laurent.lefort, kerry.taylor}@csiro.au

## Abstract

The semantic integration of web services requires scalable and shareable ontology engineering solutions. In this context, we need a repeatable ontology construction process to produce domain and mapping knowledge between heterogeneous knowledge resources. We need a solution to reuse the community standards already defined for specific application areas without breaking the traceability chain between the original resource and its ontology equivalent. We aim to support active communities of domain specialists which already produce useful knowledge on the web, and to encourage them to publish in the Web Ontology Language, OWL, as well as in their original format of choice. Our solution is to automatically generate XSL code for the conversion from XML to OWL out of a more compactly written stylesheet, to facilitate the work of the ontologist or domain specialist. We present here a practical example based on colours to show how our approach can be applied to a relatively large collection of concepts and to discuss some of the ontology engineering challenges and design decisions. We provide an overview of the resources we have selected, analyse key features of the resulting ontology vs. a prominent colour online resource, the Getty Art and Architecture Thesaurus and discuss the benefits of our approach.

*Keywords:* Ontology extraction, XSL Transformation, Colour.

## 1 Introduction

Our Ontology acquisition and Model Transformation tool, called XO, is a web-based environment for the specification and execution of XSL Transformations (XSLT) from XML to OWL. It supports the reuse of existing knowledge sources through the application of a small set of ontology design instructions over relatively large sets of reusable definitions. The colour application presented here, in which we manage over 5000 individual colour definitions, is used to illustrate how we can construct a result compatible with the ontology quality level required for Semantic Service Integration (Ackland et al. 2005).

This paper is organized in six parts: The first part is a review of existing solutions related to our specific ontology engineering requirements. The second part is an overview of XO. The third part describes the selected colour resources. The fourth part focuses on ontology design challenges and issues. The fifth part describes the generated colour ontology and compares it to the online service provided by the Getty Art and Architecture Thesaurus. The last part is a broader discussion on quality issues linked to the planned exploitation of the resulting ontology.

## 2 Ontology Engineering and Model Transformation technologies

The semantic integration of heterogeneous services cannot be done without robust and cost effective ontology engineering approaches with the capability to import knowledge from existing resources. The development of ontology engineering methods (Cristani and Cuel 2002) and tools (Damjanovic et al. 2004) has largely been driven by the need to give a manual tool to experts of a domain to facilitate the formalisation of their knowledge. The transformation methods which have been developed for resources such as thesauri (Wielinga et al. 2004, Gangemi 2004, and Miles 2005), UML models (Gašević et al. 2004) and tables inside HTML pages (Pivk et al. 2005, Tijerino et al. 2003) are not readily accessible to users of popular interactive tools such as Protégé.

Software engineering approaches such as Object Management Group's model-driven development fosters the creation of new transformation languages such as Query View Transformations (OMG 2004) with Eclipse-based tools relying on generalized compiler technology and source code reengineering. XSLT is also used for model transformation between XML-based UML formats (Grønmo and Oldevik 2005). Model transformation is now envisaged between standards such OWL, Topic Maps, UML, and other knowledge representations formalisms (OMG 2005).

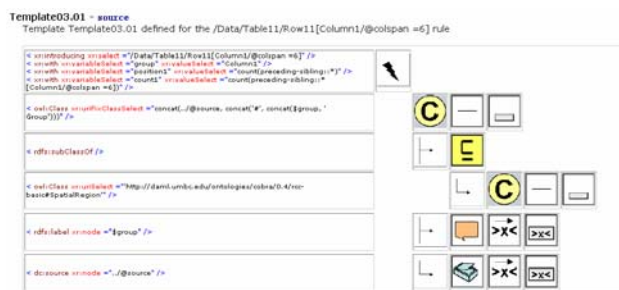
Presently, XSLT-based transformation tools are too specific to one type of input and barely compatible with each other. The development of XO has been motivated by this gap between the built-in features offered by XSLT and what is needed to facilitate the reuse of transformation instructions amongst developers wishing to generate OWL content.

## 3 Our Ontology engineering solution based on XSL

XO supports the transformation from XML into OWL, including the variants of OWL up to and including OWL

full. It is based on a compact format derived from the XSLT standard syntax. This format is preprocessed to generate XSL code which is then used to perform the intended translation. In this respect, XO has been inspired by XR, a tool for the conversion from XML to RDF developed by Sjoerd Visscher (2004). The XO syntax is closer in style to OWL making it more accessible to non-XSL specialists.

The work which is required out of the XO user is to write the design instructions for the conversion from any XML input into OWL. A typical application of XO is likely to contain between 5 and 10 “high level” ontology design patterns such as the one presented in the figure below (Figure 1). We use XSLTDoc (Birrer 2005), a javadoc-like documentation utility for XSLT to generate automatically this documentation out of the XO code. The code which is created by the XO user is shown on the left.



**Fig. 1: Graphical view of ontology design instructions**

Each pattern requires a variable number of elementary XO instructions. The majority of the instructions are mirroring the RDF syntax to specify the output according to the user’s wishes. The pattern presented here contains 5 RDF design instructions:

- owl: class (twice)
- rdfs: subclass
- rdfs: label
- dc: source

The first line in the figure above specifies how to locate and fetch the relevant information from the input file. We provide here a small fragment of XO code to show the place of XPath expressions and XSL function calls in the XO syntax.

```
<xr:introducing xr:select="/Data/Table11/Row11">
  <xr:with xr:variableSelect="s" xr:valueSelect="c2"/>
  <owl:Class xr:uriSelect="concat($documentURI,$s)"/>
```

The XSLT features provided through XO are:

- XSL templates (xr:introducing), used to access data from different parts of the input file;
- XSL variables (xr:with), used to store data so that it can be referenced elsewhere;

- Loops over input data, used to access data directly under the working node (xr:select) and data which is located elsewhere in the file;
- XSL functions to create complex expressions, enabling linking and concatenating data from various origins; and
- Calls to named templates (functions), to handle cases where recursive calls are useful.

XO may be seen by experienced XSLT users as a constraining subset of XSLT without useful features such as ordering of outputs and unique selection. The value of XO for such users is in its library of XSL utilities which provides many specific XML-to-OWL capabilities. XO offers several options for the generation of standardised identifiers (xr:uriSelect), including support for the camel case naming convention for URIs as recommended for XML Schemas, whereby concatenated words each commence with a capital letter.

For less advanced users, the main advantage of XO is its lightweight syntax: the file written by the XO user is roughly 5 times smaller than the XSLT file which implements the transformation.

XO is deployable as a servlet embedded in a Wiki Wiki environment, JSPWiki (Jalkanen 2004). This has proved to be an effective solution to managing the information required to understand and execute the transformations. XO users can define their own Wiki Wiki pages through which transformations can be executed (Figure 2). Also embedded in the environment are adapted forms of the OWL API presentation and validation servlets developed by Bechhofer et al. (2003).



**Fig. 2: A page from the Ontology Wiki Wiki**

## 4 Colours

To demonstrate our semantic integration approach, we have defined a typical e-procurement scenario offering a unified service for procurement of office stationery from independent stationers. To enable semantic integration over a range of supplier catalogue databases and colour-based querying over products such as pens or paint tubes, we need a rich reference ontology of colour. Colour is a domain which is at the same time rich enough to justify the adoption of semantic web technologies and easy enough to understand for people without prior knowledge of it.

It is also a challenging example for ontology engineering. It is at the crossroads of many domains from biology and horticulture to geology; from plastics and textile to

printing industries. Several companies (e.g. Pantone) hold proprietary colour systems and compete through their branding power and their ability to predict how a colour will be rendered on a large variety of substances (paper, textile, plastics, and other material used in interior and industrial design). These systems may form the basis of a universally accepted system, as in the example from Berners-Lee (2003). However, they are also becoming too complex for the majority of non-professional users which now take for granted the ability of the digital imaging chain to capture and reproduce colours on a computer screen (an active source of light) as well as in print (with light coming from another source).

Masolo et al. (2002) have taken into account the work of philosophers such as Gärdenfors (2000) before placing the quale concept at the top of the concept hierarchy used for colours. Quale is defined as “a property, such as whiteness, considered independently from things having the property”. In DOLCE, it is completed by quality which is used to characterize its occurrence in other entities. Bateman and Farrar (2004) suggest that representations such as the Region Connection Calculus (Cohn et al. 1997) would be useful as well, to add the spatial theory foundation which is required to describe relationships between colour regions.

Natural colour systems generally divide the colour space into three dimensions: hue, brightness (black - white scale) and chroma or saturation (mixed - vivid scale). Computer colour schemes generally use three values for screen displaying and up to six for high-quality printing.

The basis of our colour ontology is open information, based on a 3D colour space which is structured according to the NBS/ISCC system, a rigorous and user-friendly division of the colour space into 267 centroids. This information is the result of collaborative work of Mundie (1995), Foster (2004) and Bilik (2005) who have collected data from a range of public sources. We have used five different types of information from Foster's web site:

- The 267 NBS/ISCC centroid definitions (in one web page);
- The 5000 colours defined relatively to their centroids (26 pages, alphabetical order);
- The individual colours listed according to their Hue positions (360 pages: one per degree in the Hue);
- The individual colours listed according to their HTML codes (495 pages); and
- The useful notes identifying the bibliographic references.

This web site has been selected because most of the useful information is structured as HTML tables and because all the pages for each of the collections listed above are formatted with the same conventions. It is easy to transform such content into a single XML file per collection without losing track of the information's origin.

Further, the web site has been selected because of the amount of information that is available and the care that has been taken by the authors to keep the linkage between the amassed information and the sources from which it is derived. In the next section we will present some of the ontology design decisions we have made to process this set of web pages after their pre-processing into a limited number of larger XML files.

## 5 Large scale ontology generation

Our objective is to transform existing knowledge to get the best possible result. The ontology design instructions contained in the XO specification are applied repeatedly and rigorously over the inputs. This enables the XO user to tune the output to take account of factors such as the size, quality and ease of use.

To do so, the XO user must:

- Select OWL patterns to be applied over the input, to extract the useful information from the source;
- Choose between the use of abstract definitions (corresponding to T-Box modelling in description logic terminology) or instantiated ones (corresponding to A-Box modelling);
- Break the result into manageable parts, and manage the links to other sources of knowledge; and
- Decide if it is most effective to embed imported knowledge inside the resulting ontology, or instead to leave it out, to be made available as a database or service resource.

The first category of design decisions (the first and second points above) is relative to the OWL language. Writing the XO specification is a comparable task to writing directly in OWL and the same guidelines (Rector et al. 2004) can be followed. In practice, upper definitions from relevant ontology samples are good candidates to form the skeleton of the generated ontology. For relations between colours, we follow the Region Connection Calculus theory and its implementation from the SOUPA project (Chen et al. 2004).

The second category of design decisions (the third and fourth point above) is driven by the large scale factor of our colour example. We can play on two independent factors to control the generated output in terms of scalability: the selected OWL flavour and the decision to include or exclude the A-Box part from the result. OWL is designed in three flavours: OWL Lite, OWL DL and OWL Full, which are the results of a conscious trade-off in the language features between expressiveness and tractability for later exploitation through description logic reasoners. The alternative we are seeking for the OWL output and more generally the A-Box data is to provide it as part of our Semantic Integration solution through an extra resource such as a database.

Our tactic here is to generate two types of outputs simultaneously: the OWL DL ontology without the A-Box data, and the rest. In our colour example, we keep

data such as the hexadecimal colour values (HTML code) in the second output and encode it using OWL individuals.

The other design decisions are more dependent on the scale of the material we start from. It is important to mirror the intrinsic structure found in the original material to capture the choices made by the prime author to manage complexity. This has many advantages:

- It makes the transformation process simple enough and shareable by more than one person; and
- It simplifies the result validation and configuration management tasks.

## 6 The resulting colour ontology

In this paper, we highlight some of the differences between the colour ontology generated with XO from the selected colour resource and one of the major references in the colour domain, the Art and Architecture Thesaurus from the Getty Research Institute (2004). The thesaurus defines more than 130000 concepts in total. The colour hierarchy is one of the top 20 term hierarchies and is using the NBS Color Dictionary (Kelly and Judd 1997) as its main reference for colours.

This reference is also the book which has been used by the author of the web pages we start from (Foster 2004), which makes us believe that the size of our colour ontology is comparable to the colour subset of the Getty Thesaurus. Foster has annotated each colour with a coding system which points back to the secondary references cited in this original source. We have adapted the Scheme management template defined by the SKOS project (Miles 2004) to link each colour class to its relevant reference(s). We have defined 15 “schemes” in total, covering different fields with different colour needs (plastics, horticulture, and biology). This traceability information is missing from the Getty Thesaurus online service.

Our Colour ontology contains 12471 classes (almost exclusively named colours and/or colour regions) and 9657 individuals (instances of colours such as hue intervals and HTML codes), according to SWOOP (Mindswap 2005). Most of the relations between classes are presently expressed through the “proper part of” relation specified by the Region Connection Calculus. HTML Codes are defined as instances of the Colours. The ancillary information is exclusively provided through annotation properties to make the result manageable by tools such as Protégé and SWOOP.

To get this result, we are using 600 lines of XO instructions grouped into 25 templates. Most templates are using one pattern, except the first one which groups all the top class declarations. The corresponding XSL transformation is more than 11000 lines, of which roughly two thirds corresponds to the HTML documentation which is extracted out of the XO code.

We plan to further refine the ontology design instructions with the help of a reasoner to evaluate the result (or a

manageable subpart of it). In the next section, we discuss the quality issues to be addressed to tune the XO output according to our needs.

## 7 Quality issues for XO-created ontologies

XO has been developed to support the preparatory work necessary for the semantic integration of large collections of data sources and software components. To understand the objectives in terms of quality for the generated ontology, we need to describe the specific requirements of three categories of stakeholders (Ackland et al. 2005):

- Infrastructure providers supplying a consistent (and potentially large) set of reference ontologies.
- Resource providers supplying resource knowledge, including resource-specific ontologies and mapping rules that relate resources to reference ontologies.
- End users consuming the knowledge provided by the two first categories of stakeholders to define the virtual service model corresponding to their problem.

To meet the expectations of these stakeholders in terms of quality, one key challenge is to address a large range of quality objectives. The evaluation methodology presented by Gangemi et al. (2005) is a good reference to understand the relationships between “low level” quality parameters and broader quality objectives (or “principles”). Some of these quality objectives are important for the user (cognitive ergonomics, transparency, compliance to expertise). Others are important for the reasoner (computational integrity and efficiency, flexibility/modularity).

With XO, we cannot use the reasoner at the time the design instructions are created. This major difference between XO and tools such as Protégé has two major benefits:

- XO users must focus first on their ability to capture all the information from the original source before tuning the ontology design instructions to improve the result for the reasoner-related requirements.
- XO users can experiment with different ontology design strategies to better understand how to manage the concurrent quality objectives. They can even implement multiple transformations out of the same knowledge source.

XO facilitates an incremental ontology generation process without breaking the linkage to the original knowledge. This transparency of the process is critical to get user engagement and confidence in the ontology validity. Users of the XO-generated ontology can later assess the whole transformation process applied to the original sources and not just the result. The Wiki Wiki environment we provide can also help to provide the ancillary information which is too frequently missing for ontologies generated with other tools.

We have discussed above the difficulty to find the right compromise between partially conflicting quality requirements. With XO, we can create two versions of the same ontology which are coherent with each other, but designed to be used for different purposes. A larger ontology could help the resource providers and end users to interact with the system, especially during the preparation phase. At runtime, a more compact version could be actioned upon by the reasoner to handle queries and compose services from diverse network resources on the basis of the mapping statements (Cameron and Taylor 2005). This ability to dynamically generate ontologies could also be very important to manage performances and useability issues linked to the size of the generated ontologies because it is generally hard to partition large ontologies into smaller modular ones.

## 8 Conclusion

Ontology acquisition and model transformation are critical for builders of systems of systems where the challenge is the semantic integration of heterogeneous resources into a domain-rich and customer-focused network-centric application.

The colour example we have presented here illustrates the processing of relatively large resources relating different systems of references. We plan to release the generated output and to engage with the community of colour specialists to further improve the generated result. We would like other researchers to build on our work to better address issues such as the variability in the human eye perception of colours and to investigate how cultural factors influence the relationships between colour names and things which exist in the real world.

Our tool can be used to generate ontologies without losing track of the original pieces of knowledge, and to experiment with OWL patterns on a scale which is rarely seen elsewhere. With it, available knowledge can be reused more easily and more rigorously. Manual errors are less likely and the reasoning involved to put the knowledge together is more explicit. Our method also respects the initial choice made by the original expert to break out the original information into manageable parts.

We have designed XO to help us to generate the ontologies which are required for the semantic integration of services. With it, the ontologist can focus on ontology design instructions and better deal with conflicting quality issues which are hard to manage with alternative approaches.

We should thank again the contributors to the colour resources which we have used here for their effort in putting together such a mass of information in a format which is relatively easy to process. It is also possible to find such valuable input in other domains, such as Airborne Early Warning and Control where XO has been used to study the integration of civil and military resources characterising air and ground targets.

## 9 References

- Ackland, R., Taylor, K., Lefort, L., Cameron, M. and Rahman, J. (2005): Semantic Service Integration for Water Resource Management. In *Proc. of the 4<sup>th</sup> International Semantic Web Conference (ISWC 2005)*, Galway, Ireland. Lecture Notes in Computer Science 3729 (to appear), Gil, Y.; Motta, E.; Benjamins, V.R.; Musen, M. (Eds.), Springer.
- Bateman, J. and Farrar, S. (2004): Towards a generic foundation for spatial ontology. In *Proc. of the Third International Conference on Formal Ontologies in Information Systems (FOIS'04)*, Torino, Italy. Varzi, A. and Vieu, L. (Eds.), IOS Press.
- Bechhofer, S., Lord, P. and Volz, R. (2003): Cooking the Semantic Web with the OWL API. In *Proc. of the 2003 International Semantic Web Conference (ISWC 2003)*, Lecture Notes in Computer Science 2870. Fensel, D., Sycara, K., Mylopoulos, J. (Eds.), Springer.
- Berners-Lee, T. (2003): Standards, Semantics and Survival World Wide Web Consortium *SIIA Upgrade Magazine*, June/July:6-10, Software and Information Industry Association.
- Bilik Y. (2005): *Pourpre.com - les mondes de la couleur, les couleurs du monde*. <http://pourpre.com/> Accessed 17 Oct 2005.
- Birrer, I. (2005): *XSLTdoc - A Code Documentation Tool for XSLT*. P&P Software GmbH <http://www.pnp-software.com/XSLTdoc/> Accessed 26 Jul 2005.
- Cameron, M. and Taylor, K. (2005): First-Order Patterns for Information Integration, In *Proc. of the 5<sup>th</sup> International Conference on Web Engineering (ICWE 2005)*, Sydney, Australia, Lecture Notes in Computer Science, Volume 3579:173-184, Lowe, D., Gaedke, M. (Eds.), Springer.
- Cohn, A.G., Bennett, B., Gooday, J.M. and Gotts, N. (1997): RCC: a calculus for Region based Qualitative Spatial Reasoning, *GeoInformatica* 1(3):275-316, Springer.
- Chen, H., Perich, F., Finin, T., and Joshi, A. (2004): SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications In *Proc. of the First International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2004)* Boston, MA, USA. T. Finin, C. Ghidini, T. La Porta, C. Petrioli (Eds.), IEEE Computer Society.
- Cristani, M. and Cuel, R. (2005): A Survey on Ontology Creation Methodologies. *International Journal on Semantic Web and Information Systems*, 1(2). Idea Group Publishing.
- Damjanovic, V., Devedžic, V., Djuric, D. and Gašević, D. (2004): Framework for Analyzing Ontology Development Tools. *AIS SIGSEMIS Bulletin* 1(3):43-47. Association of Information Systems.
- Foster, J.C. (2004): *The mother of all HTML color charts*. <http://tx4.us/moacolor.htm> Accessed 17 Oct 2005.

- Gangemi, A. (2004): *Reusing semi-structured terminologies for ontology building: A realistic case study in fishery information systems*. WonderWeb Deliverable D16, ISTC-CNR, Rome, Italy.
- Gangemi, A., Catenacci, C., Ciaramita, M. and Lehman J. (2005): *Ontology evaluation and validation, An integrated formal model for the quality diagnostic task*. Technical Report. Laboratory for applied ontology, ISTC-CNR, Roma/Trento, Italy.
- Gärdenfors, P. (2000): Concept combination: a geometrical model, In *Logic language and Computation* 3:129-146, Cavedon, L., Blackburn, P., Braisby, N. and Shimojima, A. (Eds.), CSLI, Stanford, CA.
- Gašević, D., Djuric, D., Devedžić, V. and Damjanovic, V. (2004): Converting UML to OWL Ontologies. In *Proc. of the 13th International World Wide Web Conference (WWW 2004)*, New York, USA, 488-489. ACM.
- Getty Research Institute (2000): *Getty vocabularies program: Art and Architecture Thesaurus Online* [http://www.getty.edu/research/conducting\\_research/vocabularies/aat/](http://www.getty.edu/research/conducting_research/vocabularies/aat/) Accessed 17 Oct 2005.
- Grønmo, R., Oldevik, J. (2005): An Empirical Study of the UML Model Transformation Tool (UMT). In *Proc. of the First International Conference on Interoperability of Enterprise Software and Applications (INTEROP-ESA'05)* Geneva, Switzerland. Lecture Notes in Computer Science 3579 (to appear)., Konstantas, D.; Bourrières, J.-P.; Léonard, M.; Boudjlida, N. (Eds.), Springer.
- Jalkanen, J. (2004): *JSPWiki* <http://www.jspwiki.org/> Accessed 10 Aug 2004.
- Kelly, K.L. and Judd, D.B. (1976): *Color Universal Language and Dictionary of Names*. National Bureau of Standards special publication 440, Washington, DC: U.S. Department of Commerce.
- Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A. and Schneider L. (2002): *The WonderWeb Library of Foundational Ontologies and the DOLCE ontology*. WonderWeb Deliverable D17. , ISTC-CNR.
- Mindswap (2005): *SWOOP - A Hypermedia-based Featherweight OWL Ontology Editor* <http://www.mindswap.org/2004/SWOOP/> Accessed 23 Aug 2005.
- Miles, A. (2004): *Simple Knowledge Organisation System (SKOS)* <http://www.w3.org/2004/02/skos/> Accessed 17 Oct 2005.
- Miles, A. (2005): *Quick Guide to Publishing a Thesaurus on the Semantic Web*. <http://www.w3.org/TR/swbp-thesaurus-pubguide/> Accessed 17 Oct 2005, W3C.
- Mundie, D.A. (1995): *The NBS/ISCC Color System*, <http://www.anthus.com/Colors/NBS.html> Accessed 17 Oct 2005.
- The Object Management Group (2004): *MOF Query / Views / Transformations - Second Revised Submission*, ad/2004-01-06. Duddy, K., Lawley, L., Iyengar, S., Gerber, A., Raymond, K. and Steel, J., OMG.
- The Object Management Group (2005): *Ontology Definition Metamodel Second Revised Submission to OMG/RFP*, ad/2005-08-01. Chang, D.T., Kendall, E.F. et al., OMG.
- Pivk, A., Cimiano P. and Sure, Y. (2005): From Tables to Frames. *Journal of Web Semantics*, 3(2), <http://www.websemanticsjournal.org/ps/pub/2005-21> Accessed 17 Oct 2005.
- Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H. and Wroe, C. (2004): OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns. In *Proc. of the European Conference on Knowledge Acquisition (EKAW 2004)*, Northampton, England, Lecture Notes on Computer Science, 3257:63-81., Motta, E.; Shadbolt, N.; Stutt, A.; Gibbins, N. (Eds.), Springer.
- Tijerino, Y., Embley, D., Lonsdale, D. and Nagy, G. (2003): Ontology generation from tables. In *Proc. of the 4th International Conference on Web Information Systems Engineering (WISE 2003)*, Rome, Italy. 242-249. IEEE Computer Society.
- Visscher, S. (2004): *XR: RDF Extraction from XML*, <http://w3future.com/xr>, Accessed 5 Aug 2004.
- Wielinga, B., Wielemaker, J., Schreiber, G. and van Assem, M. (2004): Methods for Porting Resources to the Semantic Web. In *The Semantic Web: Research and Applications. Proc. of the First European Semantic Web Symposium ESWS 2004*, Crete. Lecture Notes in Computer Science 3053:299-311. Bussler, C., Davies, J., Fensel D. and Studer R. (Eds.), Springer.