ONTODELLA—A Projection and Linking Service for Semantic Web Applications

Kim Viljanen, Teppo Känsälä, Eero Hyvönen, Eetu Mäkelä
Semantic Computing Research Group (SeCo)
Helsinki University of Technology (TKK), Laboratory of Media Technology and
University of Helsinki, Department of Computer Science
P.O. Box 5500, FIN-02015 TKK, Finland
firstname.lastname@tkk.fi, http://www.seco.tkk.fi

Abstract

Content in semantic web portals is often projected along application specific navigational taxonomies and linked semantically. This paper presents a logic-based method and a server ONTODELLA for these tasks. We argue that logic rules between the content layer and the application layer add flexibility and better architectural separation of content and functionality. The system has been implemented and applied succesfully in several semantic portals.

1 Introduction

One way to utilize semantic web [2] content (ontologies and metadata) is to create semantic portals [12], which provide the users with semantically enhanced functionalities for finding and browsing content described with rich metadata and ontologies. The main functionalities of a semantic portal are semantic search and linking [13].

When using ontologies for information retrieval, ontologies may not, however, be easy and intuitive to use for the end-user. This is because the ontologies may contain irrelevent concepts, e.g., too abstract concepts (e.g., `owl:Thing`) or concepts that are not related to the user's information retrieval goal (e.g., using an extensive medical ontology when searching for information about dangers of smoking). The ontology may also have a wrong point of view compared to the user's goal. E.g., a medical ontology might be organized by the traditional scientifical way of dividing medical fields compared to the user's need to find everything that is related to dangers of smoking regardless of the medical field.

Metadata about individual resources may also create usability problems from the end-user's point of view if the user is confronted with the raw metadata which may contain, due to modelling reasons, complex structures for presenting relatively simple knowledge. In such cases, the indirect links could be more useful for the end-user as a navigational structure than if only direct relations were displayed. E.g., a person x works as professor at the department y at the university z. The relation between x and z is complex.

As shown above, semantic web knowledge is not always useful for the end-user as such. To solve the problem of mismatch between raw semantic web knowledge and the user's needs, we propose a rule-based inference service ONTODELLA with two specialized functions: ontological projection and semantic link generation.

View projections are used for transforming a given semantic content to navigational taxonomical views (facets), to be used for navigating and browsing the underlying content hierarchically in the same fashion as in, e.g., Yahoo. Projected views are designed based on the users' needs, which makes them often more usable for navigation and browsing than the original content. A projected view can be used for creating a navigational structure for a static web site [9]. The result of a projection can also be several views, which can be used for doing *multi-facet searching* of the underlying content by restricting the resultset from several orthogonal views at the same time. The benefits and usability of multi-facet search has been demonstrated, e.g., in Flamenco [6], MuseumFinland[1] [10], and SWED[2].

Semantic link generation links automatically related items based on rich metadata and ontologies used for describe the items. For example, in MuseumFinland museum artifacts related to the same life event, such as weddings, are linked [10]. Automatical linking may highlight relations that might not be discovered otherwise. For example, in the Orava case we have created links both inside the learning video collection and also inter-portal links to an other collection (MuseumFinland). Automatical linking helps refine relations between items by hiding unrelevant details be-

---

[1]http://www.museosuomi.fi/
[2]http://www.swed.org.uk/

tween related resources. Automatical linking is related to many research fields, such as recommender systems [15, 4], open hypermedia [7, 5] and semantic associations [1].

ONTODELLA helps the application designer to separate the applications from the semantic web content. Eventual changes in the underlying content do not affect the application, since only the ONTODELLA rules have to be changed. The ONTODELLA has been created as a part of the ONTO-VIEWS framework presented earlier in [13].

The paper is structured as follows: First we describe the view projection and semantic linking functionalities. Then we present some details about the implementation. In the fifth chapter we present some case applications using ONTODELLA. Section six presents related work. Finally, discussion about the results and the conclusions are presented. The work was mainly funded by The National Technology Agency's (Tekes) Fenix program.

2   View Projection

The ONTODELLA projections are defined as logical rules using Prolog[3]. They describe how the views are projected from the source content (ontologies and metadata). A view is a hierarchical index-like projection of the source content where each category is associated with a set of subcategories and data items. A view is defined by specifying a predicate `ontodella_view` with the following parameters: 1) the root URI which defines the resource to be used as the root of the view, 2) a binary subcategory relation predicate which defines the subcategories for the current category, and 3) a binary predicate that maps the hierarchy categories with the items used as leaves in the view. In addition, each view must have a label.

An example[4] of a view predicate is given below:

```
ontodella_view(
  ns:'#earth',          % root resource URI
  place_sub_category,   % subcategory relation
  place_of_use_leaf_item, % leaf relation
  [fi:'Käyttöpaikka',   % the labels
   en:'Place of Use']
).
```

The label list contains the labels for each supported language. Here Finnish (fi) and English (en).

The subcategory predicate `place_sub_category` can be based, e.g., on a containment property `isPartOf` in the following way:

```
place_sub_category(ParentCategory,SubCategory):-
  rdf(SubCategory,ns:'#isPartOf',ParentCategory).
```

The leaf predicate describes when a given resource item is a member of the given category. For example,

---

[3]http://www.swi-prolog.org

[4]The SWI-Prolog syntax is simplified due to presentation reasons. All namespaces have also been replaced with the generic ns: namespace.

`place_of_use_leaf_item` in our example above can be described as follows:

```
place_of_use_leaf_item(ResourceURI,CategoryURI):-
  rdf(ResourceURI, ns:'#usedIn', CategoryURI).
```

There may exist more than one predicate with the same name, which adds flexibility for defining the projection rules. For example, several leaf predicates may exist, which describe all the posibilities for how an item can be a member of a category. It is also possible to describe facts with predicates. E.g., the following predicate describes that `#myBird` is one leaf of the `#Birds` category:

```
place_of_use_leaf_item(ns:'#myBird',ns:'#Birds').
```

Based on the rules, the projection can be generated by iterating through the predicate `ontodella_view`, and by recursively creating the category hierarchies using the subcategory rules starting from the given root resource URI. At every category, all relevant leaf resources are attached to the category based on the leaf rules.

The result of the projection is presented as an RDF document conforming to a simple OWL ontology designed for representing the hierarchical structure of the views and the leaf items. Each category contains the following information: 1) label(s) of the category, 2) subcategories, and 3) leaf items. Additional information about the categories and about the leaf items can be added to the respective RDF resources. This information can be used for application specific purposes, such as showing the user more information than just the label of each category and leaf item. The additional information is selected based on data selection rules when creating the categories and leafs.

3   Semantic Link Generation

ONTODELLA's link generation is based on rules that describe when two resources should be linked and why. Each link rule can be arbitrary complex and is defined by a domain specialist. A linking rule p is of the form $p(Subject, Target, Explanation)$ that evaluates true when the two resources $Subject$ and $Target$ are to be linked. The $Explanation$ is bound to a human-readable explanatory label and a machine readable URI. The explanation URI can, e.g., refer to the ontological concept which connects $Subject$ and $Target$.

In the following, a linking rule used in MuseumFinland is presented in details. The rule describes the prerequisites for linking two museum artifacts based on a life event, such as weddings.

```
related_by_event(Subject,Target,Explanation) :-
  isArtifact(Subject),
```

Find all (transitive) item type classes for the subject:

```
rdf(Subject,ns:'#itemType',SubjectItemType),
rdfs_transitive_subClassOf(SubjectItemType,
    SubClassOfSubjectItemType),
```

Find all life events, e.g. weddings, that the item types above are related to:

```
rdf(SubClassOfSubjectItemType,
    ns:'#relatedToEvent',Event),
```

Find the (transitive) subclasses and the superclasses of the event:

```
(
 rdfs_transitive_subClassOf(Event,RelatedEvent)
;
 rdfs_transitive_subClassOf(RelatedEvent,Event)
)
```

Find any potential link target item type, which is related to the sub- or superclass of the event:

```
rdf(TargetItemType, ns:'#relatedToEvent',
    RelatedEvent),
```

Find all (transitive) subclasses of the target item type found above:

```
rdfs_transitive_subClassOf(
    SuperClassOfTargetItemType,TargetItemType),
```

To exclude uninteresting links, check that the subject item type is not the same (transitively) as the target type:

```
SuperClassOfTargetItemType \= SubjectItemType,
not(rdfs_transitive_subClassOf(
    SuperClassOfTargetItemType,SubjectItemType)),
not(rdfs_transitive_subClassOf(
    SubjectItemType,SuperClassOfTargetItemType)),
```

Find all the artifact items related to the target type class found above. Check also, that the subject and target items are not equivalent:

```
rdf(Target,ns:'#itemType',
    SuperClassOfTargetItemType),
isArtifact(Target),
Subject \= Target,
```

Finally, when a target link has been found, create the explanation based on the common event of the subject and target artifacts:

```
list_labels([RelatedEvent], RelLabel),
Explanation=[commonResources(RelatedEvent),
        label(fi:RelLabel)].

% (rule ends)
```

Each linking rule must be presented to ONTODELLA by creating a ontodella_relation_rule predicate, which defines the linking predicate, the label of the link and the URI identifier for the link. An example of the relation rule declaration is the following:

```
ontodella_relation_rule(
  related_by_event,        % relation predicate
  [fi:'Liittyy tapahtumaan', % label
   en:'Related by event'],
  rel:'#related_by_event'  % relation URI
).
```

The links for a specific subject are generated when ONTODELLA receives a HTTP query with the subject URI as the parameter. For example:

```
http://localhost:1234/get_relations?subject=[URI]
```

where the *[URI]* is the url-encoded subject URI.

As a result, the relevant links are returned as an RDF/XML document containing the linking information. The RDF/XML document conforms both to an RDF Schema and an XML Schema. By using both RDF and a fixed XML structure for the linking information, the application designer can choose whether to handle the information as RDF or XML when displaying the results in the user interface. For example, in the ONTOVIEWS system we used XSL transformations for displaying the links [13].

4 ONTODELLA Implementation

ONTODELLA is implemented in SWI-Prolog[5] using it's HTTP server and semantic web packages [17]. ONTODELLA can be queried via the HTTP connection for getting semantical links. The ontological projections are created as batch process due to the file sizes of the result projection and due to the time that the creation of the projection may take. E.g., in MuseumFinland the size of the projection file is about 24 MB and the creation time about 10 hours with a modern PC. The MuseumFinland contains about 10 000 ontological concepts and about 4000 artifacts and archelogical sites.

ONTODELLA's architecture (figure 1) is based on the *software hooks* software design pattern, where rules can be attached to the ONTODELLA "hooks" by defining ontodella_view and ontodella_relation_rule predicates as described above. When ONTODELLA receives a query from the user, relevant hooked predicates are executed to generate, e.g., semantical links. When ONTODELLA is started, the general settings file is read, which declares, e.g., which other files should be loaded and which port the ONTODELLA should be running at. Next, all the other files, such as rules, RDF content, ontologies, and the library rules are loaded to ONTODELLA. Settings and rules are described in Prolog syntax, RDF content and ontologies are expressed using RDF.
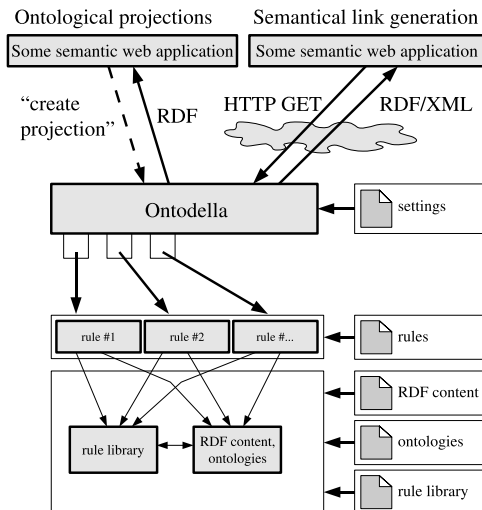
---

[5]http://swi-prolog.org/

Figure 1. The architecture of ONTODELLA**.**

Figure 2. The Orava user interface

5 A Portal for Educational Videos

ONTODELLA has been applied in several semantic portals [13, 11, 16, 14, 9]. As an example, this section provides a discussion of Orava[6], which is a prototype of a semantic portal for educational videos from the Finnish Broadcasting Company's Klaffi service[7]. The Klaffi videos were described using the Learning Object Metadata (LOM)[8]. Ontologies were modeled based on LOM and by creating ontologies based on keywords found in the original Klaffi metadata. Facets were projected directly based on the ontologies using the `rdfs:subClassOf`-relation. The Orava portal was created using the ONTOVIEWS [13] framework.

Orava contains two kind of semantical links: links to other videos inside Orava and inter-portal links to MuseumFinland. The later case demonstrates how separate portals can be connected using the semantic web technologies. Figure 2 presents the Orava user interface for one video. Navigational views are on the left. Information about the video is in the upper part of the middle section. Below it are the semantical links to other videos and the inter-portal links to MuseumFinland.

The following semantic link rules were implemented: 1) Link videos that share a common theme and the target audience. 2) Link videos based on series relation, such as the "next part of series", "previous part of series" and "same part of series". 3) Link videos to artifacts in MuseumFinland if they share a common theme. 4) Link videos to artifacts in MuseumFinland if they share the "place of manufacture" or "place of use".

A problem with creating inter-portal links between Orava and MuseumFinland was that they use different ontologies as they concentrate on different domains. It was necessary to map Orava's ontological concepts to MuseumFinland's concepts. The mapping was achieved by using string matching for matching concepts based on their labels and checking validity of the mappings manually.

6 Related work

Semantic Web Environmental Directory (SWED)[9] portal is a prototype of a semantic web portal for finding environmental organisations based in the United Kingdom. The views in SWED are defined by declaring the root concept and the traverse property either as a *narrower-than* or *broader-than* relation[10]. SWED does not provide the possibility for defining more complex hierarchy traverse rules as in ONTODELLA, but supports inference (forward chaining) when loading new datasources to the system. Only forward chaining is supported, since the system uses a database system for maintaining the RDF content. In ONTODELLA all data is loaded to main memory, which is a viable solution as long as the amount of RDF content is less than a few million RDF triplets [17].

Dolog et al. [5] presents a rule-based system for creating links between information artifacts in the context of educational content. The content is modelled from the users point-of-view and describe, e.g., which are the prerequisited documents for each document. This information can be used for linking related documents. In ONTODELLA's

---

[6]http://demo.seco.tkk.fi/orava
[7]http://www.yle.fi/klaffi/
[8]http://ltsc.ieee.org/wg12/

[9]http://www.swed.org.uk/
[10]http://www.swed.org.uk/swed/doc/portal-customization.html

case applications the item metadata describes only the *content* of the objects, not the *relations between objects*. Both systems use logic for presenting the linking rules. Dolog et al. use the TRIPLE language for describing the rules whereas ONTODELLA uses Prolog.

Knowledge-based recommender systems recommend items based on metadata about the items combined with functional knowledge about the domain and the users' needs [3]. For example, restaurants can have qualities such as cuisine, location and price level. Using such information combined with similarity measures, links to restaurants that are "cheaper" or with "a better view" can be generated. Like Burke's system [3], ONTODELLAprovides links to relevant other items based on functional knowledge and metadata about the items. In ONTODELLA the metadata is described using semantic web languages (RDF, OWL), whereas Burke uses ordinary databases for the data.

7 Discussion and Conclusions

This paper presented a rule-based approach for creating view projections and semantic linking, two major functions needed for creating semantic enhanced web applications, such as semantic portals. Using rules and inference between the content layer and the application layer adds flexibility and better architectural separation of content and functionality. For example, when adapting ONTODELLA to the different case systems, such as MuseumFinland and Orava, only the rules had to be changed for adapting the ONTO-VIEWS framework to new annotation schemes and content.

An important advantage of using rules and ontologies is the ability to create a human-readable explanation for the semantical links. This is in contrast with techniques such as collaborative filtering where creating explanations have been difficult to create [8].

Future work includes researching possibilities for making the creation of the projection and linking rules easier for domain specialist without programming skills. Also techniques for ordering the projections and generated links by relevance should be researched.

References

[1] K. Anyanqu and A. Sheth. ρ-queries: Enabling querying for semantic associations on the semantic web. In *Proceedings of the 12th World Wide Web Conference*, pages 690–699, Budapest, Hungary, 2003.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[3] R. Burke. Knowledge-based recommender systems. In A. Kent, editor, *Encyclopedia of Library and Information Systems*, volume 69, Supplement 32. Marcel Dekker, New York, 2000.

[4] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modelling and User-Adapted Interaction*, 12(4):331–370, November 2002.

[5] P. Dolog, N. Henze, and W. Nejdl. Logic-based open hypermedia for the semantic web. In *Proceedings of 1st International Workshop on Hypertext and Semantic Web, International Hypertext Conference, August 26-31, Nottingham, UK*, 2003.

[6] J. English, M. Hearst, R. Sinha, K. Swearingen, and K.-P. Lee. Flexible search and navigation using faceted metadata. Technical report, University of Berkeley, School of Information Management and Systems, 2003. Submitted for publication.

[7] C. Goble, S. Bechhofer, L. Carr, D. D. Roure, and W. Hall. Conceptual open hypermedia = the semantic web? In *Proceedings of the WWW2001, Semantic Web Workshop, Hongkong*, 2001.

[8] J. H. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Computer Supported Cooperative Work*, pages 241–250. ACM, 2000.

[9] E. Hyvönen, M. Holi, and K. Viljanen. Designing and creating a web site based on RDF content. In *Proceedings of WWW2004 Workshop on Application Design, Development, and Implementation Issues in the Semantic Web, New York, USA*. CEUR Workshop Proceedings, Vol-105, 2004. http://ceur-ws.org.

[10] E. Hyvönen, E. Mäkelä, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnila, and S. Kettula. MuseumFinland – Finnish Museums on the Semantic Web. *Journal of Web Semantics, Elsevier*, 3(2):224–241, 2005. Full length version of a paper to be published by the Journal of Web Semantics, http://www.seco.tkk.fi/publications/.

[11] E. Hyvönen, A. Styrman, and S. Saarela. Ontology-based image retrieval. In *Towards the semantic web and web services, Proceedings of XML Finland 2002 Conference*, pages 15–27, October 21–22 2002.

[12] A. Maedche, S. Staab, N. Stojanovic, R. Struder, and Y. Sure. Semantic portal — the SEAL approach. Technical report, Institute AIFB, University of Karlsruhe, Germany, 2001.

[13] E. Makelä, E. Hyvönen, S. Saarela, and K. Viljanen. Ontoviews – a tool for creating semantic web portals. In *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, May 2004.

[14] E. Mäkelä, K. Viljanen, P. Lindgren, M. Laukkanen, and E. Hyvönen. Semantic yellow page service discovery: The veturi portal. In *Poster paper, 4th International Semantic Web Conference*, Nov 2005.

[15] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, March 1997.

[16] T. Sidoroff and E. Hyvönen. Semantic e-goverment portals - a case study. In *Proceedings of the ISWC-2005 Workshop Semantic Web Case Studies and Best Practices for eBusiness SWCASE05*, Nov 2005.

[17] J. Wielemaker, A. T. Schereiber, and B. J. Wielinga. Prolog-based infrastructure for RDF: performance and scalability. In *Proceedings ISWC'03, Florida*. Springer–Verlag, Berlin, 2003.