Tying it all together: Inter-operable Topic-Centred Information Portals

Robert H.P. Engels[1] and David Norheim[2]

[1] ESIS, Oslo, Norway
www: http://www.esis.no
[2] Computas AS, Oslo Norway
WWW: http://www.computas.com

NON-PUBLISHED FULL-VERSION DRAFT OF ISWC08 POSTER

Abstract. In order to be more flexible in publishing information and improve accessibility of information for end-users, Oslo Municipality has funded the SUBLIMA project [25]. Within this project a stack of open source Semantic Web and Content Management System (CMS) components is used in order to deliver a flexible solution for publication of meta-data from libraries. Queries from a specific front-end are automatically dispatched to the available SPARQL end-points [23] in a pool of library and archive based installations. Returning results are presented to the user in an integral manner. The final delivery consists of an open source software stack based on Semantic Web Technology and W3C standards. Finally resulting in a well-defined and approved stack of software, major efforts has been spent on licensing issues, immaturity of component parts, ill-defined documentation of software and conversion of older bases into OWL [5]. Experiences and problems have been discussed with and reported into the respective communities.

Key words: Library, Archive, Semantic Web Portal, SPARQL, query dispatching, CMS, Jena, Apache, Cocoon, Struts2

1 Introduction

Norway's public sector has been keen on adapting open standards in governmental organisations. This is reflected in numerous examples of projects from public and non-public entities, for example Skole-Linux (Linux for Schools), adaptation of Open Document formats in the public sector, large governmental information portals based on open standards and in our case the Norwegian Archive, Library and Museum Authority (ABM-utvikling). ABM's program on the Norwegian Digital Library has led to the SUBLIMA project, in which an open standard based, open-source software for creation and drift of topic portals is developed.

Potential users of this software are libraries, archives, musea and other interestees wishing to offer inter-operable topic-based portals to their users. As of today, the library field in Norway shows a wide variety of Web Portals based

on a large diversity of technologies. Common for these platforms is that few use open standards, and even fewer are inter-operable. Typically their main function is to collect, organise, describe and link to external digital resources. In such a way the libraries are able to deliver sets of quality-checked resources for use in f.ex. education, schools or medical environments.

One requirement is that the portals generated with the developed software are inter-operable, meaning that queries posed to one portal could return answers from all accessible portals based on the same technology. A natural consequence of this requirement is the need for standardised meta data.

In order to fulfil these requirements an implementation of a complete stack, based on open source software and W3C endorsed Semantic Web standards has been executed. A significant part of the effort has been used for license discussions, but also immature software, incomplete documentation and conversion of existing databases has at times been a labour-intensive undertaking.

1.1 Implementation: Oslo Public Library (Deichmanske bibliotek)

The Oslo Public Library serves the county of Oslo and is Norways largest public library ([16]). Its services are available for individuals, institutions, students in the public school system and at the university level, the business community and public administration.

The library is spread over sixteen branches/departments in the city. In addition, there are several specialised departments, such as The Multilingual Library, which serves the entire country with its collection in approximately thirty-five languages, a music department, and a department for children and youth. The collection includes both fiction, non-fiction, and other media, in addition to a wide selection of childrens literature.

The librarys activity is based upon a testamentary gift from the estate of Chancellor Carl Deichman (1705-1780). Through this gift, the library was bequeathed a collection of handwriting and approximately 6,000 books, nearly all of which are preserved today. The library places great emphasis on The 24-hour Library, web-based services that are available around the clock, in addition to traditional on-site services.

For this reason, SUBLIMA has been chosen as technical platform for the delivery of new services, based on Semantic Web Technology and inter-operable, open standards. The current portal, called Detektor, consists of about 1850 topics and 4600 resources. This results in about 100.000 triples represented in Turtle [26].

— The library serves the county of Oslo and is Norways largest public library. Our services are available for everyone - individuals, institutions, students in the public school system and at the university level, the business community, and public administration.

Patrons can also choose from a large selection of books in Nordic and other European languages, especially English and German. The majority of departments also offer graphic novels, videos, DVDs and audio books for circulation.

1.2 Implementation at the Medical Library (University of Oslo)

Scandinavian Medical Information for Layman (SMIL) [21] is a topic oriented portal based on a Scandinavian international cooperation. Goal for the consortium, existing of partners from Norway, Denmark and Sweden, is to offer quality controlled meta-data with references to pages related to health, illnesses and treatments. The portal combines the best of information in three languages (Norwegian, Danish and Swedish) and collects this information in a single information hub, while it often happens that the available information on illnesses in Nordic countries is complementing each other. Contributing partners to the portal are librarians and nurses from the Nordic countries. The current SMIL base consists of 8500 records creating around 250.000 triples represented in Turtle.

2 Technology: defining and implementing the SUBLIMA project stack

Since a major requirement for the SUBLIMA stack is genericness, the two organisations mentioned above serve as independent domain providers. In such a way the SUBLIMA stack's independence of specific implementation domains is deemed guaranteed. In addition to this, there are a number of potential users evaluating the two implementations. Amongst them are a juridical portal, the Norwegian Architecture portal and several library portals under the flag of Oslo Public Library. Definition of the stack's components is done based on experience from previous projects, in addition to testing of several components in an in-house environment. The following sections describe design and implementation issues.

2.1 Matching the requirements on an Open Source software stack

The main requirements for the system is its flexibility when it comes to meta-data vocabularies used for annotation, ability to use independent domain model to annotation and full multilinguality. The system was to have full meta-data and free text search capabilities, as well as faceted filtering and domain ontology retrieval. The system should also be able to handle all types of resources on the Web. The users where used to a two level domain taxonomy, but wanted both more flexibility and expressiveness in the modelling. The tool to be developed also needed to be flexible in the design of user interfaces with accessibility requirements. One challenging requirement was that the various implemented portals where to be inter-operable on the query level. Finally, ABM-utvikling wanted the system to have an open source license to maximise the collective efforts in the tool.

The requirements called for a flexible data model, using vocabularies from e.g. Dublin Core [7]. The domain model needed where to be migrated from the existing taxonomy with associative relations as well as full flexibility in the depth of hierarchies. The domain model also needed to be organic, and were to be used

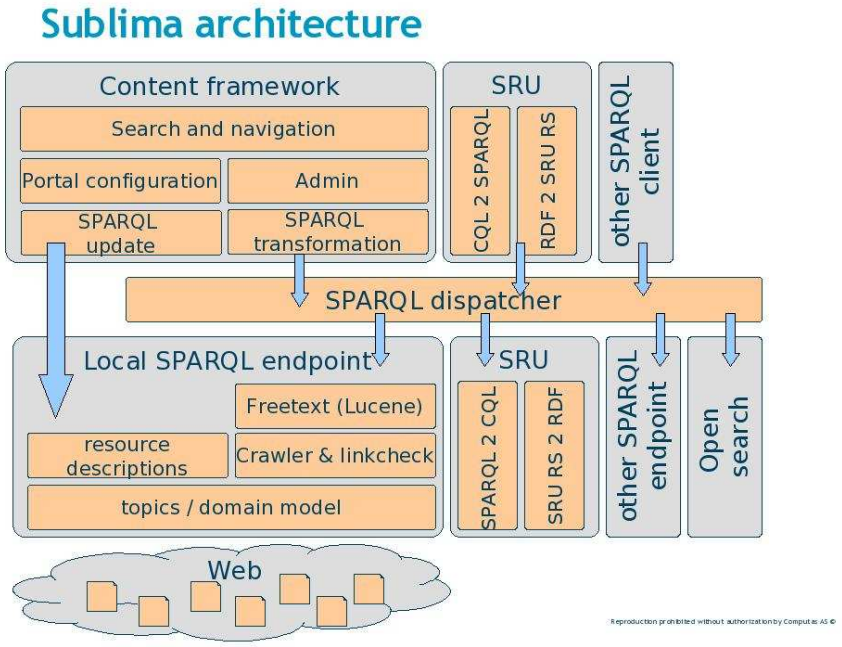

Fig. 1. the Sublima Architecture overview

and maintained by the librarian responsible for each portal implementation.

Based on this, the decision was taken to implement the system using W3C Semantic Web recommendations. Building on previous practical experiences with the use of RDF and OWL in knowledge management and search & navigation scenarios, the project partners applied this experience in the development of the architecture.

The client consists of a search interface allowing users to search using free text and advanced meta-data search. The search string is transfered into a structured SPARQL query to run against the SPARQL dispatcher.

The dispatcher allows federation of the query over various SPARQL endpoints and SPARQL wrappers/transformers to other query languages, e.g. the library sectors CQL language [4].

The architecture back-end consist of an RDF Store with SPARQL interfaces. Though the existing choice is Postgress database with Jena [10] on top (or rather Joseki) the SPARQL interface allows the replacement of the underlying storage with more scalable store when or if the performance becomes an issue. The SPARQL endpoint also makes use of free text indexing using LARQ (an extension of the SPARQL implementation of Jena using Apache Lucene [11]).The topic-ontologies are modelled using SKOS vocabulary as they are currently rather informal hierarchies with associative links. The system allows evo-

lution into a future OWL representation of the ontology. A decision was also made to do an extensive reuse of RDFS and OWL vocabularies (ref. section 2.6).

2.2 Navigating Ontologies and Resources

From experiences with earlier implementations (e.g. [8]) the conclusion was drawn that the framework GUI should be based on topic-based browsing and navigation. Especially when merging meta-data descriptions of different types of resources like books, images, sounds, movies and web pages there is a need for some ordering elements on the meta-data.

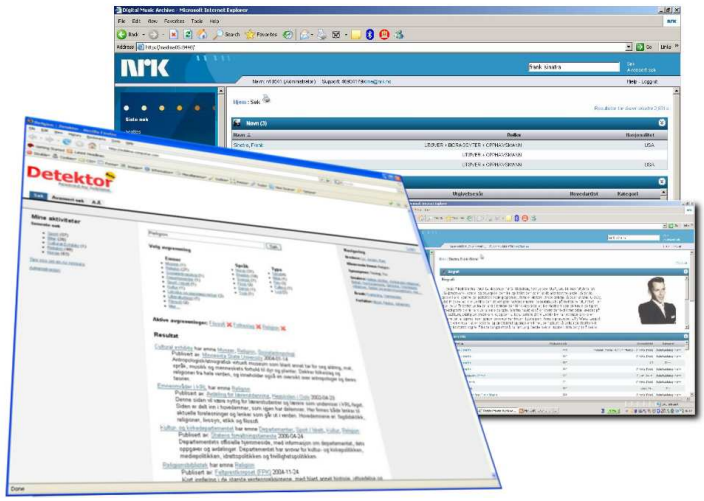

Fig. 2. Some portal based interfaces reflecting the idea of topic-based navigation and facet based filtering[4].

When analysing existing portal technology one sees a gradual shift from simple search interfaces delivering results, towards more sophisticated portal technology that also supports "drill-down" scenarios. In newer portals some of the meta-data elements serve as "facets", e.g. geographic regions, language, date and time stamps etc., which are used for interaction with search results.

In SUBLIMA, focus is on browsing and navigation of a semantic graph structure (ontology or topic-graph). This means that searching for a topic like "deïsm" results in a visualisation of the belonging part of the ontology, including superclasses ("religious philosophy"), subclasses ("agnostism"), sibling classes having the same superclass ("theïsm", "atheïsm"). In addition to ontological constructs based on class-structure, the system also provides synonyms and related

words from thesauri or domain-specific dictionaries. A user can interact with these ontology based visualisations through different devices, using standard mouse/screen interaction or touch screens. Once concepts are selected, a listing of all resources related to a each ontological concept is returned. While browsing the visualisation of the graph part, a visual feedback of resources connected to the graph nodes is provided at any time in the browsing process. This allows the user to select objects of interest and examine them in detail.

Upon finding interesting concepts and their related resources, there are two possible action types available to a user. On the one hand, the user might simply want to filter and interact based on meta-data elements represented as facets. The system is said to be "passive" in such a scenario, waiting for a user to select the right information filters. On the other hand, the system might be "active" and offer potentially interesting facts to the user. In this scenario the system traverses the graph and selects potentially interesting information to visualise to the user. If the user gets triggered by this information, a discovery process is started which might lead to new knowledge on the part of the user.

Current implementations of the SUBLIMA stack use graphical (hyperbolic-view based) visualisations as well as text-based interfaces for browsing and selecting ontology nodes, depending on application domains.

2.3 Stability and Availability

The loose data coupling using SPARQL, allows replacing the back-end storage. Initially the system has been implemented using Postgress and Jena. Future implementations of Sublima are likely to make a transition to using Virtuoso. Our main reason for this change is based on scalability issues and problems with reaction times in the current implementation.

As for the Web framework, the initial decision has been to use Apache Cocoon [1]. Although a well-defined framework, where we made good progress in building a framework for providing CMS-like functionality, we had to re-evaluate this decision. With the wish to use a CMS platform which is supported actively by a large community, new implementations will see a move to the Drupal framework [6], as such making the sublima stack a plug-in module within Drupal. We are also likely to create clients using other frameworks including Struts2 [3] and clients on OS X [17].

2.4 Licensing schemes: commercialisation vs ownership

Initially, before project start, both the supplier and customer had a deep dive into licenses and incentives. Mainly with the goal to come to an agreement on the best open source license to maximise the potential reuse and future development of Sublima, the discussion on licenses ranged from the free software license GPL to the permissive BSD license. A final agreement was reached on a CDDL license (similar to the Mozilla License) giving all parties the best incentives to further develop Sublima. The result has led to a harmonious cooperation and was successful with both research projects (and proposed research projects) and

in discussions on new commercial opportunities in the 6 months after the initial agreement.

2.5 Support and Maintenance while implementing: the role of the community

The use of new W3C specifications (e.g. SPARQL), and ditto open source tools (e.g. Jena) as well as future extensions not yet standardised (e.g. SPARQL Update) for commercial project has a high risk. However, the support from the W3C community and open source community has been impressive. The support through IRC channels, mailing lists etc has been invaluable for the project. We will here especially give credit to the HP Jena team for good and swift follow up on questions and issues.

2.6 Reuse of external modelling schemes

Having experienced conversion problems in a variety of projects, the SUBLIMA conversion process is rather simple. Existing databases with meta-data where based on MySQL [12], thus already being structured and containing clearly defined relations between objects in the information model. The implementations showed different tables for information on the "ontology" layer, consisting of topics, labels for different languages and relations between topics. This information was easily converted to an OWL based ontology. Resource descriptions in the database consisted of basic Dublin Core fields and where necessary we used additional modelling schemes (like SKOS, FOAF, etc.).

While converting the SUBLIMA data sources, we had the goal in mind to extensively reuse existing external modelling schemes. This strategic decision has been made as to prepare the implemented portals as much as possible for future integration with the Linking Open Data initiative [14]. For conversion of the before mentioned Oslo Public library database and the Medical database, the following schemes have been (re-)used while remodelling the domains:

- **RDF/RDF(S)/OWL**. because it offers the basic constructs for ontology modelling [5].
- **FOAF (Friend-Of-A-Friend)**: Name space used for representing agents (publishers) and their evt inter-relationships in the database [9].
- **Lingvoj**. Definition of language descriptors used for describing languages used in literals, but also languages on the resources the meta-data points at (referenced resource) [13].
- **DCMItype**. Dublin Core Metadata group's definition of media-types. Describes the type of media the referenced resource consists of (movie, text, image, etc) [7].
- **DCT (Dublin Core Terms)**: describes a core set of meta-data descriptors [7].
- **WDR (Powder)**: Web Description Resources. Used for describing the status of meta-data (approved, inactive, etc.) [18].

– **SIOC (Semantically-Interlinked Online Communities)**: used for description of email addresses of persons. [22]
– **XSD (XML Schema Definition)**: used in representation of DateTimeStamps in generation date [28].
– **SKOS (Simple Knowledge Organisation System)** : Used for defining resource items like preferred labels, alternative labels, broader/narrower terms and concepts [15].

After conversion of two domain specific databases, not containing any references to open standards and modelling schemes, we can say that in general we have positive experiences with their re-use. We are now working with the integration of these implementations with the Linking Open Data initiative, so that we can show the full potential of interlinked information.

2.7 Using currently available tools: Knowledge Modelling and Reasoning not compatible?

Our experiences with re-using various schemes in real-world applications are two-fold. We found that with the current state of the art there is an

– **advantage of re-use and ease of integration:** our experiences with finding and re-using existing schemes and reasoning through our domain specific models in combination with these external schemes are positive indeed. For many of the information object we had to represent, decent schemes where defined and used.
– **challenge with consistent tool-support.** Using current available tools for knowledge modelling and utilisation, we found a clear discrepancy in requirements during generation of knowledge models and interpretation of knowledge models while reasoning with the same models.

An example of the former finding is that the reuse of schemes not only made defining knowledge models easier, it also allows for a planned tighter integration with the Linking Open Data initiative and other SPARQL-based portals.

When it comes to the challenge of using current tools, we found many interesting side effects of our aim for using available Open Source tools whenever possible. After an evaluation of available editors for knowledge modelling we choose the tool which was most mature in our vision, Protégé [20]. However, despite it's perceived maturity, we encountered many cases where a decent knowledge model created in OWL and populated with information from RDBM systems or files was only editable manually in an editor like Emacs. Reasons for this where

– **the sheer size of the populated knowledge model** Whereas the Deichmanske had a model with about 4500 resources, the Medical domain model had 8500 resources. When taking into consideration all description elements for a resource, and added the ontology with concepts, these models had between 100,000 and 250,000 triples to load.

– **the way resource URI's where processed** caused troubles upon loading the model into memory. Protégé produced base URI's for each URI it encountered and checked new URI's against existing base URI's. Since these where mostly unique, the system used much memory and did not manage to load the model in over 20 minutes. After a change in representing resources, using another identifier as the originating URI that was first used, we managed to decrease loading time to under a minute. We regard the fact that we had to exchange the identifier most logical to use for the domain experts with another URI with a single base throughout the database to get this to work as a sub-optimal solution.
– **visualisation of triples in editing tools** was in many cases less then optimal for end-users wishing to change their models. It happened for example that we had to add triples to our resource descriptions for the sole reason to get a proper visualisation of important information in topics or URI's. Visualisation of the model in it's original form led to a list with e.g. 1700 entries, that all looked similar in the listing. Finding the entry to edit is not trivial in such a case.

The trade-off between model generation and utilisation was clearly an important topic of many discussions in the work group and has led to many re-modelling attempts. Often the reason was that a model created within an editor like Protégé was readable and editable by an expert, but sub-optimal for reasoning and processing with an interpreter. We ended up with fine-tuning models for optimal performance in a Emacs editors in the end. Although this might be fine for some people, our librarians where definitely not charmed by the idea and expressed a clear wish for using graphical tools instead.

3 Evaluation and Conclusion

This paper described an implementation for which we went through the full process of knowledge and software engineering in order to produce a working portal in a library setting based on two different domains.

We clearly found that the technology currently available starts to reach a certain state of maturity if it comes to functionality. At the same time, several tasks in the process where found that are in need for more research. In this respect we can mention the well-known *knowledge bottleneck*, which could be decreased by a better, more intuitive way for end-users to express knowledge in a formal language like OWL. We can also mention the absolute need we found in the project for a knowledge engineer with a deep understanding of the available standards and knowledge representation languages like Turtle, N3, OWL and RDF(S). Without such an expert it becomes nearly impossible to optimise models without loosing expressivness on the, rather important, visualisation side. Domain experts are of course always needed while converting databases and it became clear that they too had to learn many of the more advanced features of knowledge modelling in OWL, in order to be able to perform proper domain modelling.

Having said that, we got a very flexible, domain independent and highly inter-operable system back. Our total implementation time for the stack and the conversion of the two domains was 6 months, with an expected short implementation time for additional portals. With this implementation we have proven that Open Source development of Semantic Web applications is not only possible, but also leads to tangible results which are reproducible between domains. At the moment the SUBLIMA software is in use at two libraries in Oslo and a company matching portal directed towards matching and alignment of EU programs, their classification schemes on technology and projects with Norwegian companies.

Acknowledgements The authors wish to thank ABM-utvikling, Computas, ESIS, our project group members internally and external groups like HP's Jena team and community members for helping out and pointing us to solutions for our "challenges". The NRK Digital Music Archive project is a cooperation between NSA, CognIT, Norcom and Reply.

References

1. Apache Cocoon. A Spring-based framework built around the concepts of separation of concerns and component-based development. http://cocoon.apache.org/
2. Apache Software. http://www.apache.org/
3. Apache Struts2. Framework for enterprise-ready Java applications. http://struts.apache.org/2.x/index.html
4. CQL. Query language for SRU. http://www.loc.gov/standards/sru/specs/cql.html
5. Dean, M., and Schreiber, G. (eds.) OWL. Web Ontology Language Reference, W3C Recommendation http://www.w3.org/TR/2004/REC-owl-ref-20040210/
6. DRUPAL. An Open Source Content Management System. http://www.drupal.org
7. Dublin Core Metadata Initiative. http://dublincore.org/
8. Engels, R.H.P., and Tnnesen, J-R. Case Study: A Digital Music Archive (DMA) for the Norwegian National Broadcaster (NRK) using Semantic Web techniques. http://www.w3.org/2001/sw/sweo/public/UseCases/NRK/ (2007)
9. FOAF Framework. Friend of a Friend modeling scheme. http://www.foaf-project.org/
10. JENA. Java framework for building Semantic Web applications. http://jena.sourceforge.net/
11. LARQ. Lucene and ARQ (Jena SPARQL layer) combined in a single technology. http://jena.sourceforge.net/ARQ/lucene-arq.html
12. MySQL. Open Source RDBMS http://www.mysql.com/
13. Lingvoj: Multilingual Language Descriptors in RDF. http://www.lingvoj.org/
14. Linking Open Data; a W3C SWEO Community Project. http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
15. Miles, A., and Brickley, D. Simple Knowledge Organisation System (SKOS) Core Guide. W3C Working Draft, 2005. http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/
16. Oslo Public Library (Deichmanske Bibliotek). http://www.deichmanske-bibliotek.oslo.kommune.no/english/

17. (MAC) OS X. A MAC operating system http://en.wikipedia.org/wiki/Mac_OS_X
18. POWDER. Protocol for Web Description Resources. http://www.w3.org/TR/2007/WD-powder-dr-20070925/
19. PostgreSql. Open Source RDBMS. http://www.postgresql.org/
20. Protégé. Open Source Ontology editor and Knowledge Framework. Stanford University. http://protege.stanford.edu/
21. Scandinavian Medical Information for Layman. Medical Library references at the University of Oslo, Norway. http://www.smil.uio.no/
22. SIOC. Semantically-Interlinked Online Communities. http://sioc-project.org
23. SPARQL. Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/
24. SRU. Search/Retrieval via URL. http://www.loc.gov/standards/sru/
25. SUBLIMA. SUBject tool for LIbraries, Museums and Archives. Portal software homepage. http://sublima.computas.com/
26. Turtle. Terse RDF Triple Language. http://www.dajobe.org/2004/01/turtle/
27. Virtuoso. Universal Server Platform supporting SQL, RDF, XML and Web Services. http://virtuoso.openlinksw.com/
28. XML Schema Definition. http://www.w3.org/XML/Schema