Semantic BRICKS for performing arts archives and dissemination

Erik Mannens, Sam Coppens, Rik Van de Walle, IBBT-UGent-MMLab, Ghent, Belgium

Introduction

Until recently, cultural temples in Flanders had little strategy to archive and disseminate their productions. Yet, the local government wants the productions to be archived as cultural heritage, schools want material bundles for educational purposes, and other (foreign) institutions want production clips for promotional or research aims. The research project PokuMOn [1], deals with these problems and requirements of online distribution and archiving of multimedia productions of performing arts and (classical) music. In this article, we tackle the following issues: i) the institutions want an easy to use, robust, de-centralized archive; ii) the institutions want to bundle and exchange their assets; iii) the institutions want to use a common metadata schema combined with their own schemas; and iv) the institutions want their (meta)data enriched and interlinked.

The solution proposed in this article elaborates on the distributed semantic open-source BRICKS archiving and distribution architecture [2], as ease of use, robustness, independence of central authorities, low-cost, and flexibility in offered services are crucial within the cultural community. This platform allows the institutions to configure, extend, and manage their own digital depot to their needs. In order to store and exchange all the information on their productions a new layered metadata schema is developed on top of the BRICKS framework. It is an OWL DL [3], schema consisting of two layers: Dublin Core [4], and Provenance [5]. The Dublin Core layer describes the digital objects in a general way as a greatest common divisor. All the fields of Dublin Core are optional and repeatable. These characteristics allow for easy mapping to and adoption of the proposed metadata schema. It forms a common interoperability and discovery layer on top of the descriptions that are already distributed by the institutions. The second layer indicates the provenance of the Dublin Core descriptions. In most cases, the institutions have their own metadata schema which is mapped to Dublin Core. The provenance layer indicates the identifier of the original metadata description and the namespace of the original metadata schema. This information allows linking to the original descriptions, which are in most cases richer in information. To aggregate the digital objects in bundles (a/o for educational purposes) the BRICKS framework is extended with an OAI-ORE [6] web service. It describes aggregations of Web resources in a semantic way via dereferencable URI's. Furthermore, we enrich the metadata semantically following the Linked Open Data principle [7]. In our case, we apply linguistic processing on the plain text contained in some elements of the metadata such as title, contributor, subject, and description. The linguistic processing consists in extracting named entities such as persons, organizations, companies, brands, locations, and events using the OpenCalais infrastructure [8]. Once the named entities have been extracted, we map them to formalized knowledge on the Web available in GeoNames [9], for the locations, or in DBPedia [10], for the persons, organizations, and events, and feed this new knowledge back into the system. This way BRICKS is semantically adapted and extended to offer an end-to-end solution to the institutions and third parties (schools, broadcasters, etc) that can search and harvest all data via web services.

As such, this article shows how all media of performing arts productions can be archived, bundled and disseminated using distributed Semantic Web technologies. In the end, all is demonstrated within an end-to-end Proof-Of-Concept showing the feasibility of the approach in Flanders' cultural temples establishing a durable cooperation between all actors involved. Finally, we put forward some best practices, caveats, and lessons learned.

BRICKS Overview

After an initial platform evaluation [11] the distributed semantic open-source repository BRICKS was chosen as development platform. It is the outcome of the European project Building Resources for Integrated Cultural Knowledge Services (BRICKS [2]). The aim of the BRICKS project was to design an open user- and service-oriented infrastructure to share knowledge and resources in the cultural heritage domain.

The BRICKS architecture is by default decentralized. Therefore every performing arts institution can deploy its own instance of BRICKS, called a BNode. These BNodes communicate among each other and use available resources for content and metadata management. Every BNode knows only a subset of other BNodes in the system. If a BNode wants to reach another BNode that is unknown to it, it will forward the request to some of its known neighbouring BNodes that will deliver the request to the final destination. Such an approach avoids having central hubs whose failure or overload could stop the whole system. Another strong advantage of this architecture is that centralized administration costs for additional personnel and money can be avoided. That is why BRICKS was chosen as development platform.

A BNode can be seen as a collection of services that are required to manage its presence in the system and to provide services for the rest of the community. A BNode consists of three types of components: fundamental, core, and basic BRICKS components. The fundamental components are essential and needed on every BNode. The core components provide core system functionalities to the users, i.e., a minimal set of services that enable the users to use the system. The basic components are optional, and do not have to be present on all the BNodes. Most of the services are standard Web services described by WSDL documents. Since the BNode architecture is service-based, a BNode installation can be spread over several machines. In this case the fundamental services have to be present on every machine, while core and basic components could be present on only some of the machines. Figure 1 gives a schematic overview of the architecture of BRICKS. This way, BRICKS is a very heterogeneous, adaptable system without the need for a central body to maintain the system, which makes BRICKS a cost-effective solution.

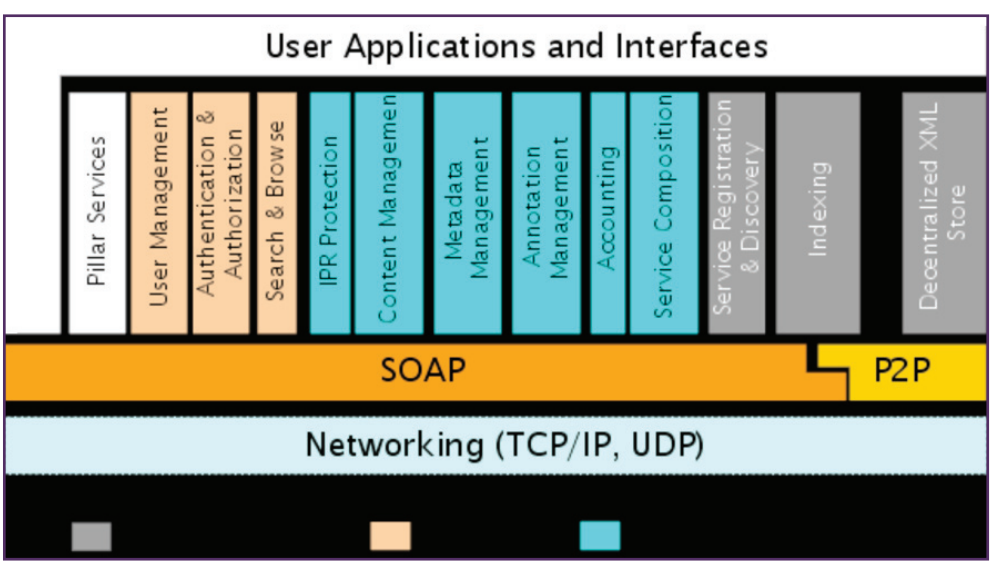

Figure 1: Overview BRICKS Architecture

Layered Metadata Schema

Introduction

The major problem we are facing is to bridge the incompatibility of the different metadata schemes used all over the arts sector in Flanders. Our proposed layered metadata schema will be used for the descriptive metadata in the project PokuMOn and is implemented in the BRICKS repository. This model not only leverages the exchange of data between the performing arts institutions in Flanders, but also the possible dissemination to the general public. The model had to be applicable in the whole performing arts sector in Flanders (and preferably beyond). In other words it had to be general enough. Many of the institutions already have descriptions of their objects. Those descriptions are described using many different metadata schemas. So those schemas that are already in use in the performing arts sector in Flanders should be able to be mapped to this proposed schema.

Requirements

The schema has to deliver all the necessary elements to the user so s/he can find information on the object of interest. When the user has found the information, s/he has to be able to link to a more detailed description of that object. In order to fulfil these requirements the model is split into two parts, a description part and a provenance part.

The first part, or common layer, describes the object. This description has to be general enough to be applicable on all the objects in use, but on the other hand it has to deliver the elements so the user can find what s/he is searching for. This part consists of an interoperability layer, a common layer above all the metadata schemas that are already in use in the field. This part then automatically offers the tools to query all those descriptions. In other words it has to be able to answer basic questions like who, what, where and when.

The second part or lower layer contains the information needed to link to a more detailed description, mostly to the complete record the first part is mapped from. This part has to reflect at least the namespace of the schema the original record is described with, a URI of the repository the record comes from and the identifier of the record in that repository.

Design

OWL DL
For the definition of the new metadata schema we used W3C's Semantic Web technology [12], more specifically the OWL ontology language [3]. The expressiveness of OWL allows us to create fine-grained property definitions by splitting the definition of properties into 'attributes' and 'relations'. Attributes (corresponding to the OWL notion of a datatype property) can take typed literals as value whereas relations (corresponding to the notion of an object property) can link to other resources like content items or concepts taken from an ontology. The sublanguage is OWL DL, not OWL FULL. OWL FULL gives the most expressiveness, but does not guarantee the support of reasoning software, while OWL DL is a little less expressive, but it is guaranteed to be completely supported by the RDF [13] reasoners. The BRICKS framework, which will make use of this schema, also requires that the schemas are described in OWL DL.

Description
The records are described in Dublin Core [4]. It is the most common metadata schema in use and it is general enough to describe all the objects of the Flemish performing arts sector. It is the largest common divider of all the metadata schemas that are used in the performing arts sector in Flanders. On top of that, all the fields of the Dublin Core model are optional and repeatable. This makes it possible to map nearly all the metadata schemas to Dublin Core. This makes the framework also OAI compliant [5], because the offering of Dublin Core descriptions is a requirement for OAI compliance of the data provider. For the implementation of the schema, all properties of Dublin Core were modelled as datatype properties, which are all optional and repeatable. This part is described by [14].

Provenance
As mentioned before, this lower layer should deliver at least three things: the metadata namespace of the originating record, the URI of the repository it comes from and the identifier of that originating record in that repository. This layer is based on a schema that is used by the OAI-PMH protocol, [5], indicating the provenance of a record. This schema is described in XML schema, so the schema has to be 'ontologized' in an OWL DL schema, which can be found at [15].

Upper Ontology
Finally, there needs to be an upper ontology that imports the two other ontologies and combines them into one ontology. This way each of the imported ontologies, the Dublin Core description (the common layer), and the Provenance description (the lower layer), can be altered independently. This is described by [16].

Implementation

DC-Description
Dublin Core consists of just fifteen properties:

- Title: A name given to the resource
- Creator: An entity primarily responsible for making the content of the resource
- Subject: The topic of the content of the resource
- Description: A description of the content of the resource
- Publisher: An entity responsible for making the resource available
- Contributor: An entity responsible for making contributions to the content of the resource
- Date: A date associated with an event in the life cycle of the resource
- Type: The nature or genre of the content of the resource
- Format: The physical or digital manifestation of the resource
- Identifier: An unambiguous reference to the resource within a given context
- Source: A reference to a resource from which the present resource is derived
- Language: A language of the intellectual content of the resource
- Relation: A reference to a related resource
- Coverage: The extent or scope of the content of the resource
- Rights: Information about rights held in and over the resource.

This ontology defines a class, DC, on which all these properties are applicable. As already mentioned, the properties are defined as datatype properties. The domain of these datatype properties is the defined class DC and the range of the datatype properties is a string. This makes the Dublin Core description unqualified.

Provenance
This layer is based on a schema that is used by the OAI-PMH protocol. The schema defines a provenance container consisting of a sequence of originDescription elements that identify the provenance of the metadata record. Each originDescription contains the following information:

- baseURL: The base URL of the originating repository from which the metadata record was harvested
- identifier: The unique identifier of the item in the originating repository from which the metadata record was disseminated
- datestamp: The datestamp of the metadata record disseminated by the originating repository
- metadataNamespace: The XML namespace URI of the metadata format of the record harvested from the originating repository
- originDescription: An optional originDescription block which was obtained when the metadata record was harvested. A set of nested originDescription blocks describe the provenance over a sequence of harvests.

Each originDescription must also have the following two attributes:

- harvestDate: The response date of the OAI-PMH response that resulted in the record being harvested from the originating repository
- altered: a Boolean value which must be true if the harvested record was altered before being disseminated again.

For the OWL DL description of this schema, a class is made up, provenanceType. An object property is defined on this class. The range of this object property is the class originDescriptionType. This object property has a minimum cardinality of one. This means that an instance of provenanceType holds at least one instance of originDescriptionType. This models the sequence of originDescription elements as described by the XML schema of the provenance.

The class originDescriptionType has six datatype properties: baseURL, identifier and metadataNamespace, which all have a URI as range, datestamp and harvestDate, which have a string as range, and finally altered, which has a Boolean as range. All these six datatype properties are required and have a cardinality of one.

The class originDescriptionType has one object property, originDescription, which relates an instance of originDescriptionType to another instance of originDescriptionType. This property is optional, so it has a maximum cardinality of one.

Upper Ontology
This ontology imports the two other ontologies and combines them. For this a class Metadata is defined. This class has two object properties, dcDescription and provenance. They have as range respectively the imported class DC and the imported class provenanceType. This way the two ontologies are combined in a new ontology. The schema and OWL DL description can be found at [16].

OAI-ORE

Introduction

Besides archiving audio, video, photo, and text, the repository has to be able to store aggregations of these objects. The performing arts institutions disseminate, beside their performances, introductions to performances, interviews with artists, programme brochures, reviews, etc. These aggregations also have to be stored, disseminated, and exchanged. For this, we developed an ontology based on the Open Archives Initiative Object Reuse and Exchange (OAI-ORE [6]).

Today, many information systems, like content management systems, support the storage and identification of aggregations, and the access to the aggregations and aggregated objects. In most systems these objects vary in semantic type, e.g., article, book, video, dataset, etc, and in metadata file format, e.g., PDF, XML, MP3, etc. These objects can also be stored on different network locations, i.e., aggregated objects can be stored locally or externally. Information systems store, identify, and deliver access to these compound objects in an architecture-specific manner. Unfortunately, the way these information systems disseminate their compound objects is far from perfect and without any broadly accepted standard. In many cases, a lot of the advanced functionalities get lost when publishing the compound objects to the Web. Mostly, the publication is aimed at the end-users and at agents, e.g., web crawlers. The structure of the object is often embedded in splash pages, user interface widgets, etc. This approach makes the structure of the compound object unclear for machine-based applications like browsers, web crawlers, etc. Consider the example of a scanned book, where all the pages get an HTTP URI. A web crawler can come across one of these pages and find links to the other pages of the book, to the chapter containing that page or to the book. A web crawler cannot distinguish between these links. For the web crawler these are untyped links or links that do contain information, but this information remains unreadable to the web crawler. So, the order of the pages gets lost, etc.

OAI-ORE Specification

The OAI-ORE standard tackles this problem by developing a standardized, interoperable and machine-readable mechanism that can express the information of compound objects. The standard makes sure that the logical boundaries of the aggregated objects and their mutual relations remain intact when publishing the compound object to the Web. To achieve this, OAI-ORE makes use of resource maps, which are in fact RDF descriptions of the compound objects. These resource maps are identified by a URI, which contains a set of RDF declarations. These declarations instantiate an aggregation as a resource with a URI, and list the aggregated resources, their mutual relations and the web context of the aggregation. Actually, these resource maps are named graphs. These graphs are RDF graphs, sets of triples, extended with a name, URI, for the graph. The named graph is not the RDF graph itself, but a representation of the set triples encoded in Atom or RDF/XML, as depicted in figure 2. To talk about aggregations on the Web, they have to have a URI. The ORE model demands that a resource map describes just one aggregation. An aggregation, on the other hand, can have multiple resource maps, each with its own representation. Clients and applications need to determine the URI of the resource map from the URI of the aggregation, so they could refer to that aggregation. This can happen in two ways: one way is to append a fragment identifier to the URI of the resource map; another solution is offered by cool URIs, e.g., by appending the '.rdf' extension to the URI of that aggregation.

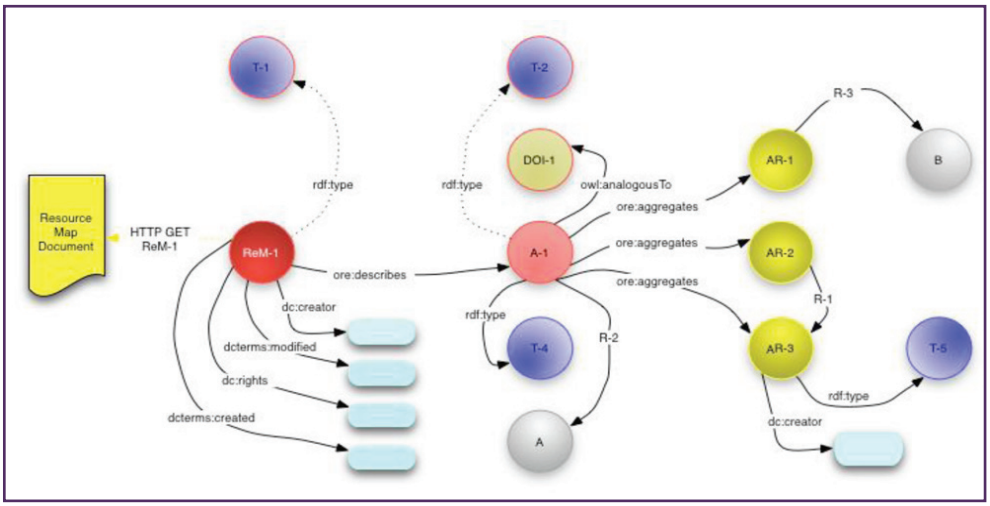

Figure 2: Schematic Representation of an OAI-ORE Aggregation

Semantic OAI-ORE Schema Implementation

The RDF schema for the OAI-ORE model consists of two classes: ResourceMap and Aggregation. The class ResourceMap has three mandatory properties: rdf: type, indicating that the resource map is of the type ore: ResourceMap; ore:describes, referring to the (URI of the) aggregation; and dc:creator, referring to the authoring authority. Other optional properties of the ResourceMap class are: dcterms:modified, indicating the modification time of the resource map; dc:rights, describing the rights pertaining; and dc:created, for the original creation time of the resource map. The class Aggregation has only one mandatory property: rdf:type, indicating the resource is of the type Aggregation. Another optional property for this class is: ore:aggregates, referring to the aggregated resources.

Shortcomings of BRICKS

BRICKS has no problems storing the resource maps, but cannot handle the cool URIs. This problem is solved by publishing the records from the JENA triple store [17] from BRICKS as Linked Open Data [7], as will be fully described hereafter. Publishing the records as Linked Open Data offers the opportunity to use cool URIs to redirect the client (web crawlers, HTTP browsers, RDF browsers) to the appropriate representation. This way, clients that come across the HTTP URI of an aggregation can be redirected to a representation they understand, preserving the typed links between the aggregated resources. So, storing the resource maps and publishing the resource maps as linked data makes the repository OAI-ORE compliant [6]. This allows the BRICKS repository to manage, exchange, and share aggregates of resources, e.g., a video of a performance, accompanied by a program brochure and a transcription of the performance, conforming to the OAI-ORE standard.

Linked Open Data

Introduction

Sir Tim Berners-Lee first introduced the term Linked Open Data in 2006 [7]. Linked Open Data lets people share structured data on the Web as easily as they share documents today. It refers to a style of publishing and interlinking structured data on the Web. Linked Open Data lets you use RDF data models to publish the structured data on the web and uses RDF links to interlink data from different datasets. This makes the Web one giant database, the Web of Data.

Linked Open Data Basics

Linked Open Data stipulates four basic principles. The first principle is that we first have to identify the items of interest in our domain. Those items are the resources, which will be described in the data. The next principle is that those resources have to be identified by HTTP URIs and avoid schemes such as URNs [18] and DOIs [19]. The third principle is to provide useful information when accessing an HTTP URI. The fourth rule is to provide links to the outside world, i.e. to connect the data into the Web of Data.

In practice, this means that every resource described by an RDF schema has to be identified by an HTTP URI, e.g., http://dbpedia.org/resource/Berlin. Every resource should also have two representations: an XHTML and an RDF representation. Every representation also has to be identified by an HTTP URI, e.g., http://dbpedia.org/page/Berlin for the XHTML representation, and http://dbpedia.org/data/Berlin for the RDF representation. When coming across the HTTP URI of a resource, the Linked Open Data server determines which representation should be served, based on information in the accept header of the user's client, and redirects the client to the appropriate representation using 303 redirect and content negotiation.

Linked Open Data vs. OAI-ORE

Publishing resources as Linked Open Data conforms to the way OAI-ORE offers to publish aggregations. OAI-ORE demands that aggregations have to be identified by a URI, and have to be described using an RDF schema, i.e., a resource map, which also has a URI. When clients consume the URI of that aggregation, they should be able to automatically detect the URI of the resource map with the appropriate representation for the client. This principle conforms to the way Linked Open Data is published.

Linked Open Data Implementation

For publishing the records from a triple store as Linked Open Data, the open-source tool Pubby [20] is used. Pubby is actually a Linked Data frontend for SPARQL endpoints. Such a SPARQL endpoint is a webservice that can handle SPARQL queries. These SPARQL queries can be seen as semantic SQL statements. BRICKS does not provide such a SPARQL endpoint. That is why the triple store in the BRICKS framework was replaced by the open-source OpenLink Virtuoso triple store [21]. This triple store offers by default a SPARQL endpoint. By configuring Pubby for the SPARQL endpoint, provided by Virtuoso triple store, the records stored in the triple store are published as Linked Open Data. This means, providing HTTP URIs for all the records served by the SPARQL endpoint, providing a simple HTML interface showing the data available about each resource, and taking care of the 303 redirects and content negotiation.

Metadata Enrichment

Finally, the stored records, constructed via our metadata schema and published as Linked Open Data, are extended with links to information from datasets likeGeoNames [9], and DBpedia [10]. This way, the records are enriched with information from external datasets, weaving that extra information into the Web of Data.

For interlinking the data automatically, the choice was made to provide extra information on the title of the resource, the persons, organizations, events, and the places involved. In practice this means for the persons, organizations, and events iterating for every Dublin Core description of the records and querying the DBpedia dataset for the values of the datatype properties: dc:creator, dc:contributor, dc:publisher, dc:subject. For the places information the values of the datatype property dc:coverage of the Dublin Core descriptions are queried against the GeoNames dataset. The results, returned from these queries, are added to the Dublin Core description via the object property: rdfs: see Also.

The descriptions of the resource (values from the dc:description datatype property from the Dublin Core description) are also examined. These strings, describing the resource, are investigated for persons, organizations, companies, brands, locations, and events. For this we rely on the OpenCalais webservice [8], which is able to investigate strings and return certain concepts mentioned in the description. The results for the persons, organizations, or events concept are forwarded to query the DBpedia dataset. The results for the places concepts are forwarded to query the GeoNames dataset.

By applying our metadata enrichment algorithms, the records are enriched with links to external dataset. This puts the records on the Web of Data and enriches the record with extra information.

Conclusion

This article shows how performing arts institutions can disseminate their content using semantic web technologies, like RDF, OWL, and Linked Open Data. The Semantic Web is an evolving extension of the World Wide Web in which the semantics of information and services on the web is defined, making it possible for the web to understand and satisfy the requests of people and machines to use the web content. To benefit the search and discovery

of the records, these records have to be described by a uniform metadata model. This model has to be applicable for a variety of data: text, audio, video, and aggregations of them. For this, three semantic models were designed and implemented: a Dublin Core description, describing the resource in a very generic way, a provenance description, referencing the original record, which can give a more detailed description of the resource than the Dublin Core description, and an OAI-ORE model to describe aggregations. This way, the performing arts institutions can share and exchange their (aggregations of) information, avoiding a lot of interoperability issues. By publishing the records in a Linked Open Data way, the server can redirect clients (people or machines) to the appropriate representation, XHTML for people and RDF for machines, which is compliant to the way OAI-ORE publishes aggregations. By enriching the data with links to information coming from e.g. DBpedia and GeoNames, the records are weaved into the Web of Data, making the Web of Data one huge database.

This is a new approach for disseminating records coming from the performing arts sector. Mobilising the sector to adapt this approach is not a trivial task, although the awareness comes from the sector itself. That is why VTi, Flemish Theatre Institute, as a coordinating body for the performing arts institutions in Flanders, was chosen to implement this approach firstly and to offer this approach as a service to the other institutions in the performing arts sector. This way, the sector can be more easily mobilised and encouraged to adopt this way of disseminating records from the performing arts sector in Flanders.

References

1. Project PoKuMOn. Available at: http://projects.ibbt.be/pokumon/

2. Project BRICKS. Available at: http://www.BRICKScommunity.org

3. M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, L.A. Stein: OWL web ontology language reference. W3C Working Draft, 2003. Available at: http://www.w3.org/TR/2003/WD-owl-ref-20030331

4. The Dublin Core Metadata Initiative, DCMI, 2009. Available at: http://dublincore.org/

5. C. Lagoze and H.V. de Sompel, The open archives initiative protocol for metadata harvesting – version 2.0, 2002.
Available at: http://www.openarchives.org/OAI/openarchivesprotocol.html

6. C. Lagoze and H.V. de Sompel, The open archives initiative object reuse and exchange, Resource Map Impementation in RDF\XML – version 1.0, 2008.
Available at: http://www.openarchives.org/ore/1.0/rdfxml.html

7. T. Berners-Lee, Linked Data, 2006.
Available at: http://www.w3.org/DesignIssues/LinkedData.html

8. OpenCalais. Available at: http://www.opencalais.com

9. GeoNames. Available at: http://www.geonames.org/export/

10. DBpedia. Available at: http://dbpedia.org/about

11. D.0.4-Eindrapport Van Horen Zeggen III v16.doc & D.5.1-SOTA Metadatamodellen. doc. Available at: http://projects.ibbt.be/pokumon/

12. Ivan Herman, Semantic Web Activity, 2009. Available at: http://www.w3.org/2001/sw/

13. Ivan Herman, Ralph Swick, Dan Brickley, Resource Description Framework, 2009. Available at: http://www.w3.org/RDF/

(14. OAI-compliant)
14. The Dublin Core Ontology used in PoKuMOn, 2009. Available at: http://multimedialab.elis.ugent.be/users/samcoppe/Ontologies/Metadata/DCDescription.owl

15. The Provenance Ontology used in PoKuMOn, 2009. Available at: http://multimedialab.elis.ugent.be/users/samcoppe/Ontologies/Metadata/Provenance.owl

16. The Upper Ontology used in PoKuMOn, 2009. Available at: http://multimedialab.elis.ugent.be/users/samcoppe/Ontologies/Metadata/Metadata.owl

17. JENA – A Semantic Web Framework for Java. Available at: http://jena.sourceforge.net/index.html

18. John Curran, Leslie Daigle, Ned Freed, Patrik Faltstrom, Uniform Resource Names (URN), 2002. Available at: http://www.ietf.org/html.charters/OLD/urn-charter.html20.

19. The International DOI Foundation, The DOI System, 2009. Available at: http://www.doi.org/

20. Richard Cyganiak and Chris Bizer, Pubby – A Linked Data Frontend for SPARQL Endpoints, 2007. Available at: http://www4.wiwiss.fu-berlin.de/pubby/

21. Openlink Software, Virtuoso open-source edition, 2009. Available at: http://virtuoso.openlinksw.com/wiki/main/Main