Semantic Web 0 (2011) 1–0
IOS Press

An Information Model for the Annotation of Resources with Heterogeneous Metadata

Hannes Ebner * and Matthias Palmér
School of Computer Science and Communication, Royal Institute of Technology (KTH), Lindstedtsvägen 3, 10044 Stockholm, Sweden
E-mail: {hebner,matthias}@csc.kth.se

Abstract. This paper discusses the annotation of resources with heterogeneous metadata and presents an information model as well as a reference implementation and showcases depicting how the model can be applied in real world situations. The presented approach provides resource and metadata management that allows for describing relations between metadata graphs, keeping track of provenance and access control. After presenting the model and the argumentation behind, the paper also describes the architecture of an implementation of the model and a web-based application taking advantage of it. It allows for describing resources with metadata, extending already existing metadata and harvesting of metadata repositories using different protocols, and exposes all metadata according to Linked Data principles. To show the taken approach in practice, several real-world examples are presented as showcases in the context of educational metadata and learning repositories.

Keywords: Resource Annotation, Heterogeneous Metadata, Linked Data, Information Model, Provenance

1. Introduction

Several projects with a focus on exchanging metadata between learning repositories had the same problem: how would it be possible to bridge the gap between "traditional" repositories and triple stores, taking advantage of the features that Semantic Web technologies offer? It was not possible to just make a transition of the systems that should be taken advantage of, so an intermediate layer had to be introduced, being compatible with the old way of managing information and working with RDF at the same time.

The approaches and the information model described in this document are intended to be such a solution, making it possible to bring already existing (meta)data from one system into the Linked Data world. Such resources and metadata are then uniquely identifiable, accessible and modifiable using URIs and REST-based services following the Linked Data principles. In this paper the most pressing problems that occur in such situations are summarized, an architecture for a solution is described and a short summary of several showcases is presented, including some conclusions regarding the general applicability and possible future applications.

Using Semantic Web technologies to annotate resources with metadata seemed to be a logical approach as described in previous publications [7,6]. RDF can act as common format for most metadata standards. Most often it is not a problem to create an appropriate unique identifier (URI) which is a requirement for using RDF. Relations can be built between resources or (parts of) their descriptions and URI-based vocabularies are used to avoid ambiguous annotations. The use of the query language SPARQL is a powerful way of discovering relevant resources based on their metadata and complex searches can be performed even in large repositories. However, using plain triples to describe resources leads to situations where it is impossible to find out who contributed what and when. Provenance information is simply missing, which makes it impractical for most use cases. Further down it is described (among other things) how it is possible to keep track of heterogeneous metadata in distributed environments.

*Corresponding author. E-mail: hebner@csc.kth.se.

2. Problem Statement

The research summarized in this paper addresses the following problems that usually occur in the context of management of diverse information as described above. How to:

1. integrate heterogeneous information sources and harmonization of metadata expressions from different standards.
2. enrich metadata originating from other sources, e.g. adding educational metadata on top of already existing generic metadata.
3. avoid duplication of metadata when the original source is updated after the harvested metadata instance has been enriched.
4. expose the combination of resources and their metadata in a Linked Data way.
5. replace harvesting techniques with Linked Data.

Some of the problems above occur when a classical metadata harvesting approach is followed. Metadata is copied from one system into another, completely neglecting the relationship between the created metadata instances. There is also the need of providing links between related resources and resources or concepts in other systems such as DBpedia etc.

Parts of the presented solution in this paper rely on the use of Named Graphs [5] which allows sets of triples to be referenced. However, Named Graphs do not come with best practices to:

- keep track of provenance of Named Graphs.
- express that Named Graphs are related, for instance describing the same resource.
- retrieve and modify Named Graphs using a standard protocol, i.e. HTTP.

This paper starts with going through the relevant state of the art for managing and annotating metadata. After that an information model is introduced to show how the previously defined problems are solved. The following section presents a reference implementation which exposes the information model through web technologies and explains the usage of Linked Data. A user interface in the form of a web application - followed by some showcases - is described in the sections after that. The conclusions summarize the work carried out and round off with the planned next steps and possible future work.

3. State of the Art

3.1. Document- vs. Graph-centric Annotation

Traditional ways of resource annotation often take a document-centric approach and use the XML format as it is an established standard for expressing information. Unfortunately, when document-centric metadata is transferred between systems (e.g. using a harvesting protocol like OAI-PMH [13]), the metadata is copied and a fork takes place. The alternative, to reuse metadata without making a copy, requires that the original instance can be uniquely identified. This is not possible with the current approach of "traditional" learning repositories as everything is based on harvesting metadata from one system into another leading to copies and forks instead of references. Information is unnecessarily duplicated and numerous variations of descriptions of the same resource are created without being able to reconstruct their history.

3.2. RDF as Common Carrier for Metadata

To be able to create flexible annotations of metadata it is necessary to use a data model which is designed to allow multiple metadata expressions following different standards to coexist. RDF is such a (meta) data model [14]. However, expressing metadata in RDF requires a thorough mapping to be crafted, which often involves an analysis of the exact semantics of the standard. Good knowledge of RDF and related standards is required as it is good practice to reuse established terms from other RDF based standards whenever possible. There are situations where the conceptual model cannot be cleanly mapped to the RDF model and information may be lost. To avoid such situations, RDF should be considered as a basis for metadata interoperability - a common carrier - when adapting existing or creating new metadata standards. For a longer discussion on this subject see [15].

3.3. Named Graphs for Managing Sets of Triples

The Semantic Web allows statements about identifiable resources to be expressed using RDF triples which may also be made available on the Web for others to discover. When new statements are made, there is no need to duplicate information. Additional statements about the same identifiable resources can be expressed as new RDF triples and be published on the Web separately from the first set of triples. If all available triples

describing the same resource are merged into a single big graph a holistic view about a resource can be constructed. With only triples as the source of this information it is impossible to differentiate between triples or sets of them, which creates several problems. To mention only a few, it is difficult to detect which triples have been replaced in more recent revisions, it is hard to keep track of the history of a resource's descriptions, and it is almost impossible to provide information which depends on its purpose (i.e. contextualisation).

The concept of Named Graphs (NG) [5,10] enables us to work around this, by being able to uniquely identify a set of triples and build (sub-)graphs. To be able to identify and differentiate between instances of metadata a generic and unique identifier has to exist. Named Graphs provide a URI for a set of triples without creating a dependence on a specific metadata standard, as it is the case with e.g. IEEE LOM's [3] Metametadata identifier expressed in XML.

Another issue is related to searching and indexing. If a query matches one or more triples it is unclear where those triples originate from and in which context they express information about the described resource. This can be partially solved by using NGs, i.e. it allows for uniquely identifying the relevant triples.

3.4. Representational State Transfer

Representational State Transfer (REST) [9] is a software architecture for distributed hypermedia systems and is a popular design pattern used for resource based web services. REST itself is protocol-agnostic, but in this paper it is used in the context of HTTP. Its architectural elements are resource identifiers, resources, resource representations and their metadata. These elements can be modelled in RDF which makes REST a logical choice to access RDF-based systems.

"Pure" REST is difficult to achieve and most of the offered REST-ful web services are REST-oriented but also contain other concepts such as RPC-oriented methods [18]. However, an implementation taking advantage of HTTP makes it easier to align with the Linked Data principles as described below.

3.5. Linked Data

Linked Data (LD) extends the idea of the Semantic Web by adding links to explore the "Web of Data" which is constructed by documents on the web. The focus lies on links between uniquely identifiable "things" described using RDF. Linked Data implements the following four rules [4]:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL) [14,16].
4. Include links to other URIs so that they can discover more things.

LD suggests the use of URIs, HTTP and RDF, which makes it more specific than REST. However, RESTful web services can operate on the Web of Data when the offered data conforms to the Linked Data principles. The growing LOD cloud [1] is easily extended by simply providing statements which link to the existing published datasets. This is also one of the big differences to traditional repositories: instead of harvesting and copying data, it is sufficient to refer to things, lookup identifiers and fetch from the original source. To improve performance the data can be cached, but the basic principles are not affected by this. The mainstream of learning repositories [15] has not arrived in the LOD cloud yet and it is and will be necessary to provide a "bridge" between these two worlds. How this can work is also topic of this paper and described further down.

4. ReM^3 - An Information Model to Manage Heterogeneous Metadata

4.1. Conceptual overview of ReM^3

The Resource and Metadata Management Model (ReM^3) is an information model for keeping track of resources and their metadata. It is based on the concepts of Contexts and Entries where each context manages a set of entries. A context is a container for a set of entries that are managed together, at a minimum it provides default ownership of the contained entries. An entry contains a resource, descriptive metadata about the resource as well as some administrative information of the entry which will be referred to as the entry information. The entry information also keeps track of access control and provenance. Access control can be managed on both context and entry level, depending on how fine-grained access control is needed. The entry also keeps track of relationships from other entries via

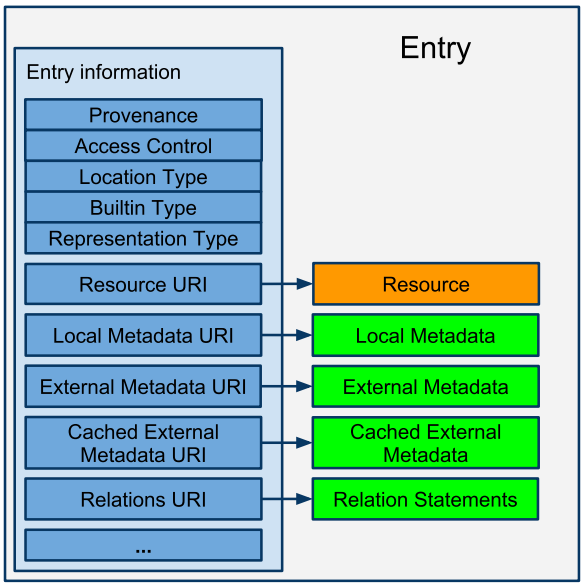

Figure 1. ReM^3 entry and linked information

a special Relation graph. See figure 1 for a conceptual representation of a ReM^3 entry.

Each entry has three different kinds of types that are more or less independent of each other:

- Location Type indicates if neither, one, or both of the entry's resource and metadata is maintained within the local system. This is the most important type for the showcases shown further down as it differentiates between local and external (remote and harvested) resources.
- Representation Type tells whether a resource has a digital representation or not.
- Builtin Type indicates whether a resource gets a special treatment within the implementation of the model.

Ideally, the entry information, that is, the information about an entry, is represented in a single RDF graph which can be requested and updated as a whole or in part. If an application needs additional information about a resource it can be represented in the same RDF graph by adding additional properties. Within the entry information the resource, the describing metadata graphs and the relation graph are URIs that are detectable via special properties from the entry URI. The URIs for the metadata, the relation graph and sometimes the resource (when the builtin type is not "none") leads to Named Graphs.

However, the availability of named graphs for an entry also depends on the location type which indicates if metadata and the resource is to be found locally or externally. More specifically, the possible values for the location type are as follows, see also figure 2.

- Local - both metadata and resource are maintained in the entry's context.
- Link - the metadata but not the resource are maintained in the entry's context.
- Reference - the resource and the metadata are maintained outside the entry's context.
- Link Reference - the resource and the metadata are maintained outside the entry's context, in addition there are complementary metadata maintained in the entry's context.

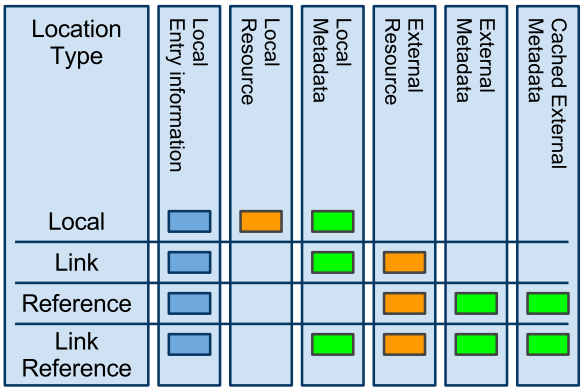

Figure 2. The ReM^3 Location Type and its implications for the location of metadata

Whenever there are metadata maintained outside of the entry's context it might be cached locally to increase reliability and performance, and to avoid pushing the responsibility of doing metadata format transformations to application developers. The entry information is always kept in the corresponding context, independently from the used location type.

The Representation Type indicates to which extent a resource is an Information Resource. A resource is anything that can be identified by a URI whereas an information resource is a resource whose essential characteristics can be conveyed in a message. Examples are documents, images, videos, etc., of various sorts which have representations, e.g. HTML, ODT, JPEG, etc., which can transferred in a message body which is the result of an HTTP request. The idea behind the representation type is based on the Architecture of the World Wide Web [12], the W3C TAG discussions on HTTP dereferencing [2] and the W3C Interest Group Note on "Cool URIs" [17].

The possible values for the representation type are:

- Information Resource - resource has a representation, in the repository or elsewhere.
- Resolvable Information Resource - The resource is an information resource but requires a resolvable step, e.g. through a lookup procedure that might be protocol specific such as urn:path or DOI.
- Named Resource - The resource is not an information resource, the resource can be referred to in communication but not transferred in a message.
- Unknown - representation type of the resource is unknown.

The Builtin Type was introduced to easily recognize resources which need special treatment by the implementation. Examples are the builtin types used for access control, namely User and Group; Context for container entries, and List to indicate an ordered list of entries within a context.

4.2. Named Graphs in ReM^3

The information model is RDF-oriented and relies on the concept of Named Graphs. As every NG is identified by a URI, it is possible to keep track of the NG provenances through the entry information as described above. The entry information contains expressions that describe the relationships between graphs. This is used to express that NGs are related, as it is the case when the same resource is described in different contexts.

In a typical situation where harvested metadata is built upon, the referenced resource is described by an additional metadata graph. The relation between the original external metadata and the newly created metadata is stored in the entry information, where the location type also indicates that there are multiple descriptions for the same resource.

The use of NGs makes contextualisation of metadata possible. Without the fourth piece of information in the quadruple it would be hard to differentiate between triples from different sources.

4.3. Expressing Provenance in ReM^3

Using resolvable URIs to avoid duplication of metadata between systems is a first step, but it is also necessary to keep track of who created or contributed what, when, where and perhaps even why. All these pieces of information are kept in the entry information and are available if the resource originates from a local ReM^3-based repository.

The following provenance-related properties are a minimum for being able to keep track of annotation cases where both local and external metadata are involved, i.e. entries with location type Link Reference:

- Creator and contributor
- Creation and modification date
- Reference to the resource
- Reference to the external (possibly original) metadata
- Date when the external metadata was cached

If the metadata originates from an external system then some restrictions apply, i.e. provenance for the resource and metadata is only known if this is known and exposed by the system where the information is fetched from. If this is not the case then the "provenance trail" starts at the time the external metadata is cached in the ReM^3 system.

One of the currently existing restrictions of the model is the lack of revisions and versioning, both for the metadata and the described resource. There is previous work which can used in this context [19] which is considered for revisions of this information model.

4.4. Expressing Access Control in ReM^3

Just as provenance is expressed in the entry information model, so is access control. The purpose of the access control in ReM^3 is to control who has rights to access entries. Access to the entry, metadata and resource is determined by specific ACL statements using the URIs of the entry, metadata and and the resource URI, respectively. The access control information for the resource is only relevant when it can be enforced by an implementation, i.e. if the resource is located in the same system (location type is local). Similarly, access control for metadata is only relevant when it is in the same system (when location type is local, link or link reference but not reference).

The access control is expressed as a set of read and write permissions for users and groups on the entry, the metadata and the resource. Any explicit permission given on entry level automatically applies to the resource and metadata and does not need not be repeated. An exception is that by default anyone has read access to the entry information, but not to the resource or metadata. Anyone who has been given write access to a entry is considered to be an owner of that entry.

Contexts are also represented as entries. Access control to a context, expressed on its entry, has a special

meaning with regard to all entries located in that context:

- Permissions given for the metadata of a context has no effect on the entries in the context.
- Permissions given for the resource of a context applies to all entries in the context who lack own access control. I.e., if an entry holds ACL information then those permissions override any permissions inherited from the contexts resource.
- Ownership of a context (write permissions on the entry level of a context) implies ownership of all entries in the context regardless of any access control specified on them.

Users and groups that can be given permissions are represented as entries with the special builtin type User and Group respectively. There are two default users and two default groups. First, "_guest" represents any user that has not authenticated himself while "_users" is the group of all users that can authenticate themselves in the system. Second, "_admin" is a predefined superuser and "_admins" is a group to which users that should have superuser privileges can be added.

There are a two special rules with regard to lists, that is, entries with builtin type list:

- Entries which are created as children of a list with custom ACL automatically inherit permissions from that list.
- An entry that belongs to a single list cannot be removed from that list (making it "unlisted") without also removing the entry itself unless the user has write permissions in the context.

5. Exposing ReM^3 using Web Technologies

5.1. Reference Implementation

The problems mentioned in the beginning were the main driver behind developing an own framework as described in [7]. It should make it possible to manage data and its metadata in an interoperable and conceptually clean way, being compatible with traditional data sources and the possibility of being part of the Linked Data cloud at the same time. The information structure of ReM^3 can be represented by a hierarchical URI model which has been implemented as a REST-ful interface as described below. The described work has resulted in an implementation of a framework called "Standardized Contextualized Access to Metadata" (SCAM) in version 4. The framework is built on top of a quadruple store (OpenRDF Sesame[1]), making it possible to identify sets of triples using Named Graphs as mentioned above.

5.2. REST-based Interface

There are three basic kinds of REST resources in a context: resource, metadata, and entry. There are two additional kinds of resources, the relations resource that contains relations from other entries, as well as the cached-external-metadata resource that contains a cache of the external metadata if the location type is reference or link reference.

The pattern below shows the URIs and allowed HTTP operations for the multiple kinds of REST resources:
{http-verb} {base-uri}/{context-id}/{kind}/{entry-id}

- *http-verb* is one of GET, PUT, POST or DELETE.
- *base-uri* is the base URI (namespace) that is specific for each system.
- *context-id* is a unique identifier for a context.
- *kind* is one of the kinds of REST resources.
- *entry-id* is a identifier for an entry that must be unique within each context.

Providing an easy-to-use and REST-oriented interface together with ReM^3 allows for enrichment of metadata as the protocol makes communication in both ways possible. Resources in other systems can be described by linking to them and building a connection between the metadata and the resource. Such connections are in turn exposed using Linked Data which integrates heterogeneous information sources.

The web API of SCAM is only summarized here, a more detailed description can be found in an earlier publication [7].

5.3. The Use of Linked Data

The main point for linking information in ReM^3 is the entry, but also lists are used to build indirect relations. Statements are used to keep resources and their metadata together. All involved entities are identified by dereferencable URIs whenever possible and HTTP is the standard protocol.

A SCAM repository can also be queried through a SPARQL endpoint. The ACL model of ReM^3 limits which metadata can be exposed. The SPARQL proto-

[1] http://www.openrdf.org

col does not support any access control, so this had to be solved on the level of the repository by exposing only public metadata. Other metadata, no matter whether completely private to the creator or restricted to groups, is not exposed at all through SPARQL. There are endpoints on two different levels:

1. A "global" endpoint for the whole SCAM repository, including all contexts and their entries.
2. An endpoint per context, including all entries of a context. This allows to restrict queries to a limited amount of entries and speeds up queries.

Information about named graphs is also exposed using the GRAPH keyword which allows to create views of contextualized resource metadata in SPARQL query results.

5.4. Additional Interfaces

The SCAM framework also has support for additional protocols, mainly aimed for harvesting and querying, such as OAI-PMH[2] and SQI[3]. SCAM supports both directions, that is, querying and harvesting other systems as well as being queried and harvested itself. The architecture of SCAM makes it possible to hook in additional protocols if required. The same applies to metadata converters, the infrastructure includes support for mapping metadata to and from RDF.

5.5. Interoperability and Implementation Experiences

The metadata editor in use allows editing of RDF graphs directly[4] and send it to the backend. Dublin Core-based application profiles (AP) are a natural choice because they map easily into RDF. To be able to do the same with Learning Object Metadata (LOM v1.0)-based profiles, a mapping from LOM to the Dublin Core Abstract Model (DCAM) was necessary. The DCMI developed such a mapping and published a draft in their Wiki[5]. On top of that, additional mappings were created to support the LRE v3.0 AP used by the Organic.Edunet project which is based on LOM and replaces respectively enhances some vocabularies. Dublin Core terms are (re)used wherever possible, only metadata properties specific to LOM were given an own identifier.

The SCAM backend supports HTTP content negotiation and performs conversions between metadata formats as needed. It is e.g. possible to send LOM/XML to the server and request RDF for the same metadata graph. The formats differ, but the information is largely the same due to a careful mapping that balances accuracy against discarding of information that cannot be translated in a good enough manner.

5.6. Scalability

Structured and scenario-oriented performance tests have not been carried out yet. However, as the SCAM version 4 framework is used in a number of projects there are experiences on how the system performs under conditions which are typical to the show cases presented further down. As an example, the installation for the Organic.Edunet project holds around 12.000 entries (each entry being a learning resource and its metadata) in several hundred contexts, which corresponds to around 0.5 million quadruples. The response times are very low and profiling showed that most of the time is spent on the serialisation of the queried information into the web client-friendly JSON format.

Another SCAM installation is used to harvest the OAI-PMH target of ARIADNE foundation[6] in order to triplify the large ARIADNE learning repository. More than a million learning resource descriptions have been stored in a single SCAM installation, resulting in around 50 million quadruples. Performance differences to the smaller Organic.Edunet installation are hardly noticeable.

However, there is one exception to the overall good performance: free-text queries on String literals. SPARQL queries using FILTER and regular expressions are very expensive. To solve this problem a Solr[7] index is used for searches in metadata literals. A listener infrastructure inside SCAM notifies Solr of events in the repository and (re)indexes entries and their metadata as soon as a change is made. This is important to keep the repository and the search index up to date and in sync. The combination of SPARQL and Solr queries allows for powerful and efficient searches even in large repositories.

[2] http://www.openarchives.org/pmh/
[3] http://www.prolearn-project.org/lori
[4] Based on RDF/JSON
[5] http://dublincore.org/educationwiki/
[6] http://www.ariadne-eu.org
[7] http://lucene.apache.org/solr/

6. User Interface: Building Rich Internet Applications using ReM^3

Since the recommended way to utilize ReM^3 is via its REST based interface we chose to focus on developing Rich Internet Applications (RIA), i.e. JavaScript applications that maintain state on the client side and use a REST-ful approach to retrieve and update data.

There are no hard restrictions on which applications can be built on top of ReM^3, in fact the information model is very generic and it should be possible to use it in a wide variety of applications. Still, certain applications are easier to build than others due to the nature of the information model. This section focuses on an application that more or less directly exposes the capabilities of the ReM^3, namely the Confolio web application. The Confolio application is by no means the only or for that matter necessarily the best way to expose the capabilities of ReM^3. However, it does sufficiently expose some of the complexity of building user interfaces that make use of the full flexibility of ReM^3.

Confolio provides portfolios for individuals and groups. Each portfolio provides a place to store resources - in the form of uploaded files, web content, physical entities or abstract concepts - together with descriptive metadata. A portfolio is represented as a ReM^3 context, and a resource together with its metadata corresponds to a ReM^3 entry. In figure 3 a work view is shown of a portfolio with a listing on the left and details of a selected entry on the right. Below two challenging user interface issues are discussed.

6.1. Presenting Entries

Entries often contain rich information, so to avoid overloading the user interface, only parts of the information are shown in each situation. The Confolio user interface contains three distinct situations where entries are shown (additional situations exists but are considered to be variations):

- S1 - non-selected entry in list.
- S2 - selected entry in list.
- S3 - entry details.

In each situation a piece of information of an entry is considered to be primary (P) or secondary (S), see table 1 for an overview. Primary information should be visible without further interaction from the user. Secondary information should be easy to reach via an interaction from the user without leaving the current view, for example via tool-tips or pop-ups triggered by mouse-clicks, gestures or keyboard shortcuts.

6.2. Presenting and Editing Metadata

The metadata expression may differ greatly between entries because:

- entries may represent different things, for example web pages or physical objects.
- entries may be described for different purposes and different target groups.
- entries may originate from different information sources which use different standards.

The use of RDF as common carrier allows these metadata expressions to coexist, both between entries and sometimes within a single metadata expression. This flexibility presents a challenge when presenting and editing metadata since very little can be taken for granted. The solution taken in Confolio is to rely on the library RForms[8] that generates user interfaces for both presentation and editing of metadata from a configuration mechanism called Annotation Profiles. The details on how RForms and Annotation Profiles are used to transform an RDF graph into a form is beyond the scope of this article and the interested reader is encouraged to look at [8] for details where also relations to other initiatives such as DCAP DSP are discussed.

To generate an editor, RForms must be told which Annotation Profile or which combination of Annotation Profiles to use. In theory, the user could be asked which Annotation Profile to use in each situation given that enough descriptive information is provided to make an informed decision. However, from a usability perspective it is often better to present users with a reasonable default and allow it to be changed into something more specific when needed. Each Confolio installation may configure a default Annotation Profile for every entry signature it wants to support. In table 2 the entry signatures of various aspects are shown in order of priority, that is, which signature aspect that has precedence when choosing an Annotation Profile.

In figure 4 we see basic metadata from Dublin Core combined with a field from IEEE/LOM regarding copyright statement.

In presentation mode the same Annotation Profile will be used, but only fields that have been filled in will be shown. If RForms detects that there are more metadata available than can be shown with the current Annotation Profile, it will look for other Annotation Profiles to use in conjunction. Such a situation can occur

[8] RForms is a JavaScript reimplementation of the older SHAME java library

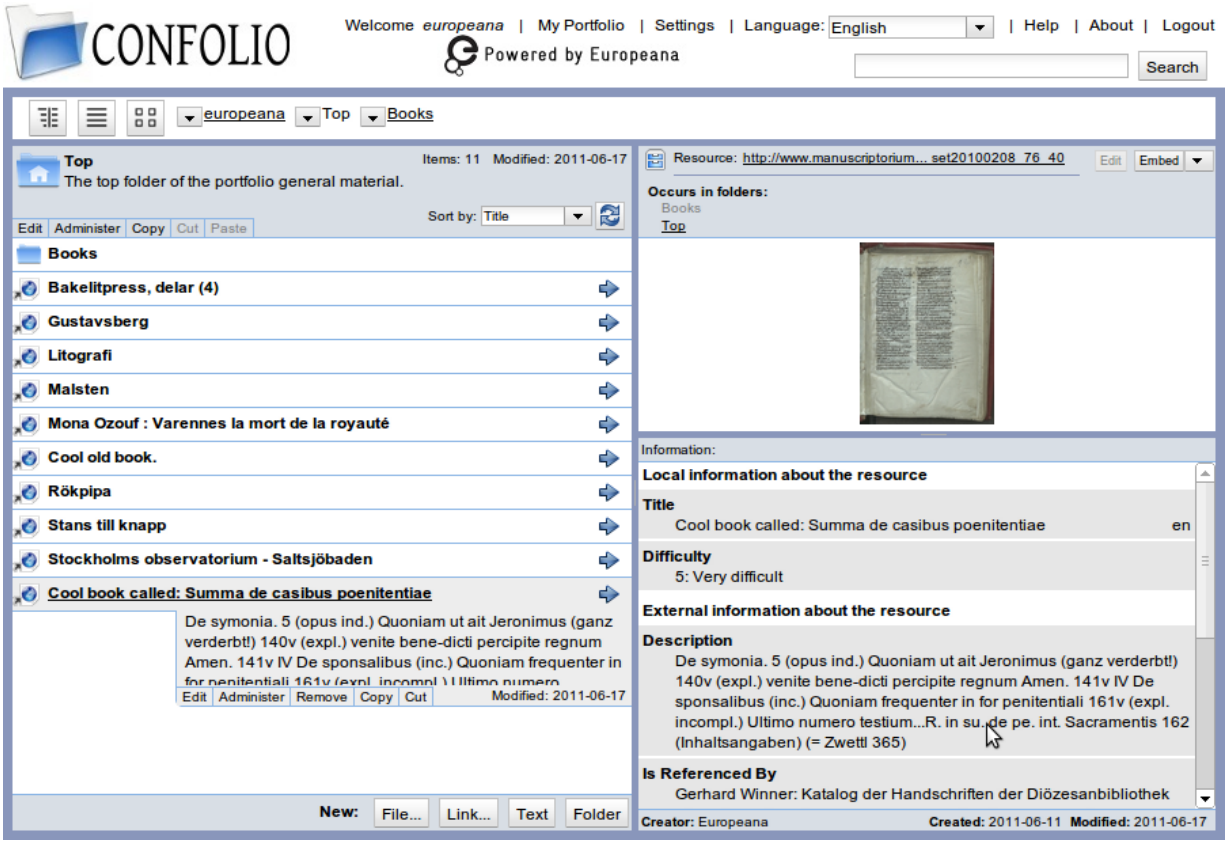

Figure 3. A screenshot of Confolio

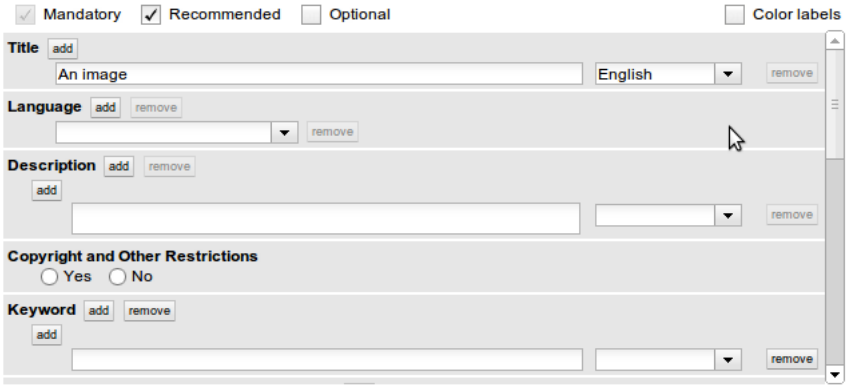

Figure 4. A metadata form which blends Dublin Core and LOM metadata

when entries originate from another system, or for that matter, the user has switched back and forth between Application Profiles or intentionally combined them.

7. Showcases

The following showcases are all centered around learning resource descriptions. They involve annotation of resources who are uploaded into or linked from the Confolio web application as well as enhancement

Table 1
Primary and secondary information of an Entry

Information	S1	S2	S3
Digital representation of the resource, for example a picture or video			P
Generic information such as title and description	P	P	
Representative icon	P	P	P
Format, if the resource has a digital representation		S	S
Application type, for example resource is an academic paper or an event		P	P*
Full local metadata, not available if location type is Reference		S	P
Full external metadata, only available if location type is Link or Link Reference		S	P
Metametadata including things like:		S	P
locationtype, e.g. if entry is a link or not	P	P	P
builtintype, e.g. if the entry is a list, person etc.	P	P	P*
representationtype, e.g. if the resource of the entry is retrievable	P	P	P*
original creator and contributors		S	P
creation and modification dates		P	P
access control referencing other users and groups			S*
Relations to other entries, including:			
one or several folders where the entry appears			S
comments	S	S	S*

*Parts planned for later versions of Confolio

Table 2
Signature aspects for choosing Annotation Profiles

Prio	Signature Aspect	Explanation
1	Builtin Type	A configuration can be given for all Builtin Types except none.
2	Service	The configuration provides a set of services identified via URL patterns that are checked against the resource's URI.
3	Application Type	If a type is given via rdf:type in the metadata this is considered to be an Application Type.
4	Mime Type	The configuration may cover the full mimetype or only the initial part to match more broadly, for example "image/jpeg" and "image" respectively.
5	Default	A default Application Profile to use when no other signature aspect applies.

and contextualisation of metadata which is harvested from other repositories.

7.1. Organic.Edunet

The goal of the now successfully completed Organic.Edunet project was to facilitate access, usage and exploitation of digital educational content related to Organic Agriculture and Agroecology. The combination of the SCAM framework and the Confolio web application was used from the very beginning of the content population process. The Organic.Edunet federation consists of numerous SCAM and Confolio installations which are harvested using OAI-PMH by the Organic.Edunet portal[9] on a regular basis. More than 11.000 educational resources have been described with educational metadata by several hundred contributors so far. Roughly half of the learning resources were already described with some basic metadata without educational information. These already existing metadata instances were harvested using OAI-PMH and converted and mapped into RDF and DCAM.

Additional educational metadata was added in the Organic.Edunet repositories. This approach is greatly supported by the ReM^3 model, which allows a differentiation between local and external resources and metadata. Such a differentiation in combination with the use of separate metadata graphs is used to enhance

[9] http://portal.organic-edunet.eu

harvested resource descriptions from e.g. the Intute[10] repository. In this case, two metadata graphs are used per resource: one with cached external metadata (in simple DC format harvested using OAI-PMH) and one with local educational metadata using LOM/DCAM. If Intute modifies the metadata in its repository this will be reflected in the SCAM repository after the next re-harvest. The locally annotated educational metadata remains untouched, which is only possible by keeping metadata from different origins in separate graphs.

7.2. ARIADNE

Following up on the results from Organic.Edunet and as proof-of-concept for the general applicability of ReM^3 and the reference implementation, the OAI-PMH target of the ARIADNE foundation[11] was harvested and triplified, resulting in around 50 million triples within 1.2 million metadata graphs in a SCAM repository. The provided LOM metadata was mapped into the DCAM and converted into RDF during the harvesting process. As in the case of Organic.Edunet, a scaffolding approach to describing learning resources can be taken. The surrounding context of a learning resource can be bootstrapped using Link References, e.g. by providing different descriptions for different learning scenarios.

Another benefit of having all ARIADNE metadata in RDF is the possibility of running SPARQL queries against a large amount of learning resource descriptions. SPARQL can be used to formulate complex queries based on the LOM/DCAM elements to query and build graphs in the repository. An example is requesting a list of all LOM Learning Resource Types that a specific person has used when annotating learning materials. More complex queries can be formulated by using additional metadata elements and advanced query logic. A use case is the contextualisation of learning resources, to get information on how different persons described the same resource with different metadata to reflect their specific use within various educational (or other) activities. The amount of triples will increase in the future as the implementation of the LOM/DCAM mapping is refined and completed.

[10] http://www.intute.ac.uk
[11] http://www.ariadne-eu.org

7.3. Europeana

The authors participated in the "Hack4Europe!" competition in Stockholm[12] which has been arranged by the Europeana project[13]. The goal of the competition was to show the potentials of the Europeana content by building applications to showcase the social and business value of open cultural data.

During the hack day the authors developed another showcase to demonstrate how heterogeneous metadata can be managed using ReM^3. Like in Organic.Edunet, a combination of the SCAM framework and Confolio is used. Both applications were extended in a way so that they can search in Europeana and extract Europeana metadata from the search results. This allows for adding resources directly from a Europeana search result to a user's personal portfolio for further annotation with contextual metadata. The demonstrated use case[14] was to search for resources which are suitable to be used in an educational context and to turn them into learning resources by annotating them with educational metadata in Confolio. Technically this means searching and caching metadata described using the Europeana Data Model and adding educational metadata (e.g. in LOM/DCAM) using a ReM^3 Link Reference in the SCAM framework. Everything is integrated into the Confolio interface and the end user does not have to know anything about where the metadata originates from or which formats that are used.

8. Conclusions

8.1. Problems in Retrospect

The first problem stated was that it is unclear how to perform an integration of heterogeneous information sources and a harmonization of metadata expressions from different standards. The RDF (meta) data model is used as a common carrier by the information model and applied by the reference implementation. Together with the ReM^3 location type this provides a solution to the problem on how to integrate heterogeneous metadata and how to enrich metadata originating from other sources, as mentioned in the second problem. The latter also benefits from a clear distinction of separate de-

[12] http://www.hack4europe.se
[13] http://europeana.eu
[14] http://hack4europe.se/information/meta-solutions-europeana-portfolio/

scriptions, implemented using Named Graphs and kept together by ReM^3 entries.

The solution to the next problem, i.e. the avoidance of duplication of metadata between systems, is based on the ReM^3 location type approach as well. An ReM^3 entry keeps track of all involved pieces of information and links back to the original source of information. Cached metadata graphs are used for performance reasons. This linking approach is also part of the solution to the problem on how to expose the combination of resource and their metadata in a Linked Data way. The ReM^3 entry graph contains statements to keep resources and their metadata together with the possibility of including additional relations. HTTP and dereferencable URIs are used wherever possible.

The last problem, replacing harvesting techniques with Linked Data, can be answered by pointing out that harvesting is not necessary but can be used if desired. ReM^3 with entry and the location type "Link Reference" allows to use harvesting together with Linked Data.

8.2. Reusing, Linking and Contextualizing Metadata

The harvested repositories are brought into the Linked Data world by giving everything an HTTP URI. In addition, everything can be identified, queried and updated using HTTP. The resources are linked with the describing metadata in the entry information of the entry. All parts that belong to an entry and are referenced by it, i.e. resource, metadata graphs, author and contributor, have a URI and are accessible via HTTP.

A contextualisation of a resource becomes possible, the same resource can be described with different metadata graphs, depending on its context of use. Basic metadata with e.g. title and description can be enhanced with educational metadata in an additional metadata graph and used in an educational context. A resource described with educational metadata by teacher A to be used in course X can be contextualized for another lecture by teacher B in course Y, just by annotating it with additional metadata, building upon the already existing descriptions. Write-access is not necessary, the metadata belong to the respective teachers, and since the LD principles are followed, anyone can point to and describe anything.

8.3. General Applicability and Extendability

A focus on Semantic Web technologies and support for SPARQL and Linked Data in general makes it possible to contribute to the LOD cloud without any additional effort. Even though the system currently is mostly deployed in the context of learning repositories with a focus on educational metadata, there are no restrictions regarding the metadata standards or application profiles in use. The supported protocols for metadata harvesting and querying can easily be extended, even though the main protocols are OAI-PMH and SQI. However, as a proof-of-concept, harvesters for proprietary protocols have been implemented and tested.

9. Next Steps

In the context of metadata and resource management it is relevant to provide means for creating additional links between e.g. learning resources. Currently, interlinking is mostly based on lists (assuming that resources contained in the same list have something in common) and entries keeping together different metadata graphs describing the same resource in different contexts. In addition to that it would be useful to provide semantics for lists, such as e.g. programmes, courses, course modules, etc. and explicit semantic relations between resources by exposing this in the user interface.

As mentioned above, no structured benchmarking with large datasets has been carried out yet. The system works well with large repositories (several million entries in one installation), but there is no knowledge about where the limits are regarding concurrent access in typical usage scenarios and the number of resources managed by the system. An elaborated set of benchmarks has to be developed to collect significant evidence regarding performance and scalability. However, this is more related to the reference implementation than to the information model.

The REST-ful interface has its limitations in certain collaborative use cases. If an update of a resource or metadata is performed by a client, all clients which have data cached e.g. in the browser cache have to make a conditional reload in order to see the changes. Due to the nature of HTTP, the clients do not get notified of any changes on the server side, they have to poll instead. This is an issue which can be solved by adding support for push technologies such as WebSockets [11] to the SCAM framework.

In addition to pushing information updates to the clients, it can be of interest to keep a version history of a resource and its metadata. Such snapshots in com-

bination with a short summary of changes can provide input for collaborators on e.g. what has been changed by whom, when and why; to put it simply, information about how a resource has evolved over time.

Acknowledgements

The work presented in this paper has been partially carried out with financial support from the EIT ICT Labs activity "Data Bridges" in the thematic action line "Digital Cities of the Future", and the EC-funded projects Organic.Edunet[15] (grant agreement ECP-2006-EDU-410012), TEL-Map[16] (grant agreement 257822) and ROLE[17] (grant agreement 231396), which the authors gratefully acknowledge.

[15] http://www.organicedunet.eu
[16] http://www.telmap.org
[17] http://www.role-project.eu

References

[1] The linking open data cloud diagram. http://richard.cyganiak.de/2007/10/lod/.

[2] W3C TAG issues list: httprange-14: What is the range of the http dereference function? http://www.w3.org/2001/tag/issues.html#httpRange-14.

[3] Final draft standard for learning object metadata (LOM) IEEE 1484.12.1-2002. Technical report, 2002.

[4] T. Berners-Lee. Linked data - design issues. http://www.w3.org/DesignIssues/LinkedData.html, 2009.

[5] J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs. *Web Semantics Science Services and Agents on the World Wide Web*, 3(4):247–267, 2005.

[6] H. Ebner, N. Manouselis, M. Palmér, F. Enoksson, N. Palavitsinis, K. Kastrantas, and A. Naeve. Learning object annotation for agricultural learning repositories. *Ninth IEEE International Conference on Advanced Learning Technologies*, (9):438–442, 2009.

[7] H. Ebner and M. Palmér. *A Mashup-friendly Resource and Metadata Management Framework*, volume 388, pages 14–17. CEUR-Proceedings Vol-388, 2008.

[8] F. Enoksson, M. Palmér, and A. Naeve. An RDF modification protocol, based on the needs of editing tools. 2007.

[9] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, 2000.

[10] F. Gandon and O. Corby. Name that graph or the need to provide a model and syntax extension to specify the provenance of rdf graphs. *Proceedings of the W3C Workshop - RDF Next Steps*, 2010.

[11] I. Hickson. The WebSocket API, W3C working draft. http://www.w3.org/TR/websockets/.

[12] I. Jacobs and N. Walsh. Architecture of the world wide web, volume one. *World*, (December):1–37, 2004.

[13] C. Lagoze. The open archives initiative protocol for metadata harvesting. *Open Archives Initiative*, pages 1–6, 2008. http://www.openarchives.org/OAI/openarchivesprotocol.html.

[14] F. Manola and E. Miller. RDF primer. *W3C Recommendation*, 10(February):1–107, 2004. http://www.w3.org/TR/rdf-primer/.

[15] M. Nilsson. *From Interoperability to Harmonization in Metadata Standardization: Designing an Evolvable Framework for Metadata Harmonization*. PhD thesis, 2010.

[16] E. Prudhommeaux and A. Seaborne. SPARQL query language for RDF. *W3C Recommendation*, 2009(January):1–106, 2008. http://www.w3.org/TR/rdf-sparql-query/.

[17] L. Sauermann, R. Cyganiak, and M. Völkel. Cool URIs for the semantic web. *W3C Interest Group Note*, 49(December 2008):1–15, 2008.

[18] R. Thurlow. RPC: Remote procedure call protocol specification version 2. *Internet Network Working Group Request for Comments*, (5531):25, 2009. http://www.ietf.org/rfc/rfc5531.txt.

[19] H. Van De Sompel, R. Sanderson, M. L. Nelson, L. L. Balakireva, H. Shankar, and S. Ainsworth. An HTTP-based versioning mechanism for linked data. *Proceedings of the Workshop on Linked Data on the Web LDOW 2010 April 27 Raleigh USA*, 2010.