Building a Semantic IntraWeb
with Rhizomer and a Wiki

Roberto García
Universitat de Lleida,
Jaume II 69,
E-25001 Lleida, Spain
+34 973702740

roberto@rhizomik.net

Rosa Gil
Universitat de Lleida,
Jaume II 69,
E-25001 Lleida, Spain
+34 973702750

rgil@diei.udl.es

ABSTRACT
IntraWebs enjoy the same technological facilities that have made the World Wide Web a success. Conversely, they can experience the same inconveniences derived from this success. Our proposal is to apply the same solutions that are being proposed in the Web context to manage huge information spaces, Semantic Web technologies. Rhizomer is a semantic metadata browsing and management component that is combined with a wiki engine in order to build Semantic Web portals. When they are combined with access control mechanisms, the resulting semantic portal can be scaled down to build a restricted access semantic intraweb. In other words, we get an easy to use HTML based solution that integrates three main components. First, there is an online content management system based on wiki. Second, a user friendly semantic metadata for searching, browsing and editing. And third, access control mechanism that allow implementing content workflows within different private publishing spaces or the public one, as it has been tested in the Rhizomik intraweb for a research community.

Categories and Subject Descriptors
H.3 [**Information Storage and Retrieval**]

H.5 [**Information Interfaces and Presentation**]

H.4 [**Information Systems Applications**]

General Terms
Management, Design, Human Factors.

Keywords
Semantic Web, IntraWeb, Corporate Web, Wiki, Content Management System.

1. INTRODUCTION
Information sharing and management in corporate or "internal" scenarios has been experiencing the same problems than those experienced in the whole World Wide Web context, although at a smaller scale as it is obvious. This is due to the fact that the same Web technologies are being applied successfully at these different scales and that their great success has produced the same manageability problems.

Therefore, it seems reasonable that the same recent contributions to the Web geared towards solving these problems, the semantic enrichment of the Web, can also provide great benefits in the more limited corporate webs or "IntraWeb" domain.

The Rhizomer project constitutes a technological framework that can be used to build up semantic web portals. Rhizomer manages RDF metadata in a user-friendly way. It facilitates not just the common Semantic Web to end-user interaction provided by semantic web browsers, it also provides the reverse interaction path: end-users can create, edit and remove semantic metadata.

The whole interaction is performed through a "classical" HTML interface in a usable way, which minimises user efforts and maximises the benefits they obtain from their Semantic Web experiences. The overall intention is to minimize the gap among computers and human beings in the context of Semantic Web.

The semantic metadata management part provided by Rhizomer is combined with an easy to use and simple content management system based on a Wiki [1] engine. The wiki provides the means to create and maintain information objects that are described with semantic metadata.

Semantic metadata and the underlying ontologies facilitate content management in the intraweb. For instance, it is possible to integrate at the ontology level the specific metadata schemes used by different enterprise parts or even different enterprises working in an integrated value chain.

Altogether, the wiki and the Rhizomer component build up an HTML solution that can be easily scaled down to restricted application domains in order to build a semantic intraweb. The wiki and Rhizomer are enriched with authentication mechanisms that allow conditioned content access.

The rest of this paper is organised as follows. Section 1.1 shows how the Rhizomer approach to Semantic Web browsing improves the user experience. Then, section 3 presents the additional functionalities that Rhizomer introduces in order to enhance end-user interaction, i.e. semantic metadata creation, edition and removal. Section 4 details the component management wiki features. Finally, section 5 presents the conclusions and the future work.

The Rhizomik web site[1] is based on Rhizomer and a wiki engine and constitutes an example of how they can be used to produce a semantic intraweb for research purposes. The Rhizomik site, which is shown in Figure 1, integrates mechanisms for the workflow from internal drafts to publicly available research reports.

---

[1] Rhizomik initiative, http://rhizomik.net

Figure 1. Rhizomik web site based on Rhizomer and Wiki

1.1 The Rhizome Metaphor

The Rhizomik initiative is inspired by the "Rhizome Metaphor" posed by Gilles Deleuze and Felix Guattari in their book "A Thousand Plateaus" [2]. The Rhizomik initiative tries to use the rhizome metaphor as its inspiration when working with knowledge from a scientific, technological but also philosophical point of view.

The rhizome serves as a metaphor for the multiplicity and infinite interconnectedness of all thought, life, culture, and language. This quote is just to pose some subtle and curious connections of the rhizome metaphor and the Semantic Web:

"...*it is odd how the tree has dominated Western reality and all of Western thought. Thought is [, however,] not arborescent, and the brain is not a rooted or ramified matter... unlike trees or their roots, the rhizome connects any point to any other point ... In contrast to centred (even polycentric) systems... A rhizome has no beginning or end; it is always in the middle, between things, interbeing, and intermezzo. The tree is filiation, but the rhizome is alliance, uniquely alliance. The tree imposes the verb 'to be', but the fabric of the rhizome is the conjunction, 'and...and...and...'. This conjunction carries enough force to shake and uproot the verb 'to be'... The middle is by no means an average; on the contrary, it is where things pick up speed... The rhizome operates by variation, expansion, conquest, capture, offshoots...*".

2. SEMANTIC METADATA BROWSING

Usability guides how end-user interaction with the Semantic Web is faced by Rhizomer. This interaction is commonly viewed, when talking about the Semantic Web, just from the Semantic Web application towards the user, i.e. what can be called Semantic Web browsing, which is detailed in this section.

This area is being extensively explored in the Semantic Web area. There are many semantic browsers but, almost in all cases, they share a browsing paradigm that is not usable enough from our point of view, we call it the "Subject-centric approach", section 2.2.

Our proposal is detailed in section 2.3 and it is employed in Rhizomer in order to construct a more user-friendly experience when browsing the Semantic Web. The Rhizomer browsing approach is based on a simple algorithm, detailed in sections 2.4 and 2.4, and it is compared to the subject-centric one in section 2.6.

However, the unidirectional experience provided by Semantic Web browsers is not enough. The other direction must be considered and the user should be able to create, edit and remove semantic metadata in a usable way. This part is explained in section 3.

2.1 World Wide Web browsing paradigm

In the Web, the browsing paradigm is based on navigating web pages and links, which constitute its basic building blocks. Web pages have content intended for human consumption and links relate web pages as shown in Figure 2.

Web pages content and links are based on HTML and derived languages. Their design should have accessibility and usability principles in mind.
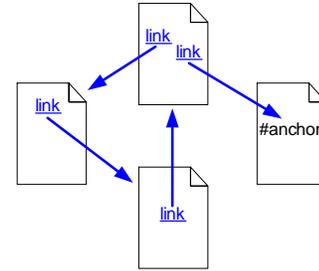

Figure 2. World Wide Web browsing paradigm

This browsing paradigm cannot be directly applied to the Semantic Web because it is based on a different ground model. The building block of the Semantic Web is the triple <*subject, predicate, object*>, which combined builds-up graphs, see Figure 3.
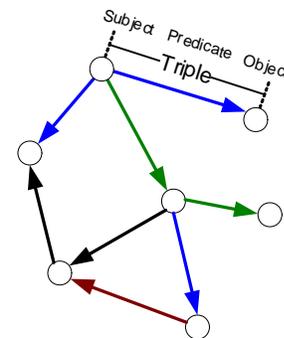

Figure 3. Semantic Web model

Semantic Web metadata is primarily intended for machine consumption. However, it is clear that it must be also accessible for human users in order to facilitate semantic web applications debugging, results presentation, querying, etc.

However, the WWW browsing paradigm cannot be applied to the Semantic Web. Semantic Web metadata can be packed in web documents, e.g. a web-accessible XML file; however it is not practical to use this approach as the basis for Semantic Web browsing.

First, not all metadata is available as web documents. Moreover, semantically related metadata might be packed in different web documents, which reduces the benefits of semantics-enabled metadata browsing.

Usually, Semantic Web metadata is available from databases. In this case, as it can be also the case for web documents, the amount of metadata is too big for human-user consumption.

Therefore, what is needed is a Semantic Web browsing paradigm and support system that can browse metadata coming from different sources through semantically coherent fragments that facilitate human consumption. Now the question is: how to define fragments?

2.2 Subject-centric Approach

The simpler approach to fragment semantic web graphs is to define the fragment as the set of all triples with the same subject. This is also semantically coherent because these triples are those describing the subject resource.

This approach is used in almost all Semantic Web browsers. For instance, Piggy Bank [3] or Brownsauce [4] generate HTML views for a given resource, i.e. a node of the graph. Table 1 shows a schematic example of such kind of view.

Table 1. HTML Table view of a metadata fragment

Rosa Gil	
EMAIL	rgil@diei.udl.es
FN	Rosa Gil
N	(anonymous item)
Show Referrers	

The view contains all the triples that have the resource as subject and it is shown as a HTML table. The table header shows the identifier of the described resource, i.e. the subject of all the triples in the metadata fragment. The following table rows contain two columns. The first one is for each triple property and the second one for the corresponding objects.

There are also other frameworks that are not explicitly oriented to RDF browsing and provide heavyweight solutions that can be adapted to navigation, e.g. the SEAL framework [5]. Moreover, there are other Semantic Web navigation tools that generate non-HTML views, e.g. Haystack [6] generates text views and [7] graphical representations.

In many cases the shown metadata fragment is augmented with triples that have the described resource as object, i.e. reverse triples that do not have the described resource as source but as destination. Additionally, more triple levels can be included, i.e. the triples that have the objects of the original triples as subject, and thus recursively. However, it is not common to include additional triple levels because it can make the fragments too big and break semantic coherence, i.e. the user looses the perspective about what is being described.

Finally, the semantic coherence is also lost when this approach is used and anonymous resources are involved. Anonymous nodes are get identified by the context in which they appear, i.e. the triples that reference them and the identified resources appearing in these triples.

However, this context is broken when the metadata fragments are built. Anonymous nodes get temporal identifiers, which are needed in order to build the graph, but they are also used in the user interface as if anonymous nodes were like other identified resources. For example, Table 1 shows a reference to an anonymous resource as the value of the "N" property. An

additional browsing step is necessary to get the triples describing the anonymous value and make it all semantically coherent, as it is shown in Table 2.

Table 2. Additional step for anonymous resource metadata

(anonymous item)	
Given	Rosa
Family	Gil
Show Referrers	

Consequently, the same approach is applied to anonymous resources. All the triples that have the anonymous resource as subject are shown. Therefore, the metadata describing the anonymous resource is shown but the identification context is lost in the presented view. There might be reverse triples to go to the identified resource that defines the identification context for the anonymous resource. In any case, however, the presented view does not include the whole context so the user might not be aware of this fact.

For example, one common examples of this situation is when RDF containers are used. They do not usually get an identifier so the resources for the containers are anonymous. Therefore, when the metadata for the resource described with container values is shown, the set of container values are not shown together with the resource to which they are associated.

In order to solve the inconveniences of the Subject-centric approach we propose the Rhizomer Approach that is detailed in the next section.

2.3 Rhizomer Approach

The Rhizomer Approach for Semantic Web browsing is also based on fragmenting the metadata graph in a subject-centric way. However, a metadata fragment generated by Rhizomer is more than just the considered resource and all the triples in which it participates as the subject.

The set of triples for a subject-centric is enlarged with all the metadata that depends on the selected subject for its identification. Therefore, the graph is traversed starting from the resource acting as the subject through all possible paths until identified resources or literals, which include all the intermediate anonymous resources. All the traversed triples constitute a Rhizomer metadata fragment and it is what is shown at a browsing step. Figure 4 shows a simple RDF graph where four fragments can be identified.
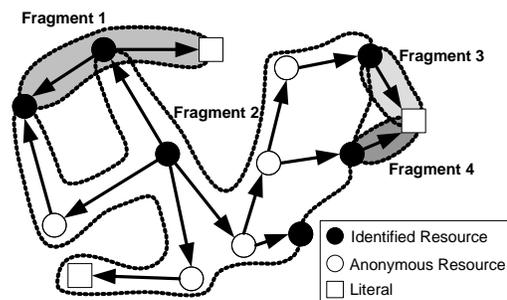
Figure 4. RDF graph and four Rhizomer fragments

As in the case of the subject-centric approach, the resources appearing as the object of the terminal triples can be navigated through new navigation steps that generate new metadata fragments describing the resources asked for detail. This has been already shown in Table 2, which shows a browsing step initiated from the Table 1 fragment.

However, Rhizomer shows all the metadata in the same identification context together. This way, a greater level of semantic coherence is maintained and the user experience is improved. Anonymous resources do no break down the metadata describing a resource and they are shown together with the resources that contextualise them, as it is shown in Table 3.

Table 3. HTML Table view of a metadata fragment

Rosa Gil		
EMAIL	rgil@diei.udl.es	
FN	Rosa Gil	
N	Given	Rosa
	Family	Gil
Show Referrers		

To conclude the presentation of the Rhizomer approach, it is important to take into consideration the presence of cycles in the metadata graph and to avoid considering triples already added to a fragment. This can be easily implemented as it is shown in the algorithm in the next section.

2.4 Algorithm

The browsing fragments are built from a set of selected resources. For each resource, all the triples where the resource plays the subject role are selected. Then, for each object of the selected triples, if it is an anonymous resource, all the triples where the anonymous resource is the subject are also selected. Then, and recursively, the same procedure is applied to the new object anonymous resources. The algorithm for building the fragment for a given resource is detailed in Table 4.

Table 4. Rhizomer algorithm for fragments browsing

```
TripleSet buildFragment(Resource r, Triples f)
{
  TripleSet rTriples = TriplesWithSubject(r)
  fragment += rTriples

  for each Triple t in rTriples
    such that NotInSet(t,fragment)
  {
    fragment.add(t)
    if anonymous(t.object)
    {
      TripleSet anonTriples =
        buildFragment(o, fragment)
      fragment.add(anonTriples)
    }
  }
  return fragment
}
```

2.5 Multi-language Support

When the browsing fragment is rendered as HTML or other format, the triple URIs are replaced with their corresponding labels if available. If there is not a defined label for the URI, it is shortened to its fragment identifier, the string after the '#', or to the substring after the last '/'.

This is done in order to improve readability. An additional improvement, which is not common in existent Semantic Web browsers, is multi-language support. The Rhizomer algorithm, in addition to the triples that compose the fragment, selects all the triples that define labels for all the involved resources, i.e. subjects, predicates and objects.

These labels use to be annotated with a language attribute. When the fragment is rendered and a preferred language specified, the preferred language labels are selected if they are available. If not, the default language has been set to English so the labels with the 'en' value for their language attribute are selected when available. If there is not any label with the preferred or default language attribute, a label without language attribute is used.

2.6 Comparative Study

As it has been previously shown, the subject-centric approach is the common method for browsing fragments building. It has been also shown how the Rhizomer approach works and how it builds more coherent fragments by incorporating all the metadata identified by the described resource, i.e. the anonymous resources related to it.

It might seem that the Rhizomer approach makes the fragments too big for human user consumption. A statistical study has been performed with different sets of Semantic Web metadata. The results of this analysis are shown in Table 5.

Table 5. Average fragment size in triples

Metadata Set	Set Size	Subject centric	Rhizomer
Musicbrainz[2]	8121	6.6	7.7
CIA World Factbook[3]	876	4.3	14.4
Wine Ontology[4]	1839	2.6	7.3

As it can be seen from the statistical analysis, there is a slight increase in the size of the metadata fragments when the Rhizomer approach is used. It is greater when there is an intensive use of anonymous resources, for instance in ontologies as class restrictions do not use to be identified or in common real world metadata because we do not use to give universal identifiers to everything.

In the case of the Musicbrainz metadata, the difference between the subject-centric and Rhizomer approach is smaller because there are few anonymous resources. Musicbrainz describes artists, albums, tracks, etc. and all of them must have an identifier in order to be easily referred. The only anonymous resources are containers, which are used, for instance, to group all the album tracks.

---

[2] http://musicbrainz.org, U2 discography

[3] Factbook for Spain, http://www.daml.org/2003/09/factbook/sp

[4] http://www.w3.org/TR/owl-guide/wine.rdf

This slight increase in size of the fragments does not put them out of the human user reach. On the contrary, it facilitates browsing because fewer steps are required. Moreover, all the metadata related to the same identifier is shown together and this increases its coherence and the usability of the whole system.
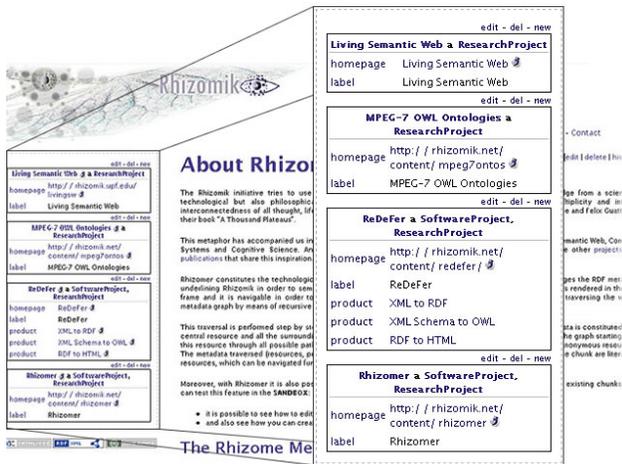

Figure 5. Rhizomik interface with Rhizomer zoom

As it has been pointed out, the inclusion of the anonymous resources together with the resource that identifies them is not common in the existing Semantic Web browsers. However, we have found a similar approach in the DBin project [8]. This project defines the term called RDFN (RDF Neighbourhood) that is based on a similar approach. The main difference is that this approach is specially tailored for producing digital signatures of pieces of a RDF graph and that it does not consider triples directionality.

Therefore, it is not suited for Semantic Web browsing but it shows that the best way of segmenting a graph is using an approach based on a subject-centric approach plus the anonymous resources identified by the subject. To conclude this section, Figure 5 highlights the metadata browsing part of the Rhizomik site, which is based on this approach.

3. METADATA EDITION
The previously presented browsing capabilities provide a simple Semantic Web user interface. It is simple because the interaction aspect is limited to selecting the piece of metadata to show next. However, the user cannot take this interaction further.

In order to improve user experience, the browsing capabilities have been enriched with editing and querying functionalities based on semantics-enabled forms. These are common HTML forms that take advantage of some simple conventions to make them semantics-enabled. Moreover, they are automatically generated from RDF metadata using a XSL transformation. Table 6 shows an example of such a form to build semantic queries. It is distinguished from an editing form because it contains a hidden field that specifies the "query-ref" performative, i.e. the pragmatics of the form is to define a query by reference speech act.

Table 6. Semantics-enabled query form

```
<form name="semform" action="RDFSOAPSender"...>
  <input type="text" name=
    "http://purl.org/dc/elements/…/title"/>
  <select name="http://purl.org/dc/…/subject">
  <option value=
    "http://www.iptc.org/subjects#034556000">
    …
  </select>
  <input type="hidden" name="performative"
    value="query-ref">
  …
</form>
```

As it can be seen in the form, the names of the form fields are URIs coming from RDF schemas and web ontologies. The intention is to make the form fields the building blocks to generate triples from forms. The fields' names are the predicates and their values the objects. The subject, in the case of a query, is not known, i.e. it is anonymous, and so it does no need to be specified. In fact, the resource URI will be the response we will get from the query.

The form fields are interpreted as the known terms we use to perform the search. For instance, the form in Table 6 can be filled with the "The Best of*" value for the input field, i.e. Dublin Core title, and the first option of the select field, i.e. Music is the RDF type. In the case of a "query-ref" form, the fields are interpreted as the triples that build up a semantic query. Therefore, an RQL [9] query, or another semantic query language, can be generated as the one shown in Table 7.

Table 7. RQL semantic query for Table 6 form

```
select graph(S) from
  {S}http://purl.org/dc/elements/…/title{Y0},
  {S}http://purl.org/dc/elements/…/subject{Y1}
where

  Y0 like "Corporate news*" and
  Y1= http://www.iptc.org/subjects#034556000
```

If the performative is "inform", then the form fields are interpreted as the building blocks for a set of triples for new metadata. In other words, this kind of forms is used to edit new or existing metadata. Now, it is necessary to specify the subject for the triples. This is done using a field named "rdf:ID" or "rdf:about". There might be also other subjects for the anonymous resources included in the edited metadata fragment.

The anonymous subjects are defined using hidden form fields named "rdf:ID" and valued with a temporal identifier just to make possible to build the graph. When all the triples for the anonymous subject have been specified, and another anonymous subject or the main identified subject is to be described, a new hidden field with the "rdf:ID" of the subject of the following triples is introduced. Table 8 shows an example of metadata edition form.

If the fields in the Table 8 form are filled with the appropriate values, e.g. "http://griho.net/people/rgil" for the non-hidden "rdf:ID" field, when it is submitted, the metadata shown in

Table 9 can be easily generated applying a direct conversion from form fields to triples.

Table 8. Semantics-enabled edit form

```
<form name="semform" action="RDFSOAPSender"...>

<input name="…rdf-syntax-ns#ID" type="text"/>
<input name="…vcard-rdf/3.0#EMAIL" type="text"/>
<input name="…vcard-rdf/3.0#N" type="text"/>
<input name="…rdf-syntax-ns#ID" type="hidden"
    value="_:anonid1"/>
<input name="…vcard-rdf/3.0#Given" type="text"/>
<input name="…vcard-rdf/3.0#Family" type="text"/>
<input name="…rdf-syntax-ns#ID" type="hidden"
    value=""/>
<input name="…vcard-rdf/3.0#FN" type="text"/>
<input type="hidden" name="performative"
    value="inform">
    …
</form>
```

Table 9. Metadata from Table 8 edit form submit

```
<http://rhizomik.net/~rosa>
  vcard:EMAIL
    <mailto:rosa.gil@diei.udl.es>;
  vcard:N [
    vcard:Family "Gil";
    vcard:Given "Rosa" ];
  vcard:FN "Rosa Gil".
```

Therefore, semantics-enabled forms facilitate a greater level of interaction with metadata through a Semantic Web browser. In addition to metadata browsing based on Rhizomer-like fragments, it is also possible to edit such fragments or create new ones using the same semantics-enabled forms.

As it has been shown, a direct parallelism can be established from form fields to triples, so the new or edited metadata can be generated from the user interaction with the form when it is submitted. Moreover, the reverse way is also direct, from triples to form fields. Therefore, the edition forms can be generated from existing metadata in order to edit it or generate new metadata based on predefined patterns.

Figure 6. Automatically generated metadata edition form

This two-ways mapping has been implemented in the Rhizomik semantic portal (http://rhizomik.net). The form fields to triples transformation is implemented when the form is submitted using JavaScript. The RDF triples to form transformation is implemented at the web server using an XSL transformation.

The implementation details are not given here due to space limitations but they can be obtained from the Rhizomik site. Figure 6 shows a form generated from RDF metadata like that shown in previous examples.

4. CONTENT MANAGEMENT WIKI

As it has been already introduced, the semantic metadata management component is complemented with a content management system based on a wiki engine. They operate independently, i.e. they are based on two separate web applications. They are coordinated just by the URL that identify the wiki contents and are referred from the RDF metadata.

From the metadata view shown by Rhizomer, wiki contents can be accessed by following URL-like resources in the underlying RDF triples. On the other hand, when a wiki content is shown it is possible to query Rhizomer for the available metadata for the wiki content URL.

The wiki engine is based on a simple Java implementation that directly stores HTML code, i.e. it does not employ any wiki scripting code for wiki pages content. This aspect facilitates the implementation and allows integrating a WYSIWYG HTML editor that facilitates enormously end-users content generation and management.

The HTML editor provides a word processor like interface and a content upload interface based on the FCKEditor[5], which is shown in Figure 7. The upload interface allows uploading non-HTML content to the wiki.
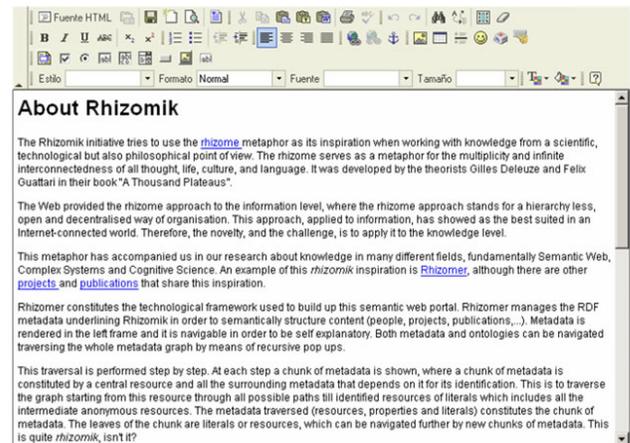
Figure 7. Wiki web-based HTML editor

The wiki engine is implemented on a Java Servlets container. The container authentication mechanisms are used to implement access control to content and metadata. This allows defining different private areas with different visualisation and edition rights based on authenticated users roles. Moreover, it is possible to maintain an area for public access.

This kind of access control implementation, although it is simple, has allowed building the Rhizomik research management site. The site has different private areas where research groups work collaboratively on draft research reports

---

[5] FredCK FCKEditor, http://www.fckeditor.net

that, ultimately, can me transferred to the public area when they are appropriate and ready for publication.

These different content areas can be deployed with common or separated metadata repositories. Semantic metadata facilitates content annotation, search, browsing and management. Moreover, all the complexity of RDF metadata and ontologies keeps hidden behind the Rhizomer interface and metadata management components as it has been shown in previous sections.

Metadata edition is also subject to controls similar to those on content. The metadata repositories share the access rights and user roles with content areas. More fine grained approaches based on Public Key Infrastructures (PKI) are planned and sketched in the future work section.

5. CONCLUSION AND FUTURE WORK

As it has been shown, our proposal uses a common HTML interface for both the content and metadata parts, which can be then fragmented through access control mechanisms and different user roles into different semantics-enabled intrawebs.

This approach is common in Semantic Web browsers but we enhance it by making it more semantically coherent. The navigation is performed by metadata chunks that are fully contextualised in the frame of at least one resource that identifies it.

Therefore, no additional browsing steps are needed in order to get to the metadata associated to anonymous resources and no metadata is presented without an identification context, as it is the common case when showing the metadata associated to anonymous resources.

In addition, in order to the improved browsing experience, the user has a more interactive experience thanks to an additional set of features. Users can create, edit and remove semantic metadata as it is browsed. This additional functionality is also available through a common web interface based on HTML forms.

The same applies to content, which is also directly editable through the wiki. Therefore, the whole interaction is browser based and the user does not have to install anything else, not even plug-ins.

The intention is to improve the user experience so all this is currently being tested with real users in the context of the GRIHO[6] usability and accessibility laboratory, the UsabiliLAB. Actually, many of the design decisions during the Rhizomer development have been based on user test results. A detailed account and study of the outcomes is also future work.

One possible inconvenience of Rhizomer is that it employs semantic metadata and it is not common in corporate networks. It is more usual to find metadata schemes based on XML.

However, it is also possible to achieve this level of metadata integration when starting from XML oriented companies using the XML Semantics Reuse methodology [10,11,12] and the XML Schema to OWL and XML to RDF tools implemented in the ReDeFer project[7], which is also part of the Rhizomik initiative.

To conclude, Rhizomer is currently under development. First of all, many bugs are being detected and new functionalities are being added to both the metadata and content parts.

For instance, we are currently incorporating the assisted addition of properties and values depending on the type of the described resource. When the user edits a chunk of metadata and tries to add a new property to describe it, a popup window presents a list of all the available properties, as it is shown in Figure 8.
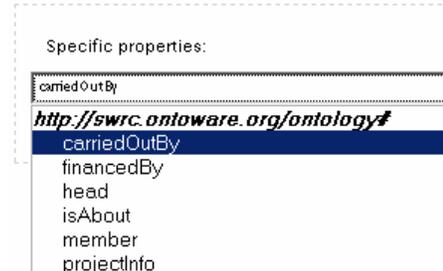

Figure 8. Assisted metadata creation popup

First, there are all the properties that are specific to the resource at hand, i.e. they are restricted to the resource type or their domain is one of the resource types. Second, all the properties that are generic, i.e. they have no domain defined or it is just restricted to any resource.

The same kind of assisted metadata creation is planned for property values. Altogether may facilitate user interaction with the underlying ontologies that structure the conceptual framework where semantic metadata is generated. In other words, the user will generate metadata following the restrictions defined by the corresponding ontologies without being aware of their existence, at least if the user does not want to.

To conclude, and as it has been introduced previously, the Rhizomer component provides a good base for implementing metadata provenance control and trust mechanism based on public key infrastructure (PKI). This will allow more fine grained mechanisms than those provided by the current servlet container authentication mechanism.

As it has been pointed out, the metadata chunks defined by Rhizomer are similar to the RDF Neighbourhoods defined in the DBin project [8]. These neighbourhoods are used to produce digital signatures of pieces of a RDF graph.

Consequently, the same approach can be employed in Rhizomer. The metadata chunks, that are used to provide the metadata navigation and edition steps, can also be used as the substrates for generating digital signatures.

When the user generates or edits a chunk of metadata, the RDF is canonicalised [13] and the digital signature is generated. For instance, this can be done using the Named Graphs API for Jena [14].

The signature is attached to the identified resource in the metadata chunk, i.e. the unique triple subject that is not anonymous. This provides a unique way to segment the whole metadata graph and a specific point for each chunk where the signature can be attached, as it is shown in Figure 9.
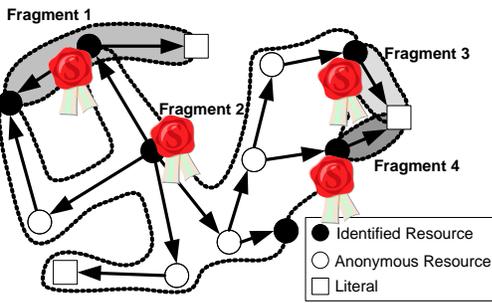
Figure 9. Rhizomer fragments

On the other hand, when the metadata graph is browsed, each metadata chunk is browsed together with its corresponding digital signature, which can be reproduced and compared in order to verify the authenticity and integrity of the metadata chunk and ensure non-repudiation of the corresponding author.

Verified signatures provide the means to build trust on the metadata. This can be done directly from the author's signature or, additionally, it is possible to rely on third parties signatures that certify metadata validity and build more complex trust networks.

In order to provide some common means to connect the metadata and the signature, the XML Signature standard has been reused. The XML Semantics Reuse methodology [10] has been employed to generate the corresponding ontologies, the XML Digital Signature and XML Encryption ontologies[8].

The classes and properties from these ontologies are then used to integrate the digital signature in the whole RDF metadata graph.

To conclude, it is important to say that the semantic intraweb provided by Rhizomer and the wiki engine is successfully satisfying the requirements of the Rhizomik research community.

Moreover, new semantic portals in the journalism and media domains are being developed. Both of them have private areas in order to build semantic intrawebs for the involved communities, in this case the stakeholders, employees and collaborators of a digital newspaper and of a music recommendation community.

6. REFERENCES

[1] Leuf, B. and Cunningham, W.: The Wiki Way. Addison-Wesley Longman (2001)

[2] Deleuze, G. & Guattari, F.: A thousand plateaus: capitalism and schizophrenia. University of Minnesota Press (1987)

[3] Huynh, D., Mazzocchi, S. and Karger, D.: Piggy Bank: Experience the Semantic Web Inside Your Web Browser. The Semantic Web – ISWC 2005. LNCS 3729, Springer (2005)

[4] Steer, D. Brown Sauce: An introduction. Technical Report HPL-2003-10, Hewlett-Packard Laboratories, Bristol, England (2003)

[5] Hartmann J. and Sure Y.: An Infrastructure for Scalable, Reliable Semantic Portals. IEEE Intelligent Systems, 19(3) (2004) 58-65

[6] Quan, D. and Karger, K.: How to make a semantic web browser. In Proceedings of the 13th International WWW Conference (2004)

[7] Sayers, C.: Node-centric RDF Graph Visualization. Technical Report HPL-2004-60, Hewlett-Packard Laboratories, Bristol, England (2004)

[8] Tummarello, G., Morbidoni, C., Puliti, P. and Piazza, F.: The DBin Semantic Web platform: an overview. Workshop on The Semantic Computing Initiative (2005)

[9] Broekstra, J. Sesame RQL: a Tutorial. http://www.openrdf.org/doc/rql-tutorial.html

[10] García, R.: XML Semantics Reuse. Chapter 7, A Semantic Web Approach to Digital Rights Management, PhD Thesis (2006) http://rhizomik.net/~roberto/thesis

[11] García, R. & Celma, O.: "Semantic Integration and Retrieval of Multimedia Metadata". Presented at the SemAnnot Workshop (2005)

[12] Gil, R.; García, R. & Jaime Delgado: "An interoperable framework for IPR using web ontologies". LOAIT'05. IAAIL Workshop Series, Wolf Legal Pub. (2005) 135-148

[13] Carroll, J.: Signing RDF Graphs. HPL-2003-142, HP Tech Report (2003)

[14] Carroll, J., Bizer, C., Hayes, P. and Stickler, P.: Named Graphs. Journal of Web Semantics, (3)4 (2005) 247-267

---

[8] http://rhizomik.net/ontologies/2004/08/xmldsig-core.owl and http://rhizomik.net/ontologies/2004/08/xenc.owl