

Semantics-Supported Cooperative Learning for Enhanced Awareness

César A. Collazos

IDIS Research Group, FIET, Universidad del Cauca
Sector Tulcan, Of. 401, Popayán, Colombia
E-mail: collazo@unicauca.edu.co

Roberto García

GRIHO HCI Research Group, Universitat de Lleida
Jaume II, 69, E-25001 Lleida, Spain
E-mail: rgarcia@diei.udl.es

Abstract: Awareness is required for supporting all forms of cooperation. In Computer Supported Collaborative Learning (CSCL), awareness can be used for enhancing collaborative opportunities across physical distances and in computer-mediated environments. Shared Knowledge Awareness (SKA) intends to increase the perception about the shared knowledge students have in a collaborative learning scenario and also concerns the understanding that this group has about it. However, it is very difficult to produce accurate awareness indicators based on informal message exchange among participants. Therefore, we propose a semantic system for cooperation that makes use of formal methods for knowledge representation based on Semantic Web technologies. From these semantics-enhanced repository and messages, it could be easier to compute more accurate awareness.

Keywords: computer supported collaborative learning; semantic web; awareness; knowledge management.

1 INTRODUCTION

Awareness is required for supporting all forms of cooperation: it is needed to coordinate and fine-tune cooperative work, to allow informal communication, and to establish conventions on the usage of shared material (Palfreyman 1996). Despite all these facts, awareness support is often neglected in CSCW, CSCL, WWW systems, or it is treated in an ad-hoc manner (Gutwin, 2002). The increasing interest in topics related to awareness has made of this one an important element, and that it has been identified like crucial aspect in the success of collaborative schemes.

In Computer Supported Collaborative Learning (CSCL), awareness can be used for enhancing collaborative opportunities reducing the meta-communicative efforts needed to collaborate across physical distances and in computer-mediated environments (Ogata, 2001).

Shared Knowledge Awareness (SKA) is a kind of awareness whose purpose is to increase the perception about the shared knowledge students have in a collaborative learning scenario. The SKA not only tries to improve and maintain the shared knowledge of a student group but also concerns the understanding that this group has about it (Collazos, 2003)

However, it is very difficult to produce accurate awareness indicators for cooperation application based on informal message exchange among participants, i.e. chat like messages, and lightweight-semantics structures for the knowledge being build. Therefore, we propose a semantic system for cooperation that makes use of formal methods for knowledge representation based on Semantic Web technologies.

The Semantic computer-supported cooperation system is based on a semantic data repository and a usable Web 2.0 interface that lets users navigate the knowledge they are sharing providing guidance in formalising the knowledge they are cooperatively building. From these semantics-enhanced repository and messages, it is easier to compute more accurate awareness mechanism as the starting point since is a formal representation instead of completely unstructured documents.

2 CSCL AND SKA

In CSCL, awareness mechanisms can be used for improving collaborative opportunities playing an important role in how the learning environment creates collaboration opportunities naturally and efficiently (Ogata, 2000). In this

scenario, Goldman have identified three types of student awareness: social, task, and conceptual (Goldman, 1992). These kinds of awareness are important for the success of efficient and effective collaboration. Moreover, authors like Gutwin (1995) also proposed workspace awareness which is the real-time knowledge a student requires about other students' interactions with the shared workspace. This kind of awareness is essential if students are to learn and work together effectively.

Collazos et al. (2002) have proposed a mechanism, named Shared Knowledge Awareness (SKA). SKA is consciousness on the shared knowledge of the students that carry out a collaborative learning activity, working in groups. This shared knowledge is more than the shared understanding of the problem. The shared knowledge is composed of the understanding of several aspects of the collaborative work, including coordination, strategy communications, monitoring, and shared comprehension of the problem.

Measuring this knowledge is a very difficult task, because it cannot be captured directly. In computer-supported scenarios there are two elements that can be analyzed to represent this knowledge. They are the actions the group members carry out during the collaborative work and the messages interchanged among them. The messages may transport the knowledge that is shared among the group members. The actions show if the transmitted knowledge was understood. Only the knowledge that is transmitted and understood could become part of the group shared knowledge. An exhaustive analysis of these two elements gives an idea about the amount of the shared knowledge. Figure 1 shows the strategy the SKA authors propose.

Basically, any collaborative application that pretends to implement SKA using this model should report all the group actions and messages to the SKA Generator Agent. It captures the shared knowledge of the messages and actions through an indicator called SKI (Shared Knowledge Indicator). Based on the value of this indicator, the agent produces awareness that can be shown in the user interface of the collaborative application. The agent uses two auxiliary modules called Actions Analyzer and Messages Analyzer to carry out these tasks. The Actions Analyzer module is in charge of studying the group actions and classifying them into one of these categories: successful, non-successful, or neutral. Then, this module generates a score based on the relatively successful actions (RSA). This score represents the shared knowledge used by the students to carry out their actions.

On the other hand, the Messages Analyzer module must study the group messages. This module uses an adaptation of the CL model proposed by Soller (2002) to do it. This adaptation consists in a redefinition of message categories, in order to provide a better classification to analyze the shared knowledge contained in the messages. The defined message categories are coordination, work, strategy, social and others. The Message Analyzer uses these categories to generate a score based on the Relative amount of Shared Knowledge in Messages (RSKM). This score represents the

shared knowledge that is transmitted in the messages. Finally, with both scores (RSA and RSKM) it is possible to calculate the value of the SKI (Shared Knowledge Indicator). The authors considered $SKI = \text{Sqrt}(RSA * RSKM)$. The value of SKI is provided as graphical awareness over the collaborative application user interface.



Figure 1 Graphical SKA

If SKI is between 0 and 0.4 then the shared knowledge is considered poor. The graphical representation of SKI presented on the user interface is shown in Figure 1a. If SKI is between 0.41 and 0.6 then the shared knowledge is considered fair and the SKA shown is depicted in Figure 1b. Finally, if SKI is between 0.61 and 1 then the shared knowledge is considered high and the SKA shown is Figure 1c.

The problems presented with this model are the informal nature of messages that do not guarantee a real analysis of the collaboration process.

3 RELATED WORK

Questions in Table 1, organized into the categories described previously, are examples of what students consider during the collaborative activity in order to be aware of what is happening in the group as they work on their task.

Table 1 Awareness in CSCL

Awareness	Questions
Social	What should I expect from other members of this group?
	How will I interact with this group?
	What role will I take in this group?
	What roles will the other members of the group take?
Task	What do I know about this topic and the structure of the task?
	What do others know about this topic and task?
	What steps must we take to complete the task?
	How will the outcome be evaluated?
	What tools/materials are needed to complete the task?
	How much time is required? How much time is available?
Concept	How does this task fit into what I already know about the concept?
	What else do I need to find out about this topic?
	Do I need to revise any of my current ideas in light of this new information?
	Can I create a hypothesis from my current knowledge to predict the task outcome?
Workspace	What are the other members of the group doing to complete the task?
	Where are they?
	What are they doing?
	What have they already done?
	What will they do next?
	How can I help other students to complete the project?

These questions are based in a classification given by Gutwin (1995). They represent the kind of information that each type of awareness should give to the students during the collaborative activity.

On the other hand, Yamagami & Seki have proposed knowledge awareness that gives feedback to the cooperative activities with an increased emphasis on sharing know-how of an organization (Yamakami, 1993). Ogata & Yano have proposed another kind of knowledge awareness. It represents a new concept for helping a learner to find an appropriate collaborator and inducing just-in-time collaboration (Ogata, 1996). Notice, however, these proposals do not refer to awareness on the construction of knowledge.

4 SKA FROM SEMANTIC WEB KNOWLEDGE REPRESENTATION

In order to facilitate the computation of the SKA, our proposal is to use formal methods for shared knowledge representation that facilitate computing the measure of SKA.

Formal methods for knowledge representation based on ontologies. An ontology is a formal, explicit specification of a shared conceptualisation. Formal means that it is an abstract model of a portion of the worlds. It is a explicit specification because it is machine-readable and understandable. Shared implies that it is based on a consensus and it constitutes a conceptualisation expressed in terms of concepts, properties, attributes, etc.

Semantic Web ontologies are based on the OWL standard, which provides a set of primitives that make possible to build a conceptualisation that can be shared in the Web.

The ontologies are complemented with Semantic Web data, i.e. expressions based on the triple pattern <subject, predicate, object>. For instance, that a given content has a defined title. These expressions are also formal because their components, the terms used in order to build the triples, are defined in the ontologies. An example of such an expression and the definitions of the used terms, with the corresponding division about data level expressions and the ontology level, are shown in Figure 2.

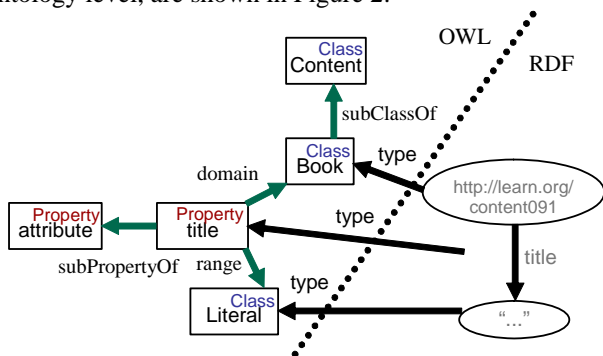


Figure 2 Semantic expression and ontology terms definitions

A set of expressions, i.e. triples, build up a graph where subject and objects correspond to nodes and properties to

the edges among them. The same applies to ontologies and the connections among data and ontologies. Consequently, the overall result is a graph.

Each user, in a Semantic CSCL environment, uses his/her own ontologies in order to convey the intended meaning of the messages she generates. Therefore, users can both input expressions and ontologies into the system.

However, this expressions and ontologies are initially isolated and correspond to graph models heavily connected for each user but potentially without any connection among users. This corresponds to a situation where the level of shared knowledge is low and that can be easily computed using graph theory methodologies (Albert and Barabási, 1992).

Our choice is the clustering factor measure, which measures the probability that two neighbours of a given node are also neighbours of one another. This is equivalent to a measure of the “island-ness” of a graph, i.e. it is possible to clearly determine isolated areas where connections are dense while connections to other areas are low.

When the clustering factor is high, the underlying graph shows isolated areas with a high level of internal connection. On the other hand, if the clustering factor is low, it is difficult to identify isolated areas.

Consequently, we interpret a shared knowledge model with a low clustering factor as a situation where a high level of shared knowledge has been achieved. On the other hand, a high clustering factor corresponds to a situation where the level of shared knowledge is low.

For instance, Figure 3 shows the graph model for a semantic CSCL scenario. On the left and the right, showing a great level of interconnections, there are the knowledge models for the two involved users. On the centre, there is a third model, an ontology that the users have used in order to connect their models and build a shared knowledge space.

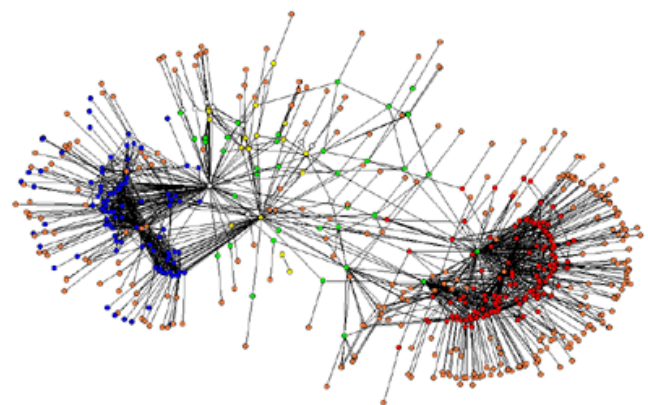


Figure 3 Graphical representation of a shared knowledge graph

The clustering factor corresponding to this situation is 0.0354. The starting situation, when users have input their own knowledge models into the systems but have not provided any assertions for integrating their models, has a clustering factor of 0.0618.

In order to produce shared knowledge, and low the clustering factor, users should make assertions about the

connections among their different conceptual models, i.e. ontologies, and produce the required formal relations about concept equivalences and subsumption that will produce the integration of conceptual models and the a shared knowledge space.

Moreover, as the knowledge model is based on formal methods and is enriched with inference mechanisms, it is possible that just one user-made connection can trigger a chain of inferences that assert more connections and provoke a great change in the clustering factor measure.

However, to make this kind of formal assertions is not an easy task for average users. In order to build a usable system that follows a semantic-approach to cooperative learning, we propose a Web 2.0 based implementation as it is detailed in the next section.

5 IMPLEMENTATION

One of the greatest problems of a Semantic Web-oriented CSCL system is users are not used to the semantic annotations. All the messages that users input into the system are highly structured messages which must follow a formal model, i.e. an ontology.

However, common users are not Semantic Web and Knowledge representation experts, even experienced users might be unfamiliar with the knowledge models they must interact with in the Semantic CSCL system.

In order to help users, a Web interface for semantic metadata browsing and edition has been implemented. It is called Rhizomer (<http://rhizomik.net/rhizomer>). Usability guides how end-user interaction with the Semantic Web is faced through 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.

This area is being extensively explored in the Semantic Web area. Many approaches are based on a graph paradigm, i.e. the user interacts with a nodes and links graphical representation. As it has been pointed out, this is not the best choice as it is not natural to force the user to interact with semantic data through the same model that is used for its representation (Karger and schraefel, 2006).

The graph model might be useful for the user in some very specific scenario, e.g. in order to get a quick view of how some data is distributed, but even this is not usually the case for Semantic Web data.

Consequently, the latest developments for semantic metadata browsing are based on different paradigms. Almost in all cases, they are based on a browsing paradigm that can be called the “Subject-centric Approach”, cf. section 5.1.

The “Subject-centric Approach” has some usability problems that have motivated us a slightly modified approach. Our proposal is detailed in section 5.1 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 5.1.1 and 5.1.2.

There are other browsing approaches like the table paradigm used in mSpace (schraefel et al., 2003). However, this table-based system must be configured in order to operate with a concrete set of data. Consequently, it does not constitute a generic paradigm for semantic metadata browsing.

5.1 Knowledge Browsing

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

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 $\langle \textit{subject}, \textit{predicate}, \textit{object} \rangle$, which combined builds-up graphs.

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?

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.

The simpler approach to fragment semantic web graphs is to define the fragment as the set of all triples with the same subject, i.e. the “Subject-centric Approach”. This is also semantically coherent because these triples are those describing the subject resource. Figure 4 shows a table rendering of a subject-centric approach metadata fragment.

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

Figure 4 Table view for a subject-centric approach fragment

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 loses the perspective about what is being described. Consequently, in order to get all the metadata related to the resource described in Figure 4, and additional browsing step is required to get the metadata shown in Figure 5.

(anonymous item)	
Given	Rosa
Family	Gil
Show Referrers	

Figure 5 Additional step for anonymous resource metadata

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 6 shows a simple RDF graph where four fragments can be identified.

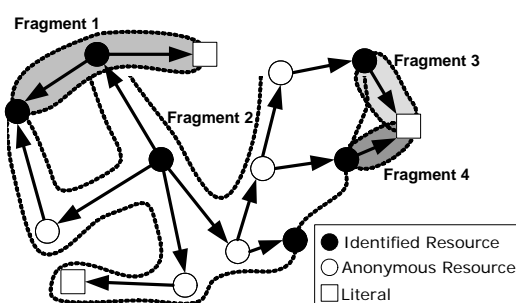


Figure 6 RDF graph and four Rhizomer fragments

As in the case of the subject-centric approach, the resources appearing as the object of the terminal triples can be browsed through new navigation steps that generate new metadata fragments describing the resources asked for

detail. This has been already shown in Figure 5, which shows a browsing step initiated from the Figure 4 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 not break down the metadata describing a resource and they are shown together with the resources that contextualise them, as it is shown in Figure 7.

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

Figure 7 Table view of a Rhizomer approach metadata fragment

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.

5.1.1 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 2.

Table 2 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
}

```

5.1.2 Multi-language Support

When a metadata 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.

5.1.3 HTML Metadata Rendering

Once the metadata fragments have been generated, they are shown to the user as an HTML rendering in the web browser. This rendering allows getting a user interface users are comfortable with. It looks like the kind of webs they are used to interact with.

In order to produce this rendering, a generic XSL transformation from RDF/XML to HTML has been developed. This approach produces consistent results as long as the metadata fragments to render are generated with the Rhizomer approach and an abbreviated RDF/XML serialisation of them is produced.

The abbreviated serialisation produces an XML stream that keeps all the related triples grouped. Consequently, it is possible to render them as a set of HTML tables, one for each resource being described, with nested tables corresponding to the descriptions of the anonymous resources they refer to.

Resource and property names are rendered as text using the appropriate label for the preferred language if available, as it has been detailed in Section 5.1.2. Literals are also rendered as text and, if different language versions are available, the preferred one is selected. Finally, there are HTML links for all resource and property names that allow browsing the metadata describing them. These links correspond to calls to a SPARQL endpoint, concretely they are DESCRIBE queries. This kind of queries is resolved in a proprietary way in order to generate Rhizomer metadata fragments and to enrich them with the RDF labels that make the multilingual rendering possible.

The resulting HTML pages look like the simple example previously shown in Figure 7. Moreover, it can be tested from the ReDeFer web page (<http://rhizomik.net/redefer>) for arbitrary metadata sets.

A related approach to render RDF as HTML is Fresnel (<http://www.w3.org/2005/04/fresnel-info>). Fresnel lenses are specifications about how to render some resources, classes and properties as HTML or other presentation languages. They allow a great level of personalisation but they require that the corresponding lens has been specified

in order to generate a rendering. However, it is also possible to create a generic lens for unforeseen kinds of metadata.

On the other hand, the XSL-based approach used in Rhizomer does not allow this level of personalisation but it is capable of dealing, in a very generic way, with any piece of metadata it encounters. In any case, if a greater level of personalisation is required, the Fresnel lenses rendering engine can be integrated in the Rhizomer platform.

Consequently, the Rhizomer rendering approach added value is that it is very easy to implement, it just requires an XSL processor so this work load can be put away from the web server and passed to the user web browser. Moreover, it can be managed with AJAX (Crane and Pascarello, 2005) so a greater level of interaction through the browser can be achieved.

5.2 Metadata Edition

The browsing capabilities already shown provide a simple Semantic Web user interface. It is simple because the interaction 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. Therefore, it is possible to generate a query form from an example piece of metadata. Table 3 shows an example of a form to build semantic queries.

Table 3 *Semantics-enabled query form*

```
<form method="QUERY" onSubmit="buildSPARQL">
  <input type="text" name=
    "http://purl.org/dc/elements/..../title"/>
  <select name="http://purl.org/dc/..../subject">
    <option value=
      "http://www.iptc.org/subjects#01011000">
      ...
    </select>
    ...
</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 and so it does not need to be specified. In fact, the resource URI will be the response we will get from the query.

The filled form fields are interpreted as the known terms we use to perform the search. For instance, the form in Table 3 can be filled with the "Corporate news" value for the input field, i.e. Dublin Core title, and the first option of the select field, i.e. music is the subject. In the case of a query form, the fields are interpreted as the triples that build

up the query pattern of a SPARQL query (<http://www.w3.org/TR/rdf-sparql-query>). A query like the one shown in Table 4 will be generated by a Javascript from the form values as a result of an “on submit” event.

Table 4 SPARQL semantic query for Table 3 form

```
DESCRIBE ?s
WHERE { ?s http://purl.org/dc/.../title ?y0;
        http://purl.org/dc/.../subject ?y1.
FILTER regex(?y0, "Corporate news") &&
        ?y1=<http://www.iptc.org/subjects#01011000> }
```

If the objective is to edit metadata, then the form fields are interpreted as the building blocks for a set of triples for the new metadata. In other words, edit forms are 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”. Moreover, 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 5 shows an example of a metadata edition form.

Table 5 Semantics-enabled edit form

```
<form method="POST" onSubmit="buildTriples"... >
<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"/>
...
</form>
```

If the fields in the Table 5 form are filled with the appropriate values, e.g. “<http://rhizomik.net/~rosa>” for the non-hidden “rdf:ID” field, when it is submitted, the metadata shown in Table 6 is generated applying a direct conversion from form fields to triples.

Table 6 Metadata from Table 5 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.

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 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 8 shows a form generated automatically from the RDF metadata shown in Figure 7 in order to edit it.

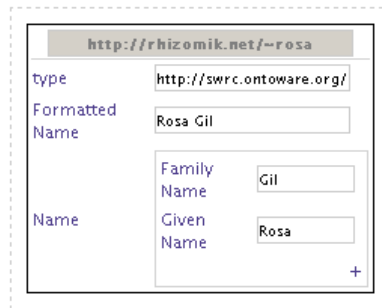


Figure 8 Automatically generated metadata edition form

From this form, the user benefits from assisted addition of properties and values depending on the type of the described resource. When the user edits a metadata fragment 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 9. The available properties are deduced from the restrictions that the conceptual models defined by the used ontologies impose.

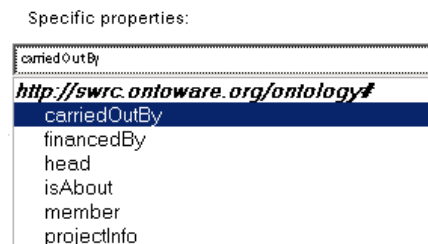


Figure 9 Assisted metadata creation popup

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 available 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.

All the required calls for assisted metadata creation are done in the background using AJAX in order to make this feature totally transparent to the user.

6 CONCLUSIONS AND FUTURE WORK

As it has been shown, the proposed semantic approach to CSCL can produce more accurate SKA measures based on well-established graph theory statistical methods. However, this requires that users follow very structure patterns for knowledge elicitation, based on the ontologies that shape the semantic metadata at hand.

Average users are not Semantic Web and Knowledge representation experts, even experienced users might be unfamiliar with the knowledge models they must interact with in the Semantic CSCL system. Therefore, there is the requirement for an usable interface for semantic metadata browsing and edition.

Rhizomer is the proposed implementation for a Semantic CSCL system and uses a common HTML interface. This approach is usual in Semantic Web browsers but it is enhanced by making it more semantically coherent. The navigation is performed by metadata fragments 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. Forms are enriched with AJAX in order to provide an assisted metadata creation mechanism that operates in the background and remains transparent to the user.

The intention now is to improve the user experience so all this is currently being tested with real users in the context of a usability and accessibility laboratory. Actually, many of the design decisions during the Rhizomer development have been based on user test results.

For instance, all links in the HTML rendering that allow browsing new metadata fragments are not underlined. Just links that point to new HTML content are underlined. This approach helps user differentiate among metadata and content browsing links. As they are used to underlined links while navigating through html documents, the same behaviour is maintained in the HTML rendering.

Additionally, accessibility is being tested with screen readers. The results are quite user friendly because the

generated HTML content is based on the available RDF labels and preferred languages and this makes it easier for text-to-speech applications.

Finally, the SKA measure mechanism based on calculating the clustering factor for the graph model is being incorporated into Rhizomer in order to make it a ready to use Semantic CSCL system.

7 REFERENCES

- Albert, R., Barabási, A.L. (2002) 'Statistical mechanics of complex networks', *Reviews of Modern Physics*, Vol. 74, pp. 47-97
- Collazos, C., Guerrero, L., Pino, J., and Ochoa, S. (2002) 'Introducing Knowledge-Shared Awareness', *Proceedings Of IASTED International Conference: Information and Knowledge Sharing*, St. Thomas, USA, pp.13-18.
- Collazos, C., Guerrero, L., Pino, J. (2003) 'Knowledge Construction Awareness', *Student-Centered Learning Journal*, Vol. 1, No. 2, pp.76-86.
- Crane, D. and Pascarello, E. (2005) 'Ajax in Action', Manning, Greenwich, USA.
- Goldman, S.V. (1992) 'Computer resources for supporting student conversations about science concepts', *SIGCUE Outlook*, Vol. 21, No. 3, pp. 4-7.
- Gutwin, C., Stark, G., Greenberg, S. (1995) 'Support for workspace awareness in educational groupware'. *Proceedings of the CSCW'95*, Lawrence Erlbaum Associates, pp. 147-156.
- Gutwin, C. and Greenberg, S. (2002) 'A Framework of Awareness for Small Groups in Shared-Workspace Groupware', *Computer-Supported Cooperative Work*, Issue 3-4, pp. 411-446.
- Karger, D. and schraefel, m.c. (2006) 'The Pathetic Fallacy of RDF', *Proceedings of 3rd International Semantic Web User Interaction Workshop*, Athens, Georgia, USA.
- Ogata, H., Matsuura, K., Yano, Y. (1996) 'Knowledge awareness: Bridging between shared knowledge space and collaboration in Sharlok', *Proceedings of Educational Telecommunications'96*, pp. 232-237.
- Ogata H, Matsuura, K., Yano Y. (2000) 'Combining knowledge awareness and information filtering in an open-ended collaborative learning environment', *Int. Journal of Artificial Intelligence in Education*, Vol. 11, pp. 33-46.
- Ogata, H., Yano, Y., Furugori, N., Jin, Q. (2001) 'Computer Supported Social Networking For Augmenting Cooperation', *CSCW*, Kluwer, Vol. 10, No. 2, pp. 189-209.
- Palfreyman K. A. and Rodden T (1996) 'A protocol for user awareness on the World Wide Web', *Proceedings of CSCW'96*, ACM Press, pp. 130 - 139.
- schraefel, m.c., Karam, M. and Zhao, S. (2003) 'mSpace: interaction design for user-determined, adaptable domain exploration in hypermedia', *Proceedings of Workshop on Adaptive Hypermedia and Adaptive Web Based Systems*, Nottingham, UK.
- Soller, A. L. (2002) 'A machine learning approach to assessing knowledge sharing during collaborative learning activities', in G. Stahl (ed.), *Proceedings of Computer Supported Collaborative Learning*.
- Yamakami, T., Seki, Y. (1993) 'Knowledge Awareness in Asynchronous Information Sharing', *Proceedings of the IFIP TC8/WG8.4 Working Conference on The Open Systems Future*, pp. 215-225.