Value-added Metatagging: Ontology and Rule based Methods for Smarter Metadata

Marek Hatala and Griff Richards

Simon Fraser University Surrey
2400 Central City
Surrey, BC, Canada, V3T 2W1
{mhatala|griff}@sfu.ca

Abstract. In this paper we describe an ontology and rule based system that significantly increases the productivity of those who create metadata, and the quality of the metadata they produce. The system suggests values for metadata elements using a combination of four methods: inheritance, aggregation, content based similarity and ontology-based similarity. Instead of aiming for automated metadata generation we have developed a mechanism for suggesting the most relevant values for a particular metadata field. In addition to generating metadata from standard sources such as object content and user profiles, the system benefits from considering metadata record assemblies, metadata repositories, explicit domain ontologies and inference rules as prime sources for metadata generations. In this paper we first introduce the basic features of metadata systems and provide a typology of metadata records and metadata elements. Next we analyze the source of suggested values for metadata elements and discuss four methods of metadata generation. We discuss how the operations on objects the metadata are describing affect suggested metadata values and we present decision tables for the metadata generation scheduling algorithm. Finally, we discuss the use of our system in tools developed for creating e-learning material conformant with the SCORM reference model and the IEEE LTSC LOM standard.

1 Introduction

E-learning is a major industry with growing applications in many sectors, including industrial training, higher education, the military, individual training etceteras. This wide range of applications has given rise to a whole ecology of specialized systems for managing learning and learning resources [18]. Interoperability between systems and between content and systems has become a major issue and resulted in several standardization efforts. In e-learning, the first result has been a standard for learning object metadata (LOM) developed by the IEEE [12] and standardization of other aspects of e-learning, including content packaging, messaging between content and systems, or sequencing of learning resources. The Shareable Content Object Reference Model (SCORM) from the Advanced Distributed Learning Network (ADL) [19] pulls together these standards and specifications and defines how they work together. Standardized metadata descriptions enable content providers to describe different aspects of the learning resources and promote both interoperability and sharing of resources.

The IEEE LTSC LOM is intended to be comprehensive and to cover most situations in which metadata are applied to learning resources. In actual implementations it is often useful to enrich the standard by applying taxonomies or ontologies to elements which are specific to the particular application. We refer to such implementations as application profiles [5].

One of the main obstacles to the widespread adoption of systems which make intensive use of metadata is the time and effort required to apply metadata to multiple resources and the inconsistencies and idiosyncrasies in interpretation that arise when this is a purely human activity. A typical three-hour course may be composed of several dozen content objects each of which may have several included media elements which can generate anywhere from 1,000 to 5,000 separate metadata values for a three hour course [15]. Some of these metadata values can change when the structure of the course changes, making upkeep an expensive proposition.

This problem is not limited to learning resources. Learning resources are increasingly being organized as learning objects [23] and in turn learning objects can be a part of a transformational set of content objects which includes knowledge objects (for use in knowledge management systems), performance objects (for use in performance support systems), collaboration objects (for use in collaboration systems) and so on. Learning objects, and content objects generally, tend to proliferate rapidly once introduced, and the amount of metadata that needs to be applied grows exponentially.

Recent studies by Greenberg et al [7] show increased awareness of the importance of creating metadata for objects even in industrial settings, despite worries about the quality of the metadata [21]. In a follow on study [8] Greenberg et al looked at collaborative metadata generation and found that of all the metadata fields, authors most frequently sought expert help when generating "subject" metadata. The study was conducted for a simple 12-element set from Dublin Core [4]. As the number of elements and requirements for richly structured metadata sets increase, the need for systems that support metadata generation and management will also grow.

There is an important relationship between ‘subject’ metadata, ontology construction and the goals of the semantic web. The semantic web initiative [2] is working towards a vision of having data on the web defined and linked in a way that can be used by machines for various applications and not just display purposes as is generally the case today [11]. Ontologies are a basic component of the semantic web. There is a direct connection between semantic web and metadata systems. The values for the metadata elements can be successfully represented using formal ontologies supporting computational reasoning [22].

Several systems for ontological metadata have been proposed. Weinstein and Birmingham [22] have proposed an organization of digital library content and services within formal ontologies. They focused on created ontological content metadata from existing metadata. In Jenkins et al. [13] an automatic classifier that classifies HTML documents according to Dewey Decimal Classification is described. An ontology-based metadata generation system for web-based information systems is described in Stuckenschmidt et al. [20]. The latter approach relies on the existence of a single ontology to which all the web pages can be related.

In this paper we present a system that helps metadata creators generate high quality metadata by suggesting values for the metadata fields. The main strength of the system is its ability to specify the rules for a particular metadata schema and application domain ontologies based on the sound analysis of the domain. In addition to generating metadata from standard sources such as object content and user profiles, the system benefits from considering metadata record assemblies, metadata repositories, explicit domain ontologies and inference rules as prime sources for metadata generation.

The paper is organized as follows. In Section 2 we introduce the basic features of metadata systems and provide a typology of metadata records and metadata elements. In Section 3 we analyze the source of suggested values for metadata elements. Section 4 presents four methods of metadata generation and classifies operations on objects affecting the suggested metadata values. In Section 5 we discuss the implementation of the system using inference engine and discuss different types of rules based on the four methods presented in the previous section. We conclude with a discussion of our approach and provide direction for further projects.

2 Metadata Systems

Interest in metadata stems from a belief that these problems can be solved by focusing on the actual meaning of the words in the web document and providing a textual meaning for non-text-based documents. In this sense, metadata function in a manner similar to a card record in a library providing a controlled and structured description of resources through a searchable ‘access points’ such as title, author, date, location, description, etc. [6]. There are also compelling arguments against metadata which are based mainly on the nature of human behavior [3]. Interestingly enough, both metadata proponents and opponents agree that it is the *quality of metadata* that is essential for high quality retrieval.

The quality of the metadata is the underlying factor that affects the overall performance of the system. Standardization processes try to increase quality by providing sound technical solutions and best practices. However, it is the human factor in the process of metadata creation that affects a metadata quality and usefulness the most. Friesen et al. [6] make an excellent point that the metadata approach effectively inserts a layer of human intervention into the web-based search and retrieval process. Documents in this new approach are not determined as relevant to a specific subject or category as a direct result of their contents, but because of how a metadata creator or indexer has understood their relevance. By focusing on the metadata creation process and facilitating the metadata creator we aim to help overcome some of the well-known problems [3].

Metadata Standards and Record Types. Metadata can fulfill its purpose only when it complies with some standard that a group of people agreed on. Metadata standards can be generic and support broad range of purposes and business models, for example Dublin Core [4], or they can be specific for a particular community, for example IEEE LTSC Learning Object Metadata (IEEE LOM) [12]. A metadata standard provides a set of elements, defines their meaning, and provides guidelines and constraints on how to fill element values. An *individual* metadata record complying with the standard represents one object or document.

This approach can be extended to consider an *assembly* of documents or objects. For example, the SCORM initiative from the Advanced Distributed Learning Network[1], which uses the IEEE LOM, describes a reference model for describing a collection of learning resources, their aggregations and their underlying assets.

Finally, a metadata *repository* is a collection of many metadata records, typically complying with the same standard, or with a way to map between standards. In addition to effective storage, a repository provides search and metadata creation tools and capabilities.

Typology of Metadata Elements. The elements of metadata schemas[2] can hold different types of values. The simplest element type is a *free text* field, for example a 'title' element. Typically, the standard provides a set of guidelines for what the values should represent. A second common element type has an associated *vocabulary* of possible values that metadata creator has to choose from. For example, the IEEE LOM element 'status' can have one of the four values {draft, final, revised, unavailable}. A third type of element uses an *external taxonomy* or classification schema. For example, Dublin Core enables the user to specify a classification schema for its 'subject' element, such as 'Library of Congress' and provide a term from the library's classification taxonomy. Finally, the element values can refer to concepts and objects in *ontologies* as they are defined in the semantic web initiative [2].

3 Source of Suggested Values

The number of elements can vary greatly between metadata schemas, for example, Dublin Core defines 15 elements, and IEEE LOM defines over 80 elements. There are two main obstacles in creating high quality metadata records: 1. the amount of work and time required for applying the metadata, and 2. the expertise required for this task – especially since in many situations, the metadata creator is either a subject-matter expert, a learning object designer/programmer, or a clerical member of a production team. With experience, creators can usually master the guidelines for the metadata standard in use, however references to external taxonomies and ontologies are more problematic as they are often beyond the creator's expertise [7]. Not only must the creator be familiar with the content of a particular ontology, s/he has to be able to navigate to the appropriate concepts quickly. It is therefore of great help to the creators if the system can suggest the most probable values to select from for as many elements as possible. This can improve both the speed with which metadata can be applied and the quality and consistency of the metadata itself. By constraining the

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

[2] As not all element sets are standardized, metadata schema provides for a more generic name for an element set.

values of the metadata elements using an ontology and by providing a system that can propose values, the individual biases described above can be moderated.

3.1 Sources Based on Record Types

The suggested values for metadata records can be computed from different sources. A different set of sources can be used for different record types.

Individual records. Individual metadata records (related or not related to other records) can have three main sources of suggested values for metadata elements:

- the user's profile,
- the application profile, and
- the object itself.

The metadata creation tool can enable the *user* to create a profile keeping information such as the user's name, affiliation, email, etc. This information can be used to automatically pre-fill the values for some metadata elements, for example element 'creator' in the Dublin Core standard.

Similarly, the *application* using the metadata is typically designed with some specific purpose, and in many cases several records are sequentially created for the same application. The application specific information can be preset in the application profile and used as suggested values for some elements. For example, the 'TypicalAgeRange' element in the IEEE LOM can be preset to value '18-24' for all the metadata if a creation tool is used for undergraduate post-secondary education.

The above sources are relatively static with regard to the individual metadata record. However, a quite significant amount of the metadata can be harvested from the *object or document* itself. The information retrieved directly from the object for which the metadata record is created can be size (in bytes), MIME type (e.g. text/html), location (e.g. URL), title (e.g. from the <TITLE> tag), etc.

Assemblies. Objects which are part of an *assembly* create together a whole and therefore it is possible that they share several element values. Although the objects in the assembly and their metadata records are distinct, a value set for one metadata element in one objects can propagate itself as a suggested value to other objects in the assembly. If the assembly is organized hierarchically some of the values can be inherited form the ancestor nodes or aggregated from the child nodes. For example, technical requirements to execute an assembly of objects are a union of requirements to execute each object which is a member of the assembly. A SCORM package is a good example of an assembly representing an e-learning course consisting of several learning objects.

Repositories Metadata records have to be managed just as any other data type. The *repository* typically contains many records which are available as a source of suggested values for some elements. The basic idea uses a notion of 'similar objects' (we will define similarity later). If we can find an object similar to the object for which we are creating metadata we can assume that they will have the same or similar values for at least some of the metadata elements.

Two basic type of similarity can be used. First, two objects can be similar in their content. To be able to compute this similarity, some method of measuring the content similarity has to be used. Currently, the methods for the text-based documents are quite well documented [1][10][14], however methods for other media types are not generally available.

The second notion is similarity defined by some set of rules. For example, let’s assume there is a dependence between the values in the element describing an ‘economic sector’ in which a resource is used and ‘prerequisite knowledge’ for learning resource. A simple rule could capture following heuristic: if a user specifies a value for ‘economic sector’ in a new record then find other records with the same or ‘similar’[3] values for this element and use values from found records from their ‘prerequisite knowledge’ element as suggested values for this element in the new record.

3.2 Sources Based on Types of Elements

With respect to the element typology presented in the previous section, values can be suggested for each type of element: free text elements, vocabulary elements, and external taxonomy/ontology elements. For example, the value for the ‘title’ element (free text) can be suggested from the object itself, the value for the ‘TypicalEndUser’ element can be pre-selected from the defined vocabulary based on the application or user profile, and the value for the ‘EconomicSector’ element can be suggested from the ontology as a result of an inference using all records in the repository.

It is also important to note that for some elements it is very problematic if not impossible to suggest a value. This is especially true when completely new content is being brought into a system.

4 Generation of Suggested Values

In the previous section we identified several sources which can be used to generate a set of suggested values (*SSV*) for metadata elements. This paper concentrates on generating suggested values for objects in assemblies and repositories, i.e. using values in metadata records for ‘similar’ objects. Although our implemented system makes use of all the techniques mentioned in the previous section we trust an interested reader can consult the available comprehensive literature and tools on generation of metadata from documents and profiles (e.g. available from http://dublincore.org).

By *assembly* we mean a collection of relatively independent units, each having its own metadata record. The assembly may have an associated metadata record of its

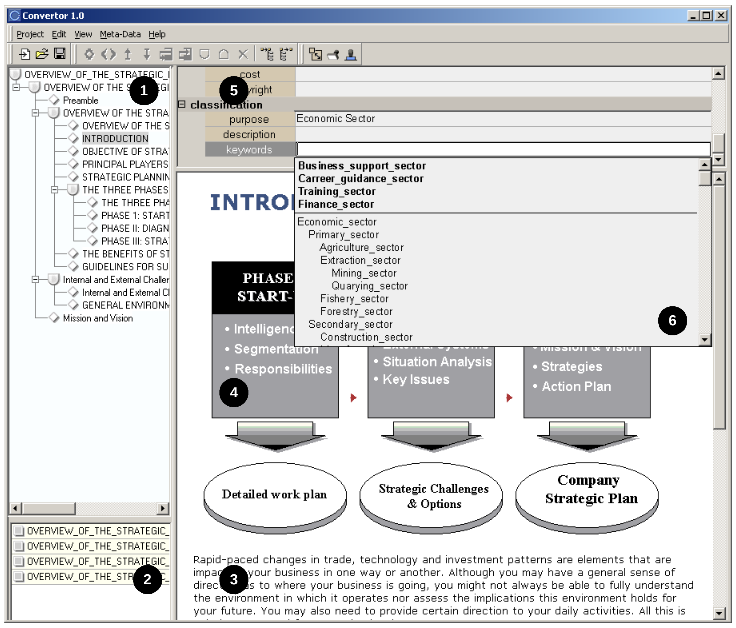
[3] In our example similarity for sectors can be defined either explicitly through related_to relation or implicitly for sectors within the subtree of primary sector in the sector ontology.

own. These units are organized into some sort of structure, for example a hierarchical structure or a structure based on IMS Simple Sequencing. For example, an e-learning course can consist from several lessons which in turn can consist of several units. In the e-learning domain, the IMS Packaging schema used by SCORM makes it possible to define an organization of units (called Shareable Content Objects or SCOs) into a hierarchy of aggregations. Fig.1 displays a hierarchical organization of an e-learning course within an application for managing the structure and its metadata.

We define an *aggregation* as a content object which contains other resources called assets. The assets can be media resources which are reusable (i.e. it makes sense to apply metadata to them) but to demonstrate their value they have to be included into the context providing object. For example, a webpage including an animation and image can represent an aggregate which has its own metadata record. The animation and the image can each have their own metadata records as well. Fig.1 displays an aggregate with 2 objects included.

Inheritance Method. This method is applicable to the metadata records of objects organized into assemblies and aggregations. The suggested values for an element exhibiting inheritance property are accumulated from the elements of records on the path to the top level record.

For assets in the aggregations, a set of suggested values generated through the inheritance is a union of SSV for their aggregate and metadata values of the particular element for the aggregate object.



OVERVIEW_OF_THE_STRATEGIC_I
OVERVIEW OF THE S GI
Preamble
OVERVIEW OF THE STRA
OVERVIEW OF THE S
INTRODUCTION
OBJECTIVE OF STRA
PRINCIPAL PLAYERS
STRATEGIC PLANNIN
THE THREE PHASES
THE THREE PHA
PHASE 1: START
PHASE II: DIAGN
PHASE III: STRA
THE BENEFITS OF ST
GUIDELINES FOR SU
Internal and External Challer
Internal and External Cl
GENERAL ENVIRONM
Mission and Vision

OVERVIEW_OF_THE_STRATEGIC_
OVERVIEW_OF_THE_STRATEGIC_
OVERVIEW_OF_THE_STRATEGIC_
OVERVIEW_OF_THE_STR

purpose: Economic Sector

Business_support_sector
Career_guidance_sector
Training_sector
Finance_sector
Economic_sector
Primary_sector
Agriculture_sector
Extraction_sector
Mining_sector
Quarying_sector
Fishery_sector
Forestry_sector
Secondary_sector
Construction_sector

• Intelligenc
• Segmentation
• Responsibilities
• Situation Analysis
• Key Issues
• Strategies
• Action Plan

Rapid-paced changes in trade, technology and investment patterns are elements that are impac your business in one way or another. Although you may have a general sense of direc s to where your business is going, you might not always be able to fully understand the environment in which it operates nor assess the implications this environment holds for your future. You may also need to provide certain direction to your daily activities. All this is

Fig.1 Snapshot of the Recombo assembly and metadata creation tool containing: assembly hierarchy (1), list of assets in the selected object (2), object (3), asset included in the object representing an aggregation (4), metadata fields (5), and list of suggested values in front of the vocabulary (6).

It is interesting to note, that although both aggregates and assets can be viewed as parts of the same hierarchy in an assembly, the rules for the SSV can differ for the same element in each aggregate, SCO or asset. For example, the ‘creator’ element for the aggregates (i.e. course content) exhibits the inheritance property and we can include into *SSV* each value found in the ‘creator’ element in records on the path to the root record of the hierarchy. The situation is different for the ‘creator’ element for an asset: in addition to the values collected by inheritance from the root the second set of set of values is collected from all assets in the assembly with the ‘similar’ media type. This heuristics is based on an assumption that it is likely that a creator of a specific media type is a media developer specializing in a particular type of media object[4].

Accumulation Method. This method works uniformly through assemblies and aggregations. The suggested values for metadata elements that exhibit an accumulation property are collected from same metadata elements in records forming a subtree of the current record.

An example of an element with an accumulation property is ‘technical requirements’ element containing a system specification required to execute an object. The technical requirements to execute an object representing a sub-hierarchy of objects and assets are a union of technical requirements of each object and asset in the hierarchy.

Content Similarity Method. The content similarity method makes use of all accessible metadata records in the repository. A set of suggested values for the elements exhibiting this property is calculated as the union of element values from metadata records of objects exhibiting content similarity with the object under consideration.

The method can employ different algorithms for calculating content similarity and different algorithms can be used for calculation of the similarity for different elements. For example, to generate *SSV* for the ‘technical requirements’ element for assets the algorithm to calculate content similarity can simply compare MIME types of the objects. However, to suggest values for the metadata element ‘subject classification’ a content analysis of the textual content of the object has to be performed as a statistical text analysis.

Semantically Defined Similarity Method. Semantically defined similarity is the most powerful and complex of the methods presented. This method operates purely on metadata records in the repository and takes into consideration both metadata element values of finalized records as well as the values in the record being created. Fig.2 demonstrates the main principle. Based on already filled values in the metadata elements in the current record one or more of the inference rules are triggered. The inference rules calculate the values for a set of metadata fields which characterize similar records. The similar records are retrieved and a set of suggested values for another field(s) is generated as a union of values from the similar records.

[4] The second set of values would be generated using semantically defined similarity.

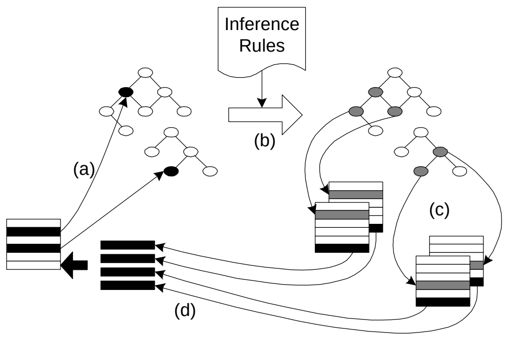


Fig.2. Generating suggested value through semantic similarity: values mapped to ontology concepts (a) are consumed by inference rules (b) which define semantically similar concepts. The records with similar concepts are found in the repository (c) and their values are used as similar values for a specific metadata field.

This method is most suitable for elements which use vocabularies derived form formal ontologies. The set of inference rules then operates on the ontologies and can use powerful inference techniques such as forward chaining or constraint satisfaction.

The customized metadata systems (metadata profiles) benefit from this method the most as standards typically provide vocabularies in only a limited form. Customized metadata systems can add elements significant for a particular domain and define domain ontologies and domain specific inference rules.

4.1 Effects of Operations on Assemblies

Creating metadata records is a highly interactive activity. Typically, users fill in metadata values in some form-type interface which can display several dozens of fields[5]. When creating an assembly, the total number of fields can easily reach hundreds or thousands. A graphical interface, as the one shown in Fig.1, allows the user to quickly reorganize the structure of an assembly and change focus from one object to another. The suggested values generation module has to be able to respond to these changes in real time. Some of the methods presented in the previous sections can be computationally expensive, such as the semantic similarity method. Therefore, it is important that the system regenerates suggested values only for the elements affected by the change in the assembly structure or changes to the values of metadata elements.

We categorize possible operations into three main categories: changes to the object content, changes to the assembly structure, and changes to the metadata values. This categorization helps us analyze an influence of operations on the set of suggested values generated by methods presented above. The operations affecting suggested values are listed in Table 1.

[5] A field is a representation of the metadata element in the form interface.

Table 1. Operations influencing suggested metadata values

Operation	**On**	**Description**
Join	OC	Join operation concatenates content of two or more leaf objects in the hierarchy by replacing both objects with the final product. For non-leaf objects the object is replaced with concatenation of all its children.
Unjoin	OC	Unjoin operation restores the state before the Join operation was performed.
Split	OC	Split operation separates the content of a leaf node into two objects and by successive splits into multiple objects.
Group	AS	Group operation introduces and extra level into the object hierarchy in the assembly. It replaces a set of nodes with one node having an original set as its children.
Ungroup	AS	Ungroup operation restores the state before the Group operation was performed.
Promote	AS	Promote operation moves a node and all its subtree to the next level in the hierarchy. The new node is positioned after the node it belonged to.
Demote	AS	The node is moved into the first preceding group at the same level. The node will become the last child in the node.
Value filled	ME	Metadata value is filled or selected from the vocabulary for the metadata element.

OC – object content, AS assembly structure, ME – metadata element

There are three distinct significant places where changes affecting the set of suggested metadata values for a particular object can occur:

- In the object itself.
- In the path from the object to the root of the hierarchy.
- In the object's subtree in the hierarchy.

We have analyzed operations with respect to already generated suggested values and implemented an appropriate scheduling algorithm for re-calculation of suggested values affected by operations. For limited space here we will report these results elsewhere.

5 System for Metadata Generation

The module for generation of suggested metadata values is a component which can be plugged into a larger system for metadata management. The main purpose of the system is to support the user in his or her task of creating metadata descriptions for sets of the objects. The system is generic and independent of the application domain and can work with various domain ontologies.

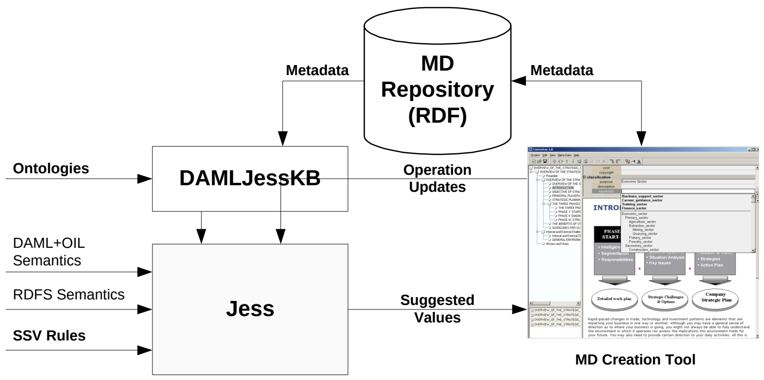


Fig.3 Architecture of the system for generation of suggested metadata values

The architecture of the system is shown in Fig.3. The generation module responds to requests generated as a result of user's actions in the metadata creation tool such as the one shown in Fig.1. In the process of generation of suggested values the generation module communicates with other components to fulfill its goals. In this paper we concentrate only on the generation component and show how metadata standards, inference rules and ontologies are used to generate suggested values.

5.1 Metadata and Ontology Representation

As shown in Fig.3 the core of the generation module is a forward chaining inference engine (Jess). Jess operates on a knowledge base which consists of facts and rules. We have represented our input data (metadata schemas, metadata records, ontologies and currents state of the problem) as facts. To transform our input data expressed in RDF and DAML+OIL to Jess facts we have used DAMLJessKB[6]. DAMLJessKB transforms RDF triples into facts with a dummy prefix PropertyValue. Below are two examples of how IMS LOM metadata RDF statements are represented as Jess facts[7].

```
(PropertyValue lom_edu:intendedenduserrole
     http://www.sfu.ca/courses/some_lo/ lom_edu:learner)

(PropertyValue lom_edu:context
     http://www.sfu.ca/courses/some_lo/ lom_edu:HigherEducation)
```

The DAML+OIL statement

```
<daml:Class rdf:about="my_ont:Accounting_sector">
   <rdfs:subClassOf>
       <daml:Class rdf:about="my_ont:Business_support_sector" />
   </rdfs:subClassOf>
</daml:Class>
```

[6] http://plan.mcs.drexel.edu/projects/legorobots/design/software/DAMLJessKB/

[7] In shown Jess representation we have replaced full URIs with qualified notation for clarity.

was converted into

```
(PropertyValue rdfs:subClassOf
     my_ont:Accounting_sector my_ont:Business_support_sector)
```

DAMLJessKB defines semantics of the RDFS and DAML+OIL as a set of rules which are loaded into the Jess engine and fired at the startup of the system. In addition to basic semantics we have also preloaded all the metadata schemas and ontologies used in the existing records in the repository. Finally, as the rules during inference process operate on metadata of existing records we needed to load metadata records into Jess. However, as not all metadata elements participate in the inference process, we have developed a filter which analyses the rules and loads only those metadata elements of existing records which are referred to by inference rules. Other record elements can be accessed by external subsystems directly from the repository as needed[8].

5.2 Inference Rules

The rules represent the knowledge in the system. The knowledge base contains several types of rules. As mentioned before, some of the rules representing semantics of RDFS and DAML+OIL are part of the DAMLJessKB. The rules we have developed generate suggested values for metadata elements and can be grouped according to the methods described in Section 4. Another set of rules is triggered by the operations described in Section 5 and maintains consistency of SSVs.

Rules for Inheritance and Accumulation Method. These simple rules collect values from one particular element from records belonging to the same hierarchical assembly as a currently created record. The rules use the information about the hierarchy to simply collect values. A rule collecting values from all objects in the path to the root of the hierarchy is shown below:

```
(defrule suggest-intededenduserrole-based-on-inheritance
  "Inherit values of intendedenduser from hierarchy above."
  (current_LO ?cur)
  (object-from-path-to-root ?cur ?obj)
  (PropertyValue lom_edu:intendedenduserrole ?obj ?val1)
=>
  (assert (SuggestedValue STRONG lom_edu:intendedenduserrole ?cur
?val1)))
```

Rules for Content Similarity Method. The content similarity is used in our system in two different ways. First, we use content similarity to compare values of some metadata elements such as ‘description’. Secondly, we apply content similarity to the

[8] For example, the ‘description’ element is accessed directly from the repository by the algorithm evaluating content similarity which is implemented outside of Jess engine and is called when user modifies the text in the element exhibiting content similarity property and facts indicating content similarity are asserted into the Jess knowledge base.

objects themselves. The algorithm to evaluate content similarity is external to the inference engine and only facts indicating content similarity and its level are used as the following rule shows:

```
(defrule suggest-subject-based-on-description-similarity
  "Object with similar descriptions can have same subject."
  (current_LO ?cur)
  (similar-content dc:description ?cur ?obj ?level)
  (PropertyValue dc:subject ?obj ?val1)
=>
  (assert (SuggestedValue ?level dc:subject ?cur ?val1)))
```

Rules for Semantic-based Similarity Method. The rules in this category implement heuristics reflecting interdependencies between values in metadata records at several levels of specificity. The most generic rules capture the dependence between metadata elements regardless of the values they have. An example is a rule suggesting value for ‘technical requirements’ based on the ‘media format’:

```
(defrule suggest-requirements-based-on-formatr
  "Objects with same format can have same technical requirements."
  (current_LO ?cur)
  (PropertyValue dcq:medium ?cur ?med)
  (PropertyValue dcq:medium ?obj ?med1)
  (test (same-medium ?med ?med1))
  ;;look for specific subclass of TechnologyRequirement
  (PropertyValue rdfs:subClassOf ?trq lom_tech:technologyrequirement)
  (PropertyValue ?trq ?obj ?val1)
=>
  (assert (SuggestedValue STRONG ?trq ?cur ?val1)))
```

A second type of the rule makes use of the taxonomical information and captures the dependence between classes of values for the particular metadata elements and other elements. The notion of similarity in these rules is based on subClassOf properties between values which are expressed as taxonomies. For example in the IEEE LOM RDF binding the specific classifications are subclasses of the lom_cls:classification. The following rule collects suggested values for specific classifications from objects created by the same content provider:

```
(defrule suggest-classification-based-on-contentprovider
  "Same person is working within some domain so he can classify
objects using the same classification as same objects within the same
package. "
  (current_LO ?cur)
  (PropertyValue lom_life:contentprovider ?cur ?val)
  (object-from-same-package ?cur ?prev)
  (PropertyValue lom_life:contentprovider ?prev ?val0)
  (test (same-entity ?val ?val0))
  ;;look for specific subproperty of classification
  (PropertyValue rdfs:subPropertyOf ?cls ims_cls:classification)
  ;;subject is used for purpose and idea
  (test (neq ?purpose dc:subject))
  (PropertyValue ?cls ?prev ?val1)
=>
  (assert (SuggestedValue STRONG ?cls ?cur ?val1)))
```

A third type of semantic similarity rules uses a generic notion of similarity. It relies on the use of ontologies as underlying conceptual structures for the values forming a metadata element vocabulary. The similarity is expressed by other set of rules and uses a full expressive power of ontologies. For example, the following rules suggest values for the classification element prerequisite based on the similarity between economic sectors. The sectors are considered to be 'similar' if they are related with each other. The relation is defined either by a property 'related_to' that is specified in the ontology directly for specific sectors or by the rule below covering a generic case where sectors are considered 'related' if they reside in the same subtree with the primary economic sector (level 2 and below):

```
(defrule related-within-main-sector
   "Industries within the main sector are related by default"
  (PropertyValue daml:subClassOf ?medSec es_ont:Economic_sector)
  (PropertyValue daml:subClassOf ?medSec2 ?medSec)
  (PropertyValue daml:subClassOf ?sec ?medSec2)
  (PropertyValue daml:subClassOf ?sec2 ?medSec2)
=>
  (assert (PropertyValue es_ont:related_to ?sec ?sec2)))

(defrule suggest-prerequisites-if-object-used-in-related-sectors
  "Objects used in related sectors can have same prerequisites."
  (current_LO ?cur)
  (PropertyValue md_sec:used_in ?cur ?sec)
  (PropertyValue md_sec:used_in ?obj ?sec1)
  (PropertyValue es_ont:related_to ?sec ?sec1)
  (PropertyValue lom_cls:prerequisite ?obj ?val1)
=>
  (assert (SuggestedValue MEDIUM lom_cls:prerequisite ?cur ?val1)))
```

It is interesting to note that the first of the above two rules depends purely on the ontology and all related sectors are inferred at startup time. The use of es_ont:related_to in the second rule is therefore straightforward pattern matching which is done very effectively in forward chaining rete algorithm.

5.3 Interaction Between Knowledge Base and External System

As the creation tool is an end-user application the user's actions change the current state of the metadata records. When the user changes a metadata value in the creation tool, this information is inserted into the knowledge base and several rules can be activated and suggest new values for some fields. As these changes are caused by operations described in Section 4.1 another set of rules monitors the user's actions and keeps a set of suggested values consistent with the current status of the metadata record.

5.4 Implementation

The generation module was implemented within the content integration framework developed by Recombo Inc. in Vancouver, British Columbia (www.recombo.com). Recombo is developing a suite of applications to support emergent XML content standards. The initial implementation is in the application domain of eLearning. One of the applications, Convertor for Word, which imports an RTF file, chunks it into small pieces (learning objects), and provides tools to manipulate the assembly structure, in this case an IMS manifest, through a set of operations such as those described in Section 5.1. Tools for creating and managing the required IEEE LOM metadata and for exporting a SCORM package are also provided.

In the current implementation the rules operate on the IEEE LOM metadata expressed in its RDF binding[9] which is compatible with Dublin Core RDF specification. In addition to metadata schemas we have implemented some of the IEEE LOM vocabularies as lightweight DAML+OIL ontologies and used ontology concepts instead of the plain string values. The tool was tested on the set of local college training material. For that purpose, we have expanded the IMS metadata schema with a set of elements specific to the economic sectors and adapted or built ontologies for these elements (including 'economic sector', 'basic skills', and 'market demand' ontologies).

Fig.1 shows a snapshot of Convertor with the suggested values for the field 'Economic Sector' (marked with number 6). Although the selection looks simple, the suggested values listed above the single line have the potential to save the user from searching through the list of 84 concepts in the full Economic Sector ontology.

6 Conclusions

Metadata is widely recognized as the key to the more effective retrieval and use of information, but most efforts to date have foundered on the resistance of the average content creator to developing and applying accurate and consistent metadata [3]. Anyone who has tried to apply metadata to a large collection of content objects realizes just how time consuming this can be. We believe that the solution to this lies in three related developments: (i) the introduction of XML-based metadata standards, of which the IEEE LOM is one example, (ii) metadata profiles which customize these standards for specific groups of users and (iii) tools that support metadata generation and management. In this paper we have focused on an approach in which the system generates suggested values for the metadata with the user either selecting the value from the range of proposed choices or using the suggested values as a pointer to the potentially relevant part of larger ontologies. As this does not reduce the amount of fields the user has to fill in we are considering weighting suggested values and setting up a threshold for automatic value assignment. This will need more experimentation and testing.

[9] http://www.imsglobal.org/rdf

Many current content object repositories, learning content management systems and content management systems assume a rich metadata environment. One reason for the delay in the widespread adoption of these systems is the difficulty, the cost, and in fact the reluctance of users to invest in the creation of accurate and consistent metadata. Systems that have been successful are typically those in which a very structured and controlled authoring environment can be created and enforced. This excludes many of the most interesting and creative sources of content, such as individual subject matter experts, the results of collaborative sessions, and the enormous wealth of legacy material that exists in the world today. We believe that the effective implementation of a system such as the one described here is the way forwards for widespread metadata creation and use.

Several approaches to the metadata generation have been tried so far. The metadata generation tools as SPLASH [9] or Aloha [16], or Recombo's Convertor can easily extract and fill values for the technical metadata fields. However, when we look at the systems generating values for other fields we find that they are either limited to a particular ontology domain [20] or they are specifically designed to work with some existing taxonomy [13]. In our approach we look at the object and its metadata record in its context – an assembly the object is part of and the repository the object's metadata record is stored in.

Considering repositories as source of the data is a basis of machine learning approaches. We have not done any comparison of our approach with these approaches yet and will explore some of these techniques in our future work.

Our research continues in several directions. Although anecdotal evidence gives us a positive feedback from the users our work is not complete until we complete an extensive empirical evaluation. By applying the datamining techniques for association rules mining we want to help the system designer in writing inference rules. As the suggested values are generated based on the values already filled in by the metadata creator the order in which values are filled determines how helpful the system can be. Based on the analysis of the rules we are investigating adaptable user interfaces for metadata creation and their usability acceptance. However, the most challenging task is to expand the system into distributed environment and specifically to integrate it with our infrastructure for connecting peer-to-peer repositories [9].

7 References

1. Aas, K., and Eikvil, L. Text categorisation: A survey. Technical report, Norwegian Computing Center, June 1999
2. Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. Scientific American, May. 2001.
3. Doctorow, C. Metacrap: Putting the torch to seven straw-men of the meta-utopia, Version 1.3: 26 August 2001, http://www.well.com/~doctorow/metacrap.htm
4. Dublin Core, www.dublincore.org
5. Friesen, N., Mason, J., and Wand, N. Building Educational Metadata Application Profiles. DC-2002: Metadata for e-Communities: Supporting Diversity and Convergence, Florence, October, 13-17, 2002. 63-69

6. Friesen, N., Roberts, A., and Fisher, S. CanCore: Learning Object Metadata, Canadian Journal of Learning and technology, Special issue on Learning Objects, Volume 28, Number 3, Fall 2002, 43-53
7. Greenberg, J., Pattuelli, M.C, Parsia, B., and Robertson, W.D. Author-generated Dublin Core Metadata for Web Resources: A Baseline Study in an Organization. Journal of Digital Information (JoDI), 2(2). 2001
8. Greenberg, J., and Robertson, W.D. Semantic Web Construction: An Inquiry of Authors' Views on Collaborative Metadata Generation. DC-2002: Metadata for e-Communities: Supporting Diversity and Convergence, Florence, October, 13-17, 2002. 45-52
9. Hatala, M.., and Richards, G. Global vs. Community Metadata Standards: Empowering Users for Knowledge Exchange, in: I. Horrocks and J. Hendler (Eds.): The Semantic Web – ISWC 2002, Springer, LNCS 2342, pp. 292-306, 2002.
10. Hearst, M.A. Untangling Text Data Mining. Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics, University of Maryland, June 20-26, 1999
11. Horrocks, I., and Hendler, J. (eds.) The Semantic web – ISWC 2002 Springer, LNCS 2342, 2002..
12. IEEE P1484.12.2/D1, 2002-09-13 Draft Standard for Learning Technology - Learning Object Metadata - ISO/IEC 11404, http://ltsc.ieee.org/doc/wg12/LOM_1484_12_1_v1_Final_Draft.pdf
13. Jenkins, C., Jackson, M., Burden, P., and Wallis, J. Automatic RDF Metadata Generation for Resource Discovery, Proceedings of Eighth International WWW Conference, Toronto, Canada, May 1999.
14. Kan, M.Y., McKeown, K.R., and Klavans, J.L. Domain-specific informative and indicative summarization for information retrieval. Proc. Of the First Document Understanding Conference, New Orleans, USA, pp.19-26, 2001
15. Leacock, T., Farhangi, H., Mansell, A., and Belfer, K. Infinite Possibilities, Finite resources: The TechBC Course development Process. Proceedings of the 4th Conf. on Computers and Advanced Technology in Education (CATE 2001), June 27-29, 2001, Banff, Canada, pp.245-250.
16. Magee, M., Norman, D., Wood, J., Purdy, R, Iwing G. Building Digital Books with Dublin Core and IMS Content Packaging. DC-2002: Metadata for e-Communities: Supporting Diversity and Convergence, Florence, October, 13-17, 2002. pp.91-96
17. Qin, J., and Finneran, C. Ontological representation of learning objects. In: Proceedings of the Workshop on Document Search Interface Design and Intelligent Access in Large-Scale Collections, JCDL'02, July 18, 2002, Portland, OR
18. Richards, G. The challenges of the Learning Object Paradigm. Canadian Journal of Learning and technology, Special issue on Learning Objects, Volume 28, Number 3, Fall 2002, 3-9.
19. Shareable Content Object Reference Model (SCORM), www.adlnet.org
20. Stuckenschmidt, H., van Harmelen, F. Ontology-based Metadata Generation from Semi-Structured Information. Proceedings of the First Conference on Knowledge Capture (K-CAP'01), Victoria, Canada, October 2001
21. Weinheimer, J. How to Keep Practice of Librarianship Relevant in the Age of the Internet. Vine (Special Issue on Metadata, Part 1), 116: 14-27.
22. Weinstein, P., and Birmingham, W.P. Creating ontological metadata for digital library content and services, International Journal on Digital Libraries, 2:1 pp. 20-37. Springer-Verlag, 1998
23. Wiley, D. (ed.) The Instructional Use of Learning Objects, Association for Educational Communications and Technology, 2001