

A COMPREHENSIVE SYSTEM FOR COMPUTER-AIDED METADATA GENERATION

Marek Hatala
Simon Fraser University Surrey
2400 Central City
Surrey, BC, Canada, V3T 2W1
+1 (604) 586-6053
mhatala@sfu.ca

Steven Forth
Recombo Inc.
Suite 240 - 1737 West Third Ave
Vancouver, BC V6J 1K7, Canada
+1 (604) 736-2272 x 131
steven@recombo.com

ABSTRACT

In this paper, we describe a system that generates suggested values for metadata elements. The system significantly increases the productivity of metadata creators as well as the quality of the metadata. The system is applicable to any metadata standard both for single metadata records and collections of related metadata records. Instead of aiming for automated metadata generation we have developed a mechanism for suggesting the most relevant values for a particular metadata field. The suggested values are generated using a combination of four methods: inheritance, aggregation, content based similarity and ontology-based similarity. The main strength of the system is that it provides a generic solution independent of the metadata schema and application 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 generations. First, we describe the generic system and then provide examples of how the system can be implemented and used in tools developed for creating e-learning material conformant with the SCORM reference model and the IEEE LTSC LOM standard.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces, I.2.3 [Artificial Intelligence]: Deduction and Theorem Proving – *Deduction, Inference engines*, J.1 [Computer Applications]: Administrative data Processing,

General Terms

Algorithms, Management, Performance, Design, Human Factors, Standardization

Keywords

Metadata systems, User tools, Ontologies, Knowledge-based systems, Semantic web, IEEE LTSC LOM, SCORM

1. INTRODUCTION

E-learning is a major industry with growing applications in many sectors, including industrial training, higher education, the military, individual training etc. This wide range of applications has given rise to a whole ecology of specialized systems for managing learning and learning resources [26]. 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 [14] and standardization of other aspects of e-learning, including packaging of learning resources using IMS Content Packaging [15], or messaging between content and systems using the AICC CMI (at the time of writing this was being formalized as an IEEE standard) and a proposal for the sequencing of learning resources known as IMS Simple Sequencing [7]. The Shareable Content Object Reference Model (SCORM) from the Advanced Distributed Learning Network (ADL) [8] 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 both constrain and enrich the standard. The standard is constrained by specifying which of the metadata elements in the standard are required and by limiting the possible values of these elements. It can be enriched by applying taxonomies or ontologies to each of these elements. We refer to such implementations as metadata profiles (for example, Cancore [7] is a metadata profile for sharing learning resources based on IMS LOM).

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. There can be as many as ninety metadata elements applied to each of these, though twenty would be more common for the learning objects and five for the media elements, generating anywhere from 1,000 to 5,000 separate metadata values for a three hour course [20]. As discussed below, certain 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 [9] and the authors have developed a model in which learning objects are 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 [10] show increased awareness of the importance of creating metadata for objects even in industrial settings, despite worries about the quality of the metadata [29]. As a follow on study to [10] on collaborative metadata generation, the results have shown the authors most often seek expert help for generating ‘subject’ metadata [9]. The study was conducted for a simple 12-element set from Dublin Core [29]. As number of elements and requirements for richly structured metadata sets increase the need for systems that support metadata generation and management will grow.

There is an important relationship between ‘subject’ metadata, ontology construction and the goals of the semantic web. The semantic web initiative [3] 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 [13]. 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 [31]. A more specific area of semantic web research related to our work is that of mapping between ontologies [12][23][24][30]. The overall process of finding mappings between ontologies presented in Prasad et al. [24] is similar to the process we use to find similar objects for metadata generation. The two approaches also draw on the common set of underlying techniques from text categorization, summarization and text mining [1][12][19].

Several systems for ontological metadata have been proposed. Weinstein and Birmingham [31] 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. [18] 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. [28]. The last approach relies on the existence of a single ontology all the web pages can be related to.

The system presented in this paper helps metadata creators to create high quality metadata by generating suggested values for metadata fields. The main strength of the system is that it provides a generic solution independent of the metadata schema and application 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. The four methods of metadata generation are presented in Section 4. Section 5 analyzes the operations on the objects that affect the suggested metadata values and presents decision tables for the metadata generation scheduling algorithm. In Section 6 we present a design of a module for metadata generation and review the current status of the system implementation. We conclude with a discussion of our approach and provide direction for further projects.

2. METADATA SYSTEMS

One of the main design principles for the World Wide Web was to make it possible to find information quickly. Although the linking of web pages makes it possible to move from one page to another, most users rely on search engines and directories to find information they are not already aware of. In the simplest terms, search engines use string keywords and phrases to discover web pages by matching keywords against the textual content of web documents themselves. The problems of this approach are well known: the thousands of matches retrieved for almost any search string. Although some search engines use different heuristics to provide better results [4] the root of the problem lies between string representation of the underlying meaning and the meaning itself.

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 or record in a library catalogue, providing a controlled and structured description of resources through a searchable “access point” such as title, author date, location, description, etc. [8]. There are also compelling arguments against metadata which are based mainly on the nature of human behavior [5]. It has been suggested that metadata are skewed in the interests of the metadata author, which results in poorer results for the user. This can happen when the metadata author tries to intentionally mislead the user, it is curious how many web searches using innocent search strings lead to pornographic sites, or when the metadata creator is locked within their own conceptual framework and has difficulty imagining how a user might categorize the world. 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. [8] 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 at helping to overcome some of the well-known problems [5].

2.1 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 [6], or they can be specific for a particular community, for example IEEE LTSC Learning Object Metadata (IEEE LOM) [14]. A metadata standard provides a set of elements, defines their meaning, and provides guidelines and constraints on how to fill element values. For example, the IEEE LOM defines elements for general object descriptions, technical interoperability, lifecycle, rights management and others. 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 Network¹, which uses the IEEE LOM, describes a reference model for describing a collection of learning resources, their aggregations and their underlying assets. The ADL is in the process of extending this model through the IMS Simple Sequencing proposal [17].

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.

2.2 Typology of Metadata Elements

The elements of metadata schemas² 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 [3].

3. SOURCES 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. Eventually, metadata creators can master the guidelines of the standard in use. It is the use of external taxonomies and ontologies, which is often beyond the creator's expertise [9]. 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 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.

3.1.1 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.

3.1.2 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, setting some values in any one of the objects in the assembly 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 from 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.

3.1.3 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][12][19], however methods for other media types are not generally available.

The second notion is similarity defined by some set of rules. For example, if a new record specifies some value for a particular element we can find other records with the same or similar values for this element and use these records as sources for suggested values in the new record.

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.

3.2 Sources Based on Types of Elements

With respect to the element typology presented in Section 2.2, 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 have identified several sources which can be used to generate a set of suggested values (SSV) for metadata elements. The work presented here 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 Section 3.1.1 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 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)

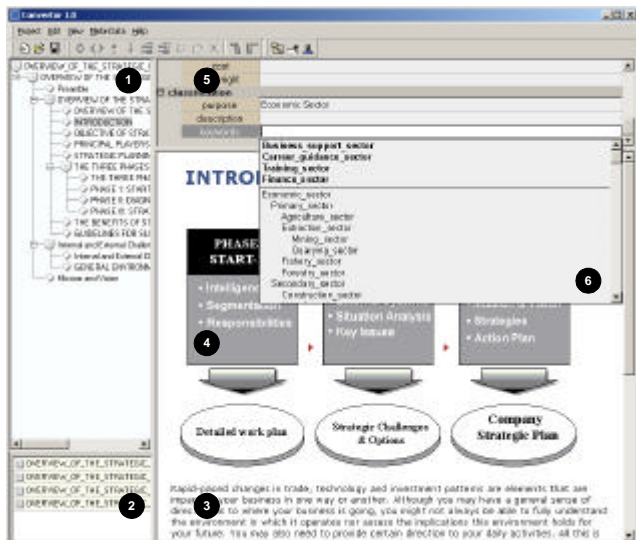


Figure 1 Snapshot of the Recombo assembly and metadata creation tool containing: assembly hierarchy (1), list of assets in the selected object (2), object (3), asset included in the object representing an aggregation (4), metadata fields (5), and list of suggested values in front of the vocabulary (6).

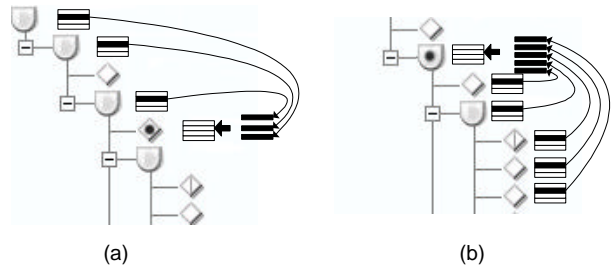


Figure 2. Generation of suggested values through inheritance (a) and accumulation (b)

into a hierarchy of aggregations. Figure 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 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. Figure 1 displays an aggregate with 2 objects included.

4.1 Generating Suggested Values Through Inheritance

This method is applicable to the metadata records of objects organized into assemblies and aggregations. The inheritance method is applicable only to those elements which exhibit an inheritance property. A set of suggested values for metadata element E for object O_n in the hierarchy H is defined as (Figure 2a):

$$ISSV_{O_n}(E) = \bigcup_{i=0}^{n-1} M_{O_i}(E) \{O_i | O_i \in \text{path from } O_0 \text{ to } O_n\}$$

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.

$$ISSV_{A_n}(E) = I_A SSV \cup M_{O_n}(E)$$

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. However, for the same ‘creator’ element from the asset, in addition to the values generated by the formula above, another set of values would be generated based on the values collected from all the assets in the assembly with the ‘similar’ media type 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 object³.

³ The second set of values would be generated using semantically defined similarity.

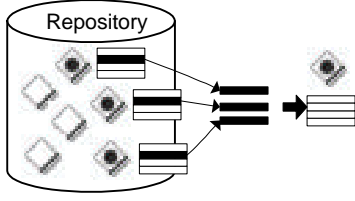


Figure 3. Generating suggested values through content similarity.

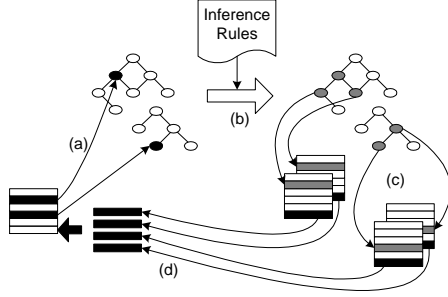


Figure 4. Generating suggested value through semantic similarity: values mapped to ontology concepts (a) are consumed by inference rules (b) which define semantically similar concepts. The records with similar concepts are found in the repository (c) and their values are used as similar values for a specific metadata field.

4.2 Generating Suggested Values Through Accumulation

This method works uniformly through assemblies and aggregations. The accumulation method is applicable only to those elements which exhibit an accumulation property. A set of suggested values for metadata element E for object O_n in the hierarchy H is defined as (Figure 2b):

$$ASSV_{O_n}(E) = \bigcup_i M_{O_i}(E) \{O_i \mid O_i \in \text{subtree of } O_n\}$$

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.

4.3 Generating Suggested Values Through Content Similarity

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. Figure 3 shows a diagram demonstrating the principle. The following formula captures this definition:

$$CSSV_{O_n,R}(E) = \bigcup_i M_{O_i}(E) \left\{ O_i \mid \begin{array}{l} O_i \in \text{repository } R_n \text{ and} \\ O_i \text{ and } O_n \text{ have similar content} \end{array} \right\}$$

The formula above includes a specification of the algorithms for calculating the content similarity. As the notation indicates, different mechanisms can be used for calculation of the similarity for different elements. For example, to generate SSV for the 'technical requirements' element (mentioned above) 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.

4.4 Generating Suggested Values Through Semantically Defined Similarity

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. Figure 4 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. Formally, we specify the set of the suggested values as follows:

$$SSSV_{O_n,R,E_1,\dots,E_k}(E) = \bigcup_i M_{O_i}(E) \left\{ O_i \mid \begin{array}{l} O_i \in \text{repository } R \wedge O_n \text{ and } O_i \text{ is similar} \\ \text{to } O_n \text{ based on similarity of } M_{O_i}(E_1) \\ \text{and } M_{O_i}(E_1), M_{O_i}(E_2) \text{ and } M_{O_n}(E_2), \\ \dots, M_{O_i}(E_k) \text{ and } M_{O_n}(E_k) \end{array} \right\}$$

This method is most suitable for elements which use vocabularies derived from 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. This method is most suitable for customized metadata systems (metadata profiles) 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.

One set of domain ontologies defines the use of the objects and object aggregation. An example here is the IMS Learning Design Information Model [30] which can be represented as ontology and used to define relations between objects. Another set of domain ontologies defines the relationships between concepts in a specific subject matter. For example in the professional training domain the 'economic sector', 'basic skills', 'market demand' ontologies are all interrelated.

4.5 Combination of Sets of Suggested Values

The four methods above generate four sets of suggested values. The final set of suggested values is a combination of all four sets:

$$SSV_{O_n,R}(E) = ISSV_{O_n}(E) \cup ASSV_{O_n}(E) \cup CSSV_{O_n,R}(E) \cup SSSV_{O_n,R}(E)$$

It is possible, that not all components in the formula above will be present for each element. If more than one set is present, it would be desirable to consider a weighting of values originating from different sets.

$$SSV_{O_n,R}(E) = w_I ISSV_{O_n}(E) \cup w_A ASSV_{O_n}(E) \cup w_C CSSV_{O_n,R}(E) \cup w_S SSSV_{O_n,R}(E)$$

Although all generated values can be presented to the user, the implementers must always take into account user interface issues. If too many values are suggested to the user the utility of the system declines. This is especially true if the suggested values are in a random order. The weighting of the values can help to resolve these issues.

5. INFLUENCE OF OPERATIONS ON SUGGESTED METADATA VAUES IN 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 fields⁴. When creating an assembly, the total number of fields can easily reach hundreds or thousands. A graphical interface, as the one shown in Figure 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.

5.1 Typology of Content and Metadata Operations

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 the influence of operations on the set of suggested values generated by methods presented in the previous section. We have identified the following operations which affect the suggested values⁵:

Join (object content)

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 (object content)

Unjoin operation restores the state before the Join operation was performed.

Split (object content)

Split operation separates the content of a leaf node into two objects and by successive splits into multiple objects.

Group (assembly structure)

Group operation introduces an 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 (assembly structure)

Ungroup operation restores the state before the Group operation was performed.

Promote (assembly structure)

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 (assembly structure)

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 for Metadata Element

Metadata value is filled or selected from the vocabulary for the metadata element.

5.2 Rules for Change Propagation

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.
- In the object's subtree.

Table 1-Table 3 summarize effects of the operation on a particular object. The effects are represented in four columns; each showing the necessity to regenerate a set of suggested values obtained by a particular method. The tables are a basis for implementing a re-computation scheduling logic in the metadata generation module.

ISSV – Inheritance

ASSV – Accumulation

CSSV – Content Similarity

SSSV – Semantic Similarity

Table 1. Influence of the operations on the object's SSV occurring in the object itself

Object itself	ISSV	ASSV	CSSV	SSSV
Join	N	Y	Y	N
Unjoin	N	Y	Y	N
Split	N	Y	Y	N
Group	Y	N	N	N
Ungroup	Y	N	N	N
Promote	Y	Y	N	N
Demote	Y	Y	N	N
Value filled	N	N	N	Y

Table 2. Influence of the operations on the object's SSV occurring on objects in the path above the object

Path Above	ISSV	ASSV	CSSV	SSSV
Join	N	Y	Y	N
Unjoin	N	Y	Y	N
Split	N	Y	Y	N
Group	N	N	N	N
Ungroup	N	N	N	N

⁴ A field is a representation of the metadata element in the form interface.

⁵ In addition to the presented set of operations two more operation are necessary to be able to manipulate the assembly: move node up and move node down.

Promote	N	Y	N	N
Demote	N	Y	N	N
Value filled	N	N	N	Y

Table 3. Influence of the operations on the object's SSV occurring on objects in the subtree

Subtree below	ISSV	ASSV	CSSV	SSSV
Join	Y	N	Y	N
Unjoin	Y	N	Y	N
Split	Y	N	Y	N
Group	Y	N	N	N
Ungroup	Y	N	N	N
Promote	Y	N	N	N
Demote	Y	N	N	N
Value filled	N	N	N	Y

6. 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 set of the objects. Although domain ontologies can be plugged into the system, the system is generic and independent of the application domain.

An architectural diagram in Figure 5 shows the role of the generation module in the larger metadata management system. The generation module responds to requests generated as a result of user's actions in the metadata creation tool. In the process of generation of suggested values the generation module communicates with other components. In the subsequent sections we describe both the external communication as well as internal design of the generation module.

6.1 Services Provided by the Generation Module

The generation module responds to requests from the metadata creation tool. As the creation tool is an end-user application it is crucial that the response time from the generation module is immediate. Therefore, we have designed a generation module functioning in parallel to the creation tool. The generation module is notified about the events in the creation tool via asynchronous messages, namely about:

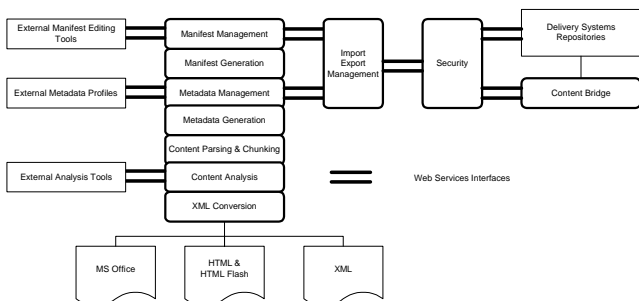


Figure 5. Metadata management systems components.

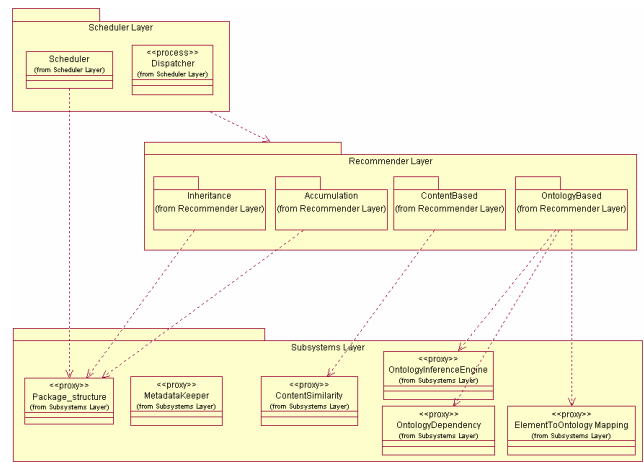


Figure 6. Internal structure of the generation module

- Changes of object focus
- Changes to the object assembly (operations presented in Section 5.1 above)
- Changes to the metadata values (values filled during an operation in Section 5.1 above)

The generation module responds to the changes by scheduling re-generation of the suggested values based on the rules for change propagation described in Section 5.2.

The only synchronous message sent from the creation tool to the generation module is a request for suggested values for a particular metadata element field. The generation module responds with two values: 1. a set of suggested values generated up to that point, and 2. a boolean value indicating the set is complete. If the generation has not been finished we take advantage of the fact that the user will select all fields at once but visit them in some sequential order. Therefore we can use that time to finish generation and the creation tool will issue a second request for suggested values when the user actually selects a specific field.

6.2 Generation Module Design

The generation module consists of three main layers (Figure 6). In the scheduler layer, the Scheduler is responsible for processing messages from the creation tool and scheduling generation of suggested values. The Dispatcher operates on the list of scheduled generation tasks and dispatches them to the appropriate methods on the Recommender layer. The methods in the Recommender layer are concurrent and the Dispatcher has the ability to change the priority of the tasks based on the messages received from the creation tool.

The Recommender layer contains modules implementing methods described in Section 4. The methods in the Recommender layer make use of the external components in the Subsystems layer.

6.3 External Support Components

The generation module used is communicating with several external modules. The external modules are represented by proxies in the subsystem layer which provides for the pluggable architecture. The following components are essential to support the generation module:

AssemblyStructure

The generation component needs to be aware of the relation between the objects in the assembly to determine the effects of the operations on set of suggested values (see Section 5.2).

MetadataKeeper

This is a facade to the metadata repository providing the generation module with the ability to retrieve values for metadata elements.

ContentSimilarity

This external component provides the generation module with the capability to compare the content of two objects.

ElementToOntologyMapping

This component keeps track of ontology assignments as vocabularies for metadata elements.

OntologyDependency

The dependencies between different elements using ontologies are expressed using the knowledge rules. This component represents a knowledge base for the semantic similarity method.

OntologyInferenceEngine

The external inference engine is used by the semantic similarity method. The engine operates on the knowledge rules maintained by OntologyDependency component.

In addition to these specific components a whole set of third-party support tools and applications is required to prepare and maintain essential data for generation modules. These include database management tools, ontology editors, knowledge-base editors and debugging tools, and others.

6.4 Implementation

The generation module is being implemented within the content integration framework developed by Recombo Inc. in Vancouver, British Columbia (<http://www.recombo.com>, see Figure 7). 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 is Converter for Word, which imports a .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.

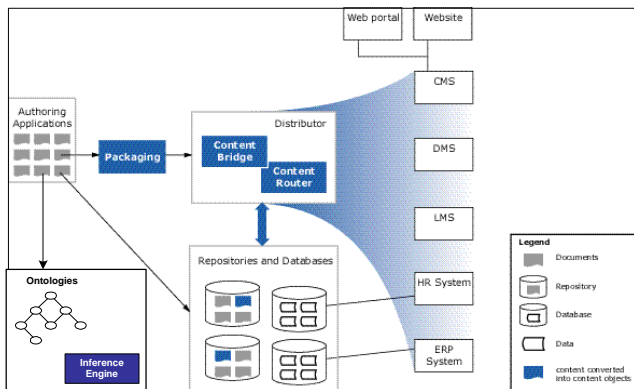


Figure 7. Enterprise content integration framework

In the current implementation we have classified each metadata element with regard to the four methods for generation of suggested values presented in the Section 4. We have converted suitable vocabularies into ontologies (e.g. 'educational context' element) and used the ontology concepts instead of the plain string values. We have tested or methods on the set of local college training material. For that purpose, we have expanded IMS metadata schema with a set of elements specific to the economic sectors and adapted or built ontologies for these elements (e.g. 'economic sector', 'basic skills', 'market demand').

In the current version, we have implemented only three methods of generating metadata: inheritance, accumulation and semantic similarity. The implementation of inheritance and accumulation method was fairly straightforward. We have based the implementation of the semantic similarity method on DAML+OIL ontologies⁶ and used a set of publicly available tools: ontology editor OilEd⁷ [2] and DAMLJessKB⁸ inference engine and rules. We have also experimented with several other tools, namely an RDF repository Sesame and RQL language⁹ and backward chaining engine SWI-Prolog¹⁰ but we have found that they don't provide a direct support for DAML+OIL and are not as suitable for the generation tasks.

We have found the use of a forward-chaining inference engine favorable over backward-chaining as it allowed us to do the required inference offline and store the results in the lookup table used during the generation tasks. However, it would be premature to decide on the general applicability of this approach as the ontologies and especially the knowledge rules we have used were of low to moderate complexity.

We have not fully implemented the content similarity method in the current implementation due to the unavailability of a good component for computing similarity of two objects based on their content. We have performed several experiments with publicly available text summarization and text classification toolkits but the results were less than satisfactory. However, there is a great deal of expertise available in this domain and we plan to revisit this issue in the future.

Figure 1 shows a snapshot of Converter 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 a potential to save the user from searching through the list of 84 concepts in the full Economic Sector ontology.

Note to reviewers: at the time of writing we have tested the implemented version only internally. We will include the results of testing within the wider audience into the final version and report on them during the conference.

7. DISCUSSION

Metadata is widely recognized as the key to the more effective retrieval and use of information, but most efforts to date have floundered on the resistance of the average content creator to

⁶ <http://www.daml.org/2001/03/daml+oil-index>

⁷ <http://oiled.man.ac.uk>

⁸ <http://plan.mcs.drexel.edu/projects/legorobots/design/software/DAMLJessKB/>

⁹ <http://sesame.aidministrator.nl/>

¹⁰ <http://www.swi-prolog.org/>

developing and applying accurate and consistent metadata [5]. 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 to generate their own values. We believe that in most cases this mode will lead to more useful results than a fully automated system and that it will be easier to use, less time consuming and more consistent than a system in which a user must input metadata without support.

The rules of how content objects can be manipulated within a tree structure (what we call content object algebra) and how this affects metadata values is another important contribution of this work. In real-life contexts, metadata is used in interaction with changing content aggregations and structures, and for the metadata to be durable across changes and transformations it must be supported by a system such as the one described here.

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 [11] or Aloha [22], 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 [28] or they are specifically designed to work with some existing taxonomy [18]. In our approach we look at the object and its metadata record in its context. The context is formed by an assembly the object is part of and the repository the object's metadata record is stored in. Our approach is comprehensive as it makes use of all available sources of information: object content, relation of the object to other objects in the assembly, domain knowledge about the area represented by a particular set of metadata elements, and the formalization of this domain knowledge within ontologies. The proposed design of the metadata generation system guarantees fast response and makes it suitable as an embedded component for end-user metadata creation tools.

The robustness of the solution poses an extra demand on the deployment of the system into the target application setting. Some parts of the solutions are transferable into any domain without modification. The only requirement is the proper configuration of the system and specification of the properties of the metadata

elements. However, to get the biggest benefit from the system some specialized expertise is required. The semantic similarity method requires a set of knowledge modeling skills which might be available for larger projects only. On the other hand, it is exactly the larger organization setting which benefits the most from formalization of their domain into a set of ontologies which can be used as the values for metadata elements.

8. CONCLUSION

In this paper we have described a system that generates suggested values for metadata elements. The system significantly increases the productivity of metadata creators as well as the quality of the metadata. The system is applicable to any metadata standard both for single metadata records and collections of related metadata records. The generated metadata values are used as suggestions for the metadata creator. The suggested values are generated using a combination of four methods: inheritance, aggregation, content based similarity and ontology-based similarity. The main strength of the system is that it provides a generic solution independent of the metadata schema and application domain. In addition to generating metadata from standard sources such as object content and user profiles, the system treats also other sources as prime sources for metadata generation: metadata record assemblies, metadata repositories, explicit domain ontologies and inference rules.

The next phase of research and development on this project is to develop a fully functioning system and to integrate within Recombo's line of products for creating content that complies with XML content standards (specifically SCORM) and for integrating the various systems and forms of content used to deliver learning and training. On the research side, once the full implementation is available we will proceed to develop and apply domain-specific ontologies from several different domains. The system will be compared with systems supporting (unmediated) human development and application of metadata and various systems for automated development and applications of metadata under development by Recombo and other organizations. The different approaches will be evaluated according to three criteria: (i) time required to develop and apply metadata, (ii) consistency of the metadata and (iii) utility of the metadata for search, lifecycle management and integration support.

9. ACKNOWLEDGMENTS

A National Research Council of Canada grant to Recombo Inc supported part of the work in this paper.

10. REFERENCES

- [1] Aas, K., and Eikvil, L. Text categorisation: A survey. Technical report, Norwegian Computing Center, June 1999
- [2] Bechhofer, S., Horrocks, I., Goble, C., Stevens, R. OilEd: a Reason-able Ontology Editor for the Semantic Web. Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, September 19-21, Vienna. Springer-Verlag LNAI Vol. 2174, pp 396--408. 2001.
- [3] Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. Scientific American, May. 2001.
- [4] Brin, S. and Page, L.: The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems, 30 (1-7), 1998, 107-117.

- [5] Doctorow, C. Metacrap: Putting the torch to seven strawmen of the meta-utopia, Version 1.3: 26 August 2001, <http://www.well.com/~doctorow/metacrap.htm>
- [6] Dublin Core, www.dublincore.org
- [7] 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
- [8] 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
- [9] 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
- [10] 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
- [11] 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.
- [12] 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
- [13] Horrocks, I., and Hendler, J. (eds.) The Semantic web – ISWC 2002 Springer, LNCS 2342, 2002..
- [14] IEEE P1484.12.2/D1, 2002-09-13 Draft Standard for Learning Technology - Learning Object Metadata - ISO/IEC 11404, http://tsc.ieee.org/doc/wg12/LOM_1484_12_1_v1_Final_Draft.pdf
- [15] IMS Content Packaging Information Model. Version 1.1.2 Final Specification. http://www.imsproject.org/content/packaging/cpv1p1p2/imscp_infov1p1p2.html
- [16] IMS Learning Design Information Model. www.imsproject.org/learningdesign
- [17] IMS Simple Sequencing Information and Behavior Model, Version 1.0 Public Draft Specification http://www.imsproject.org/simplesequencing/v1p0pd/imsss_infov1p0pd.html
- [18] 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.
- [19] 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
- [20] 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.
- [21] Maedche, A., Motik, B., Silva, N., and Volz, R.: MAFRA - An Ontology Mapping FRamework in the context of the Semantic Web, Workshop on Knowledge Transformation for the Semantic Web (KTSW) Lyon, France, July 23, 2002
- [22] 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
- [23] Noy, N.F., Musen, M.A.. Anchor-PROMPT: Using non-local context for semantic matching. In Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA, 2001.
- [24] Prasad, S., Peng, Y., and Finin, T. Using Explicit Information To Map Between Two Ontologies. AAMAS 2002 Workshop on Ontologies and Agent Systems, Italy, July 2002.
- [25] 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
- [26] 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.
- [27] Shareable Content Object Reference Model (SCORM), www.adlnet.org
- [28] 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
- [29] 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.
- [30] Weinstein, P., and Birmingham, W.P. Comparing Concepts in Differentiated Ontologies Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management (KAW'99). October 1999, Banff, Alberta, Canada
- [31] 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
- [32] Wiley, D. (ed.) The Instructional Use of Learning Objects, Association for Educational Communications and Technology, 2001