International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:3, No:1, 2009

A Metametadata Architecture for Pedagogic Data Description

A. Ismail, M. S. Joy, J. E. Sinclair, and M. I. Hamzah

***Abstract*—**This paper focuses on a novel method for semantic searching and retrieval of information about learning materials. Metametadata encapsulate metadata instances by using the properties and attributes provided by ontologies rather than describing learning objects. A novel metametadata taxonomy has been developed which provides the basis for a semantic search engine to extract, match and map queries to retrieve relevant results. The use of ontological views is a foundation for viewing the pedagogical content of metadata extracted from learning objects by using the pedagogical attributes from the metametadata taxonomy. Using the ontological approach and metametadata (based on the metametadata taxonomy) we present a novel semantic searching mechanism.These three strands – the taxonomy, the ontological views, and the search algorithm – are incorporated into a novel architecture (OMESCOD) which has been implemented.

***Keywords*—**Metadata, metametadata, semantic, ontologies.

I. Introduction

THE World Wide Web is the *raison-d'être* for the hypertext format that the Internet supports. Hypertext systems are particularly practical for organizing and browsing through large databases or corpora that consist of disparate types of information. Current research into frameworks and models of hypertext has involved both the web infrastructure and embedded link structure. The Semantic Web [3] is a grand vision that supports conveying metadata about resources in an explicit, understandable and machine-processable way for searching and organizational purposes.

In this era of the digital world of information, there are issues regarding searching and finding relevant and potentially useful learning materials related to users' needs. The advent of the

World Wide Web [2] has contributed towards standardized access to diverse data sources.

A. Ismail is with the Department of Information Science, Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia (National University of Malaysia) (phone: +603-89216712; fax: +603-89216184; e-mail: amira@ftsm.ukm.my).

M. S. Joy is with the Department of Computer Science, University of Warwick, United Kingdom (e-mail: M.S.Joy@warwick.ac.uk).

J. E. Sinclair is with the Department of Computer Science, University of Warwick, United Kingdom (e-mail: J.E.Sinclair@warwick.ac.uk).

M. I. Hamzah is with the Faculty of Education, Universiti Kebangsaan Malaysia (National University of Malaysia) (phone: +603-89216238; fax: +603-89254372; e-mail: cher@ukm.my).

Web service technology allows a consistent access via web standards to software and applications on many computer platforms, and has supported the transformation from a static document collection to an intelligent and dynamic data integration environment.

Recently, new phrases have become common in this area of research, such as "Learning Objects", "Learning Object Metadata" and "Learning Object Repositories". These terms have mainly been defined and applied due to their general meaning in the Educational Technology field and this is appropriate due to the interdisciplinary nature of the subject.

In this paper, we focus on metadata instead of learning objects themselves. Metadata is "structured data which describes the characteristics of a resource" [9]. .Metadata can be described as structured information that describes resources or learning materials to support the searching, discovering and managing activities to display extracted information in some way.

Metadata can be categorized depending on certain functions such as administrative, descriptive, technical usage, nature, technique of creation, category, structure, and semantics levels [5].

This also means that a few issues relating to Learning Objects, such as learning object management, creation, quality and granularity, will not be regarded as main topics for discussion, although certain requirements for handling the learning process and instructional theories in the field of E-Learning may be addressed.

This research work may be regarded as a test bed for presenting meta modelling languages, metadata sets, metadata organization and searching mechanisms with the help of *ontologies* for educational purposes. Ontologies outline the vital infrastructure of the Semantic Web [3]. This means that, as "ontology", any formalism will be considered within a well identified mathematical framework which supports user-defined relations and concepts and a subconcept taxonomy [4].

II. Aim and Research Novelty

A. Research Aim

The major research question has been designed as follows:

- How can pedagogic metadata adaptation be handled effectively?

The aim of the current research is to explore, design and evaluate a model for describing and identifying the pedagogic semantic relationships of learning objects by using tagged metadata.

These could be expressed by generating educational metadata using a semantic search engine and novel reference mechanisms for semantic relationship metadata, later known as *Metametadata,* by using SCO (*Sharable Content Object*) to represent the learning objects, according to the SCORM (*Sharable Content Object Reference Model*)[1].

B. Novelty of this Research

The novelty of this research is as follows:

- A Novel Metametadata taxonomy has been developed which provides the basis for a semantic search engine to extract, match and map queries to retrieve relevant results.
- Search algorithms have been developed which include semantic search of capturing metadata instances which determine the relevancy of the retrieved results when measured against the search criteria.

The use of ontological views is a foundation for viewing the pedagogical content of the extracted metadata by using the pedagogical attributes from the metametadata Taxonomy.

C. Why Metametadata

A principal motivation for using metametadata in the context of a pedagogic architecture which uses learning objects is that if the designer or administrator wishes to integrate metadata from various repositories or sources, the format and content of the metadata may vary considerably. A "high-level" view of the metadata, in the form of metametadata, will assist the process.

Metametadata are structured descriptions about a set of metadata which intelligently describe and capture relevant identified characteristic properties and relationships between metadata to aid locating, managing and retrieving data.
Metametadata are useful for the following purposes:

- providing sufficient information about metadata to enable intelligent searching;
- implementing flexible dynamic semantic mappings between metadata vocabularies;
- processing and displaying different explicit and implicit characteristics of the stored data sets;
- associating sets of related data by identifying semantic relationships between the associated metadata;
- providing consistent semantics and structures for metadata in the repositories or database schemas, browsing interfaces and presentation of content [7].

A classification scheme for pedagogic metametadata has been designed in order to provide a strong foundation for the future implementation of a pedagogic architecture supported by metametadata.

Implicit metadata for leaning objects is often used for administration purposes and can be captured through the context of the learning objects. Explicit metadata are normally straightforward metadata that are coded in a simple format (such as XML).

We can consider metametadata as also being either implicit or explicit. As an example of implicit metametadata, we might have a relationship that states that "Adam *wrote* 'LearningJava.org'" is similar to "Adam *created* 'LearningJava.org'".

In terms of metadata, we might have the following two related tags for the learning objects stored in LearningJava.org:

```
<creator name="Adam" />
<writer name="Adam" />
```

In other words, there is a semantic similarity between the tags and attributes stored in the metadata, and an element of metametadata might capture that similarity. Such implicit relationships might be queried by users through a database interface browser, so that 'LearningJava.org' would be selected by a query "web resources *authored* by Adam".

In another learning context, a programmer is developing and testing software for two projects, and the files are marked up with metadata. Using metametadata the similarities between classes of files with equivalent functionality may be represented.

For explicit metametadata to be viable, we need to understand how to identify (address) individual metadata elements external to a specific metadata instance. These can be linked with connector metametadata types that will identify metadata for specific locations, such as URIs included in structured metadata in other documents.

In the context of this paper, relationship types for metametadata are proposed to connect with these three metadata types: *semantic* metadata that can be used to describe the subject matter of the resource or document; *context* metadata which characterize relationships with external entities or the meaning of the learning objects or documents (for example, author, publisher); and *structural* metadata which indicate a description of the internal media type, structure and presentation layout, such as text, sound, image, simulation, video, etc.

We propose a taxonomy of metametadata in order to provide a common framework containing semantic definitions together with further contextual expression.

D. Metametadata Concept

The work on the Metametadata taxonomy is focused on the identification of the required metadata elements consisting of *Class, Property* and *Representation.*

Metametadata Element Concept (MeMeC) = ObjectClass + Property

Metametadata element (MeMe) = Metametadata Element Concept + [Representation]

Fig. 1 presents the Metametadata Element Concept to view the relationship between metametadata element, *representation, object classes, property* and *value domain.* A class is a set of clearly defined ideas, abstractions, or "things" in the real world which have common behaviour and properties.

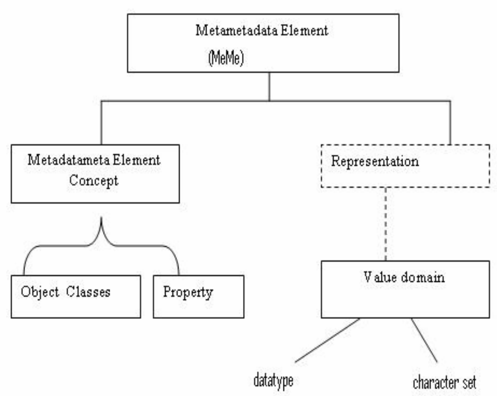

Fig. 1 Metametadata Element Concept

A property is an attribute common to all members of a class. A representation of data describes a value domain, data type, and a character set. *Object classes* can be described as the entity (the 'thing') for other objects specialization. Specialization may permits object classes to be grouped and subtyped to help users browse and locate relevant object classes.

A property describes the particular characteristic or attribute of that entity. Examples for broadly defined object classes include *Person*, *programmer* and *organization* or specific object classes example such as *Client* or *Child. An object class can be* related *with a single parent object class*. A child object inherits features of its parent object class which may contain unique features.

The metametadata concept is based on pedagogical selection by having type-based logical representations that will be used as vocabularies the common kinds of learning object features. However, the educational category does not describe the significant connections or relationships between each of the following metadata elements: Interactivity type, Learning resource type, Interactivity level, Intended end user role, Context, Difficulty, Typical learning time, Description and Language of the typical intended user [6].

The proposed metametadata relationship defines the semantic relationship between pedagogical metadata elements. Educational metadata from one category in the IEEE LOM (Learning Object Metadata) specification cover the pedagogical aspects or elements for the learning objects. Other elements listed – the interactivity type or level, semantic density and difficulty – have not been elaborated further here.

There is a need to improve the semantic relationships between metadata under the educational metadata category in LOM in order to improve learning object reusability. Therefore, it is necessary to find a semantic definition by describing each metametadata type that would link pedagogical aspects of chosen learning objects.

III. Metametadata Taxonomy

We propose a taxonomy as shown in Fig. 2 for pedagogic metametadata which uses the IEEE LOM metadata specification elements, together with key pedagogic characteristics, and metametadata elements for relational and classification purposes. The distinction between data and metadata is well understood, and metadata models may be described by classes, relationships and properties, known collectively as *types*. Our proposed taxonomy consists of a collection of types of metametadata, analogous to types of metadata, which we refer to as *connector*.

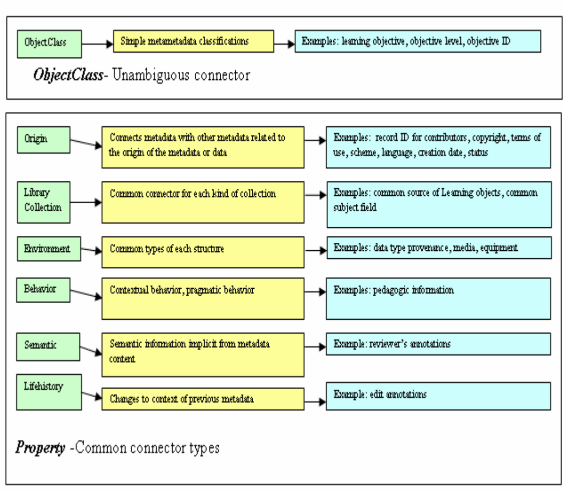

Fig. 2 Metametadata Element Concept (MeMeC)

Fig. 2 shows a proposed *Metametadata Element Concept (MeMeC)* to show the element commonalities that are able to provide an organized structure for interactors, and are by subdivided into two distinct categories, *unambiguous connectors* and *common connectors*.

1. *ObjectClass – Unambiguous connectors*. These are classification metametadata, such as identifications for types of metadata which might be used for cataloguing purposes. There is only one type of unambiguous connector, which we refer to as the *Class* type of metametadata.

2. *Property – Common connectors:* These represent any instances of relationships between selected metadata and other metadata, for example, instances of all classes that may be connected by a generic form interface for displaying object data. We can subdivide these into six generic abstract classes that we refer to as *types* (based on the IEEE LOM educational metadata elements), as shown below.

- *Origin Type*: an attribute of the origin of the records. For example, two documents sharing a common author might use origin metametadata to store that relationship.
- *Library Collection Type*: information about commonality of a group of metadata. For example, the fact that a set of learning objects is sourced from a common repository might be represented by library collection metametadata.
- *Environment Type*: information about commonalities in the administrative or technical metadata. For example, a set of learning objects which share a common type of interface, which could be identified by the authoring tools (as specified in their metadata), would be linked by environment metametadata.

- *Behavior Type*: information about metadata behavior, such as contextual or pragmatic. For example, a set of learning objects which contains metadata indicating the cognitive abilities of the target students might be identified through behavior metametadata.
- *Semantic Type*: information about semantic content of metadata. For example, if a set of learning objects contains metadata which are reviews of each object, then a subset of those objects with positive reviews might be identified through semantic metametadata.
- *Lifehistory Type*: information about changes in metadata. For example, two Learning objects whose metadata had been edited at a similar time might be linked using lifehistory metametadata.

IV. System Framework

The design of architecture, OMESCOD, is shown in Fig. 3. The process and development of Metametadata commences with parsing the data that are the stored learning objects (documents), and metadata from the documents.

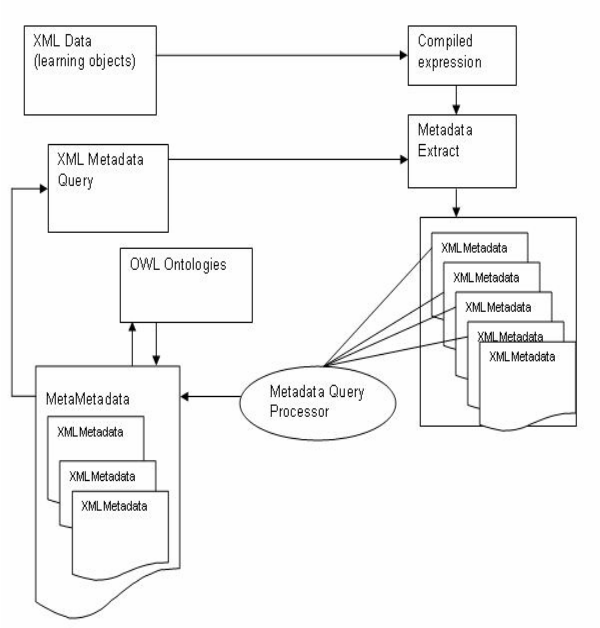

Fig. 3 The OMESCOD architecture work flows

Metadata are stored as XML, and correlate with data elements by matching the attribute ID in the data element, <metaRef> with specified <metaID>. Each instance of the metadata is parsed with a conventional parser in order to get the semantic relationship based on the proposed Metametadata taxonomy.

Each identified relationship within the XML metadata is matched with the ontologies using Protégé-2000[8] as an ontology editing environment used to manage domain models and knowledge-modelling structures with ontologies.

This can be accomplished by firstly, identifying the domain and scope of the ontology by developing an initial small ontology of classes and slots. The classes and the class hierarchy of the can then be defined, followed by the learning objects content (domain) and the properties of classes by describing the internal structure of concepts.

Forms to acquire slot values for the test instance ontology can then be generated. If a particular slot value is the same as an instance of a class, it is defined as a slot default value.

V. Architecture Design and Implementation of OMESCOD

A. *OMESCOD Architecture*

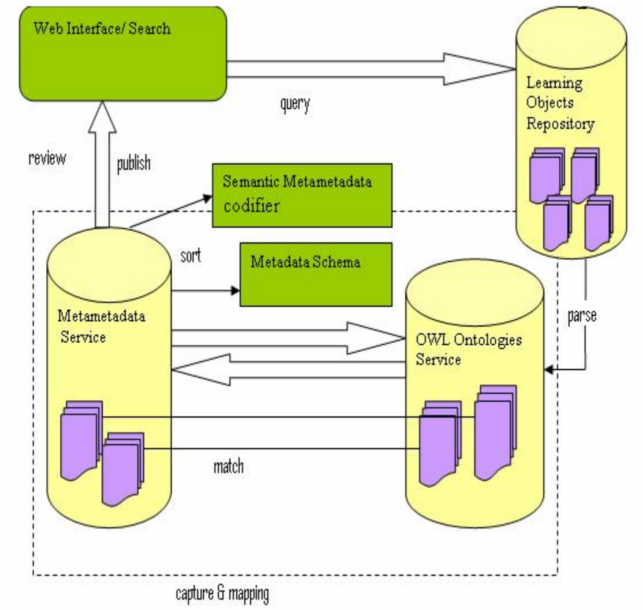

Fig. 4 OMESCOD Architecture

Fig. 4 shows architecture of the system. Firstly, a simple search interface has been designed to which accesses relevant information about the "Introduction to Java Programming" module designs.

Secondly, database tables store all the information about the resources, and the learning objects repository contains stored learning objects related to learning object-oriented programming.

Thirdly, a database supports the metadata extraction process captured in the Metametadata Service using full text searching based on the proposed Metametadata taxonomy. Finally, a database is used for storing and manipulating the ontologies captured and coded using OWL and managed in the OWL ontologies database.

Metadata tagged from the learning objects are categorized based on the IEEE LOM schema, and focus on the educational metadata elements. Resource type elements have been chosen from similar attributes within the educational metadata elements (resource type, difficulty level and interactivity level) to represent the semantic information about the learning objects.

International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:3, No:1, 2009

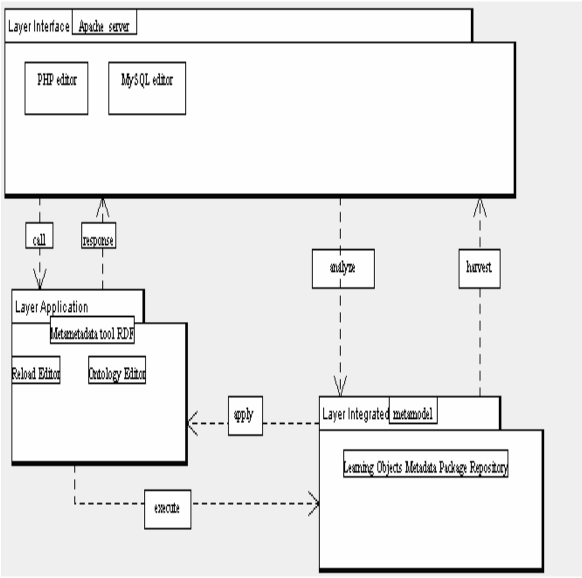

Fig. 5 Overview of the OMESCOD System

Fig. 5 shows an overview of the system. This architecture is designed using a layered approach to preserve the stability of the system if it is modified.

This design supported by Apache as a server to assist the system developments. Learning Objects metadata Package repositories designed to support the activities for analyzing and harvest information from the queried learning objects. The metadata is applied to the Layer application supports by RDF, for extracting metametadata information about metadata. Each metametadata instance will need to be linked to ontologies, support by the OWL language to gather the semantic relationship between multiple metadata.

B. Metametadata Implementation

We present several examples that illustrate the process of extracting semantic relationships, using RDF to represent the metametadata. Resource type is used to represent the query context and related requirements for describing the learning objects based on the Metametadata Taxonomy(refer to Fig.2). The following examples are shown as XML fragments using a simplified RDF syntax.

<MetametaOntoOp:hasContextrdf:resource=”Tutorial” />

This identifyies the resource’s pedagogical attributes. The level of difficultiy contains three values – ‘easy’, ‘medium’, and ‘difficult’ – and is tagged with instance, ‘behavior’.
<MetametaOntoOp:hasBehavior>
<MetametaOntoOp:hasBehavior rdf:tutorial=“easy”/>

Here is an example of XML serialization for the generated queries:

<type:hasLifeHistory
rdf:resource="http://http://www.abc.ac.uk/OoPJava_ontology#10042008"/>
<type:hasLibraryCollection
rdf:resource="http://http://www.abc.ac.uk/OoPJava_ontology# warwick"/>
<type:hasBehavior
rdf:resource="http://http://www.abc.ac.uk/OoPJava_ontology# average"/>
<type:hasObjectClass
rdf:resource="http://http://www.abc.ac.uk/OoPJava_ontology# warwick collection source"/>
<type:hasOrigin
rdf:resource="http://http://www.abc.ac.uk/OoPJava_ontology# Bob Hart"/>

VI. Query Search

A. Semantic Search

We present the semantic search method to evaluate the performance of metametadata and ontology searching by looking into two scenarios to utilize the semantic relationship between tagged metadata based on the Metametadata taxonomy(refer to Fig. 2).

The data set consists of XML documents that are used for querying by using keyword controlled vocabularies. A typical document may be a list of elements stored in specific domains.

Queries can be made through a simple keyword based search form, or can be submitted as SPARQL queries, optionally containing extensions that can specify the degree of confidence required for each term in the query. Keyword based queries are expanded into SPARQL queries, so all searches use the same internal process. The most basic search is for a set of keywords, where the results will list ontologies containing all the keywords. The query can be made more specific by adding search directives to the query.

B. Query Analysis

This test compares how different query engines behave when the same information is placed into different positions in the document. This can give insights into where data should be placed to enhance performance of searching the relevant documents. The possible positions of semantic nodes include *element name* to tag information on title of learning objects, *attribute name* to tag level of difficulties, *attribute value* to tag level of interactivity, and *element value* to tag the semantic meaning of each metadata. Level of queries for this test will situate the semantic information in each location.

<Element_name>
<Name>While Loops</Name>
<Difficulty>Hard</Difficulty>
<Type>tutorial</type>
</Element_Name>

Sample XML fragment for Metametadata

<element_name="Objectoriented_Programming" difficulty_value="easy" interactivity_value="easy"/>
<title>While Loops</title>
<title>Java Preparation for 6.170</title>
<title>Preparing table of a number by using loops</title>
<difficulty_value>easy</difficulty_value>
<interactivity_value>easy</interactivity_value>

```
  <language_value> English</language_value>
<same_type= “tutorial” item “while loops” item Java
Preparation for 6.170 item “Preparing table of a
number by using loops”</same_type>
<description_value= “easy” URL = “http:
www.abc.com/OoProgramming/difficulty.html”</descr
iption_value>
 </element_name>
```

These dynamic controls characterize the common and significant properties of an XML document in the context of query evaluation to obtain the semantic description of each metadata tagged. We also chose dynamic controls that are *fundamental* and do not depend on other dynamic controls. With these control factors, we are able to precisely control the document generation and isolate the impact of an individual factor on query evaluation.

VII. Conclusion

Much recent works in educational technology area are more towards designing framework for adapting metadata while still lacking on the needs to enhanced the pedagogical values for metadata. This paper has focused on the designing and implementation of the novel Metametadata framework as part of solution to retrieve and achieve better relevant result for pedagogic data in computer science domain.

References

[1] ADL. (2004). "Sharable Content Object Reference Model (Scorm)." Accessed on 10th August 2007. Retrieved from http://www.adlnet.org

[2] Berners-Lee, T. (1999). "The Semantic Toolbox: Building Semantics on Top of Xml-Rdf." Accessed on 9th April 2007. Retrieved from http://www.w3.org/DesignIssues/Toolbox.html

[3] Berners-Lee, T., Hendler, J. and Lasila, O. (2001). "The Semantic Web." Scientific American. **284**(5): 35-43.

[4] Garshol, L. M. (2002). "Topic Maps in Content Management: The Rise of the Itms." Accessed on 10th November 2007. Retrieved from http://www.ontopia.net/topicmaps/materials/itms.html

[5] Gilliland-Swetland, A. (2000). "Introduction to Metadata,Pathway to Digital Information." Accessed on 28th May 2007. Retrieved from http://www.getty.edu/research/institute/standards/intrometadata/2_articles/index.html

[6] IEEE. (2002). "Standard for Learning Object Metadata." Accessed on 7th April 2007. Retrieved from http://ltsc.ieee.org/wg12

[7] Ismail, A. and Joy, M. (2007). "A Novel Approach for Metametadata Views in Learning Domains". ED-MEDIA - World Conference on Educational Multimedia, Hypermedia and Telecommunications, Vancouver, AACE.

[8] Protege. (2007). "Protege." Accessed on 21st December 2007. Retrieved from http://protege.stanford.edu/

[9] Taylor,C. “An Introduction to Metadata”.(2003). Accessed on 11th January 2007. Retrieved from http://www.library.uq.edu.au/iad/ctmeta4.html