Database and Metadata Support of a Web-based Multimedia Digital Library for Medical Education

Jianting Zhang [1], Le Gruenwald [1], Chris Candler [2],
Gary McNutt [1], Wei Shung Chung [1]

[1] The University of Oklahoma, School of Computer Science, Norman, OK, 73019
[2] The University of Oklahoma, College of Medicine, Oklahoma City, OK, 73104
Contact author email: ggruenwald@ou.edu, Phone: 1-405-325-3498

Abstract. Digital Libraries play an important role in education; they are beneficial to both instructors and students. Many Web file-based course material systems have been built, but most of them do not have effective management mechanisms. Instead, they often have broken links and do not support metadata. To transfer these course materials from a file repository to an integrated and searchable educational digital library, database and metadata supports are needed. In this paper, we report our design and implementation of a Web-based multimedia digital library for medical education with database and metadata support. Our system is built on top of SQL Server, a relational database management system. It adopts the Instructional Management System (IMS) standard for metadata representation and the National Library of Medicine's Medical Subject Heading controlled vocabulary to remove ambiguity and inconsistency in metadata definitions. It includes a sophisticated search engine for both database content and metadata. It also supports the IMS Content Packaging Specification for learning materials export and exchange.

1 Introduction

Hippocrates [9], the online curriculum website of the College of Medicine at the University of Oklahoma hosts a diverse collection of curricular materials organized by courses. The acquisition of thousands of images, videos, animations, and the construction of numerous web pages has resulted in an immense repository of educational resources occupying 1.5 Gigabytes of space that has grown unwieldy and is difficult to manage. In its current file-based system implementation, it lacks general and domain specific metadata support, which makes it difficult to search for data, share it with other organizations, and add new functionality. It does not allow search on multimedia attributes, medical terminologies and other important information that are not explicitly stored in course material (e.g., intended end user, user hits, etc). Thus, it does not support queries like “Find all heart images for first year medical students that have horizontal and vertical size greater than 200 pixels”.

Supported by a grant from the University of Oklahoma Bioengineering Center (OUBC), we converted this static file repository into a database-based multimedia digital library. In this paper, we present our conversion design and implementation. The rest of this paper is arranged as follows. Section 2 gives a brief overview of related work. Section 3 presents the overall architecture of the system. Section 4 discusses the mapping of web contents and metadata attributes to database schemas. Section 5 discusses metadata based searching and content packaging. Finally Section 6 presents conclusions and future work directions.

2 Related Work

The differences between a web-based system and a digital library are well recognized in [10]. The authors argued that digital libraries can be considered islands of specialized collections on the Web which have their own management policies for publishing and access control. Link consistency and metadata support are the other two factors that distinguish a digital library from a Web-based system. Our solution to these problems is to use database technologies and build our system on top of a RDBMS to ensure data and link consistency, reduce redundancy and provide access and security. We also use RDBMS to store metadata and support metadata queries.

The National SMETE Library program [11] is among the largest efforts in developing digital libraries targeted for educational purposes. A panel session on the NSDL program was held at the first ACM/IEEE-CS Joint Conference on Digital Libraries [1]. Although developed independently, we share most of the design principles for the information architecture of a SMETE education digital library proposed in [2]. There are studies on developing domain specific education digital library systems, such as the Biology [3] and Earth System [4]. However, no detailed design and implementation information is available.

Metadata can be defined as data about data [5] or information about the information [6]. There exist several popular metadata standards, such as the general purposed and simplified Dublin Core metadata [12], MPEG-7 for multimedia [13], XML related metadata standard which includes Document Type Definition (DTD), XML schema and Resource Description Framework (RDF) [14]. In addition, IEEE Learning Object Metadata (LOM) specifications are specifically designed for learning material [15]. The specifications have been adopted by IMS (Instructional Management System) Global Learning Consortium [16], a leader in online distributed learning activities. We adopt IMS metadata content packaging specifications that support XML binding automatically. While our current system only supports limited multimedia attributes, we plan to support MPEG-7 multimedia content description standard with XML binding in the future.

Unified Medical Language System (UMLS) [17] is one of the most prevalent systems in representing medical knowledge; it includes Metathesaurus, Specialist Lexicon and Semantic Network. Many applications using UMLS, ranging from automatic metadata generation to medical knowledge inference, can be found at the UMLS research websites and American Medical Informatics Association (AIMA) annual meetings [18]. Different from other projects that use UMLS, in this system we primarily use only one of the UMLS Metathesaurus called MeSH (Medical Subject Heading, [19]) controlled vocabulary as the source of the taxopath specified in the LOM metadata model [15] to remove ambiguity and inconsistency in defining metadata of medical educational materials. We plan to use other components in UMLS to develop a more semantic-rich system in the future.

We have two primary goals in this study, one is to convert an existing operational file-based repository into an integrated web-enabled educational digital library on top of a mainstream relational database, and the other is to provide metadata services from educational, multimedia and medical perspectives. We will explore these topics in the subsequent sections.

3 Multimedia Medical Educational Digital Library: Proposed Architecture

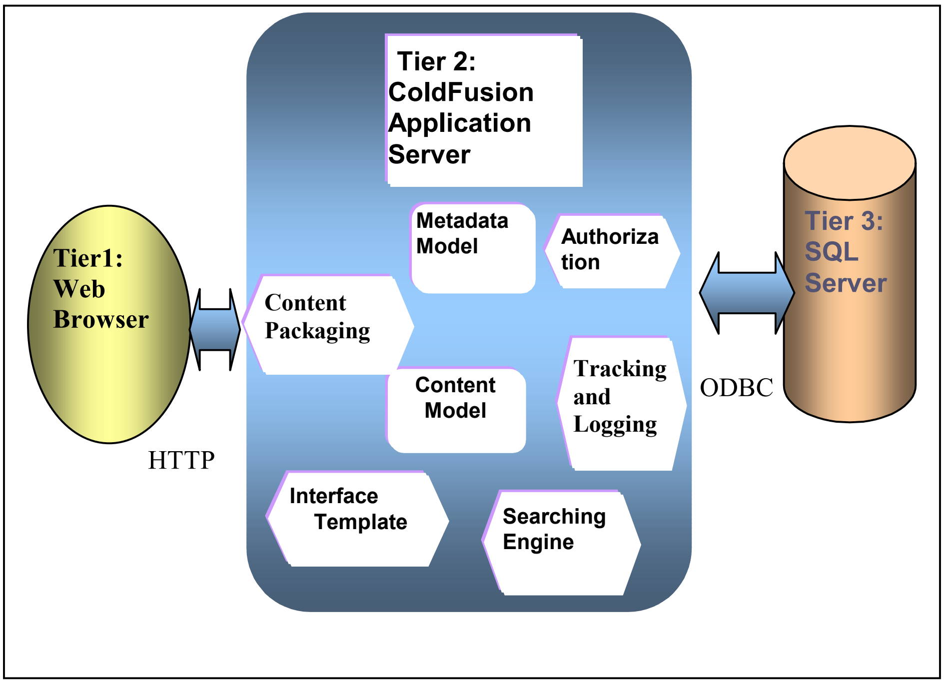

Fig. 1. System Architecture

The proposed architecture shown in Fig. 1 adopts a three-tier architecture based on web-database technologies. The first tier is Web Browser. Although any mainstream Web browser can be used, Microsoft Internet Explore 5.0+ is recommended to reduce compatibility problems. The second tier is the middleware tier or business logic tier and is implemented using ColdFusion Markup Language (CFML). Based on the content model and metadata model that we will explain in detail in Section 4, several modules are developed. The third tier is the database server. Although major DBMS systems can be used in the architecture, we use SQL Server 2000 Standard Edition for its high performance/price ratio.

There are totally five logic components in the proposed architecture based on content and metadata models, namely Authorization, Tracking and Logging, Interface Template, Searching Engine and Content Packaging. Three types of users, Learner, Instructor, and Administrator, are identified. The privileges of a Learner is a subset of the privileges of an Instructor. For example, a Learner is only allowed to view the course content while an Instructor is able to edit the course content and view usage statistics for which he/she is responsible. Similarly, the privileges of an Instructor is a subset of the privileges of an Administrator, An Administrator is responsible for all course material and system usage statistics. The tracking and logging module is designed for data mining purposes to discover learning patterns for future system improvement. The Interface Template module generates Web content dynamically according to the templates an authorized user chosen. All dynamically generated Web pages are context-aware, i.e., different GUI and content will be rendered for different users logging in under different roles. The search engine module divides metadata elements into different logic groups and allows users to combine them arbitrarily. Within each group and for each metadata element in the group, the search engine allows users to specify metadata element specific values or value ranges to query against the metadata database. Finally, the content packaging module extracts explicitly stored metadata elements as well as those that are implicit between tables, and generates a zipped package compatible to IMS Content Packaging Specifications for export and exchange. Since our goal in this paper is on database and metadata related topics, we will focus on the metadata generation, search engine and content packaging modules.

4 Metadata and Database Design

4.1 Conceptual Design of Metadata Management

We define the smallest reusable course material as the Minimum Reusable Learning Object (MRLO). The whole system contains MRLOs through different levels of aggregation. Although the architecture allows an arbitrary number of hierarchical levels, we currently have five levels of learning objects (LO) in our system: Course, Subheading, Resource, Section and Multimedia. Fig. 2 shows a typical example.

```
-Course (Biochemistry)
        -Subheading (Metabolic Pathway Animations)
                -Resource (Interactive Slide Atlas)
                        -Section (Cytology)
                                -Multimedia File (Slide91.jpg)
```

Fig. 2. Hierarchy of Learning Objects

An upper-level LO can have multiple lower-level LOs while a lower-level LO can be used by multiple upper-level LOs. Therefore their relationship is many-to-many. In our application, the classification of Course, Subheading, Resource, and Section objects is more of semantic because the content of LO is refined from an upper level to a lower level. These LOs are all in the form of HTML documents and can be generated using any popular HTML authoring tools by instructors. On the other hand, the classification of multimedia files is more of syntactic, i.e., based on media types. Usually a multimedia file is a sub-LO of a Section LO, while in some cases, a large multimedia file itself could be a standalone Resource LO. All types of LOs have a set of common mandatory metadata attributes and each type of LO has its specific ones. For example, "size" (in kilobytes) is common to all LOs while "number of pages" and "number of words" are specifically designed for text LOs. Fig. 3 shows the LO hierarchy in our system and specific metadata attributes for each type of LOs. Fig. 4 shows a set of common metadata attributes and their relation with related entities, such as contributor, funding agency and most importantly, the MeSH based taxonomy system.

Most of the metadata attributes shown in Fig. 3 and Fig. 4 are self-explanatory. We put Microsoft Word/PowerPoint/Excel, PS/PDF along with HTML as text LO. Text LO has two basic specific metadata attributes, namely number of pages (slides) and number of words. For Image type LO, we have Horizontal Size (HSize) and Vertical Size (VSize) to denote width and height of an image. ColorDepth refers to the number of bits used for a single pixel. For video type LO, we add Number of Frames and Run Time besides HSize and VSize commonly shared with Image type LO. For audio type LO, currently we are only concerned with Sample Rate and Run Time.

Metadata attributes shown in Fig. 3 and Fig. 4 can be classified into two categories, those that can be automatically obtained when the LO is created and those that must be explicitly specified (the later category is bolded in the two figures). Metadata attributes specific to multimedia types of LOs can be retrieved when they are uploaded to the system, such as using Microsoft COM object to retrieve the number of pages/slides/words of Microsoft Word/Excel/PowerPoint [20], Java Advanced Imaging (JAI) package [21] to retrieve HSize/VSize/ColorDepth. Similarly metadata attributes for video/audio can be retrieved using commercial or open source packages. Some of the metadata attributes common to all types of LOs can also be automatically generated by recording system events, such as creation date, last edit date, last access date, number of user hits, format (in terms of MIME type) and physical file size. For the common metadata attributes that has to be explicitly input by users, such as Title, MeSH IDs, we have several GUIs, such as standard Web forms and Java Applets, to help users input metadata attribute values conveniently and efficiently.

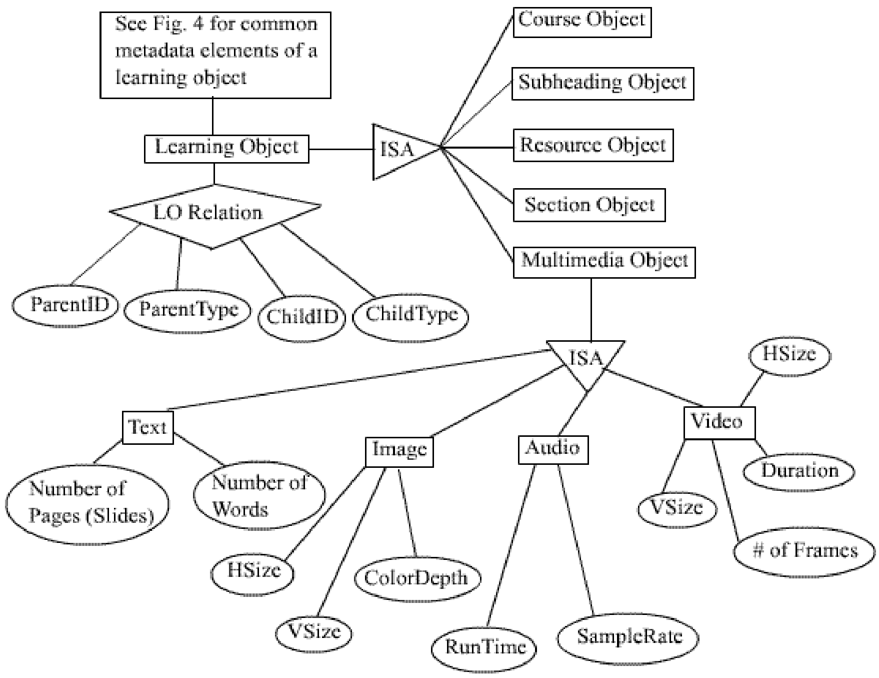

Fig. 3. LO Hierarchy and Special Metadata Attributes Associated with a LO

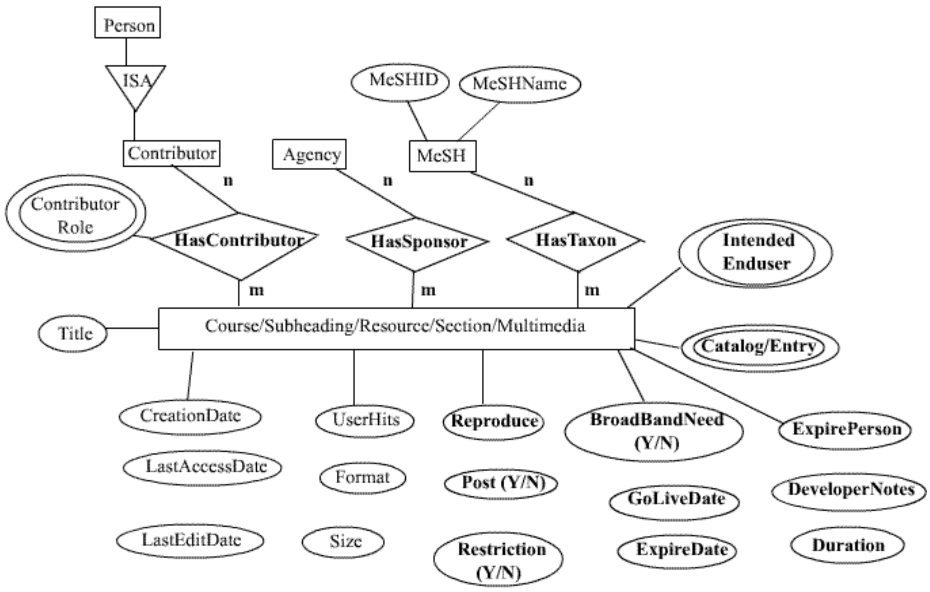

Fig. 4. Scheme for Common Metadata Attributes of a LO

4.2 Mapping Conceptual Design to RDBMS Implementation

There are two major advantages for using a relational database to manipulate metadata over flat-file based metadata (such as HTML page and XML document):

1) Technology maturity: RDBMS provides matured indexing methods and efficient query processing against metadata items while XML-Query is still under discussion [13], not to mention XML-query efficiency.

2) Storage: Metadata stored in RDBMS is in binary format and thus can save space. In addition, some of the relationships which are required to be listed explicitly in an XML document are naturally implicit in RDBMS tables and do not need extra storage.

Learning objects have recursive many-to-many relationships. This reflects the requirements that a higher-level LO has multiple lower-lever LOs while a lower-level LO might be reused in multiple higher-level LOs. We put the metadata elements common to all LOs into one table and metadata elements specific to a particular type of LO into a separate table. We call this paradigm “Vertical Segmentation” of metadata attributes. Tables may be joined together through a common unique ID in both tables. The relationship is recorded in a separate table that has four attributes (ParentID, ParentType, ChildID, ChildType). Given ID and Type we may refer to the corresponding tables to retrieve the metadata elements specific to a LO.

The Vertical Segmentation design has the following advantages:

1) Conceptual Simplicity: In many cases, queries are only on generic metadata elements. Thus we only need to work on the base table and no union operation is needed.

2) Coding Simplicity: we greatly reduce the length of query strings by only querying on the single base table in most cases instead of performing UNION operations on multiple tables.

3) Flexible Extensibility: It allows user to add additional LO types and redefine their relationships and the SQL query code on the base metadata table can be reused without any modification.

Although Video learning objects can be extended from Image type, our design uses the conventional “Horizontal Segmentation” of metadata attributes instead, i.e., all metadata attributes specific to Image type LO are put into one table and all those specific to Video type LO are put into another table. The primary concern is that most queries on Video need to retrieve all metadata elements of Video type LO, including elements shared with Image Type LOs. We would need costly JOIN operations if we use Vertical Segmentation design that put metadata attributes specific to Video Type LO into a separate table. Our decision on whether to have separate tables for base-type LOs and sub-type LOs is based on their degree of association in frequently used queries.

4.3 Mapping RDBMS Table Structure to IEEE/IMS Metadata Specifications

The mapping between IMS/IEEE metadata specifications and our metadata tables is shown in Table 1. It is clear that the mapping is an intersection between IMS metadata specifications and our internal database schema. The reason is twofold. Firstly, not all metadata fields are needed by the specifications. Some of them are for internal use only, such as the number of user hits, GoLiveDate and expiration related attributes. Secondly, not all metadata attributes in the specifications are needed and supported in our application. This mapping will be used for content packaging and export of course materials (Section 5.2).

Table 1. Mapping between IMS/IEEE Metadata Specifications to Database

IMS/IEEE Metadata (No/Name)	RDBMS Table/Data Fields
1.1 Identifier	LearningObject/LOID
1.2 Title	LearningObject/Title
1.5 Description	LearningObject/DeveloperNote
2.2 Status	LearningObject/Post
2.3 Contribute (Role/Entity/Date)	RLOCont/(LOID,userID,ConType,Date)
4.1 Format	LearningObject/sFormat
4.2 Size	LearningObject/nSize
4.4 Requirement	LearningObject/BoardBandNeeded
5.5 intendedenduserrole	LearningObject/IntendedEndUser
5.9 Typicallearningtime	LearnongObject/Duration
6.2 copyrightandotherrestrictions	LearnongObject/(Restriction)
7 Relation	LORelation(ParentID, ChildID)
8Annotation (Person/Date/Description)	Annotation/(LOID, UserID, Date, Description)
9.2.2 taxon	KeyWordMapping/(LOID,MeSHID)
9.4 Keyword	KeyWordMapping/(LOID, Keyword)

5 Metadata and Database Supported Applications

In this section, we show how to build a metadata searching engine and content packaging module for learning object export and exchange.

5.1 Building an Online Metadata Searching Engine

We first divide the searchable metadata attributes into several logical groups as shown in Table 2. The user will first select on the metadata attributes which he/she wants to search in each group. The system then renders the metadata attribute specific criteria for the user to specify. The MeSH keywords will be mapped to MeSH IDs. For the string based attributes, the user will be asked to specify the sub-string they want to be contained in the metadata attributes. For the number and date based attributes, the user will be asked to input a lower bound limit and an upper bound limit. Fig. 6 captures a screenshot illustrating the three steps of the metadata based searching process.

Table 2. Logical Group of Metadata Searchable Attributes

	Group	Attribute Name	Attribute Value Type
Basic	1	MeSH Keyword	Single String, need to map to MeSH IDs
	2	Title DeveloperNote Annotation	Single String
	3	Size UserHit	Range of numbers
	4	CreationDate LastEditDate LastAccessDate	Range of Date
Multimedia	Text (Word/Excel/ PowerPoint)	Number of Pages (Slides)	Range of numbers
		Number of Words	Range of numbers
	Image (GIF/JPEG)	Hsize	Range of numbers
		Vsize	Range of numbers
		ColorDepth	Multiple Selections
	Audio (WAV/AU) (To be implemented)	SampleRate	Multiple Selections
		RunTime	Range of numbers
	Video (Real/ QuickTime/Flash) (To be implemented)	Hsize	Range of numbers
		Vsize	Range of numbers
		RunTime	Range of numbers
		Number of Frames	Range of numbers

5.2 Content Packaging

The IMS Content Packaging Specification provides the functionality to describe and package learning materials, such as an individual course or a collection of courses, into interoperable, distributable packages [22]. It is closely related to the IMS metadata information model [13][23]. In our implementation, when an Instructor/Administrator sends a content packaging request, the system will extract related explicit or implicit information from the database tables and generate a Manifest in the form of Extensible Markup Language (XML). The Manifest together with the related learning object content are added to a Zip file by calling a server-side Java class. The Java class is able to zip a whole directory by recursively calling itself. It can also zip the content of a memory variable by specifying a file name that is associated with the variable.

6. Conclusion and Future Work Directions

In this paper, we summarized our work in transferring a file repository into a multimedia digital library on top of a database system for medical education. We presented our design and implementation of database schemas, metadata management, metadata-based searching engine and IMS compliant content packaging. In our future work, we want to add more semantic-based metadata attributes, such as incorporating UMLS semantic networks, to LOs. We also want to use standardized multimedia description scheme, such as MPEG-7, as multimedia metadata. Finally, as an integrated system, we want to provide better GUI and web-content authoring tools in developing medical education resources to enrich our digital library.

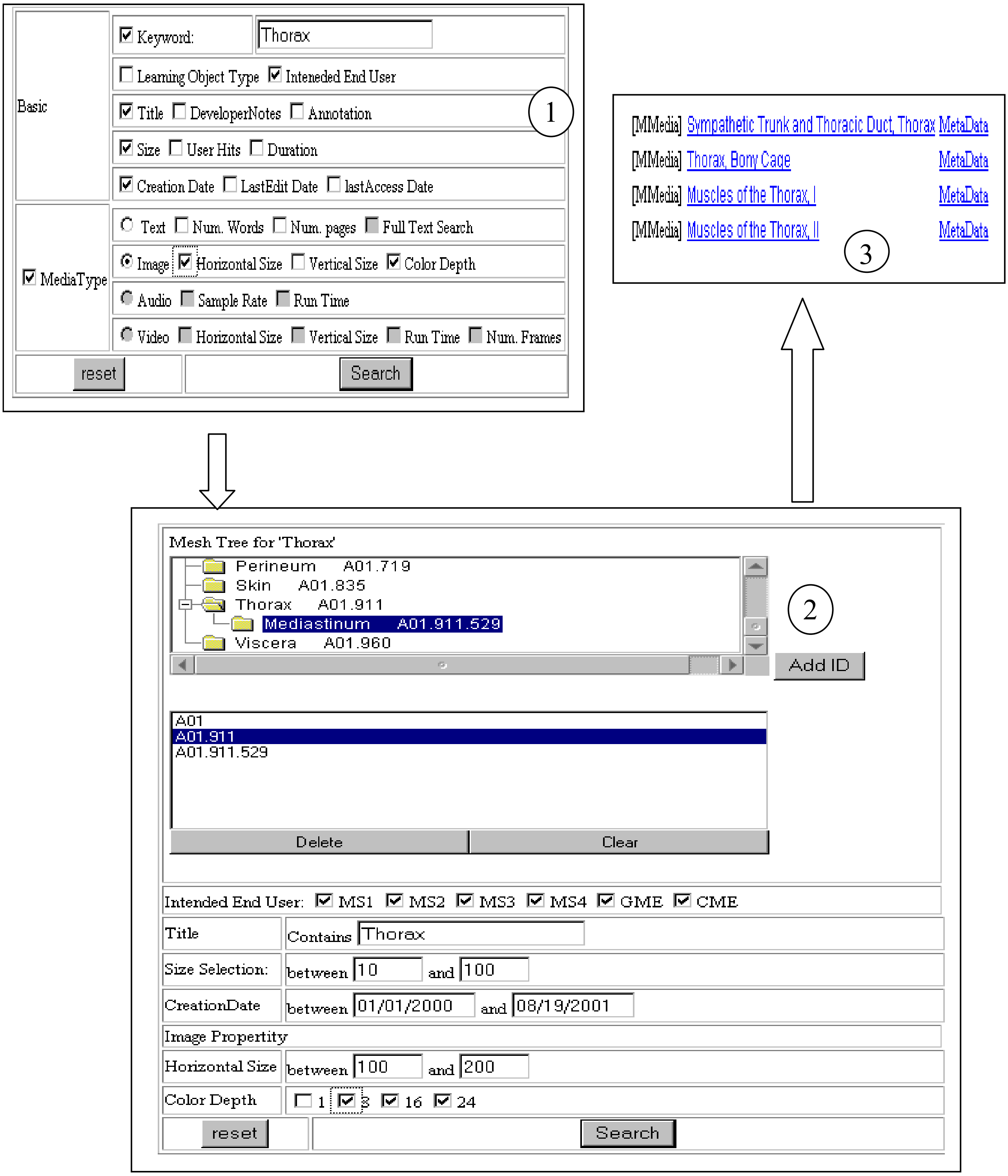

Fig. 6. Steps of the metadata based searching process

References

1. B.Muramatsu, C.A.Manduca, M.Mardis, J.H.Lightbourne, F.P.McMartin, The national SMETE digital library program (panel session), ACM/IEEE JCDL'01, 278-281, 2001
2. A.Dong, A.M.Agogino, Design principles for the information architecture of a SMETE education digital library, ACM/IEEE JCDL'01, 314-321, 2001
3. J.Futrelle, S.Chen, K.C. Chang, NBDL: a CIS framework for NSDL, ACM/IEEE JCDL'01, 124-125, 2001
4. M.R.Marlino, D.W.Fulker,G.Horton, The Digital Library for Earth System Education: implementing the DLESE Community Plan, IEEE IGARSS'00, 46 -48 vol.1, 2001
5. A.T.McCray, M.E. Gallagher, M.A. Flannick, Extending the role of metadata in a digital library system, IEEE Forum on Research and Technology Advances in Digital Libraries, 190 –199, 1999
6. A.Steinacker, A.Ghavam, R.Steinmetz, Metadata standards for web-based resources, IEEE Multimedia, 8(1), 70 –76, 2001
7. F.Cirillo, A.Cozzolino, M.De Santo, M. Marsella, S.Salerno, A metadata based distance learning platform, IEEE Systems, Man, and Cybernetics, 2000, vol. 1, 44 -48
8. S.B.Goschnick, Design and development of Melbourne IT Creator--a system for authoring and management of online education, Technology of Object-Oriented Languages (TOOLS), 187 –201, 1998
9. Hippocrates System, The University of Oklahoma, College of Medicine, http://hippocrates.ouhsc.edu/
10. K.Maly, M.Zubair, S.Shen, S.Zeil, Planning Grant for the Use of Digital Libraries in Undergraduate Learning in Science, http://dlib.cs.odu.edu:8000/documents/descriptionOnly/descriptionOnly.pdf
11. National SMETE Digital Library Program, http://www.smete.org
12. Dublin Core Metadata Initiative (DCMI), Dublin Core Metadata, http://dublincore.org/
13. ISO/IEC JTC1/SC29/WG11, Overview of the MPEG-7 Standard, http://www.cselt.it/mpeg/standards/mpeg-7/mpeg-7.htm
14. W3C, Extensible Markup Language (XML), http://www.w3.org/XML/
15. IEEE Learning Technology Standards Committee (LTSC) wg12, Draft Standard for Learning Object Metadata , http://ltsc.ieee.org/doc/wg12/LOM_WD6-1_1.pdf
16. IMS Global Learning Consortium, http://www.imsproject.org
17. Unified Medical Language System (UMLS), http://www.nlm.nih.gov/research/umls/
18. American Medical Informatics Association (AIMA), http://www.amia.org/
19. Medical Subject Heading, http://www.nlm.nih.gov/mesh/MBrowser.html
20. Microsoft OLE File Property Reader, http://support.microsoft.com/support/kb/articles/q224/3/51.asp
21. Java Advanced Imaging, http://java.sun.com/products/java-media/jai/index.html
22. IMS Global Learning Consortium, IMS Content Packaging Specification, http://www.imsproject.org/content/packaging/index.html
23. IMS Global Learning Consortium, IMS Learning Resource Meta-data Specification, http://www.imsproject.org/metadata/index.html