E-LEARNING CONTENT MANAGEMENT:
AN ONTOLOGY-BASED APPROACH

Nophadol Jekjantuk, Md Maruf Hasan
School of Technology, Shinawatra University
Shinawatra Tower III, 1010 Viphavadi-Rangsit Road, Chatuchak, Bangkok, Thailand
nophadol_bkk@hotmail.com, maruf@shinawatra.ac.th

ABSTRACT
Scarcity of E-Learning content being a barrier for E-Learning is no longer true on today’s Internet. The current concerns are how to effectively annotate and organize available content (both textual and non-textual) to facilitate effective sharing, reusability and customization in an intelligent fashion. In this paper, we explain a component-oriented approach to organize content in an ontology. We also illustrate our 3-Tier E-Learning Content Management Architecture and relevant Web Services and Interfaces. We use a simple yet intuitive example to successfully demonstrate the current working prototype which is capable of compiling personalized course materials on a particular topic (e.g., “Database”) on-the-fly. The prototype uses the Pellet Semantic Web Reasoner as an inference engine to satisfy the constraints and criteria specified by a user (through browser based interface) or an agent (via Web Service API), and retrieves relevant content from the domain ontology in an organized fashion.

KEY WORDS
Learning Objects, E-Learning Content Management, Ontology-driven Inference, Semantic Web Services

1. Introduction
The ubiquity of the Internet and E-learning with their new educational tools and applications are rapidly changing the old way of learning. In the past, we simply distributed E-learning content on the WWW in a semi-structured fashion – with HTML tags and links. In principle, content available on the WWW is accessible ubiquitously. However, in reality, due to the limitations of keyword-oriented hit-or-miss search engines, we often find it hard to locate the desired content. Moreover, the absence of explicit metadata or annotations reflecting the pedagogical facets about these contents (such as content-dependency) often makes it harder to organize the discovered content in a self-sufficient courseware for personalized teaching and learning. Only expert users can go through the content and validate or organize heterogeneous content in a sensible manner. With the advent of Semantic Web technologies, E-Learning content annotated with proper metadata (including pedagogical attributes) and organized in an ontology may effectively facilitate efficient dissemination, discovery and reuse of content in a new way. In this new approach, not only humans but also artificial agents can discover and organize contents from heterogeneous sources and combine them into customized courseware that satisfies specific criteria and constraints stipulated by users or agents. Customized courseware refers to a collection of content (possibly from heterogeneous sources) where the content-related dependencies and pedagogical constraints are preserved.

In this paper, we explain a component-oriented approach to organize E-Learning content in domain ontology. This component oriented approach is inspired by the concept of Learning Objects (LOs) and surrounding technologies [1][2]. We also illustrate our 3-Tier E-Learning Content Management Architecture and relevant Semantic Web Services. We use a simple yet intuitive example to successfully demonstrate our current working prototype which is capable of compiling personalized course materials on “Database” and “Thai Language Learning” on-the-fly. The Web-based prototype is available at http://trf.shinawatra.ac.th/ecms/. The prototype uses a generic Semantic Web Reasoner, Pellet [3] as an inference engine to satisfy constraints and criteria specified by a user (through browser-based interface) or an agent (via Web Services API), and retrieves relevant content from the domain ontology in an organized fashion.

2. Background
In this section, we provide background information on E-Learning, and the concept of Learning Objects (LOs). LOs are the basic foundation of a component-oriented E-Learning system. The E-Learning Content standards are also reviewed. We also review relevant state-of-the-art Semantic Web technologies and their advantages. We also refer to related work as they appear relevant.

2.1 E-Learning Content Issue
E-Learning Content is any digtal resource that we can use to support learning. E-Learning Content can be divided into two categories: textual (text-based content such as, plain-text, PDF, etc.) and non-textual
Textual content can be effectively located by using a keyword search engine, such as Google or Yahoo, producing a number of results. Only human experts can make sense of such a set of retrieved content and organize it into a customized courseware.

With the present state-of-the-art technology, non-textual content is still difficult to locate - even popular search engines, like Google, cannot find them if that content is given irrelevant filenames or surrounded by unassociated textual context. Ontology-based metadata annotations for non-textual content pave the way for locating and organizing non-textual content efficiently.

It should be noted that for both of the above cases (textual and non-textual content), search engines generally produce a set of links (URLs); and each URL subsequently points to other URLs or contents (URIs) in a nested fashion. How much of the retrieved contents are useful largely depends on the user’s expertise and patience to understand and explore the property and nature of the contents.

### 2.2 Learning Object, Metadata and Domain Ontology

The IEEE Learning Technology Standards Committee (LTSC) describes Learning Objects (LO) as “any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning” [4]. Learning objects include multimedia content, textual content, instructional software and software tools, organizations. Each learning object must have a description that enables human and computer agents to search relevant contents efficiently. This means, objects must be wrapped in metadata. As the metadata is machine-readable, it must be possible for a specific system to interpret the metadata from other sources and then reuse the learning objects. However, the metadata has limitations because it is only a wrapper for the search engine to identify one object from another. However, the content interrelationship or dependencies between the LOs can not be fully described by only metadata. Therefore, a domain ontology which conceptualizes a particular domain is essential. In our research, we captured pedagogical attributes and content relationships using metadata and respective domain ontology.

### 2.3 E-Learning Standards

In this section, we will summarize some of the popular E-Learning standards, such as those specified by the Learning Technology Standards Committee (LTSC), the Instructional Management System (IMS), and the Advanced Distributed Learning (ADL).

#### 2.3.1 LTSC

The LTSC was founded by the Institute of Electrical and Electronic Engineers (IEEE) which has developed many technological standards for electrical and information technologies and sciences. The purpose of LTSC is “to develop accredited technical standards, recommended practices, and guides for learning technology” [4].

#### 2.3.2 IMS

The IMS is the most essential organization in the e-Learning community since it is a consortium of distinguished academic, corporate, and government organizations for developing and promoting open specifications to facilitating online distributed learning and to address interoperability issues, such as locating and reusing content, tracking learner improvement and exchanging student profiles between the systems [5].

#### 2.3.3 ADL

The ADL developed the Sharable Courseware Object Reference Model (SCORM) [6]. SCORM is a specification for standardizing the reusability and interoperability of learning content.

SCORM focuses on two critical aspects of learning content interoperability:

1. It defines an aggregation model for packaging learning content.
2. It defines an API for enabling communications between learning content and the system that launches it.

SCORM also divides the world of learning technology into functional components. The key components are

1. Learning Management Systems (LMS)
2. Shareable Content Objects (SCOs)

SCOs are a standardized form of reusable learning object. An LMS is (for the purposes of SCORM) any system that keeps learner information, can launch and communicate with SCOs, and can interpret instructions that tell it which SCO comes next. Additional components in the SCORM model are tools that create SCOs and assemble them into larger units of learning.

#### 2.4 Semantic Web Technologies

Semantic Web is new WWW architecture that supports content with formal semantics. Therefore, the content is suitable for automated systems [7]. Such an architecture will enable automated agents to reason about Web content, and perform intelligent inferences about that content to develop customized courses delivered just in time to the user, according to their preferences and needs.
To achieve these goals we need to express the meaning (in terms of attributes) of the content by using Semantic Web technologies in several layers. The following layers are the basic ones:

- the XML layer, which represents data;
- the RDF layer, which represents the meaning of data;
- the Ontology layer, which represents the formal common agreement about the meaning of data;
- the Logic layer, which enables intelligent reasoning with meaningful data.

The Semantic Web technologies help us to develop systems that gather E-Learning content from diverse sources; process, organize and share content with other humans or artificial agents using ontology. Such an approach makes contents machine-understandable and it becomes possible to develop automated Web services with those heterogeneous contents. Three important technologies for developing the Semantic Web are eXtensible Markup Language (XML), the Resource Description Framework (RDF) and Ontology Web Language (OWL).

2.5 Relate work

In this section, we will refer to E-Learning content management approaches that are relevant to our work. Stojanovic et al. [8] examined the comprehensive benefits of E-Learning by using Semantic Web technology which provides an E-Learning portal architecture that uses ontology to address the description of learning material (content). The form of presentation (context) and the dimension of learning materials (structure) provide flexible and personalized learning materials. Tane et al. [9] developed tools called “Courseware Watchdog” which use ontology to address the different needs of teachers and students in organizing their learning materials. Dolog et al. [10] used personalized service architecture to address the gap between an adaptive educational system and a personalized functionality. It brings personalization to the semantic web to help the user find learning materials, courses or learning paths that are suitable for that user. However, there are still gaps in the semantic relationship between contents and user perception and affordance of those contents since LOs themselves are not capable of capturing individual or group perspectives. In conjunction with addressing those issues, we use domain ontology and declarative query language to represent LOs and their relationships and support navigation at the conceptual E-Learning space. Knight et al. [11] and Amorim et al. [12] use ontologies to facilitate the representation of learning object context and expressiveness limitations found on the current XML-Schema. In addition, Agarwal et al. [13] provide the model of intelligent agents that can make E-Learning efficient since artificial agents tries to make inference across contents. In our research, we focused on a 3-Tier Architecture, where Content Tier (contents and their structures) is clearly separated from the Inference Tier as well as the Interaction Tier so that we can deal with any type of contents and user perspective using a generic inference engine as explained in subsequent Sections (cf. Figure 1).

3. System prototype

In this section, we explain the 3-Tier Content Management Architecture after introducing the tools and techniques we used in developing our prototype. We also give a specific example of how the system works. Although, we restrict ourselves in a smaller domain (i.e. contents of a “Database Book”), our approach is equally effective for general purpose E-Learning content management as long as a domain-ontology is available to annotate contents.

3.1 Protégé, Jena2, RDQL and Pallet Reasoner

We use Protégé and OWL Editor [14] to construct domain ontology for its simplicity and popularity. We use Jena2 APIs and RDQL to interact with a generic Semantic Web Reasoner, Pallet (as explained below) to implement our prototype.

Jena is a Java framework for constructing Semantic Web applications and supports major ontology languages such as RDF/RDFS, DAML+OIL, and OWL (except OWL-Full). In particular, as of writing, Jena2 supports OWL-Lite, some constructs of OWL-DL and OWL-Full such as hasValue and partial unionOf. Some of the significant constructs that are not supported in Jena2 are complementOf and one of [15]. RDQL is a query language for RDF within the Jena framework. The purpose of RDQL is to extract information from RDF graphs. This means that RDQL only retrieves information stored in the model which contains a set of N-Triples statements. RDQL can process ontology in a number of languages including OWL. A typical RDQL query has the following form:

```sql
SELECT ?x
WHERE (?x shortPrefix:localName "value")
USING shortPrefix FOR <URIprefix>
```

?x is a variable. In the WHERE clause, a set of N-Triples define the pattern of a query. The USING clause defines an alias for the prefix of a URIs to simplify the URI. RDQL can also query about predicates or objects too. The limitation of RDQL is that there is no disjunction in the query. Though RDQL is relatively simple in syntax, it is efficient for most of the ontology queries.

Pellet is an open-source pure Java implementation of OWL-DL reasoner. It can be used in combination with both Jena and OWL API libraries and also provides a DIG interface. Pellet API provides functionalities validate and check consistency of ontologies, classify the taxonomy,
check entailments and answer a subset of RDQL queries (known as ABox queries in DL terminology). Pellet is an OWL-DL Reasoner based on the tableaux algorithms developed for expressive Description Logics. It supports the full expressivity OWL-DL including reasoning about nominals (enumerated classes). Therefore, OWL constructs owl:oneOf and owl:hasValue can be used freely. Currently, Pellet is the first and only sound and complete DL reasoner that can handle this expressivity. Pellet ensures soundness and completeness by incorporating the recently developed decision procedure for SHOIQ (the expressivity of OWL-DL plus qualified cardinality restrictions in DL terminology).

3.2 The 3-Tier Content Management Architecture

The overview of the 3-Tier E-Learning Content Management Architecture is shown in Figure 1. The content server stores both the content and its structure in an ontology (contents may be distributes across servers and reachable with hyperlinks). The Generic AI Reasoner engine is in between the content server and user/software agent interface. The user or the agent interact with the content server through the Reasoner by specifying criteria (attribute-value pairs) and constraints; then the Reasoner locates the relevant contents (based on those conditions), and deliver the contents to the user/software agents in an organized fashion.

![Figure 1: The 3-Tier E-Learning Content Management Architecture](image)

**a. The Content Server**

We use OWL to represent and link the content on the E-Learning content server. The content dependencies and other pedagogical attributes of each object (content) are needed to be annotated in advance. Annotation may also be done in a collaborative fashion.

The domain ontology is to represent the domain knowledge. As an illustrative example, in this paper, we annotate the content of a *Database Book* in an ontology. The book consists of 4 modules: *Foundation*, *Applications*, *Systems*, and *Advanced*. The *Foundation* module consists of only 1 Part, *Applications* Module has 2 Parts, the *Systems* module has 3 Parts and the *Advanced* module has only 1 Part. Each Part consists of one or more Chapters. The Chapters across this book have direct and transitive relationships (Figure 2). We capture the content dependency and other attributes (such as number of hours required to deliver the content) in the domain ontology.

![Figure 2: Organization of Database Course](image)

**Figure 2: Organization of Database Course**

```
Module
ModuleName xsd:string
description xsd:string

hasPart inModule:Inverse

Part
PartName xsd:string
description xsd:string

hasChapter inPart:Inverse

Chapter
title xsd:string
hours xsd:int
description xsd:string

hasContent inChapter:Inverse

Content
ContentName xsd:string
url xsd:string
author xsd:string
description xsd:string
```

**Figure 3: Content Schema for the Database Course**

To illustrate the relationships or dependencies in this domain, we use 2 types of relationships, first is the direct
relationship between content such as hasPart, hasChapter and hasContent including its inverse relationship such as inModule, inPart and inChapter, respectively. Second is the indirect or transitive relationship such as prerequisite which means that it has inherited relationships. For example, the content of Chapter 5 is transitively related to contents of all four Chapters from Ch1 to Ch4. The content schema of the content ontology is shown in Figure 3.

b. The Generic AI Reasoner

We integrate Jena2 and Pellet OWL Reasoner and develop a Content Management Module (CMM). The CMM interacts with users or agents and make use of the Generic AI Reasoner (Jena and Pallet) to locate contents (based on user criteria) from the content server.

We use Jena2 and RDQL to query the content from the content server. The inference mechanism in Jena 2 does not support transitive relationships. Therefore, we resort to the Pellet OWL Reasoner as an external Reasoner. Pellet provided a Jena interface which can reduce overhead time instead of using other external Reasoner such as Racer Pro and Fact++. The structure of Generic AI Reasoner along with CMM is explained in Figure 4.

<table>
<thead>
<tr>
<th>API or Web Service</th>
<th>Intelligent Contents Management Module (CMM)</th>
<th>Jena 2</th>
<th>Pellet OWL Reasoner</th>
</tr>
</thead>
</table>

Figure 4: Organization of the Generic AI Reasoner

The system loads the ontology model in OWL from the Content Server. Jena 2 binds the model containing the original ontology. Then an RDF graph-base is ready for RDQL query. The Content Management Module (CMM) generates queries based on user’s or agent’s request. The CMM passes the query to Jena2 to find out the relevant relationship by using the Pellet OWL Reasoner which uses information in the base graph to generate additional entailments from the original set of statements. Finally, Jena 2 returns the content to CMM for necessary processing. The retrieved content that satisfies the query is generated in XML format via Web Service API and presented to the user in an organized fashion. Agent-based access to the content can also be done in similar fashion.

c. The Interface (API/Web Service)

The current prototype provides a Web Browser-based Interface for human users and a Web Service API for artificial agents. The web interface is written in PHP scripts. In this design we allow people to select topics and fill in the number of hour reflecting their time constraint for example. The user is then presented with content compiled in an organized courseware for personalized teaching and learning. The Web Service API defines the methods along with parameters to access content from the content server.

3.3 Operational Details of the Prototype ECMS

Content organized or annotated in ontology (OWL file) on our Content Server in this fashion can easily facilitate intelligent content dissemination to support personalized teaching and learning. In reality, teachers and learners often search specific content to satisfy their personal needs. In the context of our Database Course example, a teacher may already possesses partial content on Foundational database topics but trying to search for Application-oriented content to offer a 20-hour application-focused Database Course to a group of students who do not have any foundation knowledge in databases. Our prototype can accept such constraints and retrieve relevant content from the content server through ontology-driven reasoning as explained in the following subsections and screenshots.

a. The ECMS Server

The ECMS Server runs at a specific TCP Port which users can interact with by using a simple telnet program. The ECMS prototype is a multithread program which can support multiple users simultaneously.

The current prototype uses a 2-Mode approach to retrieve content from the content server. In the First Mode, our CMM retrieves all content that is available in the OWL ontology through direct query matching and generates answers into XML form which user/software agents can easily optimize in terms of the utilization of the content; and in the Second Mode, constraints are further checked and validated with the help of the Generic AI Reasoner using description-logic –based inference.

b. Web-based interface

We have also implemented a Web-based interface for friendly interaction with users. The user can enter the URL of the OWL ontology in an input box and press the GO button. The Web browser then interacts with the ECMS systems via TCP/IP through a specific port. The ECMS will retrieve the collection of content which is available on the OWL ontology which it generates into XML form. Then the Web browser will convert it into HTML form and display it to the user (cf., Figure 5). The user can specify criteria and constraints based on their preferences. Responses from ECMS are generated in XML so that Web Services can be easily implemented for artificial agents in similar manner.
4. Conclusion and future work

In this paper, we explained a component-oriented approach for E-Learning Content Management using Semantic Web technologies. The prototype ECMS can efficiently organize contents for a particular domain in ontology and, therefore, with the help of a generic Semantic Web Reasoner, both users and software-agents can interact with the systems conveniently and can extract E-learning content efficiently. Contents do not necessarily have to be on a single server and annotation of contents can also be done collaboratively using a collaborative annotation too. We used a specific domain (Database Course Contents) and a specific example (a 20-hour Application-oriented database course for students without a Foundation in databases) to demonstrate the essence of our approach in the context of personalized E-learning. Other examples such as Contents for Foreign Language Learning are available online. We plan to apply our approach to broader domains [8][9] such as organizing the content available on a particular site such as the MIT’s Open Course Ware initiative (http://ocw.mit.edu) or through automatic Web crawling [13]. Integration of User Profiles with E-Learning Contents as proposed by Dolog et al. [10] is also an interesting issue to explore under our 3-Tier Framework.

Acknowledgment

This research was supported by Thailand Research Fund (Grant No: MRG4880112) awarded to Dr. M.M. Hasan.

References


