Open km-Learning: Using Semantic Web Technologies to Facilitate E-Learning

Grace Zhao¹, Xiaowen Zhang², Abdullah Uz Tansel³
¹Computer Science, Graduate Center, CUNY, New York, NY, USA
²Computer Science, College of Staten Island, Staten Island, NY, USA
³Computer Information Systems, Baruch College, New York, NY, USA

Abstract—We propose an e-learning framework that takes domain knowledge-specific Open Educational Resources (OERs) from the World Wide Web and stores the metadata of the top qualified entries into an ontology-driven structural data store. When responding to users’ learning queries, the framework constructs a cluster of related subjects around the query topic, including the topic itself, to form a knowledge map, along with the related Learning Objects¹ (LOs). We call this e-learning framework Open km-Learning (OkL), where km stands for “knowledge map.”

Keywords: E-Learning Platform, Semantic Web, Learning Object, Knowledge Map, Ontology

1. INTRODUCTION

Open Learning Objects (LOs) are available on the Internet in many forms, such as Massive Open Online Courses (MOOCs), individual LOs, discussion rooms on given subjects, or simply definitions of particular terms. This paper proposes a novel e-learning platform, Open km-Learning (OkL), to reinforce learning by providing all forms of quality open LOs relevant to a specific topic of interest, as well as a list of related topics. We explain the term ‘related’ in detail in Section 4. Semantic Web (SW) technologies, especially ontologies, make it possible to conceptualize a learning path—a graph of related topics, in other words, a knowledge map.

We will use Information Security² in this paper as an example of domain knowledge to illustrate the framework.

The remainder of this paper is organized as follows: Section 2 gives a brief overview of some related works. Section 3 identifies and describes the e-learning components of the OkL framework. Section 4 discusses the Semantic Web aspects of the platform. In Section 5, we propose our OkL framework and present some of its key features. The last section concludes our paper, addresses challenges, and offers directions for future study.

2. RELATED WORKS

Ever since the inception of the Semantic Web, many researchers have studied the benefits, possibilities, and methods for combining e-learning with SW technologies.

Keenoy et al. strove to use Semantic Web technologies to syndicate and personalize educational resources [5] [6]. The Self E-Learning Networks (SeLeNe), proposed in their work package, carried personalized learning through distributed and autonomous RDF³/RDFS³ creation and management.

Rashid, Khan and Ahmed introduced an ontology-based e-learning management system model [10] addressing course syllabus, teaching methods, learning activities and learning styles. In this model, Administration, Instructor, and Learner are inter-related through Learning Resource and Ontology-based Contextual Knowledge. However, the paper does not provide any Semantic Web implementation details.

Bansel and Chawla [1] proposed an approach to move from content-focused learning services to semantic-aware and personalized learning services. The paper focuses on knowledge representation techniques in semantic e-learning. It also briefly addresses Semantic Web-based intelligent information retrieval method.

In sum, most of the studies within the Semantic Web e-learning literature tend to use SW technologies to tackle the full spectrum of an e-learning system, from pedagogy, assessment, and user profiling to administrative activities. The painstaking efforts demonstrated in the work of Keenoy et al. show a glimpse of the complexities involved in building a semantic e-learning system. In addition, it is difficult to articulate a feasible and salient framework out of the platforms delineated in most of the studies, because they are either too abstract or too specific.

¹A “Learning Object” was defined by Beck [3] as â€œa collection of content items, practice items, and assessment items that are combined based on a single learning objective.â€ In this paper, we define an LO as any self-contained, re-usable, and web-representable unit of learning. An LO could be an hour-long video lecture, a three-webpage tutorial, a podcast, or a full-length research paper in PDF format. The metadata and supporting materials (practice, assessment, copyright, etc.) bound with a learning resource are also part of the LO.

²Information security, sometimes shortened to InfoSec, is the practice of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction.

³The Resource Description Framework (RDF) is a family of World Wide Web Consortium specifications originally designed as a metadata data model.

⁴RDF Schema (RDFS) is a set of classes with certain properties using the RDF extensible knowledge representation language, providing basic elements for the description of ontologies, otherwise called RDF vocabularies, intended to structure RDF resources.
This paper attempts an empirical approach to define a semantic e-learning platform – OkL, to achieve a presentable and feasible route to semantic e-learning.

3. THE E-LEARNING COMPONENTS OF OkL


Since the OkL platform does not manufacture LOs, but rather re-assembles and disseminates LOs that are available on the Web, we will focus our efforts only on technological, management, and interface design perspectives. Technological and interface design involves the technical infrastructure and representation of functional aspects of the system, respectively. The management of e-learning refers to the storage, distribution, and maintenance of learning data and information structure, the focal point of the OkL platform.

4. THE SEMANTIC WEB COMPONENTS OF OkL

In the OkL platform, we adopt the essential elements of SW technologies: ontology, vocabularies, Resource Description Framework (RDF) and Triple Store5.

Ontology is notoriously difficult to design and create, yet it is the ‘yoke’ of a SW platform. It usually requires domain experts to devote a tremendous amount of time and effort to building the initial version, and the maintenance of the ontology is no easier. In an effort to ameliorate the complexities and errors of building an ontology graph, we limit the nodes in an ontology graph to class (rdfs:Class, owl:equivalentClass), subclass (rdfs:subClassOf), and instance (rdf:type) only. As a result, the output graph is a hierarchical acyclic structure. We created an ontology of Information Security Encryption Algorithm (Fig. 1) to demonstrate the hierarchical structure. The root node is the super class, the child nodes are the subclasses, and the leaves are instances.

However, as we increase the scope of a knowledge domain (e.g., domain of Information Security vs. Information Security Encryption Algorithm), more concepts will be introduced. It is challenging, if not impossible, to include each additional concept into the same hierarchical tree. But a more logical and empirically sound approach is first to identify and create sub-trees within each sub-domain of knowledge while preserving the hierarchical knowledge structure. Then, we can selectively link the sub-trees, based on relationship patterns within the specific rule set.

The design guidelines for building an ontology of a knowledge domain can be generalized as follows:

We identify and construct all sub-trees in the knowledge domain. Each sub-tree represents an ontology, containing one root node and at least one subclass or instance. We then examine the relationship among the root nodes of all sub-trees. If there exists a relationship between the root nodes of two subtrees, we use a generic properties entry, :isRelatedTo, to connect the two root nodes. (We will not consider any relationship existing in non-root nodes.) After completing all operations, we should be left with either a tree or a forest.

Based on the ontology structure described above, we are ready to introduce the concept of a knowledge map.

We will map the query topic to one of the nodes in the ontology graph. We call the node the anchor node. Next, we seek the following nodes in relation to the anchor node in the ontology graph:

- **Parent node**: immediate superclass of the anchor node, usually indicating a more general concept.
- **Child node**: immediate subclass/instance of the anchor node, usually indicating a more specific concept.
- **Sibling node**: subclass/instance derived from the same immediate superclass of the anchor node; or a class with ‘:isRelatedTo’ property associated with the anchor node.

We define the above operation more rigorously below:

**Definition 1:** Let \( G \) be a rooted graph whose root is \( rN \). Let \( n \) be a node (anchor node) of \( G \). There may be more than one path from \( n \) to \( rN \). Let \( f_p : G \setminus rN \to G \) be the mapping defined by:

\[
 f_p(n) := \{ x | \text{node adjacent to } n \text{ on the path to } rN \}. \tag{1}
\]

Then \( f_p(n) \) is the parent node set of \( n \).

The **sibling node set** of \( n \) is defined as:

\[
 \Phi_s(n) := \{ x | y \in f_p(n), y \in f_p(x) \}. \tag{2}
\]

Subsequently, the **child node set** of \( n \) will be:

\[
 \Phi_c(n) := \{ x | n \in f_p(x) \}. \tag{3}
\]

Let \( \Phi(n) \) be the set of all nodes in the knowledge map of \( n \); then:

\[
 \Phi(n) := \{ f_p(n) \cup \Phi_s(n) \cup \Phi_c(n) \cup \{ n \} \}. \tag{4}
\]

Another case for the sibling node should be also included in the knowledge map – if two root nodes of two sub-trees are connected with the properties entry ‘:isRelatedTo,’ the two root nodes are siblings of each other.

**Definition 2:** Let \( G = (V, E) \) be an undirected graph, where \( V \) is the vertex set, and \( E \) is the edge set, of \( G \). Let \( G_1, G_2 \) denote two rooted graphs whose roots are \( rN_1 \) and \( rN_2 \), respectively, and \( rN_1, rN_2 \in V, G_1, G_2 \subset G \). If \( (rN_1, rN_2) \in E \), then \( rN_1 \) is the sibling node of \( rN_2 \). Since \( G \) is an undirected graph, it implies \( (rN_2, rN_1) \in E \), thus \( rN_2 \) is the sibling node of \( rN_1 \).

5A Triple Store is a purpose-built database for the storage and retrieval of Resource Description Framework metadata.
However, it should be noted that an anchor node will not necessarily have all three types of node sets, though it will have at least one of the three. Therefore, a minimum knowledge map should contain two nodes: the anchor node and one of the three node types. In addition, an anchor node may appear in more than one sub-trees in the ontology graph. (In this case, we will then construct more than one corresponding knowledge map.)

All nodes in the ontology graph will be defined in the controlled vocabularies. The metadata of each LO will be saved in RDF format in order to be stored into the Triple Store.

5. THE OkL FRAMEWORK

The framework is a logical and functional conceptual model that represents the information flow and linkages between various modules and main processes. Any subsequent system architecture should be derivable from this logical framework.

Fig.3 below illustrates the landscape of the OkL platform. We briefly explain each step delineated in the diagram, so as to present the entire workflow of the e-learning conceptual system.

Fig. 1: Ontology of Information Security Encryption Algorithm.

Fig. 2: Illustrations of the anchor node’s related nodes.
5.1 Building the Knowledge Base

Step 1: Crawling for LOs from the Web

In this step, a crawler is to collect OERs through top cross-domain knowledge search engines, such as Google, YouTube, etc, and authoritative domain knowledge-specific sites. We collect only $\omega$ number of top search results ($\omega$ is a small integer parameter) for further processing.

Prior condition: The query subject should match one of the nodes in the ontology graph.

Step 2: Parsing and processing metadata

After obtaining the LO URLs, we retrieve the related pages and extract the metadata of the LOs, including generic web metadata, custom metadata, and possible SW metadata.

Step 3: Filtering and re-ranking LOs

In this step, the Intelligence Engine kicks in. Based on the metadata of each LO, the engine decides whether the LO is relevant to the query subject and to what degree the LO meets other criteria (if any), such as a rating, length, and complexity.

The related algorithms of filtering and re-ranking LOs are outside the scope of this paper.

Step 4: Creating RDF entries

In this step, we create an RDF file for each LO, based on the predefined LO RDF template.

Step 5: Storing LO RDF in the Triple Store

We use the Triple Store API to store the LO RDF into the graph database.

Steps 1 – 5 will be a recurring process running in the background on a repeated schedule, as a means to update LO records and to ensure the validity of LO URLs.

5.2 Semantic Web Maintenance

Step 1: Maintaining the accuracy and currency of ontologies and vocabularies

This step can be managed through machine learning techniques and user contributions.

![Fig. 3: OkL platform conceptual design block diagram.](image-url)
5.3 Querying the Knowledge Base

Step 1: Ontology node mapping
After the user or the crawler issues a query, we map the query string to an ontology node.

Step 2: Parsing ontology to obtain the ‘knowledge map’ node set
In this step, we will build the mini knowledge map around the query string.

Step 3: LO retrieval
In this step, we query the Triple Store using SPARQL and get the LO result set.

Step 4: Generating Interface
We construct the knowledge map and display related LO links.

For example, based on our Information Security Encryption Algorithm Ontology (Fig. 1), if a user searches for ‘Symmetric Algorithm,’ the application will display a list of LOs concerning ‘Symmetric Algorithm.’ It will also provide the learner with a diagram of concepts related to ‘Symmetric Algorithm’: the more generic concept related to ‘Symmetric Algorithm’ is ‘EncryptionAlgorithm,’ while the more specific concepts are ‘BlockCipher’ and ‘StreamCipher.’ The correlated or parallel concepts are those sibling class/instance nodes of ‘SymmetricAlgorithm’ in the ontology, namely, ‘AsymmetricAlgorithm.’ The interface is illustrated in Fig. 4.

6. CONCLUSION
The key elements of the OKL platform are the domain knowledge ontologies and the knowledge maps deduced from the ontology graphs. We aim to construct these elements in an efficient and effective way and to provide the web learners with an enhanced learning experience.

There are many challenges ahead, such as designing the system at a more granular level in terms of defining the re-ranking and recommendation algorithms, ontology adjustment/updating algorithms, etc.

We hope the OKL framework can help to streamline the practice of applying Semantic Web Technologies into e-learning platforms.

References