Using Domain Ontology for Learning Objects

Yasemin Gültepe
Department of Computer Engineering, Kastamonu University, Kastamonu, Turkey

Lyazzat Balgabayeva
Turan University, Almaty, Kazakhstan

ABSTRACT
Learning objects are one of the innovations in domain of teaching technology. Shared and reusable learning objects are based on the understanding that content should be designed in a modular way rather than designed as a single integrated software. Learning objects are separate and independent objects, allowing each individual to create lessons in accordance with his or her own learning style or method, enabling the realization of customized learning. Since learning objects have descriptive information (metadata), they can be easily searched, so that they can access the learning content on time. The metadata definitions are made in the XML file. For this reason, learning objects can be shared with the use of XML, while the reuse of learning objects cannot be provided because XML is insufficient in the semantic definition of learning objects. Semantic Web (Web 3.0) technologies can be used to produce workable and interpretable web pages. Ontologies are being developed with the use of these technologies. Thanks to the ontology, intelligent learning environments can be developed to access the learning objects that are distributed on the web about each learning acquisition. In this study, according to the curriculum of Computer Engineering, ComputerEngineeringCurriculum learning object ontology was defined. The defined learning object ontology provides better sharing and reusability of learning objects. Protégé ontology development editor was used for this purpose. This paper shows that learning environments developed using semantic web technologies and ontologies can offer intelligent solutions for individualized instruction and rapid access to accurate instructional content on the web.

Keywords
Learning objects, ontology, semantic web.

1. INTRODUCTION
The e-content, which is divided into small pieces that can be reused in various learning environments, is called Learning Objects (LOs). It is one of the innovations in teaching technology. Learning objects will fulfill both instant learning needs and current and future non-learner learning needs [1]. It is based on the understanding that the contents are designed in a modular fashion instead of being designed as single software.

When learning objects are developed and used correctly, they can provide considerable contributions to both creators and users. In order to be able to use learning objects for various purposes or environments, it is necessary that these objects are surrounded by information describing or describing them. This is done with metadata called “learning object metadata” or “information about the data”. Metadata descriptions are recommended because of search and reusability enhancements.

There are many metadata standards published for various purposes. These standards have been developed to define and provide access to specific types of information sources. Examples of such standards include Learning Object Metadata (LOM), Dublin Core Metadata Initiative (DCMI), IEEE Learning Technology Standards Committee (LTSC), IMS (Instructional Management System) Global Learning Consortium Learning Resource Metadata, and Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIAIDNE) [2].

The LOM metadata elements contain a total of 76 tags listed under nine headers. We can list them as follows: General, Lifecycle, Meta-metadata, Technical, Educational, Rights, Relation, Annotation and Classification.

LOM provides metadata in XML (Extensible Markup Language) format. XML, both as a language and as a technology, is used to format, define, and construct a data set. XML offers computer-readable and automatically processable document structures that can be seamlessly transferred across the platform and system independent on the web. XML only describes the structure of the document, but does not deal with its interpretation by machines. However, it does not contain information on what it means semantically. For this reason, LOM, which is used for the needs of the users in the field of education, is insufficient in the interoperable ground which removes the problem of data inconsistency which is experienced in using the related learning objects. In addition, because of the keywords that are not semantically related but used in the search process, many web sites can be listed in the bottom line within the same search result. Because of the lack of text-based search engines, access to learning objects is not possible at the desired level. Semantic Web technologies can be used to produce workable and interpretable web pages. Ontologies are being developed with the use of these technologies. Through ontologies, it will be possible to establish semantic and logical connections between concepts beyond classification.

The basic idea in the semantic web is to be able to provide communication between metadata models [3]. Metadata can only contribute to the semantic web at the point where it is shared with others; otherwise the metadata has no meaning beyond defining the local associations on the available data.

In this study, ontology of software engineering course in computer engineering syllabus has been realized. Thus, learning objects, which are the smallest components from the top level concepts, are defined as ontology based. Ontology is the most basic component of the semantic web. An area on the web offers common words and meanings by making the canonical definition of objects to needy people who want to achieve shared information. Ontologies have different roles in Software Engineering. Basically, this role can be listed as follows: [4]

- Ontologies for the Software Engineering process: Modularization, distribution, reuse and integration.
of software components and systems is an important aspect of software engineering. Ontology can be defined as a dictionary containing the conceptual meanings of the processes in this area and the relations between them.

- Ontologies in Software Engineering Domain: It foresees use of ontologies at the realization time and improvement of the software development process. In this category, ontologies are defined to represent the target of a specific sub-domain. On a larger scale, domain ontology will help researchers and scientists develop a universal ontology. In addition, information in any domain is provided to be conceptualized.

In this study, according to the curriculum of Computer Engineering, Computer Engineering Curriculum learning object ontology was defined. The second chapter of the study provides information about the related studies in the literature. In the third chapter, semantic web and ontology, ontology tool, ontologies for learning objects are explained. In the fourth chapter, ontology based learning object definitions are presented for software engineering course. In the fifth chapter, an evaluation is made about the results.

2. RELATED WORK

Semantic web languages are used to identify objects in the semantic web environment. RDF (Resource Description Framework) is a data model for the semantic web. The object, property, value triples in the RDF expression forms the basis of RDF. The LOM’s RDF-based display is interactive at the level of knowledge-based content to ensure interoperability. The semantic web can produce better quality learning object metadata for learning objects.

Ontologies aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of domain, which may be reused and shared across applications and groups [3]. Ontologies can be used by various applications in different areas. One of the areas where ontologies can be used is education. Many ontologies [5, 6, 7, 8] related to learning objects are being developed in the field of education, each day creating a wider use area for themselves.

In 2013, Casali et al. (2013), LOM ontology was designed based on the LOM structure [9]. This ontology designed provides a common dictionary to share information with the user. Even if the metadata model on the ontology is changed, the greatest advantage of all applications that use this ontology is that the behavior of applications can be changed without changing the program code.

In order for traditional LOMs to have inference and reasoning capabilities, MSLF (Multi-layered Semantic LOM Framework) can be used as a sub-model of the semantic web-layer structure [10]. Under this framework; The URI layer (Learning Objects), the XML layer (LOM), the ontology layer (OWL based), and the rule layer (RuleML based) are combined. In order to make logical reasoning on LOM-based metadata documents, rule language and query language are used to perform information extraction. As an example of the use inferences in software engineering ontology: (a,r,b)(responsible(a,r)/Ause(r,b)—performs(a,b)). This axiom says that if an agent a responsible a resource r, and r uses activity b, then a also performs b.

There are many ontology-based applications provided by information integration to achieve interoperability. In the field of education, different studies have been carried out with the aim of ensuring interoperability through the information about the courses. Gas’evic, et al. (2001) have emphasized ontology based structuring of a LO and semantic annotation of both the LO as a whole and its components [5]. Different ontologies are defined to provide reusability of LOs: Metadata Ontology (MO), Content Structure Ontology (CSO), and Content or Domain Ontology (DO). The MO can be used to show all kinds of metadata schemas, while the CSO generates semantic labels for LOs. DO is designed to be an RDF source of LOs content. In Figure 1 there are ontologies designed to create a common structure that allows LO’s to be shared and reusable among different components.

![Fig 1: Ontologies describing learning objects](image)

The ability of ontologies to divide into multiple ontologies or combine multiple ontologies into ontology facilitates the maintenance of these structures. In general, different types of ontologies such as content, context, structure can be combined to define a course in an e-learning environment. In context ontology, domain specific terms, objects on the pedagogical model in content ontology, relationships between corresponding rules and learning materials in structure ontology are defined.

In an e-learning environment, there is a high risk that two authors will express the same subject in different ways. This means that semantic identical concepts (i.e., e-learning-content issues) can be expressed in different terms from the field vocabulary. The problem can be solved by incorporating a field dictionary into the ontology and thus mapping from the terms of field knowledge to the meanings defined by the ontology concepts. [11].

In 2014, Kaya and Altun (2014) have proposed ontology based learner model for e-learning systems that use instructional learning objects [12]. A learner model consists of metadata consisting of instructional decisions about a learner. The learner model will be able to relate to other learning ontologies such as learning model ontology and domain ontology.

The most important issue for success is the completion of performance graphics beyond comparative testing. In a data acquisition system, the most useful performance metrics for most work are precision, recall and average precision. Farhat and Jebali [13] have used two information extraction metrics precision and recall in first step. After they have used ontology based information extraction metrics called augmented precision and augmented recall. Experiment results show that a significant number of learning objects are needed to increase sensitivity. Indeed, extraction of semantic metadata depends on many factors. The quality of the information contained in the documents depends on the consistency of the second. It also depends on the consistency between documents and used ontology [13].
3. ONTOLOGY-BASED LEARNING OBJECTS

3.1 Semantic Web and Ontologies
The Semantic Web [14, 15] is a web environment that enables well-defined information and services to be easily understood by machines. The Semantic Web, however, is an extension of the existing web that provides semantic information that can be used by collaborators and people. The fundamental building block of the Semantic Web is ontologies. Theoretically, ontology defined as "a formal and clear specification of a shared conceptualization", consists in coding the concepts in a particular area, their properties and their relations to each other in such a way that the machine can understand [16].

Ontologies have begun to be used in many working area. Software engineering is also one of the areas where ontologies are used. Ontologies are used in software engineering for different forms and different purposes. Generally, ontologies are widely used in course content and evaluation designs in training programs on software engineering.

3.2 The Ontology Tool
The ontologies defined in this work were developed using the Protégé ontology development editor. The ontology language used for ontology designing and making real is OWL (Ontology Web Language). The reason for using OWL to develop ontology is that this language is accepted as the standard by W3C (World Wide Web Consortium).

Ontologies are visually defined through the graphical interface of the Protégé ontology development editor, so that the area to be defined can be modeled. The editor also facilitates the development of ontologies and provides great convenience in ontology development by reducing the possibility of errors. Protégé can work with many auxiliary semantic web tools and technologies developed for many processes such as ontology development, querying and deduction.

3.3 Ontologies for Learning Objects
The basic idea of the learning object concept is the division of small pieces that can be reused in various learning environments related to education. E-content that is divided into small pieces that can be reused in various learning environments is called learning object. It is important that the learning objects are designed in the smallest possible size. This reason is that they can be reused without changing the learning objects or with minor changes and they can be formed in different forms with different applications [1, 10]. Thus, both the effectiveness and productivity of instructional design process and the flexibility of learning objects will also increase [17].

A learning object consists of two parts: data and metadata. Data is the content component represented by the learning object itself and can be large and complex structures from a picture file, a video image, a piece of music or text to a whole lesson. Metadata is a definition that defines the data and allows it to be understood and classified by the people. Metadata can be defined as the "tag" of the data [18]. Metadata is used to access the source.

The purpose of this study is; implement an ontology model for undergraduate level software engineering course as e-learning application. Ontology and learning objects developed for this purpose are also expressed ontologically. Ideally, learning objects should be the smallest possible while maintaining a large number of semantic connections to other objects.

Each part of the ontologies is divided into the smallest units indexable for the purpose of searching and reusing learning material. IEEE LOM can be used as the metadata standard. The educational metadata category is like the metadata category that distinguishes the LOM standard from other standards. In this category there are statements that describe educational or pedagogical characteristics of the object. These are respectively; Interactivity Type, Learning Resource Type, Interactivity Level, Semantic Density, Intended End User Role, Context, Typical Age Range, Difficulty, Typical Learning Time, Description, Language, etc. The metadata of the learning object is information that describes, explains, locates or facilitates the management of a particular learning object.

A top-down approach has been used to determine class hierarchies in software engineering course ontology. This approach aims at conceptualizing by field experts. This approach defines meta-abstract concepts and then develops learning objects as small pieces of hierarchy.

Figure 2 shows the ontology hierarchy that starts with the Computer Engineering Curriculum and is developed using the top-down approach that ends with the learning object. The term curriculum refers to courses, lessons and academic content taught in a school or a specific course or program. Figure 2 shows Computer Engineering Curriculum as a curriculum, Software Engineering Course as a course, Software Life Cycle Models Lessons as lesson and Waterfall Model Topic as a topic.

In the context of the developed ontology; Table 1 summarizes the metadata that can be used for Curriculum, Course, Lesson, Topic and Learning Object shown in Figure 2 [7]. Metadata related to the concepts of software engineering course are created semantically using semantic web technologies and learning objects can be easily searched and found based on this upper data, thus providing the opportunity to access the learning contents in a timely manner.

Fig 2: Curriculum Ontology Hierarchy
The development of ontologies for ontology-based learning objects is very important. For this reason, ontology development processes are needed to facilitate this process, based on the ontology development editor and an ontology development methodology that meets the needs of the system. In this study, based on the ontology development methodology of Noy and McGuiness (2000) study, the ontology development processes meeting the system requirements were realized [19].

The curriculum contains detailed topics for each lesson and lessons of each course. Learning objects belonging to course subjects are at the bottom of the hierarchy according to the curriculum direction. As in many engineering fields, current course subjects and concepts are also important in the field of Software Engineering.

In this study, curriculum for ontology-based e-learning system, lessons in this curriculum, weekly lesson topics (course flow) according to lessons, and learning objects belonging to these lessons are defined as ontology based. Thus, ontology-based curriculum management is provided in a semantic integrity.

Software engineering is a knowledge domain in computer science instruction manuals supported by ACM and IEEE Computer Society [15]. This information area contains the following sub-headings: 1) software life cycle 2) design methods 3) software testing 4) performance evaluation 5) software development management [ACM/IEEE-CS Joint Task Force on Computing Curricula]. Each topic is subdivided so that the smallest units can implement the ontology model, which is defined as learning objects. And then, when necessary, the sub-topics are divided into learning objects. For each of these concepts, ontology objects will be defined separately and discussed in the subsections of this section.

### 4. THE DESIGN OF ONTOLOGY-BASED LEARNING OBJECT FOR SOFTWARE ENGINEERING COURSE

The aim of curriculum ontology is to provide a full representation of the disciplines’ knowledge areas for educators, curriculum designers, training fields and students. Thus, curriculum ontology can be used to support curriculum development through curriculum matching, interactivity, and classification between related disciplines. In addition, the curriculum covers a number of ontologies that are linked to the course classes defined in ontology.

In this study, according to the curriculum of Computer Engineering, Computer EngineeringCurriculum ontology was developed. On the ComputerEngineeringCurriculum ontology, there is a list of the curriculum examples such as the course data, the basic concepts of the courses in the curriculum, the goals of the course and the knowledge required to be gained from this curriculum. Thus, ontology-based curriculum management is provided in a semantic integrity.

#### 4.2 Ontology Classes and Class Hierarchy

The default class hierarchy view is limited to showing is-a or sub/super class relations. Some of the subclasses and classes in the ComputerEngineeringCurriculum are shown in Figure 3.

![Fig 3: Partial ComputerEngineeringCurriculum Ontology](image)

The ComputerEngineeringCurriculum ontology is divided into three sub-classes. When defining an ComputerEngineeringCurriculum ontology, first a classification is made on the basis of a hierarchical structure. Subclasses belonging to this class have also been defined to illustrate the different uses of this class. System ontology started with ‘Thing’ class. These classes can be listed as follows: Lessons, Topics and Courses.

The basic class of ComputerEngineeringCurriculum is the Courses class. In this class, metadata about the lessons, the relations with the other classes, the representational properties, the concepts related to the lessons are defined. Courses class is defined according to the lesson plan. Lessons class is used to define the course flow. Subjects under course flow are represented by Topics subclass.

In this study, applications of Software Engineering course were examined. Figure 4 shows the SoftwareEngineeringCourse class as another class in the ComputerEngineeringCurriculum ontology. The sub class relation (dfs:subClassOf property) of the Courses class in the ComputerEngineeringCurriculum ontology is used. The SoftwareEngineeringCourse class consists of seven subclasses. These are; LearningMaterial, TeachingMethod, LearningMethod, LearningObjective, Assignment, ExamQuiz and Concepts. One of the goals of the concept SoftwareEngineeringCourse is to briefly described Course Information in Bologna Process. The learning objects of SoftwareEngineeringCourse concept are LearningMaterial, TeachingMethod, LearningMethod, LearningObjective, Assignment, ExamQuiz.

There are software development models or methodologies.
Software development teams determine the most appropriate development model for their projects and create their goals. In order for learning objects to be reused at a higher level, some of the topics mentioned are divided into smaller units.

Fig 4: Graphical representation of SoftwareEngineeringCourse class

The concepts that will be used when creating weekly lessons are the lower classes created under the Lessons class. Lessons class is used to define the course flow. This class also lists the lessons and the lower lessons of the other courses in the "Courses" sibling category. In this study, weekly course subjects of software engineering course are defined. One semester is at least fourteen weeks. However, the five-week section is shown in this study.

The Lessons class is shown in Figure 5. The Lessons class consists of five sub-classes: PerformanceEvaluationLesson, SoftwareDevelopmentManagementLesson, SoftwareLifeCycleLesson, DesignMethodsLesson and SoftwareTestingLesson. The learning objects of Lessons concept are PerformanceEvaluationLesson, SoftwareDevelopmentManagementLesson, DesignMethodsLesson and SoftwareTestingLesson.

The software life cycle takes care of the software life cycle, which is exactly what happens when the software is developed. Among the topics covered are Software Engineering, structured software development techniques and software project management issues. The learning objects of SoftwareLifeCycleModelsTopic concept are SoftwareLifeCycleModelsTopic, SoftwareQualityCost, Contents, BasicConcepts and SoftwareEngineeringScopes.

Fig 5: Graphical representation of Lessons class

The Topics class will be detailed on specific topics shown in the weekly course streams defined in the Lessons class. The concepts in each Lessons class can cover several topics.

In Figure 6 the Topics class is shown. The Topics class consists of two sub-classes: SoftwareLifeCycleModelsTopic and SoftwareDevelopmentLifeCycleModelPhasesTopic.

Fig 6: Graphical representation of Topics class

The software life cycle models describe the phases of the software cycle, and the order of execution of these phases. Each stage produces output that the next stage in the life cycle requires.

The basic goal of the SoftwareLifeCycleModelsTopic concept is to define the engineering processes that must be tracked during the software development lifecycle for project success. Software Life Cycle Models are divided into traditional and agile models. In this study, we merged two models. These models are defined as sub-classes. The SoftwareLifeCycleModelsTopic concept is shown in Figure 7. Subclass of the SoftwareLifeCycleModelsTopic class include IncrementalModel, XP, ScrumModel, RAD, SpiralModel, WaterfallModel and PrototypingModel, and represent sub-topics of these classes as smaller components.

Fig 7: Graphical representation of SoftwareLifeCycleModelsTopic class

SoftwareDevelopmentLifeCycleModelPhasesTopic class is shown in Figure 8. SoftwareTesting, SoftwareDesign, are subclasses which are further granulated into subclasses, while SoftwarePlanning, SoftwareImplementation, SoftwareAnalysis, SoftwareDevelopment, and SoftwareMaintenance is not further granulated and these are represented as a learning object.

Fig 8: Graphical representation of SoftwareDevelopmentLifeCycleModelPhasesTopic class

A software lifecycle model is a standardised format for planning, organising and running a new development project. Many different models are known and used.
The SoftwareLifeCycleModelsTopic class is shown in Figure 9. The SoftwareLifeCycleModelsTopic is granulated in seven subtopics and twenty three learning objects in total. The learning objects belonging to the Software Engineering course include the necessary information in the hierarchy. The Software Engineering course content consists of learning objects. As a whole, the content can be represented by a hierarchy of learning objects.

**5. CONCLUSION**

The effective use of learning objects is that these objects can be used repeatedly. Since learning objects have descriptive information, they can be easily searched and found so that they have access to learning contents in a timely manner. This work models information concepts and relationships with an ontology based approach to describe information content.

Modeling languages such as UML used in software engineering are designed according to the closed world approach. The ontological languages used to develop ontology adapt the open world approach. In addition, the use of existing ontologies with alteration or expansion is one of the methods that can be used to reduce the costs of software development activities.

Ontology developed in this study describes the relationship between the Software Engineering course field and course flows. Representation of software engineering area models with ontologies, provides integration of ontology modeling languages with other model languages. Ontology based development envisages the use of ontologies in the software development process and in the process of improving these processes.

In this study, according to the curriculum of Computer Engineering, Computer EngineeringCurriculum ontology was developed. Computer EngineeringCurriculum ontology has been developed so that learning objects can be identified. Metadata definitions of learning objects have been made using terminology in this ontology.

ComputerEngineeringCurriculum ontology is evaluated as an ontology that has a deep taxonomy, rich relationships, a comprehensive modeling of concepts, adequate individual of class and concepts with close meaning to each other. Thus, ontology-based curriculum management is provided in a semantic integrity. Finally, it is expected that learning objects will create an environment to facilitate e-learning and will benefit for this field.

**6. REFERENCES**


