Learning Objectives Feature for the Instructional Module Development System

Ketaki Andhare, Odesma Dalrymple, Srividya Bansal
Department of Engineering
Arizona State University

Abstract

The road to effective science, technology, engineering and mathematics (STEM) instruction starts with a well-conceived and constructed plan or curriculum. STEM educators, who typically come from STEM backgrounds and have little or no STEM education training, can benefit from the use of an information technology (IT) tool that guides them through the complex task of designing an instructional module (i-mod), i.e., a single course that can span over a specified duration of time. The next generation of the World Wide Web (WWW), called the Semantic Web, promises to further improve productivity by providing meaning and intelligence to the vast data currently present on the web. The Instructional Module Development (IMoD) system, currently under development, presents a web-based framework for representing an i-mod and scaffolds users through the design process. This framework uses Semantic Web technologies to automate aspects of the complex decision making processes that are needed for designing an i-mod. One of the main components of the IMoD system is the Learning Objectives Feature (LOF). The LOF consists of an interface and the backend intelligence needed to help the user (instructor) to create specific and measurable learning objectives and link them with other aspects of the i-mod design. This paper presents the design of the backend structure of the LOF.

Keywords - instructional module, learning objectives, ontology, semantic web, web application

I. INTRODUCTION

A. MOTIVATION

To ensure that future generations of engineering, science and other technological practitioners are equipped with the required knowledge and skills to continue to come up with innovative solutions to solve societal challenges, effective courses or i-mods that incorporate the best pedagogical and assessment practices must be developed and delivered. Tertiary-level STEM educators tend to have little or no STEM education training. Their approaches to learning, instruction, and assessment mimic the experiences they were exposed to as students and are not necessarily informed by scholarship in the area of how people learn. The road to effective STEM instruction starts with a well-conceived and constructed plan. An information technology (IT) tool that can guide STEM educators through the complex task of instructional module development, provide relevant information about research-based pedagogical and assessment principles and automate aspects of the complex decision making involved in the design process,
will be of great value. Although IT tools such as Electronic Performance Support Systems (EPSSs), Knowledge Management Systems (KMSs), and Repositories, have been used to support some parts of the i-mod design and development process, none of them currently provide all of these features\(^1\).

### B. PROBLEM STATEMENT

The Instructional Module Development (IMoD) system, currently under development, presents a web-based framework for representing an i-mod and scaffolds users through the design process in a fashion similar to the way turbo tax, a tax preparation software package, provides step-by-step guidance on request, to users completing tax returns. This framework uses Semantic Web technologies to automate aspects of the complex decision making processes that are needed for designing an i-mod. The architecture of the IMoD system resembles the structural model of an i-mod\(^2\). It consists of five main components – Context, Learning Objectives, Content, Pedagogy and Assessments. This paper discusses the tool being developed for the Learning Objectives component, known as the Learning Objectives Feature (LOF). The LOF forms the heart of the IMoD system and will be built on top of a semantic web ontology that will represent the learning objectives (LOs) associated with an i-mod. An ontology formally represents knowledge as a set of concepts and their relationships in a domain\(^3\). The LOF will provide an interface to enable the user to enter learning objective data. This data will be captured and processed into metadata that will be used by the backend intelligence to create and link the LOs with data in the other components of the IMoD.

### II. BACKGROUND

This section describes the literature review that was undertaken before the LOF was designed. The system has been designed such that its foundation is based on research in areas such as curriculum and instruction design\(^2,4,5,6,8,9\) and how people learn\(^7\). The aim of this tool is to make this research accessible to instructors so that they can follow the best practices while designing their instruction.

#### A. INSTRUCTIONAL DESIGN

According to Pellegrino, curriculum can be defined as the knowledge and skills in subject matter areas that teachers teach and students are supposed to learn\(^4\). The curriculum also describes the scope of the content that is to be taught in a particular subject area. It also usually provides an order in which the content is to be taught to be most effective. Pellegrino defines a triad that is the center of an educational enterprise. This triad consists of – the Curriculum, the Instruction (learning activities used to teach the content) and the Assessments. Pellegrino believes that this triad should be centered about a subject domain and all three components of the triad should be in alignment with each other in order to achieve effective teaching. In order to achieve alignment in the triad elements, Pellegrino believes that it is important to understand the principles behind how people learn. An important aspect in the process of learning is being able to develop a “foundation of factual knowledge” and then understand and apply these facts in a practical environment\(^4\). This concept is in line with the taxonomy of learning domains developed by Benjamin Bloom in 1956. Bloom identified different levels of learning within the cognitive
domain[5]. Two more domains were later added to this classification – the affective domain and the psychomotor domain. As shown in Figure 1 these domains are further divided and arranged from the simplest learning activity to more complex activities. Bloom’s taxonomy was later modified by Anderson and Krathwohl to bring up to date with the 21st century[6].

Another important principle of learning is the environment in which the learning takes place. A learning environment can be classified into four types: learner-centered, knowledge-centered, assessment-centered and community-centered[7]. In a learner-centered environment, the focus is on the student, and teaching is adapted to the skills of the learner. In a knowledge-centered environment, the focus is on the content being taught and in an assessment-centered environment, feedback is important to adapt the learning to what the student has learned. Community-centered environments think of the classroom or the learning environment as a community and learning techniques are adapted to fit this ideology.

Another approach to developing a curriculum is known as the Backward-Design principle introduced by Wiggins and McTighe in their book Understanding by Design[2]. Usually instructors design a course based on the content presented in a textbook. Wiggins and McTighe advocate the opposite. They believe that identifying the objectives to be achieved during the course is a more effective starting point. One should “start with the end—the desired results (goals or standards)—and then derive the curriculum from the evidence of learning (performances) called for by the standard and the teaching needed to equip students to perform”[2]. To backward design a curriculum, the desired results are identified, and then assessments are designed to verify that these results have been achieved. The learning experiences and instruction are then formulated around the desired results and the assessments.
The IMoD system draws inspiration from the work done by Bloom, Anderson and Krathwohl in the area of learning taxonomies to design the ontological model and from Wiggins and McTighe to design the structure of an i-mod.

B. LEARNING OBJECTIVES

As shown by Wiggins and McTighe, the desired results form a crucial part of the instructional module design process. The desired results or the objectives are the starting point for developing an effective and successful i-mod. An objective is a way in which an instructor can inform others of what he/she intends for the students to achieve. As defined by Robert Mager, an objective is related to outcomes instead of the process followed for achieving those outcomes and the outcome is specific and measurable.\[8\]

According to Dr. Dee Fink, learning goals or objectives should be defined after identifying the context of learning. Dr. Fink has formulated a Taxonomy of Significant Learning that classifies learning activities.\[9\] This taxonomy contains six sub-categories – foundational knowledge, application, integration, human dimension, caring and learning how to learn. In order to define the objectives, the instructor should think about which one of these sub-categories the objective would belong to. The way these sub-categories are defined is such that they are interactive and each is able to stimulate any of the others.

Another way of formulating an objective is by following the structure defined by Mager. An objective can have three characteristics: Performance – what the learner should be able to do, Conditions – the conditions under which the learner should be able to do it and the Criterion – how well must it be done.\[8\] The Performance characteristic states what the learner is expected to do during the course of learning in order to display competence. This performance is to be carried out under certain Conditions. These conditions should also be specified in the objective in order to add clarity. Since the objective is an outcome that is to be attained, there has to be a way in which acceptability can be determined. This is the criterion that should be specified to describe what performance is acceptable. It would be quite simple to clutter an objective with unnecessary information, like describing the procedures to be followed for the learning instruction, the target audience, etc. Mager suggests that it is better to leave these out of an objective.

The IMoD project follows the instructional objective format defined by Mager and builds upon it to help the instructor develop clear, specific objectives.

C. EXISTING TOOLS THAT SUPPORT INSTRUCTIONAL DESIGN

This section talks briefly about the different e-learning and curriculum development tools that are available via the World Wide Web and how they differ from the IMoD system.
Curriculum Design Tools

The National Engineering Education Delivery System (NEEDS) is a library of learning resources belonging to the engineering domain and is available digitally \([10]\). These resources can be accessed by users (both students and teachers) through a web interface. Users can upload and download these resources, search for and also comment on the available resources in order to facilitate their educational needs.

Connexions is also a digital library of educational resources. It provides a repository of educational content that can be accessed over the World Wide Web. Instructors or authors can create and upload content in the form of modules of varying sizes. A group of modules forms a “collection” of knowledge that can be downloaded and used by students \([11]\).

National Science Digital Library (NSDL) provides a broad set of tools and services that help the education community to “organize, manage, and disseminate digital educational content to advance STEM teaching and learning” \([12]\). These tools can be used by developers as stand-alone applications or can be partnered with NSDL.

The Understanding by Design tool \([13]\) is based on Wiggins’s and McTighe’s Backward Design principle \([2]\). This tool helps instructors to design a course or a unit based on the Backward Design principle. It provides a user interface where an instructor can enter the desired results, design assessments to evaluate whether these results have been achieved and then plan instruction for the course.

While most of the existing tools help in the sharing and management of educational resources, the IMoD tool helps instructors to develop i-mods based on the best design principles that have arisen out of extensive research in the fields of education and instruction. It helps both experts and novices to learn to develop effective instruction without needing prior knowledge of the best practices to be followed.

Ontology-based Tools

Content Automated Design and Development Integrated Editor (CADDIE) is an e-learning tool that employs ontologies and semantic web technologies in order to support instructional design and provide personalized learning services to its users \([14]\). The learner using this tool is first profiled and then based on the profile the tool identifies the best strategies for presenting resources to the learner so that he/she can learn in the most effective manner.

The Intelligent Web Teacher (IWT) is a research project based on semantic web technologies that also provides personalized learning services. It uses semantic web ontology to represent “Domain Concepts” and relations between them. A Domain Concept is a concept that belongs to some educational domain and could be described by an educational resource or “Learning Object” \([14]\).

LOMster is a tool that has been developed to “share and reuse” educational resources and is based on peer – to – peer technology \([15]\). The tool is based on the Learning Object Metadata (LOM) standard developed by the IEEE LTSC (Learning Technology Standards Committee). The LOM standard specifies the format of the metadata that is used to describe a learning object. The tool allows users to add learning objects/ resources to the system, generates metadata for the
added content and also allows provides the ability to share this resource with peers connected in the network.

As part of the IMoD project we will develop an ontology for the Learning Objectives Feature and its other components. This ontology will contain the classes that will be used to define the structure of a learning objective and also the structure of the metadata that will be generated for the LOs. The ontology would also hold the relations between the different classes.

D. SEMANTIC WEB TECHNOLOGIES

Semantic Web Ontology: At the heart of semantic web technologies lie ontologies. Ontologies provide a formal representation of knowledge by specifying knowledge as concepts belonging to a domain. An ontology also provides relationships between the different concepts that have been specified. Ontologies are usually defined by creating a hierarchy of classes to represent data entities and linking these classes by creating relationships between them. The languages used to specify ontologies are called ontology languages. Some of the XML based ontology languages are DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer), RDF (Resource Description Framework), OWL (Web Ontology Language), RDF Schema and SHOE (Simple HTML Ontology Extensions).

Resource Description Framework (RDF): The Resource Description Framework, as the name suggests, is a language for describing web resources. It used for representing information, especially metadata, about web resources [16]. RDF is designed to be machine-readable so that it can be used in software applications for intelligent processing of information.

Web Ontology Language (OWL): The Web Ontology Language is a markup language that is used for publishing and sharing ontologies [17]. OWL is built upon RDF and an ontology created in OWL is actually a RDF graph. Individuals with common characteristics can be grouped together to form a class. OWL provides different types of class descriptions that can be used to describe an OWL class. OWL also provides two types of properties: object properties and data properties. Object properties are used to link individuals to other individuals while data properties are used to link individuals to data values.

Protégé: Protégé is an open-source ontology editor [18]. Protégé provides a Protégé-OWL editor that allows users to create semantic web ontologies in W3C’s Web Ontology Language.

III. LOF FRAMEWORK

A. IMoD STRUCTURE

The IMoD tool would help to create new i-mods, store and manage already defined i-mods and share them if needed. The structure of the i-mods would also facilitate re-use of existing i-mods and sharing of i-mods among users.
In order to understand the structure of the Learning Objectives Feature (LOF), it is important to understand the structure of the IMoD system. The IMoD system is comprised of five components – Context, Learning Objectives Feature, Content, Pedagogy and Assessments. The Context component holds information about the instructor, the i-mod being designed and the schedule that will be followed by the i-mod. The Content component would contain information about the educational content of an i-mod, i.e., what content is to be taught as part of this i-mod. The Learning Objectives component or the LOF would store all the learning objectives to be achieved. Pedagogy would help to define the learning activities or instructional activities that the instructor would utilize to convey the subject matter, i.e., content. Assessments would contain all the assessment pieces designed for the objectives defined in the LOF. Each of the components would be linked to one another to provide a more comprehensive and integrated i-mod. For example, an LO would specify the Content that supports it. It would also be linked to Assessments that would help to evaluate whether or not the LO was attained. Content would be linked to the Pedagogy that would specify how this content is to be delivered to the audience.

B. LOF STRUCTURE

As Wiggins and McTighe discovered, the desired results form the linchpin on which the i-mod rests. IMoD follows this understanding and provides a Learning Objectives Feature component that allows an instructor to define a learning objective that follows a modified version of the format defined by Robert Mager. Bloom’s revised taxonomy by Anderson and Krathwohl is then used to build upon the objective and add more meaning to it. The model of a LO is as shown in Figures 2 and 3.

![Learning Objective Structure Diagram](image)

Figure 2: The learning objective structure for the Learning Objectives Feature of the IMoD (Red boxes indicate user input while Blue boxes indicate metadata generated by the back end)

The Learning Objective consists of three parts – Condition, Performance and Criteria. These three parts are entered by the user while stating the LO. The back end analyzes the input and
Andhare, Dalrymple, Bansal

generates metadata for the LO, i.e., adds more meaningful information to it. The learning domain and domain category of the Performance part is based on Bloom’s revised taxonomy and is as illustrated in Fig 3. The learning domain can be cognitive, affective or psychomotor, while the domain categories are the learning activities that form the learning domains.

Consider a learning objective defined as: Given a block of wood and design specifications, the student should be able to reproduce the wood pattern with 95% accuracy. In this example, the Condition is that a block of wood and design specifications have been provided to the student, who is the target audience. This Condition is performance-based and has been explicitly provided by the user. If the condition is not specified, then the LOF would annotate the LO as containing a generic condition and pick the target audience and the i-mod type from the Context component of the IMoD system. The Performance is the act of reproducing the pattern in the wood block. The use of the learning action ‘reproduce’ indicates that the learning objective belongs to the cognitive learning domain whose domain category is ‘creating’. The system will also allow the user to enter the content area that the performance belongs to, for example, the content area in this instance could be ‘Wood Pattern-making’. If the content area has been provided then it will be linked to this content in the Content component of the IMoD system. The percentage of accuracy with which the student is supposed to carry out the performance forms the Criteria part of the LO and is of the ‘accuracy’ type.

![Figure 3: Learning domains and domain categories](image)

(Blue boxes indicate Learning Domains and Green boxes indicate Domain Categories)

C. ONTOLOGICAL MODEL

The ontology has 5 main classes: Learning Objective, Learning Domain, Domain Category, Objective Components and Content. These classes are further divided into sub-classes as shown
in Figure 4. The metadata is used to associate information about one class with another class. This association will be done using object properties used to define relations between the classes. The class hierarchy is as shown in Figure 5. Table 1 shows how the ‘criteria’ class and its sub-class ‘accuracy’ are defined in OWL. Table 2 shows the OWL code snippet for the object property class ‘hasPerformance’ and its sub-class ‘hasLearningDomain’.

<table>
<thead>
<tr>
<th>Table 1: Code snippet for an OWL class and sub-class</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CLASS</strong></td>
</tr>
<tr>
<td>&lt;owl:Class rdf:about=&quot;&amp;LearningObjectiveOntology;Criteria&quot;&gt; &lt;rdfs:subClassOf rdf:resource=&quot;&amp;LearningObjectiveOntology;Objective_Components&quot;/&gt; &lt;/owl:Class&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Code snippet for an OWL object property class and sub-class</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CLASS</strong></td>
</tr>
<tr>
<td>&lt;owl:ObjectProperty rdf:about=&quot;&amp;LearningObjectiveOntology;hasPerformance&quot;&gt; &lt;rdfs:domain rdf:resource=&quot;&amp;LearningObjectiveOntology;Learning_Objective&quot;/&gt; &lt;rdfs:range rdf:resource=&quot;&amp;LearningObjectiveOntology;Performance&quot;/&gt; &lt;rdfs:subPropertyOf rdf:resource=&quot;&amp;owl;topObjectProperty&quot;/&gt; &lt;/owl:ObjectProperty&gt;</td>
</tr>
</tbody>
</table>

The learning domain class has the 3 sub-classes as shown in Figure 3: Cognitive Domain, Affective Domain and Psychomotor Domain. The domain category class is further divided into the 18 categories shown in Figure 3.

Consider a learning objective LO1 that has performance P, a generic condition GCO and criteria C. P belongs to the learning domain D and domain category DC, and the condition GCO has
target audience TA and i-mod type TY. LO1 also specifies the content that P belongs to as, CON. Figure 6 shows how classes are associated with one another based on the properties being used with respect to LO1. The circles in Figure 6 indicate the classes in the ontology and the links between the classes illustrate the relations (object properties) between them.

Figure 4: Ontological model (class hierarchy)

Figure 5: Object properties
D. BACK END ALGORITHM

The back end algorithm will be responsible for connecting the ontology to the user interface and providing the intelligence needed by the tool for processing user data. The back end algorithm will analyze the data entered by the user, compare it to the ontology and generate metadata for it. This metadata will be stored along with the user data to make connections and associations between the different components of an i-mod. The generated metadata will also help the user by providing suggestions or tips that will make the LO more effective.

One of the algorithms that form the back end intelligence, currently under development, reads in the user input and analyzes it to identify the Performance part of a LO and determine the learning domain and the domain category that it belongs to. Table 3 contains the pseudo-code for this algorithm. Other algorithms that would be part of the back end would be those that analyze the input to identify the Condition and Criteria and then generate metadata for them, create associations between different components by identifying the commonalities between them, and eventually, even algorithms that will analyze a LO or an i-mod, evaluate its quality and provide suggestions to the user to help improve the i-mod.
Table 3: Pseudo-code for algorithm for generating Performance metadata

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Get user input and create an ontology instance document.</td>
</tr>
<tr>
<td>2.</td>
<td>Parse the ontology using an OWL parser.</td>
</tr>
<tr>
<td>3.</td>
<td>Extract and load semantic relations from the ontology. Identify the performance component of the LO.</td>
</tr>
<tr>
<td>4.</td>
<td>Compare user data with the extracted relations to derive the learning domain and domain category.</td>
</tr>
<tr>
<td>5.</td>
<td>Add the derived metadata for the learning domain and the domain category to the ontology instance document.</td>
</tr>
</tbody>
</table>

IV. SUMMARY AND FUTURE WORK

This paper focuses mainly on the Learning Objectives Feature of the IMoD tool. The same concept can be extended to include all the components that form the IMoD system. The ontology built during this project will be extended to include data needed by the Content, Pedagogy and the Assessments components to form a comprehensive whole, and also to make strong associations between all the components. The aim is to fully utilize semantic web technologies to integrate all the components in the IMoD system and reveal the relations between the components to the user, so that designing instruction becomes not only simple and convenient, but also helps to improve the quality of instruction. The web interface will also be developed further to allow users to add Content, Pedagogy and Assessment pieces to an i-mod. Future work would also involve the management and sharing of multiple i-mods and the ability to allow users to obtain feedback for a constructed i-mod.

ACKNOWLEDGEMENTS

We would like to express our gratitude to all those who have helped us through the conception and realization of this project. We would like to thank Shashank Balasubramanian, John Houghtelin, Vishnu Menon, Vrushali Moghe and Michael Sheppard for their assistance in the development of this project. We would also like to extend our thanks to the College of Technology and Innovation at Arizona State University for supporting this project.

REFERENCES


