IMoD: SEMANTIC WEB-BASED INSTRUCTIONAL MODULE DEVELOPMENT SYSTEM

Srividyaa Kona Bansal, Odesma Dalrymple, Vishnu Menon, Ketaki Andhare, Vrushali Moghe
Arizona State University – Polytechnic campus, Dept. of Engineering, Mesa, AZ 85212, USA
srividya.bansal@asu.edu, odesma.dalrymple@asu.edu

ABSTRACT

To ensure that future generations of engineering, science, and other technological practitioners are equipped with the required knowledge and skills to continue to innovate solutions to solve societal challenges, effective courses or instructional modules that incorporate best pedagogical and assessment practices must be designed and delivered. Science, technology, engineering and mathematics (STEM) educators typically come from STEM backgrounds and have little or no formal STEM education training. An information technology (IT) tool that can guide STEM educators through the complex task of course design and development, ensure tight alignment between various components of an instructional module, and provide relevant information about research-based pedagogical and assessment strategies will be of great value. This paper presents a framework and prototype implementation of a semantic web-based software system called the Instructional Module Development (IMoD) system that guides instructors, step-by-step, through an outcome-based education process as they define learning objectives, select content to be covered, and define the learning environment and context for their course(s). The IMoD software system uses Semantic Web technologies to provide intelligent interactions with users; dictate a course design process in conformance with the underlying framework and provides feedback to the user on their course design.

KEY WORDS


1. Introduction

The road to effective science, technology, engineering, and mathematics (STEM) instruction starts with a well-conceived and constructed plan or curriculum. Most new STEM educators have little to no formal STEM education training. They often utilize the instructional techniques they were exposed to as students, and may not be familiar with best practices in instructional design. Developing a well-conceived curriculum is a labour-intensive process and is often only achieved after numerous iterations of teaching the course. The aim of the semantic web-based Instructional Module Development (IMoD) system is to present a framework-informed by the scholarship on curriculum design- for conceptualizing and representing an instructional module (i-mod), particularly in the areas STEM disciplines. This system will provide a scaffold via various help and support features to guide the users through i-mod development process.

1.1 Example User Story

Consider a typical user, Jane, a recent Ph.D. graduate and a new Software Engineering faculty member who has been assigned to teach Introduction to Object-oriented Programming. Like many new faculty, Jane really wants to do a good job, and make a positive impact on the students she teaches. Given that this will be her first time teaching, and a new course at that, she can’t help but feel a bit anxious. The Chair of the department wants to provide support for Jane to be successful at the task. He submits a recommendation for her to attend a 3-day workshop on effective teaching. Unfortunately, by the time his recommendation is received, there was no more availability. Unguided, Jane approaches the design of the class by attempting to model the structure that she had been exposed to when she took a similar class.

Revising this scenario given the availability of the IMoD system. Jane goes to IMoD site and creates a user account. She then creates a new course design and signs-off. The next day she resumes use of the system and continues the self-paced course design process. She is initially prompted to enter information about the learning context and environment for the course, e.g., type of course, meeting days and times, anticipated number of hours that will be spent on in-class and out-of-class activities, instructor(s) information, course policies, etc. As she enters the information in response to the prompts, she realizes that she is uncertain about which course policies to choose, and needs more information. She clicks on the help button next to the policy section and a panel appears from the right side of the screen with explanations on the various policy options and their importance in course management. This information allows Jane to make an informed selection. She then observes a pop-up message on her screen that recommends she explore the learning objectives section next, and heed this advice. As she navigates to the learning objective interface, an option to view a short video presentation on learning objectives appears. When
2. Related Work

To justify the need for the development of the IMoD system we conducted a competitive analysis to determine the strengths and weaknesses of tools and approaches currently used to support course design and related training. The tools and approaches that were evaluated were categorized into five groups based on primary functions and features. Each of these groups is described in the following section, and compared in Table 1.

2.1 Knowledge/Learning Management System (KMS/LMS)

This group contains a number of proprietary and open-source solutions that are delivered either as desktop or web-based applications. These tools mainly facilitate the administration of training, through the (semi-) automation of tasks such as: registering users, tracking courses in a catalog, recording data, charting a user’s progress toward certification, and providing reports to managers. These tools also serve as a platform to deliver eLearning to students. In that context, their main purpose is to assemble and deliver learning content, personalize content and reuse it. Examples: Blackboard [1], Moodle [2], Sakai [3], Canvas [4], WebWorK, and Olat [5].

2.2 Educational Digital Libraries

These tools contain collections of learning and educational resources in digital format. They provide services that support the organization, management, and dissemination of the digital content for the education community. Examples: National Engineering Education Delivery System (NEEDS) [6], National Science Digital Library (NSDL) [7] and Connexions [8].

2.3 Personalized Learning Services

There are a number of e-learning tools that leverage semantic web technologies to support personalized learning services for their users with an ontology based framework [9]. Some of these tools function by initially profiling the learner and then, based on that profile, identifying the best strategies for presenting resources to them. They can also provide feedback to instructors on student learning, so improvements to the content and structure of the course can be incorporated. For many of these tools the ontology framework is used to bridge learning content with corresponding pedagogy; however, they seldom address assessments and learning objectives. Examples: Content Automated Design and Development Integrated Editor (CADDIE) [9], Intelligent Web Teacher (IWT), LOMster [10], and LOCO-Analyst [11].
Table 1: Competitive Analysis - Current Tools & Approaches that Support Outcome-Based Course Design

<table>
<thead>
<tr>
<th>Current tools</th>
<th>Provides step-by-step guidance on the execution of the course design process</th>
<th>Provide access to research-based assessment strategies</th>
<th>Provide feedback on the fidelity of course design</th>
<th>Generates course design documentation</th>
<th>Accessibility</th>
<th>No. of users supported</th>
<th>Allow sharing and peer-reviewing of course design</th>
<th>Allows collaboration among users on design tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed IMoD system</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>OL</td>
<td>UL</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>KMS / LMS</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>OL</td>
<td>UL</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Educational Digital Libraries</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>OL</td>
<td>UL</td>
<td>✗</td>
<td>❋</td>
</tr>
<tr>
<td>Personalized Learning Services</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>OL</td>
<td>UL</td>
<td>✗</td>
<td>❋</td>
</tr>
<tr>
<td>UbD Exchange</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>OL</td>
<td>UL</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Professional Development Workshops, Courses &amp; Seminars</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>FtF</td>
<td>L</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Key**
- ✗: Feature is not available
- ✓: Feature is available tool
- UL: Supports an unlimited number of users
- L: Limited / pre-determined enrollment
- OL: On-line access
- FtF: Face to face at predefined location

1 Fidelity will be assessed in terms of the cohesiveness of the alignment of the course design components (i.e., content, assessment, and pedagogy) around the defined course objectives.
2 Users can submit 1 course design / year for review and feedback from certified UbD Exchange reviewer.
3 Accessible to users who pay for a subscription.

**Figure 1. IMoD Framework**

A: PC

B: Overview
2.4 Understanding by Design Exchange (UbD Exchange)

This is a software framework based on Wiggins’s and McTighe’s Backward Design principle [12] that is used for designing curriculum, assessments, and instruction, and integrates K-12 state and provincial standards in the design of units. It provides a form-based user interface to fill in the details of the course unit that is being designed.

2.5 Professional Development Workshops, Courses & Seminars

Face-to-face training sessions in teaching and learning that are facilitated by experts in the field of instructional design. Examples: National Effective Teaching Institute [13], Connect Student Learning Outcomes to Teaching, Assessment, and Curriculum [14], ENE 695 – Content, Assessment and Pedagogy [15]

Our search identified very few tools and approaches that contained features or functionality that explicitly facilitated the design of course curriculum. This is the key deficiency that the IMoD system will address. Some of the e-learning tools such as IWT and LOMster (described in section 2.3) use the Learning Object Metadata (LOM) standard developed by the IEEE LTSC (Learning Technology Standards Committee). The LOM standard describes a ‘learning object’, i.e., an educational resource. The IMoD system uses an ontology that describes all the important components of course design – learning objectives, content, assessments, and pedagogy.

3. Theoretical Framework

Many of the leaders in faculty development programs have identified facilitation by experts as a key ingredient in increasing the effectiveness of instructional development programs [13]. For the IMoD system, which will provide professional development with the use of an online tool, expert facilitation is embedded within its design, through the application of a framework that is informed by research in the area of instructional development for STEM disciplines.

3.1 Existing Models for Outcome-Based Course Design

Outcome-based education (OBE) is an approach where the product defines the process, i.e., the outcomes that specify what students should be able to demonstrate upon leaving the system are defined first, and drive decisions about the content and how it is organized, the educational strategies, the teaching methods, the assessment procedures and the educational environment [16]. This is a contrast to the preceding “input-based” model that placed emphasis on the means as opposed to the end of instruction. OBE was used as the principal guide for the development of the IMoD framework. A number of models have been developed to represent the application of OBE in the design of effective courses. Four key models widely discussed in the engineering education literature are: 1) the Effective Course Model by Felder & Brent [13]; 2) Integrated Course Design by Fink [17]; 3) Understanding by Design Model [12]; 4) Content Assessment Pedagogy Model by Streveler, Smith, & Pilotte [15]. All of these models either directly or indirectly identify four main elements that must be tightly aligned when defining a course design, i.e., course objectives, content, assessments, and pedagogy. Therefore, one of the main challenges in adhering to an outcome-based approach is maintaining the alignment between course elements. Inconsistencies in the interrelation of these elements can lead to the overall incoherence of the course.

One approach for achieving alignment among course element is through a “backward-looking” design process where the desired results are identified first, 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 use of this approach forms the basis of the Understanding by Design model [12], and the other models also apply it. One of the key functions the IMoD system is expected to perform is the evaluation of the fidelity of the course design. To achieve this, the IMoD framework must include machine processable constructs that can be used to make inferences on the inconsistencies in the relationships between the elements of the course being designed. While the backward-looking process dictates an ideal sequencing of tasks, it is limited in its ability to support automated inferencing on course element coherence. The IMoD framework, therefore, expands on the current models with the inclusion of new constructs.

3.2 IMoD Framework

The IMoD framework adheres strongly to the OBE approach and treats the course objective as the spine of the structure. New constructs (not included in the models previously discussed) are incorporated to add further definition to the objective. The work of Robert Mager [18] informs the IMoD definition of the objective. Mager identifies three defining characteristics of a learning objective: Performance – description of what the learner is expected to be able to do; Conditions – description of the conditions under which the performance is expected to occur; and the Criterion – a description of the level of competence that must be reached or surpassed. For use in the IMoD framework an additional characteristic was included, i.e., the Content – description of the disciplinary knowledge, skill, or behavior to be attained. The resulting IMoD definition of the objective is referred to as the PC^3 model.

The other course design elements (i.e., Content, Pedagogy, and Assessment) are incorporated into the IMoD framework through interactions with two of the PC^3 characteristics. Course-Content is linked to the content and condition components of the objective. The condition component is often stated in terms of pre-cursor
disciplines, knowledge, skills or behaviors. This information, together with the content defined in the objective, can be used to generate or validate the list of course topics. Course-Pedagogy is linked to the performance and content components of the objective. The types of instructional approaches or learning activities used in a course should correspond to the level of learning expected and the disciplinary knowledge, skills or behaviors to be learned. The content and performance can be used to validate pedagogical choices. Course-Assessment is linked to the performance and criteria components of the objective. This affiliation can be used to test the suitability of the assessment strategies since an effective assessment, at the very least, must be able to determine whether the learner’s performance constitutes competency. Figure 1 shows two visual representations of the IMoD framework. The first emphasizes the alignment of course design elements through the PC³ model of an objective. The second emphasizes the constraints placed on the course design by variables defined in the learning context.

4. IMoD Semantic Model

Semantic web technologies facilitate the organization of knowledge into conceptual spaces, based on their meanings; extraction of new knowledge via querying; and maintenance of knowledge by checking for inconsistencies. These technologies can therefore support the construction of an advanced knowledge management system [19][20]. The IMoD software system uses Semantic Web technologies to provide intelligent interactions with the users, dictate a course design process in conformance with the underlying framework, check for omissions and inconsistencies in the design, provide feedback to the user on their course design. The IMoD framework is translated into a rich meaningful knowledge structure in the form of ontology, i.e., an explicit and formal specification of a conceptualization [21]. During the course design elicitation process, logical inference algorithms test the course design for consistency and adherence with the ontological model.

The ontological model for the IMoD system currently has classes and subclasses corresponding to the Learning Objective and Content components of an instructional module. The Learning Objectives feature of the IMoD system is presented in detail in our prior work [22]. The ontology consists of 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 2. Learning Objective consists of four parts – Condition, Performance, Content, and Criteria. The Learning Domain is based on Bloom’s revised taxonomy [23]. Learning domains are categorized as cognitive, affective, or psychomotor. The domain categories are the learning activities that correspond to a certain learning domain. For example, the Cognitive learning domain has domain categories that include remembering, understanding, applying, analyzing, evaluating, and creating. Table 2 shows the OWL code snippet of the ‘Criteria’ class and its sub-class ‘Accuracy’. Table 3 shows the OWL code snippet for the object property ‘hasPerformance’ and its sub-property ‘hasLearningDomain’.

![Ontological Model (class hierarchy)](image)

4.1 Inference algorithm to provide feedback to user

The IMoD system provides the ‘Evaluate’ feature on Learning Objectives tab that assesses a learning objective and its conformance to the Semantic model. It performs 3 checks on a learning objective – completeness, correctness, and alignment of a learning objective with other components. Completeness checks for any missing components by comparing the learning objective created by the user, with the ontology specified. Correctness verifies that the learning domain and domain category entered by the user match the performance component of the learning objective. For example, if the performance is ‘analyze’ then it will belong to the Cognitive domain and Analyze domain category. If the user has incorrectly selected any other domain or category, then on evaluating the learning objective, the user will be notified of the error. The evaluate function also checks the alignment between the learning objective and content components of the IMoD system. If the user has specified some content topics and wants to use these topics while creating a learning objective, then it is essential to ensure that the importance of the content topic matches the learning objective. For example, if the user has created a content topic, ‘Functional requirements’ and specified its priority to be ‘Enduring understanding’, it means that the learning objective created for this topic should belong to a higher order learning domain like ‘analyze’, ‘evaluating’ or ‘creating’ so that students understand the topic in depth. The evaluate function verifies this based on the topic priority specified by the user; and the learning objective belongs to the correct learning domain and category. In the IMoD system, ‘Evaluate’ feature generates an XML file that contains the learning objective. This file is provided as input to the inference algorithm that processes the file and produces an output XML file that contains the
results of the evaluation. Consider an ill formed learning objective: ‘Given a set of Software requirements, student should be able to produce (write) a Software Requirements document’. The contents of the XML generated for this learning objective is shown in Table 4. On evaluating this objective the resultant XML file produced, that has feedback on improvements that can be made to the learning objective, is shown in Table 5.

5. IMoD Prototype Implementation

The prototype implementation of the IMoD system is a web-based tool that allows users to login, create profile, create/edit/save/load i-mods. It consists of three distinct components: Learning Context [24], Learning Objectives [25], and Content [26]. The system is built with a client-side user interface, a server, and a database. ExtJS 4.0 was used for client-side scripting, PHP 5.3.8 was used for server-side scripting, and MySQL 5.0.8 for the database. The ontology was built using OWL and Protégé editor. Protégé OWL APIs were used to implement the evaluate feature that evaluates the structure of the learning objective; identifies missing elements of a learning objective; validates the learning domain and domain category selection; checks if performance of a learning objective is covert or overt and provides feedback to the user; and finally validates the priority of a content topic based on the domain category.

5.1 Learning Context

The IMoD system has a Learning Context tab as shown in figure 4 that describes the learning environment of an i-mod. The user can enter information such as course title, number, URL, location, semester, schedule, pre-requisites, policies, and instructors in a user-friendly manner.

5.2 Learning Objectives Feature

Learning Objectives tab as shown in figure 5 allows users to create new learning objectives, edit existing objectives, and evaluate them. This tab has four panels arranged in an accordion style. The panels are: condition panel that provides a text area for users to enter the condition for a learning objective; performance panel that gets user input on the learning domain, domain category, performance, and a performance indicator; content panel where the user can select a content topic; and criteria panel that allows the user to select a criteria and criteria type (such as accuracy, speed, quality or quantity). Example (shown in figure 3): Given a Software Requirements Specification (SRS) document of a Capstone project, students should be able to analyze the quality (identify incorrect requirements) of the Software requirements in the document with 95% accuracy.

5.3 Content Feature

The content feature of the IMoD system incorporates four user-interface sections each of which contribute something specific to the task of topic addition, prioritization, and organization. The four sections are: Topic Addition Grid, Topic Hierarchy, Topic Prioritization, and Schedule as shown in figure 7.

![Figure 3. Example Learning Objective described using PC^3 model](image3)

5.3.1 Topic Addition Grid:

The topic addition grid enables instructors to create topics/sub-topics, set an associated learning objective, set a level of learning (priority) and add resources. An instructor can click on the Add Topic button on the grid to begin creation of a new topic/sub-topic. Instructors can click on the Remove Topic button to remove a selected topic/sub-topic from the system. Once instructors are done entering information, (s)he can save the information by clicking on the Save button located on the toolbar of the IMoD system. Information stored in the topic addition grid can be dragged and dropped onto the neighbouring tree structure (on right-hand side).

5.3.2 Topic prioritization:

The content feature of the IMoD system uses a prioritization framework that classifies topics and subtopics of a particular course as one of the following: good to be familiar with, important to know or understand and enduring understanding (shown in figure 4). The IMoD content feature implements this technique by having the user set a priority for each content topic/sub-topic as they are added into the IMoD system. This feature is implemented using a drop-down field from which the user can select a priority designation. This
Table 2. Code Snippet for OWL Class Criteria and Subclass Accuracy

<table>
<thead>
<tr>
<th>CLASS</th>
<th>SUB CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;owl:Class rdf:about=&quot;#Criteria&quot;&gt;</code></td>
<td><code>&lt;owl:Class rdf:ID=&quot;Accuracy&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;rdfs:subClassOf rdf:resource=&quot;#ObjectiveComponents&quot;/&gt;</code></td>
<td><code>&lt;rdfs:subClassOf rdf:resource=&quot;#Criteria&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/owl:Class&gt;</code></td>
<td><code>&lt;/owl:Class&gt;</code></td>
</tr>
</tbody>
</table>

Table 3. Code Snippet for Object Property and SubProperty

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>SUB PROPERTY</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;owl:ObjectProperty rdf:ID=&quot;hasPerformance&quot;&gt;</code></td>
<td><code>&lt;owl:ObjectProperty</code></td>
</tr>
<tr>
<td><code>&lt;rdfs:subPropertyOf&gt;</code></td>
<td><code>&lt;rdf:about=&quot;http://www.w3.org/2002/07/owl#topObjectProperty&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/owl:ObjectProperty&gt;</code></td>
<td><code>&lt;rdfs:domain rdf:resource=&quot;#Learning_Objective&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;rdfs:range rdf:resource=&quot;#Performance&quot;/&gt;</code></td>
<td><code>&lt;rdfs:range rdf:resource=&quot;#Affective_Domain&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/owl:ObjectProperty&gt;</code></td>
<td><code>&lt;rdfs:range rdf:resource=&quot;#Psychomotor_Domain&quot;/&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/owl:ObjectProperty&gt;</code></td>
<td><code>&lt;rdfs:domain rdf:resource=&quot;#Performance&quot;/&gt;</code></td>
</tr>
</tbody>
</table>

value is then saved into the database when the user clicks on the save button on the content page. The user can also choose to leave the drop-down empty and later update the priority field as and when (s)he/she is ready.

5.3.3 Topic Hierarchy:
The IMoD system employs a tree structure to organize topic/subtopic information and also distinguish between the topics and sub-topics. Once the user is done creating a topic, they can select and drag a topic from the Topic Addition grid onto a selected node on the Topic Hierarchy tree structure.

5.3.4 Schedule:
Once instructors are done creating and organizing content topics, the IMoD system provides them with a means to plan the schedule for instruction. This section of the content feature provides a means to graphically organize the order in which the content topics are to be taught.

Table 4. XML file provided as input to ‘Evaluate’ feature

```
LEARNING OBJECTIVE CREATED BY THE USER
(CONVERTED TO XML FORMAT BY ‘EVALUATE’ FEATURE)
<?xml version="1.0"?>
<learningObjective>
  <performance>produce</performance>
  <indicator>write</indicator>
  <learningdomain>Cognitive</learningdomain>
  <domaincategory>Remembering</domaincategory>
  <content>Software Requirements document</content>
  <priority>Enduring Understanding</priority>
  <condition>Given a set of Software requirements</condition>
</learningObjective>
```

Table 5. XML file generated as output by ‘Evaluate’

```
FEEDBACK TO THE USER ON A LEARNING OBJECTIVE
<?xml version="1.0" encoding="UTF-8"?>
<evaluationResults>
  <comment>Learning objective contains Performance. Good.</comment>
  <comment>Learning objective contains a Condition. Good.</comment>
  <comment>Learning objective contains a Content topic. Good.</comment>
  <warning>Criteria is missing. Adding a Criteria to a learning objective would help in creating Assessments.</warning>
  <warning>The Domain Category selected is incorrect. The category should be: Applying</warning>
  <warning>The priority of the content topic does not align with the performance. Please re-consider.</warning>
</evaluationResults>
```

6. Conclusions and Future Work
A well-designed and constructed course plan or curriculum is an integral part of the foundation of effective STEM instruction. We have presented a framework for outcome-based instruction design that incorporates learning objectives, content, assessments, and pedagogy. We presented a prototype implementation of the IMoD system that guides a user through the design of Learning Context, Learning Objectives, and Content elements of an instructional module. This is an ongoing project and the fully functional IMoD system that includes assessments and pedagogy components is currently under development. Future work would also involve the sharing and reviewing of i-mods and the ability to allow collaboration during the design phase of
Figure 5. Learning Context Tab of IMoD System

Figure 6. Learning Objective Tab of IMoD System

Figure 7. Content Tab of IMoD System
an instructional module. The long-term goal of this project is to provide feedback to the user on the fidelity of their course design and report any omissions and inconsistencies in the design. This will be assessed in terms of the cohesiveness of the alignment of the course design components (i.e., content, assessment, and pedagogy) around the defined course objectives. The IMoD system will also serve as a repository of current best pedagogical and assessment practices, and based on selections the user makes when defining the learning objectives of the course, the system will present options for assessment and instructions that align with the type of student learning or level of student learning desired.

References


