Building Faculty Expertise in Outcome-based Education Curriculum Design

Srividya Bansal, Ashraf Gaffar
School of Computing, Informatics, Decision Systems Engg.
Arizona State University, Mesa, AZ, USA
{srividya.bansal, ashraf.gaffar}@asu.edu

Odesma Dalrymple
Shiley-Marcos School of Engineering
University of San Diego, San Diego, CA, USA
odesma@sandiego.edu

Abstract— **An information technology (IT) tool that can guide STEM educators through the complex task of course design 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. A team of researchers is engaged in a User-Centered Design (UCD) approach to develop the Instructional Module Development System (IMODS), i.e., a software program that facilitates course design. In this paper the authors present the high-level design of the IMODS and demonstrate its use in the development of the curriculum for an introductory software engineering course.**

Keywords—instruction design; outcome-based education; semantic web-based application; user-centered design

I. Introduction

Felder, Brent and Prince [1] have made a strong argument in support of instructional development training for engineering faculty. This argument, which cites among other reasons: shortfalls in graduation rates, changing demographics and attributes of the student body, and modifications in the expectations of graduates; can be extended to encompass all STEM fields. Furthermore, studies show that for 95% of new faculty members, it takes four to five years, through trial and error (the most common method of gaining expertise in teaching), to deliver effective instruction [2]. While there are a number of options available to faculty for receiving instructional development training (i.e., training focused on improving teaching and learning), most share similar format, features, and shortcomings. For example: workshops, courses and seminar series, the most common program structures, are often offered at a cost to the institution, department or individual attendee; delivered face-to-face at specified times; and accessible to a restricted number of persons. Even when interest is high, these factors can become obstacles to participation.

Outcome-based Education (OBE) is a result-oriented approach where the product defines the process. The learning outcomes guide what is taught and assessed [3], [4]. This approach contrasts the preceding “input-based” model that places emphasis on what is included in the curriculum as opposed to the result of instruction. There is a growing demand and interest in faculty professional development in areas such as OBE, curriculum design, and pedagogical and assessment strategies.

In response to these challenges and needs, a group of faculty researchers from two south-western universities have undertaken a project to design and develop the Instructional Module Development System (IMODS) that will facilitate self-paced instructional development training while the user creates his/her course design with the added benefits of being free to all who are interested, accessible almost anywhere through a web browser, and at any time that is convenient. Additional key features of the IMODS are as follows:

1. Guides individual or collaborating users, step-by-step, through an outcome-based education process as they define learning objectives, select content to be covered, develop an instruction and assessment plan, and define the learning environment and context for their course(s).
2. Contains 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 instruction that align with the type/level of student learning desired.
3. Generates documentation of a course designs. In the same manner that an architect’s blue-print articulates the plans for a structure, the IMODs course design documentation will present an unequivocal statement as to what to expect when the course is delivered.
4. Provides just-in-time help to the user. The system will provide explanations to the user on how to perform course design tasks efficiently and accurately. When the user explores a given functionality, related explanations will be made available.
5. Provides feedback to the user on the fidelity of the course 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.

In this paper the authors present the high-level design of the IMODS and demonstrate its use in the development of the curriculum for an introductory software engineering course. The rest of the paper is organized as follows. Background material for this research project is presented in section 2. Section 3 presents the high-level design of the IMODS software system. Section 4 presents a case study that demonstrates the use of the IMODS framework in the development of an introductory software engineering course. The paper concludes with future work and acknowledgements.

II. Background

A. Related Work

To justify the need for the development of IMODS 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.

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, Moodle, Sakai, Canvas, WeBWorK, and Olat

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), National Science Digital Library (NSDL) and Connexions

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 [5]. 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), Intelligent Web Teacher (IWT), LOMster [6], and LOCO-Analyst [7].

Understanding by Design Exchange (UbD Exchange): This is a software framework based on Wiggins's and McTighe's Backward Design principle [8] that is used for designing curriculum, assessments, and instruction, and integrates K-12 state and provincial standards in the design of units [9]. It provides a form-based user interface to fill in the details of the course unit that is being designed.

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, Connect Student Learning Outcomes to Teaching, Assessment, and Curriculum, Content, Assessment ant Pedagogy [10].

Our search identified very few tools and approaches that contained features or functionality that explicitly facilitated the design of course curriculum. Of these tools few of them facilitated the generation of design documentation and feedback to the user on the fidelity of the design. These are two key deficiencies that IMODS will address.

B. IMODS Framework – PC^3 Model

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 [11] 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 to be learned – description of the factual, procedural, conceptual or meta-cognitive knowledge; skill; or behavior related to the discipline. The resulting IMOD definition of the objective is referred to as the PC3 model [12].
The other course design elements (i.e., Content, Pedagogy, and Assessment) are incorporated into the IMOD framework through interactions with two of the PC3 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 disciplinary 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 a visual representation of the IMOD framework. Learning domains and domain categories defined by Bloom's revised taxonomy [11] are used to describe learner performance. Learning domains are categorized into Cognitive, Affective, and

Psychomotor, which are further classified under various Domain Categories (Remember, Understand, Apply, Analyze, Evaluate, Create). Each Domain Category has performance verbs associated to it. Learning objective in the PC^3 model is described in terms of Performance, Content, Condition, and Criteria. Performance is described using an appropriate action verb from revised Bloom's taxonomy based on the learning domain and domain category.

Criteria: Learning objective assessment criteria are categorized as quality, quantity, speed, and accuracy. Criteria for learning objectives are described in terms of one or more of these categories with a criteria value defined or determined later when the assessment is defined.

Knowledge Dimensions*:* The revised Bloom's taxonomy introduced an additional dimension called the knowledge dimension that was categorized as Factual, Conceptual, Procedural and Metacognitive.

Topic Prioritization: The IMODS framework uses a prioritization framework that classifies topics and subtopics of a particular course as one of the following:

- Critical
- Important
- Good to know

Achieving the right mix of the three levels of learning (priorities) is essential to planning a good course.

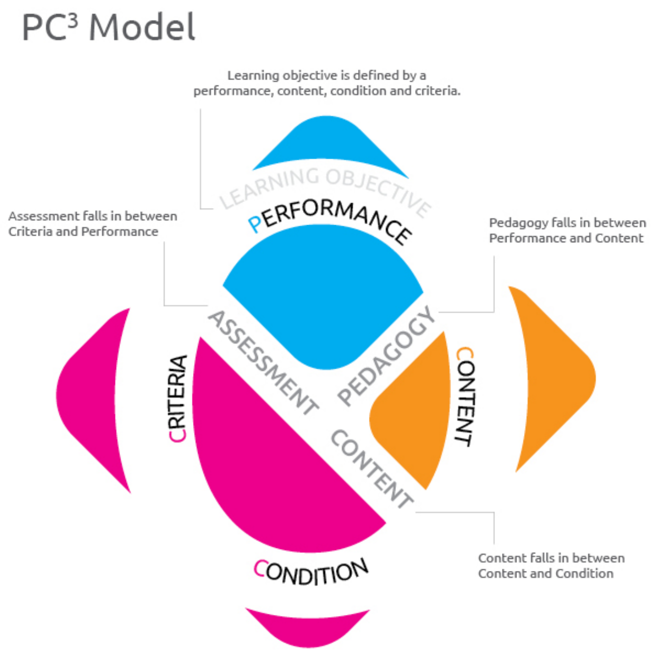

PC³ Model
Learning objective is defined by a performance, content, condition and criteria.
LEARNING OBJECTIVE
PERFORMANCE
Assessment falls in between Criteria and Performance
Pedagogy falls in between Performance and Content
CRITERIA
ASSESSMENT
PEDAGOGY
CONTENT
CONTENT
Content falls in between Content and Condition
CONDITION

Figure 1: IMODS Framework - PC^3 model

C. User-centered design methodology

The IMODS system is being developed using a user-centered design (UCD) methodology, as opposed to technology focused, methodology [13]. This approach is well suited for the project given the high cognitive nature of outcome-based course design tasks, and the high levels of interactions required between the user and the system to not only facilitate the development of course designs, but to help users build an enduring foundation of knowledge, skills and habits of mind about curriculum development.

UCD is an emerging design method that focuses on both operational and technical requirements by observing and understanding user needs and wants, as well as by prototyping and testing software throughout all phases of software lifecycle. It enables the capturing and resolution of any mismatches between users and software early on. The main objective of UCD is to allow for a closer match between users and the software, leading to a more intuitive interaction. As a design process, it also has the objective of reaching that goal in the most resource-efficient way, in terms of time and cost through careful planning and execution [14].

The UCD process can be divided into five main phases: Plan, User Research, Design, Develop, and Measure. Thus far, the research team has completed the user research and design phases of the UCD process. In this respect, 4 focus group sessions were conducted with prospective users of IMODS to gather insights on how faculty approach the task of designing a course. At the beginning of each session all participants were asked to fill an electronic background survey that collected demographic information, primary areas of interest in teaching and research, time spent on teaching, number of courses taught per year (at both undergraduate and graduate levels), and number of new courses developed (both at undergraduate and graduate levels). Participants were also asked to fill an electronic questionnaire about curriculum design tools that they currently use to create and manage their courses (e.g. preparing syllabi; communicating with students; developing teaching materials; preparing, assigning, and delivering grades, etc.). The results of this phase were published in ASEE 2014 and FIE 2014 [15], [16]. The results from the focus group helped identify the key features of the software system; an ontology that defines the relevant terms of the domain and identifies their specific meaning as well as the potential relationships between them; and mental model, which is a tool to improve understanding of the user needs and activities. Figure 2 shows the ontological concepts and relationships between concepts as a hierarchy. Figure 3 shows the mental model that depicts an affinity diagram of similar activities organized sequentially into 9 towers in the upper half with detailed relevant activities in the lower half.

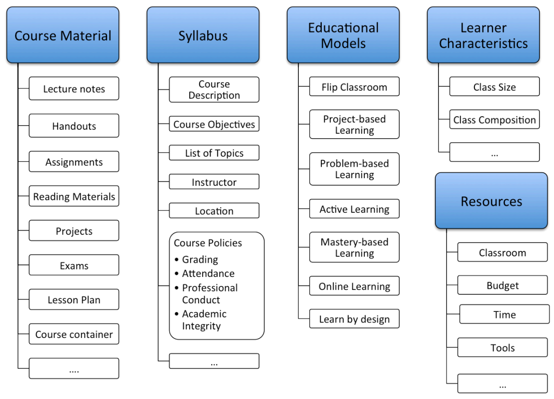
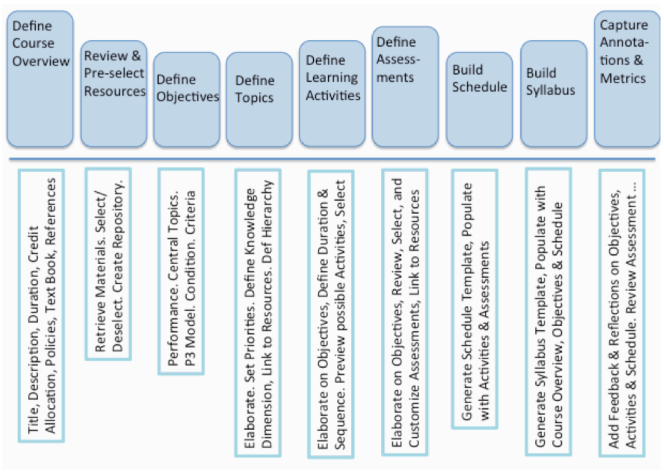


Figure 2: IMODS Ontological Concepts and Relationships



Define Course Overview
Review & Pre-select Resources
Define Objectives
Define Topics
Define Learning Activities
Define Assess-ments
Build Schedule
Build Syllabus
Capture Annota-tions & Metrics
Title, Description, Duration, Credit Allocation, Policies, Text Book, References
Retrieve Materials. Select/ Deselect. Create Repository.
Performance. Central Topics. P3 Model. Condition. Criteria
Elaborate. Set Priorities. Define Knowledge Dimension, Link to Resources, Def Hierarchy
Elaborate on Objectives, Define Duration & Sequence. Preview possible Activities, Select
Elaborate on Objectives, Review, Select, and Customize Assessments, Link to Resources
Generate Schedule Template, Populate with Activities & Assessments
Generate Syllabus Template, Populate with Course Overview, Objectives & Schedule
Add Feedback & Reflections on Objectives, Activities & Schedule. Review Assessment ...

Figure 3: Mental Model

III. High-Level Design Of IMODS

One of the biggest challenges of software design is to make sure the user has sufficient understanding to use the application successfully and accomplish the required tasks. Our User-centered design approach followed two main phases:

A. Conceptualization Phase

After the user research provided a relatively clear idea and understanding of domain- and user needs, this initial design phase provides a high-level design with concepts identification, conceptual modeling and early prototyping. During initial conceptualization and high-level design, we focused on Brainstorming sessions and contextual analysis to build an initial concept of the application. We gradually consolidated it into a set of requirements on flip charts and PowerPoint slides. The main goal of high-level design is to plot down schematic ideas and steps into visual graphs and models; an early blueprint. We started by investigating different options and providing design alternatives to make sure we have a broad view before identifying a good design. Doing this early on, at high-level, sketchy, paper-based only, and without going into details help provide several solution alternatives at a very low cost.

The IMODS system is conceptualized such that a course design is centered around the learning objectives of the course defined by the instructor (user) as shown in Figure 4. Learning objectives are directly associated with the course content, assessments, and pedagogical activities as defined in the PC^3 model (Figure 5).

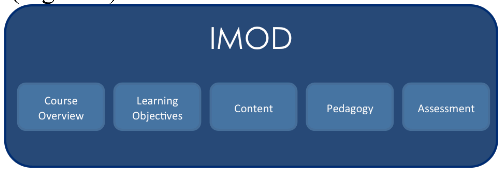


Figure 4: IMODS System Overview

The Learning Objectives component of the IMODS system was conceptualized with it various components based on the PC^3 model as shown in Figure 5.

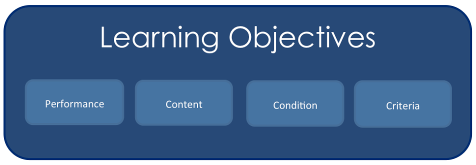


Figure 5: IMODS Learning Objective Overview

Then as we planned and executed the testing, we were able to directly ask the user to work on a "Criteria" within "learning objectives". Our dual purpose is to let the user know that those are two concepts of the IMOD, as well as knowing the structural relationship between them.

B. Development Phase

In order to validate that we were proceeding in the right direction, we ran a series of usability tests on the application. We chose between multiple good designs instead of focusing on only one early on. The high-level design sketches were discussed with the users to make sure what they said in unstructured dialogs and vague ideas and imaginations can now be concretely captured in design artifacts for further validation and clarifications [17]. Our main goal was to evaluate the simplicity and clarity of the application structure to allow for an easy-to-recognize mental model. A mental model can be loosely defined as the user perception of the application. The opposite is the developers' perception of the application. While the latter one is the actual structure that developers use to build the application, typically as their interpretation of the requirements, the user mental model is not necessarily the same. With the fact that users don't normally have access to the actual structure of the application, or detailed and prolonged access to the application to know any of it's internal structure, they can only perceive what's exposed to them from the UI, and can build an "imagination: of what the application structure might look like. In ideal situations, this "imaginary" structure should match the actual structure build by developers. In reality, though, it is rarely the case. A discrepancy or vagueness on the user mental model (we can call it a delta) is typically present and expected. The problems arise when this delta is large, indicating an application whose structure is not comprehensible by the user. We have identified 2 tools that are most suitable for this project in this phase.

C. Tools

Navigation Model is one of the essential methods of design that we used. A significant challenge in complex software is not the contents of each screen, but how the user mentally build a mental view of how all screens are connected (like a city road map), and how to navigate between hundreds of screens to accomplish their task. In this regard, we have developed an effective technique, elastic prototyping, an implementation of a participatory design to help designers and users build a navigation model together, greatly reducing time and effort needed. Figure 6 shows the navigation model for the primary application. Figures 7 and 8 show the navigation model for new user registration and user login. One of the

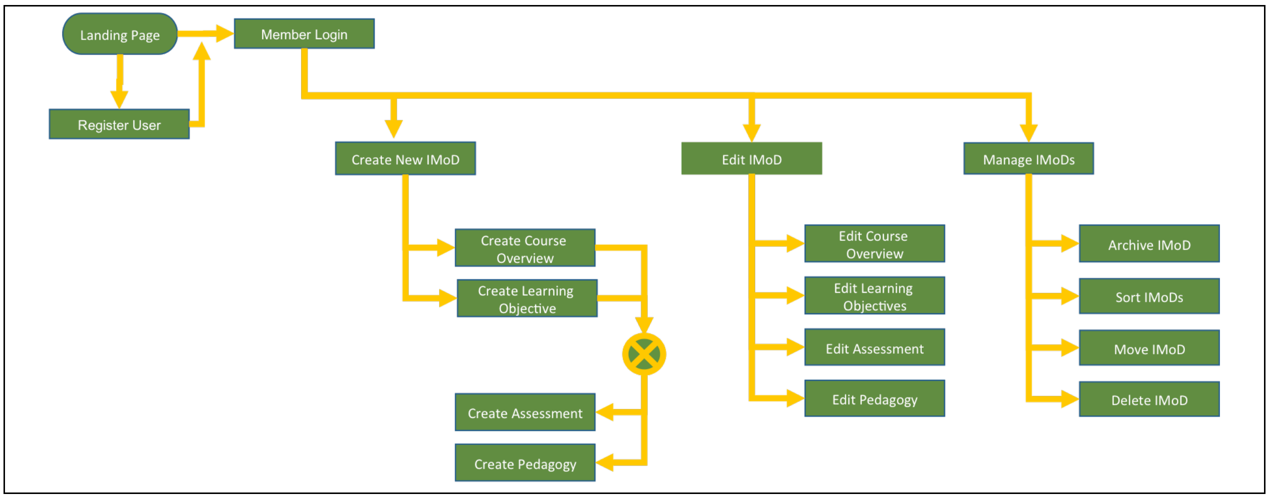


Figure 6: Primary IMODS Application - Navigation Model

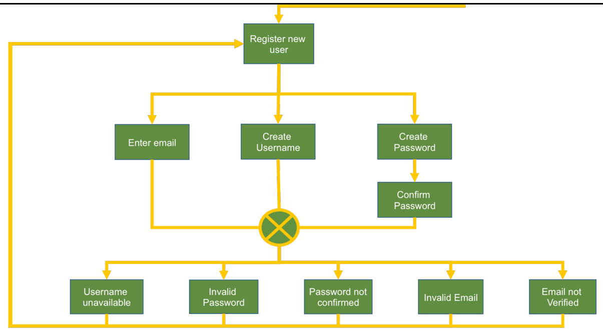


Figure 7: New User Registration - Navigation Model

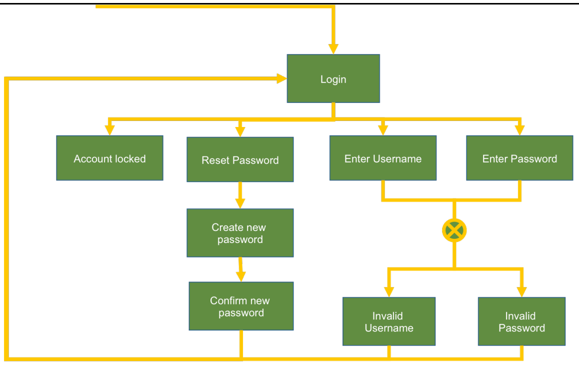


Figure 8: User Login - Navigation Model

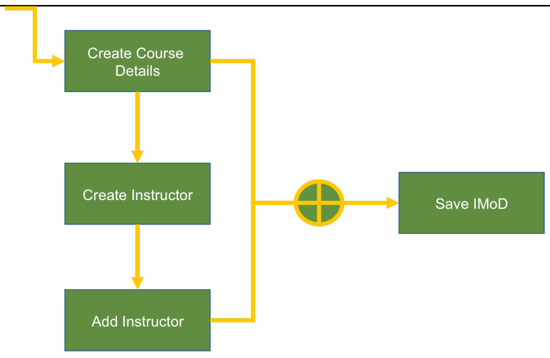
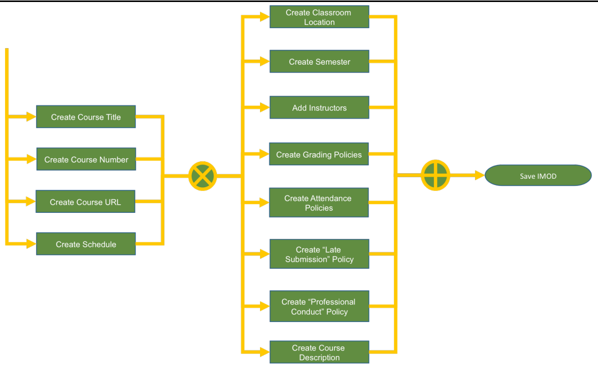
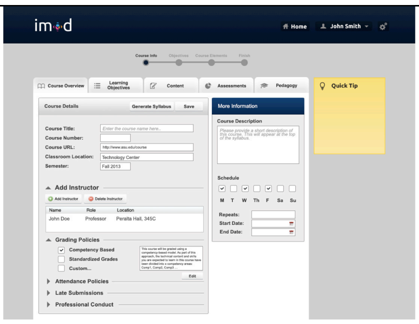


Figure 9: Course Overview - Navigation Model



Create Course Title
Create Course Number
Create Course URL
Create Schedule
Create Classroom Location
Create Semester
Add Instructors
Create Grading Policies
Create Attendance Policies
Create "Late Submission" Policy
Create "Professional Conduct" Policy
Create Course Description
Save IMOD

Figure 10: Course Details - Navigation Model



Course Title: Enter the course name here..
Course Number:
Course URL: http://www.asu.edu/course
Classroom Location: Technology Center
Semester: Fall 2013

Add Instructor

Name	Role	Location
John Doe	Professor	Peralta Hall, 345C

Grading Policies
[✓] Competency Based
[ ] Standardized Grades
[ ] Custom...

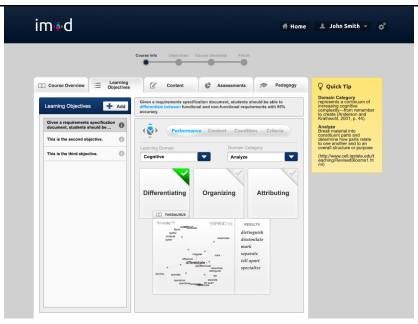
This course will be graded using a competency-based model. As part of this approach, the technical content and skills you are expected to learn in this course have been divided into x competency areas: Comp1, Comp2, Comp3 ...

Attendance Policies
Late Submissions
Professional Conduct

Course Description
Please provide a short description of this course. This will appear at the top of the syllabus.

Schedule
[✓] M [ ] T [✓] W [ ] Th [✓] F [ ] Sa [ ] Su
Repeats:
Start Date:
End Date:

Figure 11: Course Overview Mockup



Given a requirements specification document, students should be ...
This is the second objective.
This is the third objective.

Given a requirements specification document, students should be able to differentiate between functional and non-functional requirements with 95% accuracy.

Learning Domain: Cognitive
Domain Category: Analyze

Differentiating ✓ | Organizing | Attributing

Results: distinguish, dissimilate, mark, separate, tell apart, specialize

Quick Tip

Domain Category represents a continuum of increasing cognitive complexity—from remember to create (Anderson and Krathwohl, 2001, p. 44).

Analyze Break material into constituent parts and determine how parts relate to one another and to an overall structure or purpose

(http://www.celt.iastate.edu/teaching/RevisedBlooms1.html)

Figure 12: Learning Objectives Mockup

main components of course design is describing course overview information that includes data about course title, description, schedule, instructors, course policies, etc. Figures 9 and 10 show the navigation model for course overview data entry. In a similar manner navigation model for other screens of IMODS that are used for design of Learning Objectives, Content, Assessments, and Pedagogy were created.

Prototyping (PT) is extensively used in UCD to visualize and validate all otherwise vague ideas and unclear expectations at low cost and high effectiveness. We focused on three main categories of prototyping: Paper (low-level) PT, low-fidelity electronic (medium level) PT, and high-fidelity, detailed PT [18]. Paper prototypes are very inexpensive and help us capture several initial ideas and concepts, and validate them. After explaining their needs, users often change their minds when they see them on paper. Therefore multiple paper PT sessions gives a head start in validating what users actually mean and need. After initial concepts, design ideas and directions were identified, we moved into a medium fidelity prototyping stage where we provided a sketchy visualization of key screens without contents and gradually validated them and added initial contents. Figures 11 and 12 show the user interface mockups of Course Overview and Learning Objective components of the IMODS system

D. System Architecture

The development phase of the project included identifying appropriate technologies to be used for the development of the IMODS semantic web application, design of the back-end database schema, installation and configuration of the server-side and client-side technologies, and development of the user interface screens for login, registration, index, and creation of an instructional module and the connectivity of these web pages with the backend database. An Agile software development methodology called Scrum is being used for the development of this project. Scrum is an iterative and incremental framework for managing product development.

The technologies chosen included Groovy on Grails, an open source, full stack, web application framework for the Java Virtual Machine. It takes advantage of the Groovy programming language and convention over configuration to provide a productive and streamlined development experience. Grails uses Spring Model-View–Controller (MVC) architecture as the underlying framework. MVC is a software architecture pattern that separates the representation of information from the user's interaction with it. PostgreSQL was chosen as the database management system. It is a powerful, open source object-relational database system with more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. Git was chosen for source code version control. It is a distributed revision control and source code management (SCM) system with an emphasis on speed.

E. Testing

For the testing phase of the project, we opted to not have a complete discovery test, where the user would be asked to do a blind discovery of the application without any prior knowledge [19]. That would be a simulation of a completely novice user without any application background. Instead, we decided to test for an average user with some level of knowledge about the application structure. The reason is that we already have a concrete navigation model, and we can typically bring it to any user's attention in few minutes to help them in building a correct navigation model. That is one of the main advantages of using the navigation modeling method. Currently our research team is working on software development and testing phases of the project.

IV. CASE STUDY

The IMODS framework was applied to design an introductory software engineering course titled "Software Enterprise I: Personal Software Process" in B.S. in Software Engineering program. This section describes the use of IMODS – PC^3 model for course design.

A. About the Course

Software Enterprise I: Personal Software Process is a sophomore course in the Software Engineering program that introduces software engineering and object-oriented software design principles using a modern programming language. Students are introduced to Software Engineering, Software Life Cycle models, Object-Oriented Programming, Personal Software process, Effort estimation, effort tracking, defect estimation and defect tracking. Students learn personal software process for individual professionalism, time and defect estimation; yield and productivity. A project-based pedagogical model is used for delivery of all our courses in Software Engineering program. Students in this course worked on a game project using Java programming language.

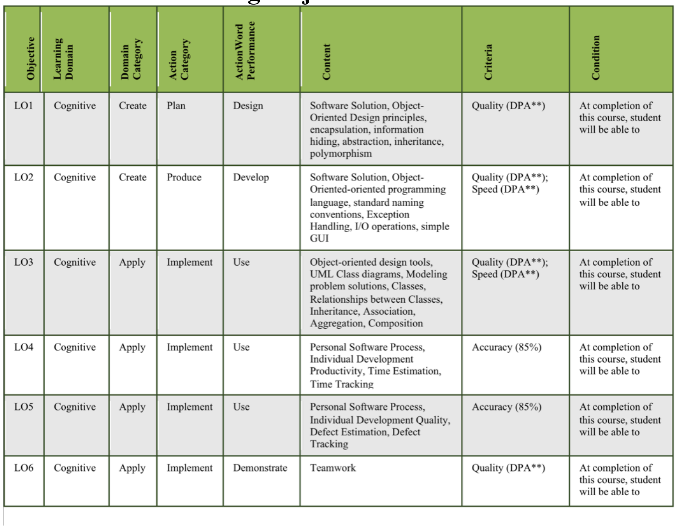
B. Learning Objectives

Learning objectives of this course were defined using the PC^3 model. The course has 6 objectives that are categorized under Performance, Content, Condition, and Criteria as shown in the Table 1. The objectives are as follows:

- *LO1*: Design a software solution using Object-Oriented Design principles of encapsulation, information hiding, abstraction, inheritance, and polymorphism
- *LO2*: Develop a software solution in an object-oriented programming language employing standard naming conventions and making appropriate use of advanced features such as exception handling, I/O operations, and simple GUI
- *LO3*: Use object-oriented design tools such as UML class diagrams to model problem solutions and express classes and relationships such as inheritance, association, aggregation, and composition

- *LO4*: Use personal software process for individual development productivity through time estimation and tracking
- *LO5*: Use personal software process for individual development quality through defect estimation and tracking
- *LO6*: Demonstrate teamwork

Table 1: Learning Objectives based on PC^3 Model



Objective	Learning Domain	Domain Category	Action Category	ActionWord Performance	Content	Criteria	Condition
LO1	Cognitive	Create	Plan	Design	Software Solution, Object-Oriented Design principles, encapsulation, information hiding, abstraction, inheritance, polymorphism	Quality (DPA**)	At completion of this course, student will be able to
LO2	Cognitive	Create	Produce	Develop	Software Solution, Object-Oriented-oriented programming language, standard naming conventions, Exception Handling, I/O operations, simple GUI	Quality (DPA**); Speed (DPA**)	At completion of this course, student will be able to
LO3	Cognitive	Apply	Implement	Use	Object-oriented design tools, UML Class diagrams, Modeling problem solutions, Classes, Relationships between Classes, Inheritance, Association, Aggregation, Composition	Quality (DPA**); Speed (DPA**)	At completion of this course, student will be able to
LO4	Cognitive	Apply	Implement	Use	Personal Software Process, Individual Development Productivity, Time Estimation, Time Tracking	Accuracy (85%)	At completion of this course, student will be able to
LO5	Cognitive	Apply	Implement	Use	Personal Software Process, Individual Development Quality, Defect Estimation, Defect Tracking	Accuracy (85%)	At completion of this course, student will be able to
LO6	Cognitive	Apply	Implement	Demonstrate	Teamwork	Quality (DPA**)	At completion of this course, student will be able to

** DPA Determined Per Assessment

C. Content

The list of Content topics and subtopics are listed in Table 2. For each topic the knowledge dimension and topic priority is defined. This information is used to find assessments and instructional activities that best fit for delivering a topic.

Table 2: Content Topics based on PC^3 Model

Content Topic	Content Sub-Topics	Knowledge Dimension	Priority
Object-Oriented Design Principles	Encapsulation	Factual (F), Conceptual (C)	Critical (3)
	Information Hiding	Factual (F), Conceptual (C)	Critical (3)
	Abstraction	Factual (F), Conceptual (C)	Critical (3)
	Inheritance	Factual (F), Conceptual (C)	Critical (3)
	Polymorphism	Factual (F), Conceptual (C)	Critical (3)
	Software Solution	Conceptual(C), Metacognitive (M)	Critical (3)
Object-Oriented Design tools	Modeling Problem solution	Procedural (P), Metacognitive (M)	Critical (3)
	UML Class diagram	Procedural (P)	Critical (3)
	UML Use Case Diagram	Procedural (P)	Important (2)
Object-Oriented Programming Language	Exception Handling	Factual (F), Conceptual (C)	Important (2)
	I/O Operations	Factual (F), Conceptual (C)	Important (2)
	Simple GUI	Factual (F), Conceptual (C)	Important (2)
	Standard naming conventions	Factual (F), Conceptual (C)	Good to know (1)
Personal Software Process	Time Tracking	Factual (F), Procedural (P)	Critical (3)
	Time Estimation	Factual (F), Procedural (P)	Critical (3)
	Defect Tracking	Factual (F), Procedural (P)	Critical (3)
	Defect Estimation	Factual (F), Procedural (P)	Critical (3)
Teamwork	-	Metacognitive (M)	Important (2)

D. Assessments

Assessments chosen for this course include a mix of both formative and summative assessments. The PC^3 model aligns assessments chosen for the course with the learning objectives by checking compatibility of learning domains, performance, and criteria requirements. Table 3 provides the list of assessments with their corresponding learning domain category, knowledge dimension, and criteria type that each method is suitable for.

E. Instructional Activities

Pedagogical activities used in this course are listed in Table 4 along with the knowledge dimension and learning domain category that they are suitable for. The list of activities includes a mix of lectures, lab activities, Q&A discussions, and problem solving activities.

Table 3: Course Assessments

Assessment	Type	Domain Category	Knowledge Dimension	Criteria
Programming exercise (write formal code)	Formative	Understand, Apply, Analyze, Evaluate, Create	Conceptual, Procedural	Speed, Quality, Accuracy
Partially guided programming exercise	Formative	Understand, Apply, Analyze, Evaluate	Conceptual, Procedural	Quality, Accuracy
Guided Lab exercise	Formative	Understand, Apply, Analyze, Evaluate	Conceptual, Procedural	Quality, Accuracy
Quiz	Formative	Remember, Understand	Conceptual, Factual	Accuracy, Speed
Project	Summative	Understand, Apply, Analyze, Evaluate, Create	Conceptual, Procedural	Quality, Accuracy
Exam	Summative	Understand, Apply, Analyze, Evaluate	Conceptual, Procedural	Quality, Accuracy

F. Results

Software Enterprise I: Personal Software Process course in the Software Engineering program in School of Computing, Informatics, Decision Systems Engineering (CIDSE) at Arizona State University was designed using the IMODS – PC^3 model and offered as a face-to-face section (with 82 students) as well as an online section (with 87 students) by the same instructor (one of the co-authors). Using the IMODS framework ensured the alignment between various course elements and thereby ensuring high-quality course design.

Table 4: Course Pedagogical activities

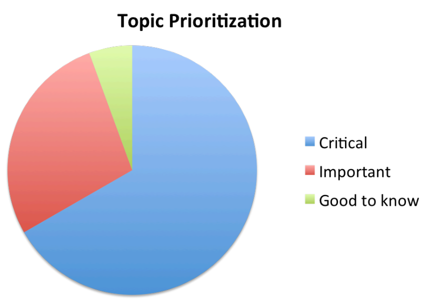
Instructional Activities	Domain Category	Knowledge Dimension
Lectures (Face-to-Face, Audio, or Video)	Remember, Understand	Conceptual, Factual
Problem Solving	Understand, Apply, Analyze, Evaluate	Conceptual, Procedural
Partially Guided Programming exercise	Understand, Apply, Analyze, Evaluate	Conceptual, Procedural
Active Reading	Remember, Understand, Apply	Conceptual, Factual, Procedural
Programming Assignment	Understand, Apply, Analyze, Evaluate, Create	Conceptual, Procedural
Guided Lab Exercise	Understand, Apply, Analyze, Evaluate	Conceptual, Procedural
Q&A Forum	Remember, Understand	Factual, Conceptual

Alignment between various course components:

The framework supports the checking of alignment between course assessments and learning objectives. The course assessments are linked to the performance and criteria elements of the learning objective as shown in Figure 1. The framework supports the checking of alignment between course instructional activities and learning objectives. The course pedagogical activities are linked to the performance and content of the learning objective as shown in Figure 1.

Topic Prioritization:

Use of the PC^3 model ensured a balanced distribution of the topics under Critical, Important, and Good to know as shown in figure below.



V. Future Work

Following the high-level design phase of the project, the next step will be software development and testing of IMODS. We will conduct usability testing of the prototype with instructors and solicit feedback using surveys, observation, and user interviews. The feedback will be incorporated into the iterative software development lifecycle model. The scope of this project will also include the evaluation of its novel approach to self-guided web-based professional training in terms of: 1) user satisfaction with the documentation of course designs generated; and 2) impact on users' knowledge of the outcome-based course design process.

Acknowledgment

The authors gratefully acknowledge the support for this project under the National Science Foundation's Transforming Undergraduate Education in Science, Technology, Engineering and Mathematics (TUES) program Award No. DUE-1246139.

References

[1] R. M. Felder, R. Brent, and M. J. Prince, "Engineering Instructional Development: Programs, Best Practices, and Recommendations.," *Journal of Engineering Education*, vol. 100, no. 1, pp. 89–122, Jan. 2011.

[2] R. Boice, *Advice for new faculty members*. Allyn & Bacon, 2000.

[3] R. M. Harden, J. R. Crosby, M. H. Davis, and M. Friedman, "AMEE Guide No. 14: Outcome-based Education: Part 5--From Competency to Meta-Competency: A Model for the Specification of Learning Outcomes.," *Medical Teacher*, vol. 21, no. 6, pp. 546–552, 1999.

[4] W. G. Spady and K. J. Marshall, "Beyond Traditional Outcome-Based Education.," *Educational Leadership*, vol. 49, no. 2, pp. 67–72, 1991.

[5] G. Adorni, S. Battigelli, D. Brondo, N. Capuano, M. Coccoli, S. Miranda, F. Orciuoli, L. Stanganelli, A. M. Sugliano, and G. Vivanet, "CADDIE and IWT: two different ontology-based approaches to Anytime, Anywhere and Anybody Learning," *Journal of e-Learning and Knowledge Society-English Version*, vol. 6, no. 2, 2010.

[6] S. Ternier, E. Duval, and P. Vandepitte, "LOMster: peer-to-peer learning object metadata," in *Proceedings of EdMedia*, 2002, pp. 1942–1943.

[7] "LOCO-Analyst." [Online]. Available: http://jelenajovanovic.net/LOCO-Analyst/index.html. [Accessed: 28-May-2012].

[8] G. P. Wiggins and J. McTighe, *Understanding by design*. Association for Supervision & Curriculum Development, 2005.

[9] J. Warren, "Changing community and technical college curricula to a learning outcomes approach," *Community College Journal of Research &Practice*, vol. 27, no. 8, pp. 721–730, 2003.

[10] R. A. Streveler, K. A. Smith, and M. Pilotte, "Aligning Course Content, Assessment, and Delivery: Creating a Context for Outcome-Based Education," *K. Mohd Yusof, S. Mohammad, N. Ahmad Azli, M. Noor Hassan, A. Kosnin and S. K, Syed Yusof (Eds.), Outcome-Based Education and Engineering Curriculum: Evaluation, Assessment and Accreditation. Hershey, Pennsylvania: IGI Global*, 2012.

[11] R. F. Mager, "Preparing Instructional Objectives: A critical tool in the development of effective instruction 3rd edition," *The Center for Effective Performance, Inc*, 1997.

[12] K. Andhare, O. Dalrymple, and S. Bansal, "Learning Objectives Feature for Instructional Module Development System," presented at the PSW American Society for Engineering Education Conference, San Luis Obispo, California, 2012.

[13] A. Gaffar, "Enumerating mobile enterprise complexity 21 complexity factors to enhance the design process," in *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research*, 2009, pp. 270–282.

[14] A. Gaffar, N. Moha, and A. Seffah, "User-Centered Design Practices Management and Communication," in *Proceedings of HCII*, 2005.

[15] S. Bansal, O. Dalrymple, A. Gaffar, and R. Taylor, "User Research for the Instructional Module Development (IMOD) System," in *American Society for Engineering Education Annual Conference (ASEE)*, Indianapolis, IN, 2014.

[16] O. Dalrymple, S. Bansal, A. Gaffar, and R. Taylor, "Instructional Module Development (IMOD) System: A User Study on Curriculum Design Process," in *Frontiers in Education (FIE)*, Madrid, Spain, 2014.

[17] J. Lazar, J. H. Feng, and H. Hochheiser, *Research methods in human-computer interaction*. John Wiley & Sons Inc, 2009.

[18] W. Lidwell, K. Holden, and J. Butler, *Universal principles of design: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Rockport Pub, 2010.

[19] N. Moha, A. Gaffar, and G. Michel, "Remote usability evaluation of web interfaces," *Human Computer Interaction Research in Web Design and Evaluation. P. Zaphiris and S. Kurniawan. Hershey, PA, Idea Group Publishing*, pp. 273–289, 2007.