International Journal *of* Designs for Learning
2019 | Volume 10, Issue 1 | Pages 131-144

IMPLEMENTING INDIVIDUALIZED LEARNING IN A LEGACY LEARNING MANAGEMENT SYSTEM: A FEASIBILITY PROTOTYPE FOR AN ONLINE STATISTICS COURSE

M. Earnest Morrow[1] & Dabae Lee[2]
[1]*Sam Houston State University;* [2]*Emporia State University*

Educators are being encouraged to shift their instructional paradigm from teacher-centered to learner-centered through the use of technology. For online courses, legacy learning management products originally designed to support and deliver teacher-centered instruction may represent a constraint to implementing the learner-centered paradigm. Yet, replacement of these systems presents a formidable hurdle to educators wishing to initiate learner-centered online courses. This hurdle could be lowered significantly by a transitional approach that allows learner-centered strategies to be delivered within the framework of existing learning management systems. This paper describes our efforts to prototype such a transitional approach for an online statistics course. Pedagogical and technological objectives were successfully achieved by combining the technologies of the Sharable Content Object Reference Model (SCORM), a legacy learning management system, and a stand-alone course authoring tool to deliver an example course demonstrating adaptive, competency-based student progress instruction that personalizes one's learning path with topic-contingent assessment feedback.

M. Earnest Morrow is completing an educational doctorate (EdD) in Instructional Systems Design and Technology at Sam Houston State University after retiring from an extensive career in information technology management at a major multinational corporation. Mike's objective is to redirect his experience in technology and corporate management practices to issues in the field of Computer Science Education. The subject of his dissertation concerns quantitative analysis of a cardsorting instrument for the assessment of computational thinking skills acquisition in a higher education setting.

Dabae Lee is an assistant professor of Instructional Technology at Emporia State University. She teaches undergraduate and graduate courses in instructional technology, program evaluation, and research methods. Her research interests include personalized learning, student collaboration in learner-centered instructional methods, roles of technology in learner-centered paradigm, active learning spaces, and research methods in Instructional Technology.

INTRODUCTION

Educators are being encouraged to shift their instructional paradigm from teacher-centered to learner-centered in order to transform education through the use of technology (U.S. Department of Education, 2010). Making such a shift requires changes to not only instructional methods and strategies, but also places new requirements on the supporting technology systems such as network infrastructure, and learning management systems (LMS) (Watson & Reigeluth, 2008). This is especially true for online education which relies on the LMS to deliver instructional content and assessments as well as provide access to collaborative activities and communications. Legacy LMS products, such as Blackboard, or Moodle, originally designed to support and deliver teacher-centered instruction may represent a constraint to implementing the learner-centered paradigm (Aslan, Huh, Lee, & Reigeluth, 2011; Yildirim, Reigeluth, Kwon, Kageto, & Shao, 2014). Yet, replacement of these systems with more collaboratively oriented products like Schoology or Edmodo is a disruptive and potentially expensive undertaking that presents a formidable hurdle for educators wishing to initiate learner-centered online courses. This hurdle could be lowered significantly by a transitional approach that allows learner-centered strategies to be delivered within the framework of existing LMSs.

This design case describes our efforts to prototype such a transitional approach and thereby demonstrate the technical feasibility of implementing a personalized learning path model in a legacy LMS for an online statistics course.

https://doi.org/10.14434/ijdl.v10i1.22500

BACKGROUND

A southern university recently began offering an online doctoral program in Instructional Technology. Course offerings and course objectives for the program's curriculum were developed to focus on leadership in instructional technology through research and experiential assignments. Part of the experiential learning aspect of the program was for the doctoral students to engage with the instruction solely online and thereby better learn to lead instructional technology by having to utilize instructional technology. Furthermore, this paper's secondary author identified the course in statistical methods as a potential learning experience in advanced instructional technology. She proposed an instructional design based on competency-based progress with flexible pacing (Johnson, 2000; Mills, Ablard, & Gustin, 1994; Reigeluth & Karnopp, 2013) Then, the first author of the study developed prototypes to test the feasibility of implementing the proposed instructional design in the legacy LMS. This paper documents our efforts to demonstrate the technical feasibility of implementing competency-based instruction with flexible pacing for use in the online statistical methods course.

Pedagogical Objectives

Learning statistics can be more effective with a sequential approach that allows the learner to gain understanding of the fundamentals, and build connections between topics before advancing to more complex topics (Bunderson, Wiley, & McBride, 2009; Kulik, Kulik, & Bangert-Drowns, 1990; Posner, 2011; Stodolsky, 1988). This sequential nature of the subject lends itself to flexible pacing of online instruction with competency-based assessment of student progress (Johnson, 2000; Mills, Ablard, & Gustin, 1994; Reigeluth & Karnopp, 2013). The model we propose in this paper we call Personalized Learning Path (PLP), which is an individualized learning approach that enables alternative learning paths based on students' learning needs. As one element of a personalized learning approach to instruction, individualized learning is designed to be paced to an individual learner's specific learning needs (Software & Information Industry Association, 2010; U.S. Department of Education, 2010).

The PLP model allows Competency-Based Student Progress (CBSP), which derives from Bloom's work on mastery-based learning (Bloom, 1968). Three principles guide the design of CBSP instruction: 1) use ongoing formative assessments to identify difficulties in mastering learning objectives; 2) take corrective action following assessments; and 3) require learners to demonstrate mastery of a learning objective in order to proceed to the next objective (Lee, 2014). Based on these principles, the following instructional strategies were developed for the online doctoral statistical methods course.

- Specifically, it is intended that the instructional content for the statistical methods course will be presented sequentially in single topic chunks which have been identified and sequenced through domain-mapping (Bunderson, Wiley, & McBride, 2009). The presentation of each topic is to be immediately followed by a diagnostic quiz to evaluate the student's progress towards competency of that topic. If sufficient mastery is demonstrated, the instructional sequence moves forward to the next topic. If not, the sequence reverts back to the specific topic that can correct the misunderstood concept.
- Immediate and topic-contingent feedback to each assessment item allows learners to identify and correct their errors and misconceptions early in the learning process (Mason & Bruning, 2001). It also promotes learner motivation as it provides immediate and specific feedback (Cohen, 1985).
- As the instructional sequence progresses, more advanced topics may require the learner to construct relationships with concepts from previous topics. Therefore, assessing the learner's understanding and mastery of these new topics requires diagnostic tests that are crafted to explore the relationships among the new and prior concepts. The topic-contingent feedback associated with evaluating these multi-concept relationships may, as a result, require a learner to revert to specific earlier topics multiple times from multiple advanced topics to develop the targeted understanding. Due to this dynamic sequencing feature, individual students are likely to follow different paths through the course content and assessments. We refer to such progressions through the material as PLPs. This is illustrated in Figure 1.

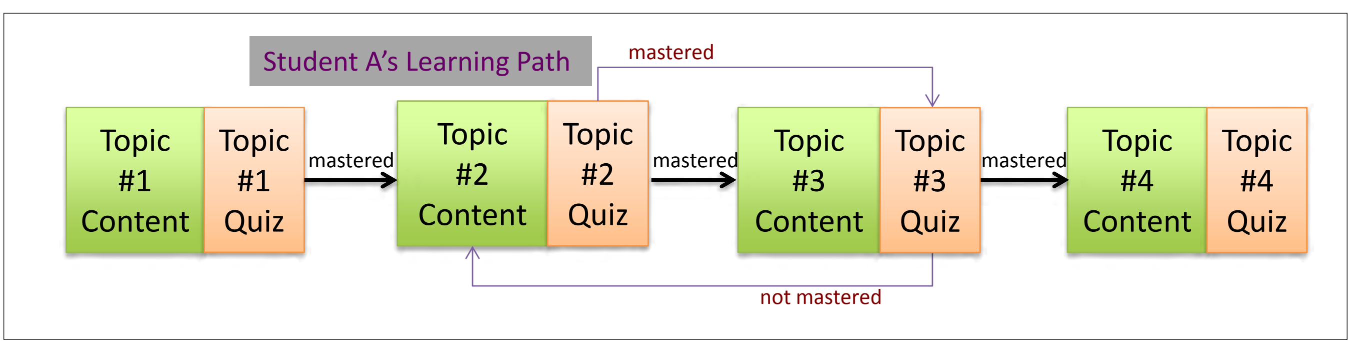


FIGURE 1. Personalized Learning Path (PLP) example.

Technical Objectives

The University's LMS (Blackboard Learn) provides the online learning service. Therefore, Blackboard is the default choice for our delivery of the CBSP oriented statistics course. Interviews with management of the learning service confirmed their understanding of the instructional strategies and support for hosting the technical implementation. However, to their knowledge, no other courses in the University's online learning service have utilized adaptive instructional technology. Therefore, before proceeding to the development of the statistical methods course for the doctorate of instructional technology program, we wanted to establish confidence in the technical feasibility of delivering competency-based student progress online instruction through this existing LMS.

To gain this confidence, we set a goal of developing an example course for delivery in Blackboard which would demonstrate the following capabilities:

- Sequence learners through single topic chunks of instructional content.
- Follow the presentation of each topic with a formative assessment of the learner's mastery of the topic and its relationships to previous topics.
- Initiate remedial action for incorrect assessment item responses.
- Randomly draw the questions for each topic assessment from a pool of questions for that topic because some topics may be assessed multiple times. Therefore, no assessments will be the same for any two learners, and questions are unlikely to be repeated for any individual learner.
- Record attainment of competency for each topic presented in the course in the LMS gradebook, each time that topic is assessed. Evaluation and analysis of the gradebook should be by these attainment objectives, as opposed to solely by the aggregate scores of each quiz.
- Collect or derive data on the actual sequence through the instructional topics followed by each learner. Topics which a learner fails to master the first time will be listed additional times within these sequences. If many learners show difficulty with the same topic it will be an indication that the instructional design for that topic needs to be examined for improvement. Constructing these personalized learning paths from the LMS gradebook can be accomplished by sequencing every access to each topic by the date and time stamps.

PROCESS

Our goal in this design case was to determine a technically feasible approach for delivering CBSP instructional sequencing with topic-contingent feedback through the University's existing LMS, Blackboard Learn. We began the process with the secondary author creating example course materials to serve as an illustration of the pedagogical objectives. The primary author then followed an iterative, rapid development approach beginning with the existing features of Blackboard, and then sought ways to extend the LMS with third-party add-ins for Blackboard, the use of custom code (i.e., Building Blocks), and web-based interfaces, It was recognized that use of existing add-in or interfaces would be preferable to the development of custom code. Successive iterations of various technical approaches were constructed, evaluated, and adapted until the desired features of the example course were fully realized.

Example Course Material

We developed example course material as outlines of four topics in Statistics (Topic 1: Average, Topic 2: Mode, Topic 3: Median, and Topic 4: Standard deviation). Each of these topics is followed by a diagnostic quiz containing closed ended question and answer options. Topic-contingent actions to be taken for each possible answer are specified. Most of the questions and actions are like those in Figure 2.

2-1. What is the mode of the following data: 2, 3, 4, 5, 6, 6, 7, 7, 7, 8, 8, 9

a. 5
b. 6
c. 7
d. 7.5
e. 8

Correct answer: c
Other responses
- All other responses: Prompt to try one more time. When a wrong answer is chosen again, ask to Review Topic 2.

FIGURE 2. Sample question with single contingent response.

However, some of the questions have multiple possible actions depending upon the student's answer, such as those in Figure 3.

4-4. Which of the following is most sensitive to outliers?

a. interquartile range
b. standard deviation
c. median
d. mode

Correct answer: b
Other responses
- a: Review Topic 4 - Other measures of spread
- c: Review Topic 3
- d: Review Topic 2

FIGURE 3. Sample question with multiple contingent responses.

There is also the question shown in Figure 4, which includes nested questions as another aid to building the student's understanding of the problem and the concepts involved.

These last two examples illustrate the granularity of the decision points and the variety of possible remedial actions to be taken that can be required to provide topic-contingent feedback. Such factors can pose significant technical challenges for many commercial course authoring tools. So

3-4. A test was given to 60 individuals. Half answered 80% of the questions correctly, the other half answered 90% correctly. Which of the following statements is correct?

a. mean > mode
b. mean = median
c. mean < median
d. mode < median

Correct answer: b

Other responses:

- a: Ask: What is the mode value?
  - 90
    - Ask: What is the mean value?
      - 85 -> Try Q 3-4 again.
      - Anything else -> Review Topic 1.
  - Anything else: Review Topic 2.
- c: Ask : What is the mean value?
  - 85
    - Ask: What is the median value?
      - 85 -> Try Q 3-4 again.
      - Anything else -> Review Topic 3.
  - Anything else: Review Topic 1.
- d: Ask: What is the mode value?
  - 90
    - Ask: What is the median value?
      - 85 -> Try Q 3-4 again.
      - Anything else -> Review Topic 3.

Anything else: Review Topic 2.

FIGURE 4. Sample question with nested contingent responses.

we first had to determine if the native features of Blackboard Learn could meet these needs.

Iteration 1: Blackboard Native Features

Our initial implementation effort for the example course focused on identifying Blackboard's inherent capabilities for meeting our technological objectives through a review of literature (Bremer & Bryant, 2005; Dobre, 2015), online documentation (Blackboard, Get started, n.d.), and course development help articles for higher education faculty (Bernardi, 2015; Northeastern University, 2014; University of Massachusetts, 2014). These capabilities were then explored by building quick, simple course structures using Blackboard's native builder functions. Successive attempts implemented alternative approaches for adapting Blackboard capabilities to the project objectives. We soon identified Blackboard's Adaptive Release feature as the most promising native capability. Several test versions were developed using this feature. With each attempt, deficiencies were identified and new adaptations designed to address them.

Blackboard's Adaptive Release feature allows content items to be hidden until certain conditions are met (Bernardi, 2015). For the example course, we started by placing an adaptive release rule on Topic 2 and Quiz 2 causing them not to display until the student achieves a passing score on Quiz 1. All following topics and quizzes could be hidden likewise. So, upon first entering the course, a student would only see Topic 1 and Quiz 1. After passing Quiz 1, Topic 2 and Quiz 2 would appear.

If the student then incorrectly answers a topic 1 related question within Quiz 2, the LMS needs to direct the student back to review the Topic 1 content. However, the adaptive release rule in Blackboard Learn does not have the granularity to identify the question in Quiz 2 with the incorrect answer unless the questions for each topic are isolated to their own quiz. So Quiz 2 has to be replaced with two quizzes: Quiz 2–Topic 1 and Quiz 2–Topic 2. Likewise, Quiz 3 might be replaced with 3 topic specific quizzes.

Adaptive release rules could then be applied to take action when a topic needs to be presented for review. However, the only action available with the rule is to show an initially hidden content item. The presentation to the student cannot be moved forward or backward in the list of course content items. That is, there is no mechanism to alter the presentation sequencing to the learner. To address this limitation, we hid the previous content items needed for review prior to each quiz. Adaptive release rules could then be applied to these review items so they only display if the score for the following quiz is non-passing. Thus, if the student fails a quiz, then an associated topic review content item immediately preceding the student's current position will become available. The learner must then manually select this "released" content for review. This leads to a course structure as shown in Figure 5.

Obviously, as more topics are added to the course, such a course structure becomes more complex and redundant.

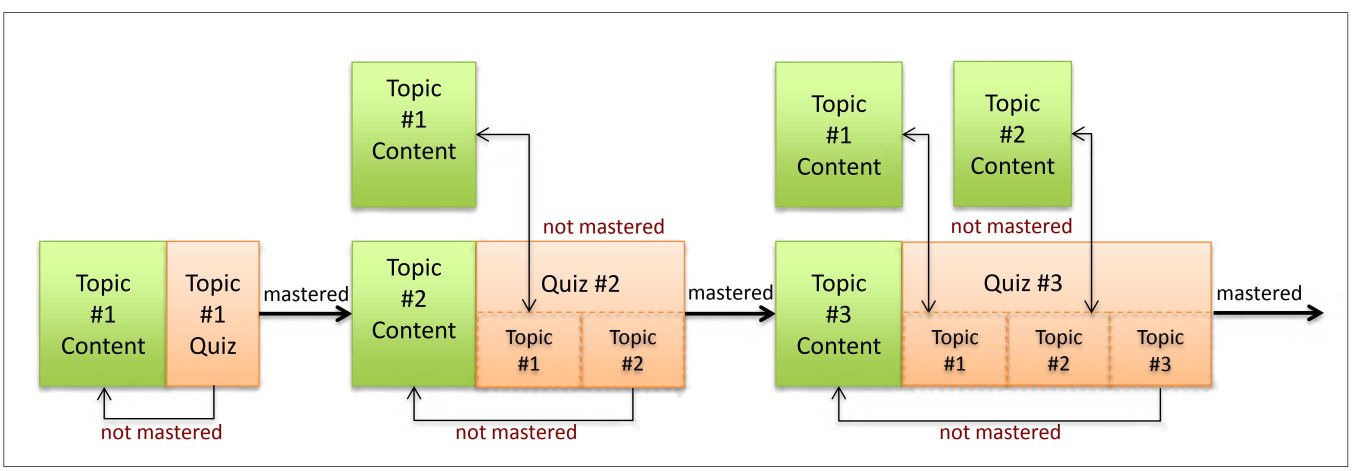


FIGURE 5. Example PLP constructed with Blackboard Adaptive Release rules.

Although the content pages for each topic (and its associated review instances) can be drawn from a Blackboard content resource pool to minimize redundancy in composing and editing content, the course structure still quickly grows to become unwieldy.

This course structure also results in a user experience that is not intuitive. Previously shown course content items will disappear, and then reappear if needed for review. The user will then have to "back-up" to this previous topic in order to review the material. Some users may be frustrated by this visible indication of backward movement.

Iteration 1 Summary

The comparison between the capabilities of Blackboard and the design objectives concluded that Blackboard's native course building features were not adequate to create an acceptable adaptive learning experience. The quiz results were not sufficiently granular to identify the specific topics requiring review when multiple topics are included in a single formative assessment. We worked around this limitation by splitting an assessment into multiple, topic specific, quizzes. However, this added complexity to the construction and navigation of the course. Also, the coarse granularity made it impossible to implement a single quiz item that tests on multiple topics. Furthermore, only gradebook scores can trigger a release rule. Other variables cannot be defined, set, or referenced and this limits the logic design of a course structure.

We were not surprised by finding these limitations, as we had not expected native Blackboard capabilities to meet our needs. However, this iteration was valuable in adding depth to our understanding of the design challenges.

Iteration 2: Extending Blackboard

By brainstorming on potential technical means of extending Blackboard to address the limitations of Iteration 1, we identified a need to embed programmable sequencing logic into Blackboard's presentation of instructional content and assessments. This approach would allow selection and presentation of a topic based upon the evaluation of preceding assessment items by topic. One way we considered implementing this capability was to utilize a dynamic web-browser page (i.e., HTML, which is a Blackboard allowed type of content) with JavaScript coding to present content and quizzes developed in third-party tools, and to embed an application programming interface that allows grades and variables to be exchanged between the gradebook and these third-party tools. This approach would allow for the presentation of richer content (developed in a third-party HTML tool); for a more granular response to assessment items (through custom program logic and data interchange with the gradebook); and for more elaborate logic for the sequencing of the instructional content (again through the use of custom program logic). Our Blackboard installation could be extended to provide such a capability by coding a "Building Block" module, which is the mechanism for extending and customizing Blackboard (University of Massachusetts, 2014).

Extending an LMS to provide adaptive learning capabilities has precedence. Sodoke, Riopel, Raîche, Nkambou, & Lesage (2007) extended Moodle to adapt assessment based upon item response theory. However, while such an approach is feasible, we considered it to be a complex and time consuming effort. Therefore, before proceeding with custom development, we searched for existing Blackboard extensions that might already provide such functionality. And indeed, the envisioned capabilities were found to already exist in every LMS that implements the industry standard Sharable Content Object Reference Model (i.e., the SCORM). This discovery was a great relief for us. Blackboard implements the SCORM up through the 2004, version 4 specification (Blackboard, Add content packages, 2014). Utilizing the SCORM content feature in Blackboard allows an instructional designer to present the results of one or more content and assessment authoring tools through the LMS; have detailed assessment data recorded in and reported by the LMS; and specify logic for dynamically determining the sequence of content that the LMS will present.

Therefore, our second design iteration explored use of the SCORM content package feature of Blackboard. A sample list of SCORM compliant authoring tools is presented in Appendix A. We chose to author our content with the Articulate StoryLine product, per the preference of the course instructor.

SCORM Features

The SCORM was written at the beginning of the millennium by the Advanced Distributed Learning Initiative (ADL) under contract to the U.S. Department of Defense. It is intended to enable the production of reusable training modules (i. e., learning objects) across a large and diversified government curriculum as a way to lower training development costs and to improve the consistency of instruction (Advanced Distributed Learning, SCORM overview, 2015). The model is a set of specification documents that define in detail a standard interface between instructional content objects and hosting LMSs. Training software suppliers implementing the interface according to the specifications are said to be SCORM compliant. Instructional designers can utilize one or more SCORM compliant authoring tools and be assured that the resulting product will operate as intended in a SCORM compliant LMS. Due to the federal government's sponsorship for the standard, there are many SCORM compliant authoring tools and LMSs on the market today.

The SCORM defines several separate specifications, including: 1) the *run-time environment*, 2) the *data model*, and 3)

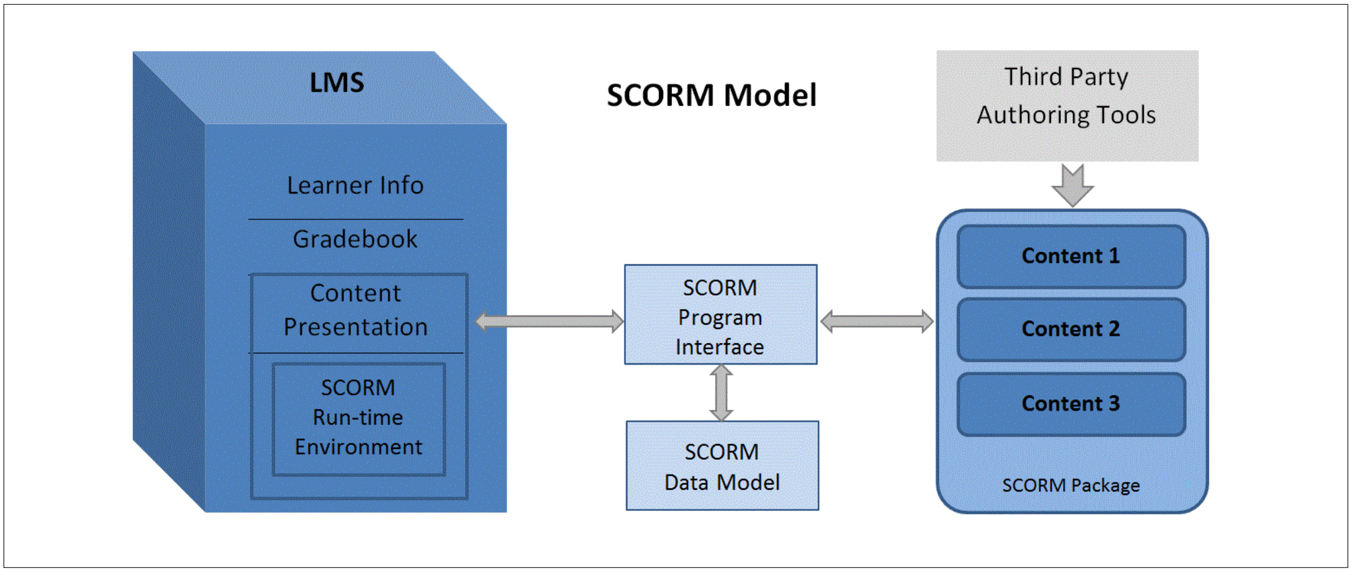


FIGURE 6. SCORM Components.

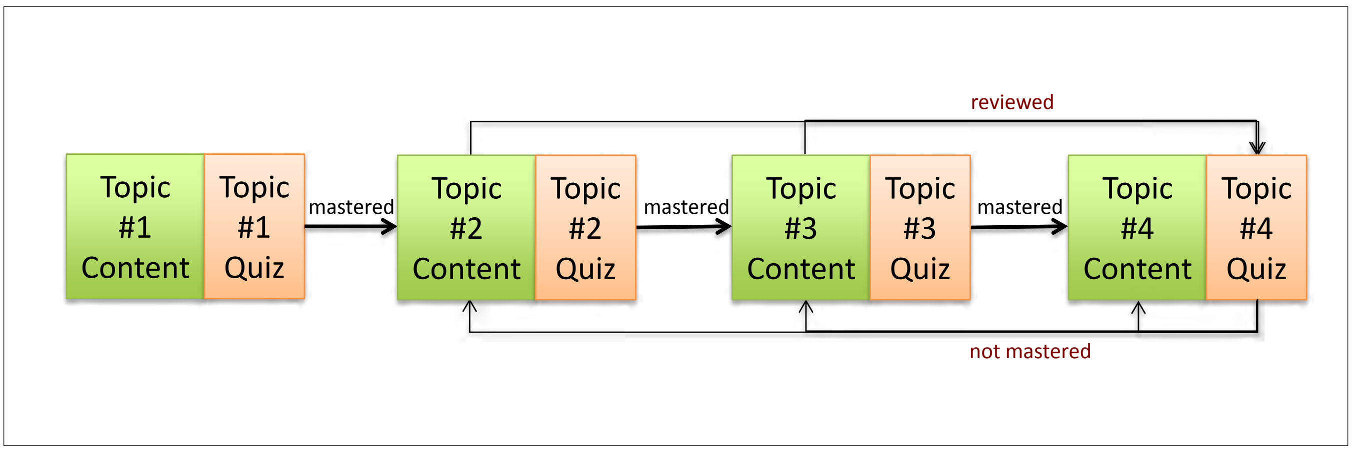


FIGURE 7. Example PLP with multiple contingent responses.

the *packaging model*. These are illustrated in Figure 6. The run-time environment defines how a hosting LMS is to present SCORM content. The data model defines the information a LMS is to store and make available to SCORM content. The packaging model defines how SCORM content is to be physically structured, described, and packaged so that a host LMS can process it (Rustici Software, SCORM explained, n.d.).

Using these specifications, the delivery of instruction in Blackboard can be extended in several useful ways. Third party authoring tools can be used to supply instruction that has richer content, and quizzes that have finer granularity for assessment and response, than can be created natively in Blackboard Learn. This instruction and assessment content can then be packaged as discrete learning objects for efficient reuse or review. In SCORM terminology, the packaging of several discrete learning objects into an instructional topic is known as embedding multiple Shared Content Objects (SCOs) within an activity tree. A learner's progression through the activity tree can be dynamically determined during the instruction either through sequencing rules within the packaging specifications, based upon that learner's performance measures (either scores or progress indicators); or through the use of JavaScript programming logic embedded within the learning objects. Thus, the learning can be highly individualized.

By utilizing the SCORM specifications for the data model and the API, each learning object (i.e., SCO) can exchange information with the hosting LMS. For example, the SCO may retrieve the student's name for incorporation in the course presentation. The SCO can also report back to the LMS its learning objectives, and each learner's test scores, training progress and completion status. Recorded assessment information can be detailed to the level of individual questions asked, the student's response, and whether the response

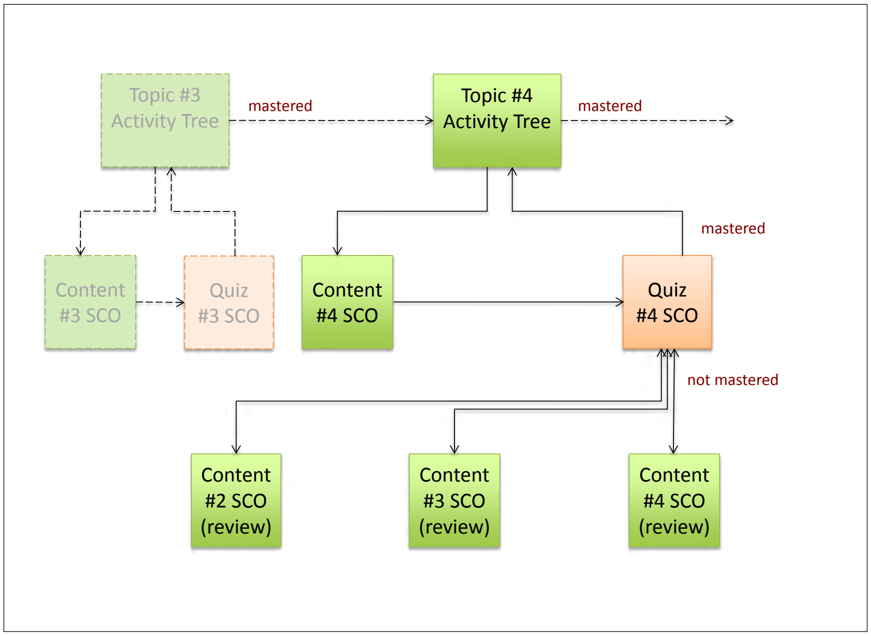


FIGURE 8. Example SCORM Activity Tree for multiple contingent responses.

was correct or incorrect. All information exchanges are timestamped. This enables the host LMS to produce detailed analytic reports for each student.

Designing with SCORM

We analyzed our example course anew from the perspective of the SCORM specification for activity trees. We selected an approach using multiple SCOs, and multiple activity trees with one for each of the four topics to be presented. Each of these activity trees contain one SCO for the topic's instructional content, a second SCO for the topic's diagnostic quiz, and possibly one or more SCOs for the instructional content of other topics needed for review. For example, in question 4-4 shown earlier (Figure 3), incorrect responses could cause a review of Topics 2, 3, or 4 as shown in the PLP model in Figure 7.

As an activity tree, this same flow is represented hierarchically as shown in Figure 8.

An important feature of activity trees is that they rely on references to a single instance of an SCO and not a repetition of the SCO itself, unlike our experience with Blackboard. Thus, the SCO for Content #3 only has to be created and defined one time in the SCORM packaging description (a file called imsmanifest.xml) but can be referred to from the activity trees for both Topic 3, and Topic 4. Furthermore, these activity tree references can also convey a context for accessing the SCO, such as reviewing all, or only a part of the content, as required for Question *4-4* answer *a*, (Figure 3). For example, Figure 8 shows SCOs for both Content #3 and Content #4 appearing twice: once within the context of presenting instructional content for their related topic; and once in the context of providing review material as topic-contingent feedback for the Quiz #4 SCO. Such a capability represents significant reusability.

Designing with StoryLine

Articulate's StoryLine product can be used to author instructional content and assessments for publication as stand-alone modules for a website, or as SCORM compliant SCOs. Instructional content can be a combination of static text and graphics, audio and video clips (including screen- motion capture), animations, and interactive graphics (for gaming or simulations). This makes it a very capable tool for building content. Assessments can randomly draw questions from question banks, and can report detailed information through the SCORM to a LMS. Programming logic can be added through "if-then" conditional "triggers" in order to control the flow of the presentation within StoryLine, or through JavaScript code to interact with the LMS through the SCORM application programming interface (Articulate, 2015).

The design of our example course imposes sequencing requirements upon the activity trees which are implemented through program logic within the SCOs. Specifically, the design indicates a need for each Quiz SCO within an activity tree to specify the next topic SCO for presentation (either the next sequential topic, or a review of a previous topic), and for each topic SCO to either continue the sequence to its associated Quiz SCO, or to return to the Quiz SCO from which it was invoked (as a review item). Furthermore, to exploit its reusability, a topic SCO, if invoked as a review item, might need to be entered at one of several possible internal section points and likewise contain multiple possible exit points. For example, the Topic 4 content SCO may be entered at the beginning and exited at the end when invoked as the initial presentation of the topic, but be entered at the section on *Other measures of spread* and exited after that section when invoked from question *4-4* answer *a*, as shown previously in Figure 3. The slide map for the Topic 4 Content SCO is displayed in Figure 9.

In order to implement this program logic, it was helpful for us to design standard ways for StoryLine to interact with the SCORM run-time environment, both to retrieve the context information embedded in the activity tree definitions, and to control the sequencing through the SCOs. The primary author developed several new JavaScript functions for addition to the API libraries provided by Articulate. These functions added to the control and communication

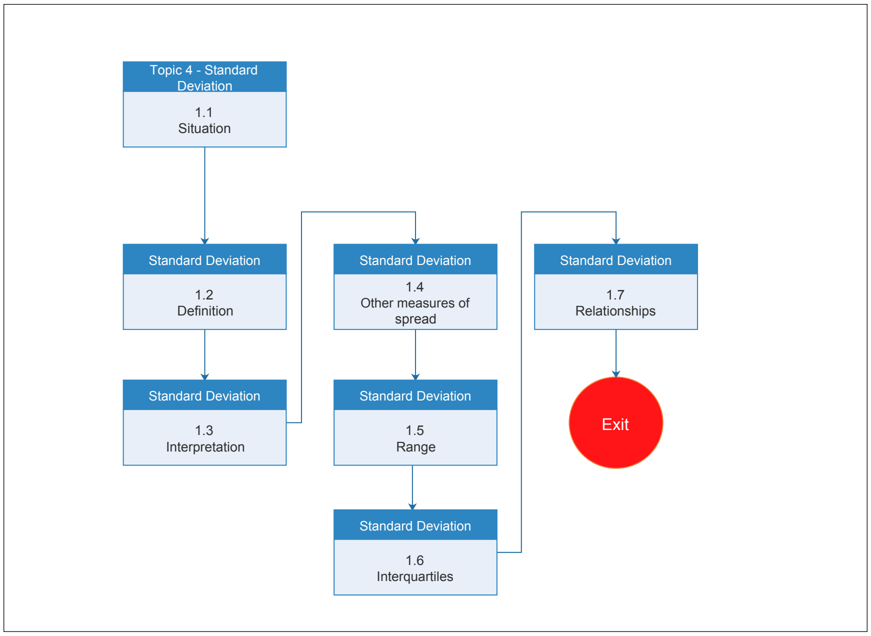


FIGURE 9. Slide Map of Topic 4 SCO.

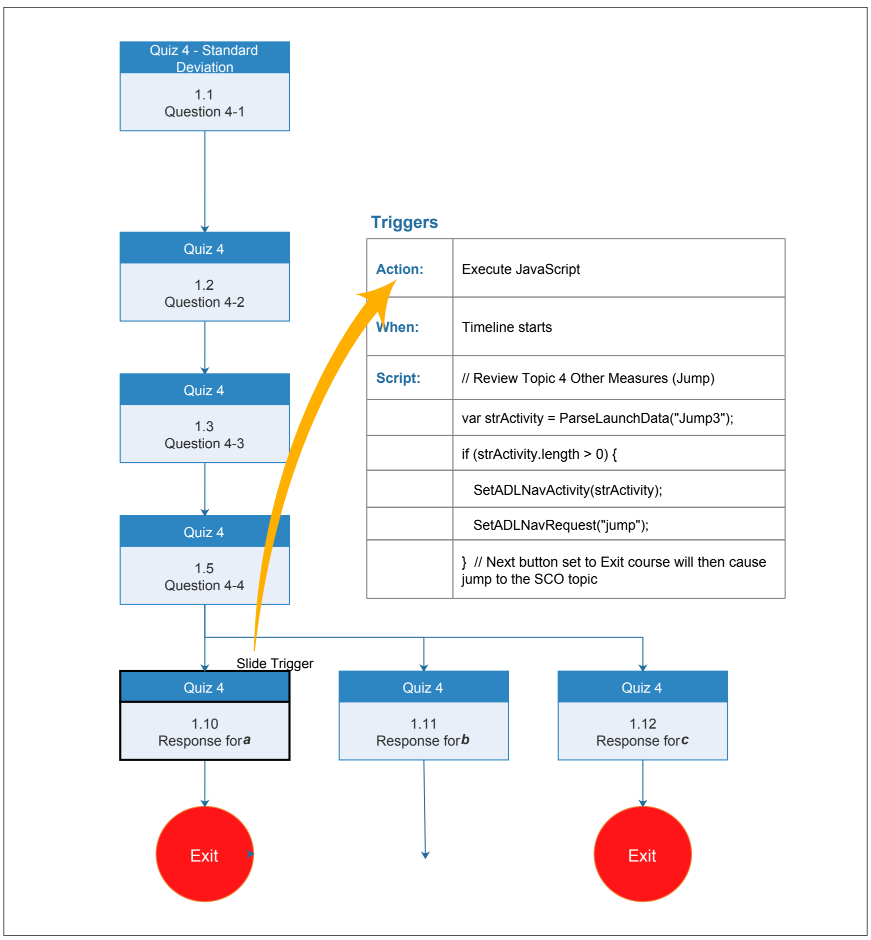


FIGURE 10. Quiz 4 slide navigation exit trigger.

capabilities between a Storyline SCO and the SCORM run-time environment. For example, Figure 10 shows how several of these JavaScript functions are invoked from a StoryLine trigger to programmatically specify the next topic SCO for presentation as required for question 4-4, answer *a*.

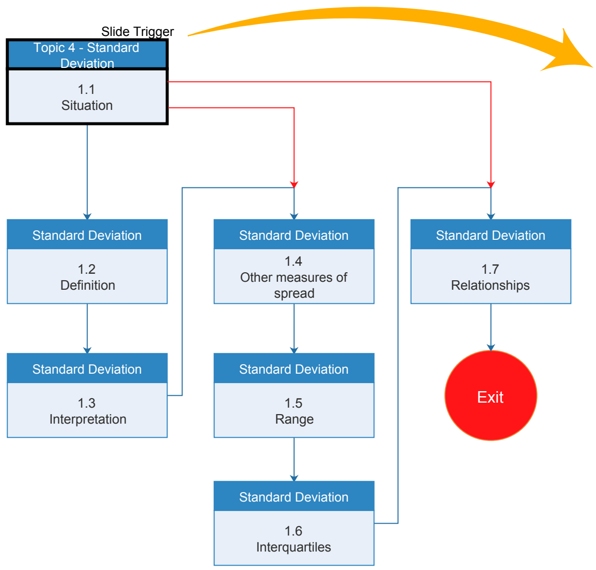
On the learner's screen, Slide 1.10 (the outlined box in Figure 10) instructs the learner to click on the Next button to proceed to review Topic 4, *Other Measures of Spread*. When the learner clicks on the Next button, StoryLine exits the Quiz 4 SCO, and the SCORM run-time environment then executes the specified navigation request to return to the Topic 4 SCO.

This design approach centralizes all of the specific details of the possible instructional sequences in a single location: the activity tree definitions of the imsmanifest.xml file.

Figure 11 illustrates how the Topic 4 SCO responds in a context-appropriate manner following invocation by SCORM as the response to question *4-4* answer *a*. A timeline start trigger on the opening slide (box 1.1) of the SCO retrieves the context specific reference for the desired section to display (*Other measures of spread)*. This trigger follows a convention within StoryLine to utilize internal variables and triggers to immediately jump to the desired alternative entry point (i.e., a section). This alternative branching to each of the sections within Topic 4 is indicated in Figure 11 by the multiple arrows flowing from box 1.1 to the different sections of the SCO, such as box 1.2, 1.4, and 1.7.

Iteration 2 Summary

Our first iteration development attempted to use only the native course builder capabilities of Blackboard Learn, but failed to meet the pedagogical and technical objectives we had set out. The prototype we developed for the second iteration, using the SCORM content package feature of Blackboard, successfully allowed us to deliver the improved capabilities for instruction, assessment,



Action:	Execute JavaScript
When:	Timeline starts
Script:	// Retrieve and set return jump
	var strActivity = ParseLaunchData("Jump0");
	if (strActivity.length > 0) {
	SetADLNavActivity(strActivity);
	SetADLNavRequest("jump");
	};
	// Check for a Topic parm and if found, trigger a jump to that screen
	var strTopic;
	var strTopic0=ParseLaunchData("Topic");
	strTopic = strTopic0.toLowerCase();
	var player = GetPlayer();
	player.SetVar("Topic", strTopic);
	var strVariable;
	if (strTopic.length > 0) { // flip strTopic to fire a trigger
	strVariable = player.GetVar(strTopic);
	if (strVariable === "A") {
	strVariable = "B";
	} else {
	strVariable = "A";
	};
	player.SetVar(strTopic, strVariable);
	};

FIGURE 11. Topic 4 slide navigation entry trigger.

and individualized sequencing required by our objectives. While this second iteration prototype served as a proof of concept for this technical approach, we concluded that further refinement might be needed before we could begin development of the actual statistics course for inclusion in the doctoral program.

We noted several overall concerns about using this approach for actual course development:

- The details of the SCORM specifications are technically complicated and provide multiple variations for implementing communication and control between an SCO and a hosting LMS. Development with SCORM may require a level of programming expertise that is only suitable for organizations dedicated to course development, such as firms that produce training for the government or industry. In our case, we limited our prototype to a subset of SCORM features that proved to be quick and easy to use for our requirements. Expanding the prototype to a full course would require establishing conventions and procedures for the construction of the SCOs and the activity trees.
- Bugs with JavaScript code or logic can be difficult to find, especially once a content or quiz module has been integrated into SCORM and loaded into Blackboard. Debugging this code requires experienced technical skill. Our prototype attempted to address this issue by generalizing the JavaScript coding into callable functions and establishing conventions for their use in StoryLine. However, we felt that the JavaScript functions and their use were still not as easy to understand as we needed.
- There is flexibility in the definition of a SCO in terms of how much instructional content is included. One option is to present all topics and diagnostic assessments within one SCO. That would place all branching and sequencing logic within the StoryLine development environment. The downside to this approach is that the logic within StoryLine becomes complex to test and to maintain. Our prototype took the alternative approach, for the sake of testing boundaries, to create separate SCOs for each topic and each quiz. This resulted in a development environment more suitable for testing, at the expense of a more complex SCORM packaging structure and the need to define more of the sequencing logic through the imsmanifest.xml file. The techniques developed in this design iteration also enable an ability to choose an option that lies between these two extremes, such as combining each topic and its associated quiz into a single SCO. Such an approach would seem to add very little additional logic to StoryLine and would reduce the number of elements in the SCORM activity trees and associated imsmanifest.xml file.
- The execution of SCORM components within Blackboard is not as reliable as we had expected. Demonstrations of the prototype in different settings revealed sensitivities to different web browsers and operating environments. On the campus, it was discovered that the prototype required the use of a specific web browser. We were surprised that support for resolving issues of this nature could not be found within the Storyline or SCORM communities, such as Articulate's E-Learning Heroes (https://community.articulate.com) and Rustici Software's Ask Us Anything (https://scorm.com/blog/)

Overall, the prototype produced by this second design iteration demonstrated the feasibility of our pedagogical and technical objectives. We were able to construct an example course that combined the technologies of StoryLine, the Sharable Content Object Reference Model (SCORM), and Blackboard to deliver competency-based student progress (CBSP) adaptive instruction with topic-contingent assessment feedback.

Iteration 3: Refining the Technical Design

The instructor for the doctoral level statistical methods course planned to hire a graduate assistant for a semester to create the actual course content. Based upon our experience in assembling and testing the example course, and given the concerns listed earlier, we concluded that an experienced JavaScript developer was required. Unfortunately, such a candidate could not be found within the price range permitted by the limited development budget for this course. Therefore, the primary author agreed to undertake a third iteration of the design to evolve the prototype into a more complete sample which would enable a graduate student to develop the course with less experience in JavaScript and StoryLine.

In order to capitalize on the technical development already accomplished, it was decided to focus on improving the design of the StoryLine SCOs. Development of the third iteration focused on accomplishing the following technical goals:

- Combine the instructional content for a single topic with its diagnostic quiz into a single SCO. This approach would halve the number of SCOs to be created, and simplify the activity trees. This is represented in Figure 12 with its four SCOs compared to the activity trees in Figure 8 with their seven SCOs.
- Create a single generic StoryLine SCO with placeholders for the instructional content, while containing all of the background programming logic (JavaScript code and StoryLine triggers) required for operation. The developer could start with this generic SCO, and then "drop in" the appropriate content for the instruction and the assessment.

This generic StoryLine SCO would contain all the programming logic necessary to:

- Present the entire topic to the learner from the initial sub-topic through to the last quiz item.
- Directly implement a review of that topic's instructional material as a topic-contingent response to a quiz item within that topic. For example, a quiz item in Topic 4 that indicates a need to review Topic 4 instruction should be handled within the Topic 4 SCO. It should not require exiting the SCO and subsequent processing in the activity tree. This eliminates one branch in the Figure 8 activity tree for Topic 4.
- Connect the SCO's need to review instructional content within another SCO with the appropriate context parameter in the activity tree. For example, an incorrect response on question 4-4 might require a review of material in Topic 2. This requires exiting the Topic 4 SCO, and invoking the Topic 2 SCO through a context specific entry in the Topic 4 activity tree.
- Present one or more instructional sub-topics as a review item. This requires receiving a context specific parameter from the activity tree, and using it to skip directly to that portion of the instructional content. After presenting the requested review material, the SCO would need to exit and have the presentation return to the quiz that invoked the review. For example, if the Topic 2 SCO is invoked from the Topic 4 quiz, then once the learner has completed the Topic 2 review, the learner should be returned to the Topic 4 SCO.
- Re-initiate the quiz following a return from a review of a prior topic. For example, Topic 4 question 4-4 causes a review of a Topic 2 item, which then returns the learner to the Topic 4 quiz for re-assessment.

By using a StoryLine feature that allows division of an SCO into multiple scenes, a modular design for the sample was constructed. Individual scenes were defined for typical sub-topics in the instructional content. Scenes were also created for various types of assessment items and

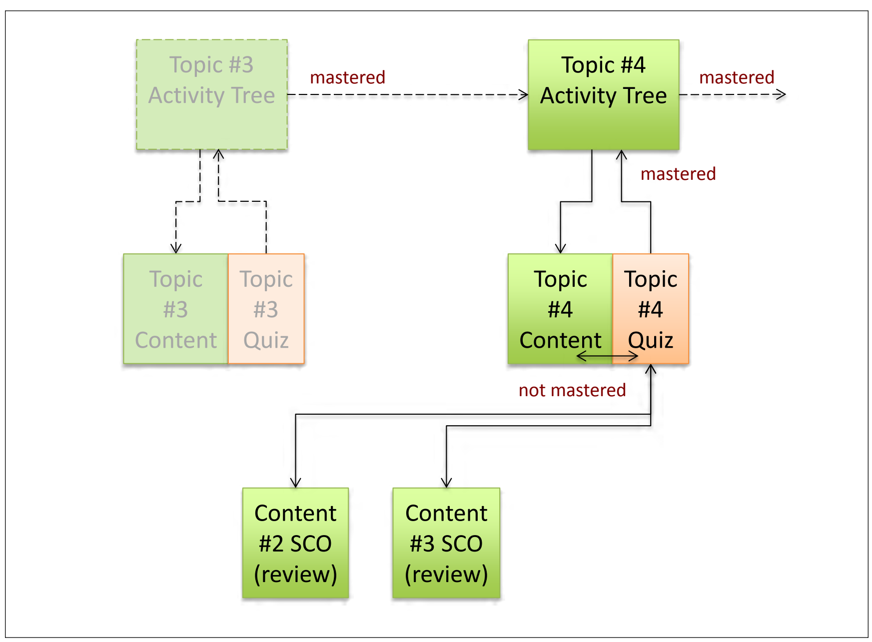
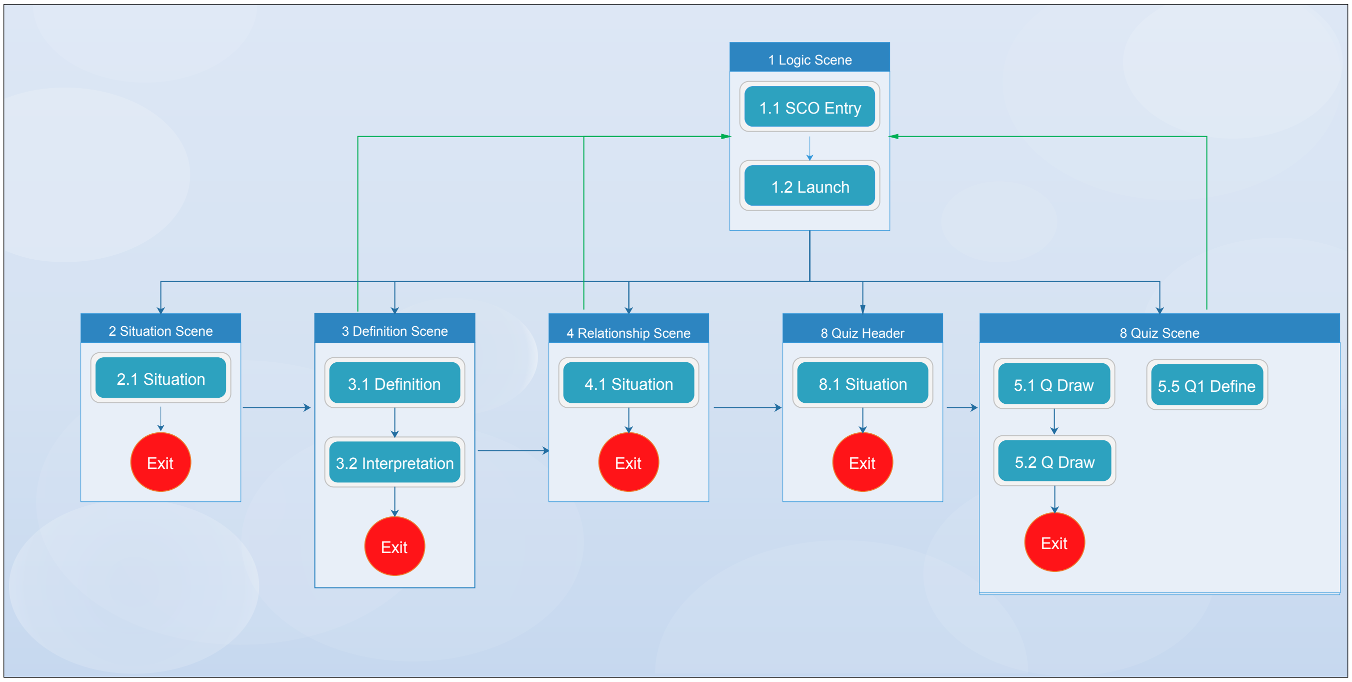


FIGURE 12. Activity Tree example for refined SCOs.



IGURE 13. Refined Slide Map of Template SCO.

topic-contingent responses. The resulting design is shown in Figure 13.

This design embeds the necessary programming logic into the start and end points (i.e., slides) of each scene so that the content developer need not be concerned with coding JavaScript or StoryLine triggers. The developer need only populate the middle of a scene with the desired content. The developer initiates desired actions at the end of a scene through manipulation of a defined set of StoryLine variables. The Logic Scene slide 1.1 contains triggers based upon these variables that execute the desired action (e.g., jump to scene *x*, or exit this topic and go to a specified SCO). This slide does not display to the learner.

This third iteration design presents a highly structured and consistent approach to delivering the desired Competency-Based Student Progress form of instruction envisioned by the secondary author. By removing the need to create or alter JavaScript code and StoryLine triggers, it presents a minimal learning curve to content developers. Also, because the JavaScript is not modifiable by the developer, it should operate in a more reliable and predictable manner, and not require technical expertise in debugging the programming code.

Iteration 3 Summary

The example course was successfully implemented using the template materials from this third design iteration. It demonstrated the use of the Sharable Content Object Reference Model (SCORM) capability of Blackboard (and many other LMSs) to import a SCORM package consisting of multiple Sharable Content Objects (SCOs) authored in the Articulate StoryLine product.

CONCLUSION

The example course developed in this design case met the following objectives:

- *Each individual learner is sequenced through a series of single topic chunks of instructional content*. Unique SCOs were created for each topic chunk. SCORM activity trees specified the sequential flow through all of the SCOs.
- *The presentation of each topic is followed with a formative assessment of the learner's mastery of the topic and its relationships to previous topics through the diagnostic quizzes. Incorrect responses to individual assessment items trigger an immediate remedial, topic-contingent action to review the appropriate content*. Triggers and JavaScript code implement programmable decisions to alter the learning path by immediately jumping to the proper SCO needed for review of a specific concept, and then returning back to the diagnostic quiz for another attempt at demonstrating mastery.
- *Questions for each topic assessment are randomly drawn from a pool of questions for that topic. Therefore, no assessments will be the same for any two learners, and questions are unlikely to be repeated for any individual learner*. StoryLine allows the creation of Question Banks and the random drawing of assessment items from a specific Question Bank.
- *Attainment of competency by topic can be recorded in the LMS gradebook*. The SCORM Data Model allows the

definition of Learning Objectives, and the Application Programming Interface allows these Learning Objectives to be set by StoryLine and recorded in the LMS Gradebook.

- *The LMS gradebook can be analysed by attainment objectives as opposed to aggregate quiz scores, in order to derive the personalized learning path for each student*. Information about assessment items and results in StoryLine can be very granular, with specific answers reported for every attempt. Blackboard can provide detail analytical access to these results.

In addition to meeting all of the pedagogical and technical objectives, the technologies developed for this prototype provide the following benefits:

- The approach and technology demonstrated in this paper can be applied to implementing additional features of adaptive or personalized learning without the need to replace or extend an existing, in-place LMS. For example, content selection and sequencing could be modified based upon the results of a pre-test assessment (Khan, 2006). Likewise, the method of instruction could be differentiated based upon student learning styles or preferences (Truong, 2015; Manochehr, 2006).
- While instructors and instructional designers who desire to deliver an online course with learner-centered components may have no choice in their selection of an LMS, they can have a choice in the tools used to author and present their instructional activities. For this project we chose StoryLine. However, other product choices are available, as shown in Appendix A. At a minimum, an authoring tool needs to be SCORM-compliant with support for SCORM 2004. This will ensure that all content types produced by the tool, such as slides, audio and video clips, animations and interactive simulations and games are properly packaged for delivery through the LMS. This also ensures that performance and assessment data can be recorded in the LMS gradebook. Although not a requirement, it is an added benefit if the product supports the creation of multiple-SCO SCORM packages. If sequencing between content objects needs to be data-driven, then the tool should provide an easy method to invoke custom JavaScript code.

Industry Standards Subsequent to SCORM

As noted earlier, the SCORM standards were developed at the start of the millennium. This was prior to the widespread adoption of Web 2.0 and mobile technologies by the general population. Many educational tools and resources that are available today (e.g., geo-tagged mapping, conferencing, social media sharing) do not adhere to the technology structure envisioned by the SCORM where the integration between the third-party content and the LMS occurs on the client-side (Holmesglen Institute of TAFE, 2013). Instead, these websites and apps are stand-alone, cloud-service offerings with their own hosting sites and presentation interfaces. In the case of mobile applications, the presentation may not even be based on HTML. Thus, SCORM cannot be utilized to bring these new resources into instruction delivered through a Legacy LMS.

To address this need, two new, independent standards have more recently been issued by separate organizations. Advanced Distributed Learning drew upon their experience with the creation of the SCORM and created a new specification called the Experience API (xAPI) also known as TinCan (Advanced Distributed Learning, Experience API, 2015). Meanwhile, the IMS Global Learning Consortium (IMS) developed the Learning Tool Interoperability (LTI) standards (IMS Global Learning Consortium, 2016). Both of these efforts utilize a server-side interface between an LMS and third-party resources. With this approach, the third-party content remains external to the LMS, so there is no uploading of content. Only a URL link is required. Learner information, security credentials, and performance data can be exchanged in real-time, as with SCORM content, but navigation and sequencing of multiple SCOs occurs solely within the logic of the external service. Desire2Learn takes this approach with its LeaP LTI module for Adaptive Learning (Desire2Learn, 2015).

The choice between using the SCORM, xAPI, or LTI specifications should be based upon the third-party tool to be used to create the content and the specification to which it conforms.

Adopting a learner-centered approach to instruction necessitates the creation of engaging and interactive resources. However, online educators who desire to shift their instruction to the learner-centered paradigm may find that existing teacher-centered LMS implementations represent a formidable hurdle to the delivery of such resources. This paper chronicled the design and prototyping of a transitional approach to reducing this hurdle by using the SCORM to integrate and deliver content created by a third-party authoring tool within the framework of an existing LMS

REFERENCES

Advanced Distributed Learning. (2015). *Experience API.* Retrieved from Advanced Distributed Learning: https://www.adlnet.gov/adl-research/performance-tracking-analysis/experience-api/

Advanced Distributed Learning. (2015). *SCORM overview.* Retrieved from Advanced Distributed Learning: http://adlnet.gov/adl-research/scorm/

Aslan, S., Huh, Y., Lee, D., & Reigeluth, C. M. (2011). The role of personalized integrated educational systems in the information-age paradigm of education. *Contemporary Educational Technology, 2*(2), 95-117. Retrieved from http://www.cedtech.net/articles/221.pdf

Articulate. (2012, September 24). *Using javascript to tell the storyline flash player to go to slide*. Retrieved from E-learning heroes discussion blog: https://community.articulate.com/discussions/articulate-storyline/using-javascript-to-tell-the-storyline-flash-player-to-go-to-slide

Articulate. (2015). *StoryLine 2 - all features*. Retrieved from Articulate: https://www.articulate.com/products/storyline-all-features.php

Bernardi, J. (2015, August 6). *How to setup adaptive release for an assessment in Blackboard Learn*. Retrieved from University of Houston Knowledge Base: http://innovate.uh.edu/edtech/knowledgebase/adaptive-release-for-an-assessment-on-blackboard-learn/

Blackboard. (2014, April). *Add content packages.* Retrieved from Blackboard Help: https://en-us.help.blackboard.com/Learn/9.1_2014_04/Instructor/090_Course_Content/010_Create_Content/025_Add_Content_Packages

Blackboard. (n.d.). *Get started.* Retrieved from CourseSites: https://www.coursesites.com/bbcswebdav/institution/coursesites-files/www/getstarted.html

Bloom, B. S. (1968). Learning for mastery. *Instruction and Curriculum, 1*(2), 1-10.

Bremer, D., & Bryant, R. (2005). A comparison of two learning management systems: Moodle vs Blackboard. *Proceedings of the 18th Annual Conference of the National Advisory Committee on Computing Qualifications*, pp. 135–140.

Bunderson, C., Wiley, D., & McBride, R. (2009). Domain Theory for instruction: Mapping attainments to enable learner-centered education. In C. M. Reigeluth, & A. A. Carr-Chellman (Eds.), *Instructional-design theories and models: Building a common knowledge base* (pp. 327-347). New York, NY: Routledge.

Cohen, V. B. (1985, January). A reexamination of feedback in computer-based instruction: Implications for instructional design. *Educational Technology*, pp. 33-37.

Desire2Learn. (2015). *D2L resource center - LeaP.* Retrieved from D2L: https://documentation.desire2learn.com/en/LeaP

Dobre, I. (2015). Learning management systems for higher education - an overview of available options for higher education organizations. *Procedia - Social and Behavioral Sciences, 180*, 313-320.

Holmesglen Institute of TAFE. (2013). *Beyond SCORM - New interoperability standards.* Retrieved from E-Standards for training: http://e-standards.flexiblelearning.net.au/research/funded_projects/emerging_technology_trials/2013/beyond_scorm_-_new_interoperability_standards.php

IMS Global Learning Consortium. (2016). *Learning tools interoperability.* Retrieved from IMS Global Learning Consortium: https://www.imsglobal.org/activity/learning-tools-interoperability

Johnson, D. T. (2000). Teaching mathematics to gifted students in a mixed-ability classroom. *ERIC Clearinghouse on Disabilities and Gifted Education, the Council for Exceptional Children*.

Khan, B. H. (2006). *Flexible learning in an information society.* Hershey,PA: Information Science Publishing.

Kulik, C. -L., Kulik, J. A., & Bangert-Drowns, R. L. (1990). Effectiveness of mastery learning programs: A meta-analysis. *Review of Educational Research, 60*(2), 265-299. https://doi.org/10.3102/00346543060002265

Lee, D. (2014). How to personalize learning in K-12 schools: Five essential design features. *Educational Technologies, May-June*, 12-17.

Manochehr, N. N. (2006). The influence of learning styles on learners in e-learning environments: An empirical study. *Cheer Virtual Edition, 18*, 10-14.

Mason, B. J., & Bruning, R. (2001, January). *Providing feedback in computer-based instruction: What the research tells us.* Retrieved from Center for Instructional Innovation: University of Nebraska-Lincoln: http://dwb4.unl.edu/dwb/Research/MB/MasonBruning.html

Mills, C. J., Ablard, K. E., & Gustin, W. C. (1994). Academically talented students' achievement in a flexibly paced mathematics program. *Journal for Research in Mathematics Education*, 495-511. https://doi.org/10.2307/749487

Northeastern University. (2014, October). *Choosing your publish settings in Storyline for use with Blackboard gradebook.* Retrieved from Northeastern University Academic Technology Services: http://www.ats.neu.edu/wp-content/uploads/2014/01/Storyline_SCORM_and_Blackboard_grading.pdf

Posner, M. A. (2011). The impact of a proficiency-based assessment and reassessment of learning outcomes system on student achievement and attitudes. *Statistics Education Research Journal, 10*(1), 3-14.

Reigeluth, C. M., & Karnopp, J. R. (2013). *Reinventing schools: It's time to break the mold.* Lanham, MD: Rowman & Littlefield.

Rustici Software. (n.d.). *SCORM explained- business of SCORM.* Retrieved from Rustici Software: http://scorm.com/scorm-explained/business-of-scorm/

Sodoke, K., Riopel, M., Raîche, G., Nkambou, .., & Lesage, M. (2007). Extending Moodle functionalities to adaptive testing framework. *World Conf. on E-Learning in Corporate, Government, Healthcare & Higher Education (E-Learn), 2007.*

Software & Information Industry Association. (2010). *Innovate to educate: System [Re]design for personalized learning; A report from the 2010 Symposium.* In collaboration with ASCD and the Council of Chief State School Officers, Washington, DC

Stodolsky, S. S. (1988). *The subject matters: Classroom activity in math and social studies.* Chicago, IL: University of Chicago Press.

Truong, H. M. (2015). Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities. *Computers in Human Behavior, 2015*, 1-9. https://doi.org/10.1016/j.chb.2015.02.014

U.S. Department of Education. (2010). *Transforming American education: Learning powered by technology.* Washington DC: Office of Educational Technology.

University of Massachusetts. (2014). *Blackboard Learn Building Blocks*. Retrieved from UMassOnline: https://confluence.umassonline.net/display/UMOLTT/Blackboard+Learn+Building+Blocks

Watson, S. L., & Reigeluth, C. M. (2008, Sept.-Oct.). The learner-centered paradigm of education. *Eductional Technology, 48*(5), 39-48.

Yildirim, Z., Reigeluth, C. M., Kwon, S., Kageto, Y., & Shao, Z. (2014). A comparison of learning management systems in a school district: searching for the ideal personalized integrated educational system (PIES). *Interactive Learning Environments, 22*(6), 721-736. https://doi.org/10.1080/10494820.2012.745423

APPENDIX A

A partial list of SCORM compliant authoring tools

- Adobe Presenter / Captivate
- Articulate Quizmaker / Studio / StoryLine
- Brainshark Presentations
- Camtasia
- Composica
- Engage
- Impatica OnCue
- iSpring QuizMaker
- learningMaker
- mLearning Studio
- myUdutu Course Authoring Tool
- ProForm
- Quest Express
- THESIS
- Unison
- Vivo

Source: eLearning Atlas—http://www.elearningatlas.com