Live Catalog of Smart Learning Objects for Computer Science Education

Alexander Hicks, Clifford A. Shaffer
Virginia Tech
Blacksburg, United States
{awh4kc, shaffer}@vt.edu

Kamil Akhuseyinoglu, Peter Brusilovsky
University of Pittsburgh
Pittsburgh, United States
{kaa108, peterb}@pitt.edu

ABSTRACT

We present the initial version of a "live catalog" of LTI enabled smart learning objects that instructors and educators are able to preview and test before deciding whether to integrate these tools in their own courses. The catalog is available on the public Instructure Canvas site and currently showcases LTI tools from multiple educational institutions.

Author Keywords

LTI; tool interoperability; learning management system; smart learning content

INTRODUCTION

Many "smart" educational resources are becoming available for CS courses [2]. Two major concerns are helping instructors to locate them, and making it easy for instructors to integrate them seamlessly into a course LMS alongside resources from other providers [7, 1]. One solution to improve integration is IMS Global's Learning Tools Interoperability (LTI) protocol. While it can be highly effective, accessing LTI-enabled tools can be difficult for the average instructor.

To address these issues, we have created a "live" catalog of computer science educational resources that are LTI-compliant. By "live", we mean that the catalog allows instructors to actually use examples of the learning resource in a real LMS, along with instructions for how to integrate the tool into their own (LTI compliant) LMS, and access to the full list of exercises available from that provider. This includes demonstrating that the learners' grades are automatically passing back to the LMS gradebook from the tool. Current catalog contents include the OpenDSA, CodeWorkout, and Web-CAT systems from Virginia Tech, and the ACOS content server from a joint project between Aalto University and the University of Pittsburgh.

BACKGROUND AND RELATED WORKS

Our live LTI catalog demonstrates how powerful existing infrastructure and interoperability protocols within computer science education can be by showing examples of several tool providers as well as the instructions to integrate it into your own course. This interactive component is the key difference from other catalogs that simply aggregate currently existing LTI tools and their instructions such as IMS Global's list of certified LTI tools [4] or the University of Wisconsin-Madison's Knowledgebase aggregation of integrated and third party applications [11]. While both of those resources are valuable, they do not provide hands-on access to the tools.

We chose to use Canvas as the LMS for this catalog for several reasons, including that Canvas is the LMS that we are most familiar with, Instructure provides an easily accessible public instance that allows for anyone to view it, and as a participant in the IMS consortium, Instructure's Canvas is typically up-to-date in its support for LTI [9].

We recently presented a case study where two large educational research teams cross-shared their educational materials through the LTI standard [10]. In this paper, we leverage the benefits of a publicly accessible LMS to create a live content catalog of computer science educational tools. We believe this will help instructors to be more engaged in trying and re-using these resources.

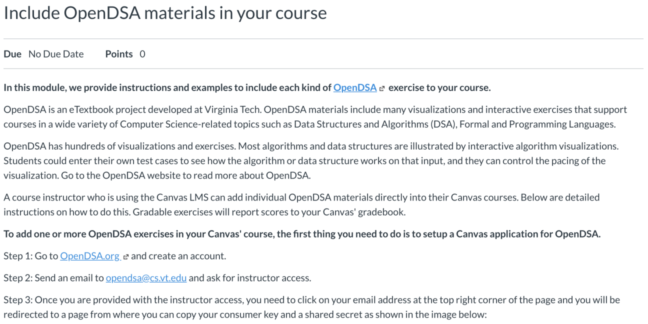
TECHNICAL DETAILS AND TOOLS INCLUDED

The live catalog exists as a public Canvas course on Instructure's Free-for-Teachers site [1]. The catalog currently demonstrates activities from OpenDSA, CodeWorkout, Web-CAT, and the ACOS Server. OpenDSA is an eTextbook project that supports using a wide variety of individual auto-graded exercises as well as full textbook chapters with multiple exercises [8]. Figure 1 demonstrates how the catalog provide instructions for how to embed materials from each catalog entry into a Canvas course. CodeWorkout is a platform that provides simple practice problems designed for students to practice Java problems and concepts [5].

Web-CAT is a software system designed to show students the power of test-driven development by testing them not only on their solution, but how well they were able to test their own code using JUnit tests [6]. ACOS server is a smart learning content server developed as a joint project between Aalto University and the University of Pittsburgh [12]. It enhances the re-usability of online learning activities by decoupling content types from interoperability protocols. ACOS server currently supports multiple communication protocols including LTI. To

Sixth SPLICE Workshop at L@S 2020 "Building an Infrastructure for Computer Science Education Research and Practice at Scale", SPLICE'20, August 12, 2020, Virtual Event

[1] The catalog can be accessed at https://canvas.instructure.com/courses/2062633



Include OpenDSA materials in your course

Due No Due Date **Points** 0

In this module, we provide instructions and examples to include each kind of OpenDSA exercise to your course.

OpenDSA is an eTextbook project developed at Virginia Tech. OpenDSA materials include many visualizations and interactive exercises that support courses in a wide variety of Computer Science-related topics such as Data Structures and Algorithms (DSA), Formal and Programming Languages.

OpenDSA has hundreds of visualizations and exercises. Most algorithms and data structures are illustrated by interactive algorithm visualizations. Students could enter their own test cases to see how the algorithm or data structure works on that input, and they can control the pacing of the visualization. Go to the OpenDSA website to read more about OpenDSA.

A course instructor who is using the Canvas LMS can add individual OpenDSA materials directly into their Canvas courses. Below are detailed instructions on how to do this. Gradable exercises will report scores to your Canvas' gradebook.

To add one or more OpenDSA exercises in your Canvas' course, the first thing you need to do is to setup a Canvas application for OpenDSA.

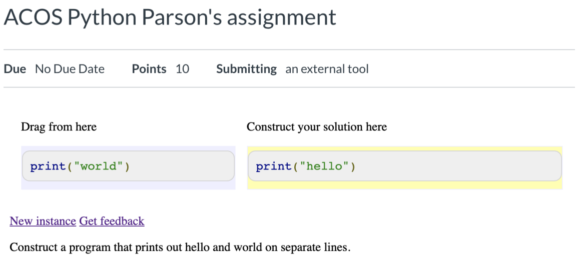
Step 1: Go to OpenDSA.org and create an account.

Step 2: Send an email to opendsa@cs.vt.edu and ask for instructor access.

Step 3: Once you are provided with the instructor access, you need to click on your email address at the top right corner of the page and you will be redirected to a page from where you can copy your consumer key and a shared secret as shown in the image below:

Figure 1. Instructions for integrating OpenDSA materials in a Canvas course on the Live Catalog

better support LTI, we upgraded ACOS server to support IMS Content-Item messages, which enables content item selection through Canvas to reduce the complexity of content addition. We also allow instructors to interact with all the available activities before selecting the correct activity as an assignment. Figure 2 shows an example of a Parson's problem as a Canvas assignment served in the catalog by ACOS server. Besides graded Parson's problems, Java and Python animated examples are also provided by ACOS server.



ACOS Python Parson's assignment

Due No Due Date **Points** 10 **Submitting** an external tool

```
print("world")
```

```
print("hello")
```

Construct a program that prints out hello and world on separate lines.

Figure 2. Python Parson's problem as an assignment on Canvas served by ACOS

DISCUSSION AND FUTURE WORK

The tools currently integrated in the live catalog all use LTI 1.1, which has been deprecated and scheduled for end of life in 2022 in favor of LTI 1.3, which uses OAuth 2.0 and OpenID Connect in order to provide for a more secure platform and the ability to pass more information back from the tool provider to the LMS [3]. Several of the currently integrated tools are currently undergoing upgrades to LTI 1.3 and the catalog will need to be updated accordingly once those upgrades are completed. This includes new instructions for adding the tool as well as details and examples of what additional information the tool may pass back to the LMS. The current work also demonstrated the need for an improved realization of the content item selection process to enable instructors to select the most appropriate learning content. Since the standards do not specify these implementation details, there are multiple ways of supporting the content item selection process. Currently, most tool providers list all the available content with limited navigation and search support. However, instructors might need additional information in order to choose the right content. For example, they might need to find an activity that concentrates on a specific learning objective, or need to know the difficulty level of the activity. This challenge should be addressed to enable easier and effective content selection. We are actively working to find and add new tools to the catalog.

REFERENCES

[1] S. Booth, S. Peacock, and S.P. Vickers. 2011. Plug and play learning application integration using IMS Learning Tools Interoperability. (2011), 5.

[2] P. Brusilovsky, S.H. Edwards, A. Kumar, L. Malmi, L. Benotti, D. Buck, P. Ihantola, R. Prince, T. Sirkiä, S. Sosnovsky, J. Urquiza, A. Vihavainen, and M. Wollowski. 2014. Increasing Adoption of Smart Learning Content for Computer Science Education. In *Working Group Reports of the 2014 on Innovation and Technology in Computer Science Education Conference*. 31–57.

[3] IMS Global Learning Consortium. 2020a. LTI Security Announcement and Deprecation Schedule | IMS Global Learning Consortium. (2020). **https://www.imsglobal.org/lti-security-announcement-and-deprecation-schedule**

[4] IMS Global Learning Consortium. 2020b. Product Certifications. (2020). **https://site.imsglobal.org/certifications**

[5] S.H. Edwards and K.P. Murali. 2017. CodeWorkout: Short Programming Exercises with Built-in Data Collection. In *ACM Conference on Innovation and Technology in Computer Science Education*. ACM, 188–193.

[6] S.H. Edwards and M.A. Perez-Quinones. 2008. Web-CAT: automatically grading programming assignments. In *13th ACM conference on Innovation and technology in computer science education*. 328.

[7] M. Ellis, C.A. Shaffer S.H., and Edwards. 2019. Approaches for Coordinating eTextbooks, Online Programming Practice, Automated Grading, and More into One Course. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. ACM, Minneapolis MN USA, 126–132.

[8] E. Fouh, V. Karavirta, D.A. Breakiron, S. Hamouda, S. Hall, T.L. Naps, and C.A. Shaffer. 2014. Design and Architecture of an Interactive eTextbook – The OpenDSA System. *Science of Computer Programming* 88 (Aug. 2014), 22–40.

[9] H. Manzoor. 2019. Disseminating Learning Tools Interoperability Standards. (2019), 91.

[10] H. Manzoor, K. Akhuseyinoglu, J. Wonderly, P. Brusilovsky, and C.A. Shaffer. 2019. Crossing the Borders: Re-Use of Smart Learning Objects in Advanced Content Access Systems. *Future Internet* 11, 7 (2019).

[11] University of Wisconsin-Madison. 2020. Canvas - Enabled Application Configurations (UW-Madison). (2020). **https://kb.wisc.edu/page.php?id=65466**

[12] T. Sirkiä and L. Haaranen. 2017. Improving Online Learning Activity Interoperability with Acos Server. *Software: Practice and Experience* 47, 11 (2017), 1657–1676.