Bringing Non-programmer Authoring of Intelligent Tutors to MOOCs

Vincent Aleven
aleven@cs.cmu.edu
Human-Computer Interaction Institute
Carnegie Mellon University

Ryan Baker
Baker2@tc.columbia.edu
Human Development
Teachers College
Columbia University

Yuan Wang
Elle.wang@columbia.edu
Human Development
Teachers College
Columbia University

Jonathan Sewall
sewall@cs.cmu.edu
Human-Computer Interaction Institute
Carnegie Mellon University

Octav Popescu
octav@cmu.edu
Human-Computer Interaction Institute
Carnegie Mellon University

L@S 2016, April 25-26, 2016, Edinburgh, Scotland Uk
ACM 978-1-4503-3726-7/16/04.
http://dx.doi.org/10.1145/2876034.2893442

Abstract
Learning-by-doing in MOOCs may be enhanced by embedding intelligent tutoring systems (ITSs). ITSs support learning-by-doing by guiding learners through complex practice problems while adapting to differences among learners. We extended the Cognitive Tutor Authoring Tools (CTAT), a widely-used non-programmer tool kit for building intelligent tutors, so that CTAT-built tutors can be embedded in MOOCs and e-learning platforms. We demonstrated the technical feasibility of this integration by adding simple CTAT-built tutors to an edX MOOC, "Big Data in Education." To the best of our knowledge, this integration is the first occasion that material created through an open-access non-programmer authoring tool for full-fledged ITS has been integrated in a MOOC. The work offers examples of key steps that may be useful in other ITS-MOOC integration efforts, together with reflections on strengths, weaknesses, and future possibilities.

Author Keywords
Intelligent tutoring systems, ITSs, MOOCs, Interoperability, Feasibility study, Log data analysis

ACM Classification Keywords
K.3.1 Computer Uses in Education*:* Distance learning

CTAT Example-tracing Tutor Features

- Provide custom-designed problem-solving interfaces that break down complex problems
- Provide step-by-step guidance within problems, not just end-of-problem feedback
- Support immediate and delayed feedback on steps
- Provide on-demand hints that give guidance
- Can follow students with respect to multiple different solution paths through problems, regardless of which path the student pursues
- Track individual student's knowledge growth in terms of fine-grained knowledge components
- Select problems adaptively, on an individual basis, so students practice un-mastered knowledge components until they master the targeted knowledge

Introduction

A recent study suggests that, in MOOCs, learning-by-doing has a stronger influence on learning outcomes than learning by reading or by watching [5]. MOOCs have previously supported learning-by-doing for example by posing questions with feedback and assignments with peer grading.

By contrast, intelligent tutoring systems (ITS) support learning-by-doing through practice on sets of complex problems. They break problems into steps, provide step-level guidance within problems, monitor individuals' knowledge growth, and adaptively select problems. There is substantial evidence that ITS can enhance student learning, compared to other methods of instruction [6][8][11][13][15]. Although at least one MOOC has incorporated an ITS [12] and others have incorporated ITS-like elements (e.g., [9][10][14]), we are not aware of prior integration into a MOOC platform of a full-fledged ITS, created with an openly available dedicated ITS authoring tool.

It makes sense therefore to investigate whether and how ITS and MOOCs can be synergistic. Given that ITS and MOOCs have evolved separately, they first need to be technically integrated. This technical integration is not straightforward, however. In this paper, we present a pilot study that tested the feasibility of this integration in the context of the edX MOOC "Big Data in Education". We focus on example-tracing tutors [1], a type of tutoring system that can be built without programming, using dedicated authoring tools.

CTAT: Non-Programmer Authoring of Tutors

The Cognitive Tutor Authoring Tools (CTAT) support the authoring of example-tracing tutors using a variety of end-user programming techniques: drag-and-drop interface building, programming by demonstration, graph editing, formula writing, and template-based mass production [2]. CTAT example-tracing tutors support key ITS behavior (see sidebar). CTAT is integrated with the Tutorshop, a web-based learning management system (LMS) we created for tutor use in classrooms. This tutor development environment has been used by roughly 650 authors for many dozens of research projects [2]. 44,000 students have used CTAT-built tutors, mostly in classrooms, at all levels of education, from elementary school through college.

We recently completed the technical integration of CTAT/Tutorshop and edX [3] by: (a) Reimplementing the "tutor engine" (which provides within-problem guidance) in Javascript, (b) implementing the Learning Tools Interoperability (LTI; [7]) Provider interface in the Tutorshop. This implementation enables us to integrate CTAT tutors in edX or any other LTI-compliant learning management system.

Pilot Study

Using CTAT, we created 8 tutored activities, each with 8-13 steps, for use in the edX MOOC "Big Data in Education" (Figure 1). The tutors provide support for assignments in which students analyze data sets using standard tools such as RapidMiner to, for example, build prediction models using different classification algorithms and validation schemes. The tutors provided brief instructions for each subtask to be carried out in RapidMiner, provided feedback when the subtask had been completed and the result entered into the tutor, provided hints upon the student's request, and revealed instructions for the next subtask when the previous had been completed. Although this tutoring behavior is simple, it was appropriate for the given assignments.

Creating an example-tracing tutor in CTAT

1. Create problem-solving interface An author creates problem-solving interfaces for the targeted problem types, which break the problem into steps. They can be built through a Flash or Java IDE, or by writing code in HTML.

2. Create a behavior graph The author then demonstrates, on the problem-solving interface, solution paths that students might reasonably take, which CTAT's Behavior Recorder records in a *behavior graph.* She generalizes the behavior graph with formulas and other means so it represents an appropriately broad range of student behavior.

3. Prepare graph for tutoring The author attaches hints, feedback messages, and knowledge component labels to the graph.

At student run time, the tutor uses the graph to interpret and guide students' problem-solving behavior, and to track students' knowledge growth.

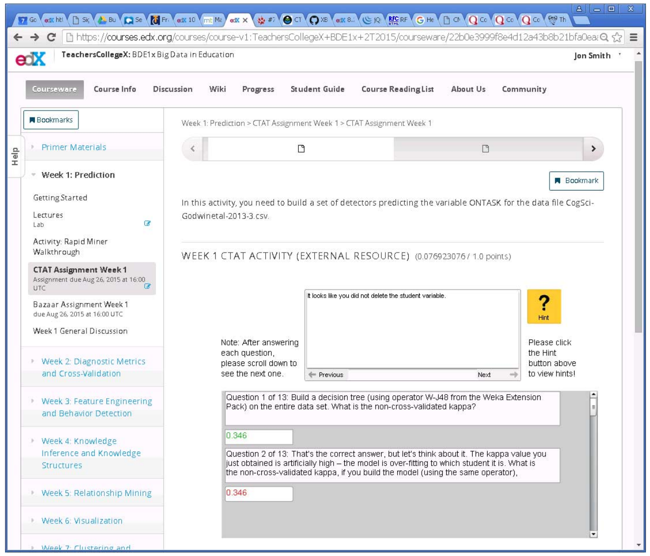

Figure 1: CTAT tutor in edX MOOC "Big Data in Education"

As is the case for all CTAT tutors, all student-tutor interactions were logged to DataShop [4], a large online repository of educational technology data sets, together with tools for analysis. The analysis that follows is based on exports from DataShop data. Of the 10,358 students who enrolled in the course, 559 started the first tutor activity and 86 started the last. The moderate hint and error frequencies (Figure 2; chart generated by DataShop) suggest that these assignments were appropriately challenging and that the tutor provided useful guidance to many students.

ITS log data and tools like DataShop play an increasingly important role in iteratively refining tutors [4]. For example, they help tutor authors identify common errors so that they can extend their tutor to give specific feedback when these errors occur. Also,

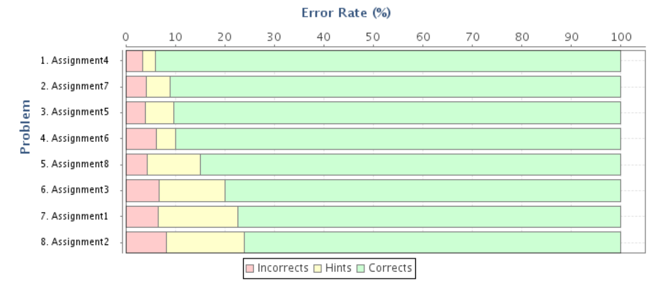

Figure 2: DataShop report displaying error rate (red) and hint use (yellow) on the tutor activities

authors can use log data to analyze what is most challenging for students and to refine the tutor's underlying knowledge component model, which can help make the tutor more effective or efficient [1].

Discussion

Our study demonstrates the technical feasibility of embedding example-tracing tutors in MOOCs. Any example-tracing tutor built with CTAT can now be embedded in any LTI-compliant MOOC or e-learning platform. The work thus brings non-programmer authoring of one class of ITSs to MOOCs. To the best of our knowledge, it represents the first integration of a MOOC and a full-blown ITS created with a free ITS authoring tool (http://ctat.pact.cs.cmu.edu). Although the tutors used in this study were rather simple, the integration supports all adaptive tutoring behaviors of CTAT tutors. The same means of integration (i.e., engineering the tutor so that it can run as a self-contained learning object in a browser; making it compatible so that an LTI-compliant LMS can launch it) could be used for other MOOCs and with other ITSs. There are some limitations, however. First, instead of launching the tutors from a separate server, it would be

ITS in edX Course "Big Data in Education"

- Eight CTAT tutors, a new one assigned each successive week
- Each tutor included 8-13 steps, some requiring analysis with tools such as RapidMiner
- Course certification required completing 70% of tutored activities
- 10,358 students enrolled, 114 received a certificate of completion
- 559 students started the first tutor assignment, 86 started the last tutor assignment (the 8th)

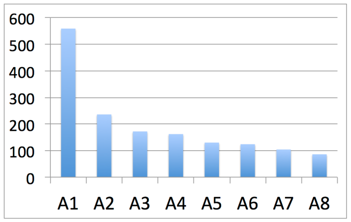

Figure 3: Number of students who started each tutor problem

preferable to host them on the edX site itself. We are working toward achieving this goal via use of edX XBlocks, but this integration would be an edX-only solution, unlike the LTI integration. Second, the LTI 1.1 standard limits the amount of data about student performance that the tutor can pass to the edX LMS, although this limitation may be loosened in LTI 2.0.

We see this work as expanding the options for learning-by-doing in MOOCs and for applying the one-on-one coaching of ITS at scale, where detailed, individualized guidance can be hard to provide. Further, it opens the door for addressing research questions on how typical MOOC activities and ITSs can best be combined and whether students learn better as a result. A key idea – not novel, but we are not aware of any implementation – is to share a student model across the ITS and the other MOOC elements. This integration could enhance the capabilities for adaptivity in both. For example, the tutor's adaptive behaviors could depend on what videos the student has watched and how well she did on quizzes embedded in those videos. Conversely, the system may recommend videos that help with content with which the student struggled in the tutor.

References

[1] Aleven, V. and Koedinger, K.R. 2013. Knowledge component approaches to learner modeling. In *Design recommendations for adaptive intelligent tutoring systems,* Army Research Lab., Orlando, FL, 165-182.

[2] Aleven, V., McLaren, B.M., Sewall, J., van Velsen, M., et al 2016. Example-Tracing tutors: Intelligent tutor development for non-programmers. *International Journal of AI in Education.* 26, 1.

[3] Aleven, V., et al 2015. The beginning of a beautiful friendship? Intelligent tutoring systems and MOOCs. *Proc. AIED 2015,* 525-528.

[4] Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., et al 2010. A data repository for the EDM community: The PSLC Datashop. In *Handbook of Ed Data Mining,* CRC Press, Boca Raton, FL, 43-55.

[5] Koedinger K.R., Kim J., Jia J.Z., McLaughlin E.A., Bier N.L. 2015. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. *Proc. Learning at Scale 2015,* 111-120.

[6] Kulik, J.A. and Fletcher, J.D. 2015. Effectiveness of intelligent tutoring systems. *Rev. Ed. Res.*

[7] https://www.imsglobal.org/activity/learning-tools-interoperability

[8] Ma, W., Adesope, O.O., Nesbit, J.C., and Liu, Q. 2014. Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology.* 106, 4, 901.

[9] Mitros P. and Sun F. 2014 Creating educational resources at scale. *Proc. ICALT 2014,* 16-18.

[10] https://www.oppia.org/

[11] Pane, J.F., Griffin, B.A., McCaffrey, D.F., and Karam, R. 2014. Effectiveness of Cognitive Tutor Algebra I at scale. *Educational Evaluation and Policy Analysis* 36, 2, 127-144.

[12] https://schoolyourself.org/

[13] Steenbergen-Hu, S. and Cooper, H. 2014. A meta-analysis of the effectiveness of intelligent tutoring systems on college students' academic learning. *Journal of Educational. Psychology.* 106, 2, 331-347.

[14] Strader, R. 2015 OLI and edX Collaborate on New Feedback Mechanisms for Learners. Open edX blog. https://open.edx.org/blog/oli-and-edx-collaborate-new-feedback-mechanisms-learners

[15] VanLehn, K. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational. Psychologist.* 46, 4, 197-221.