

A Technologist's Agenda for Scriptable, Smart, Social, and Republishable Courses

William Billingsley
University of New England
Armidale, NSW 2350, Australia
wbilling@une.edu.au

ABSTRACT

This position paper describes the personal agenda of a lesser-known technologist and academic, who has been working on smart and social education technology since the early 2000s.

Since 2015, I have also been the coordinator for a bachelor-level computer science degree at a regional Australian university with a small on-campus cohort and larger online cohort. Market pressures encourage us to pursue greater flexibility in modes and times of study, greater authenticity in how study relates to industrial practice, and greater connectivity in how our systems and studies interact with those of the wider community. As a technologist, this also motivates me to draw from four previous projects, to try to create technologies and techniques to make this ever-greater flexibility viable for a small computer science teaching team.

By combining techniques from these projects, I aim to produce a small open source set of utilities to support publishing of smart, social, courses that support authentic tasks and can run and be managed in a variety of formats.

CCS Concepts

•**Social and professional topics** → *Computer science education; Software engineering education*; •**Applied computing** → *Interactive learning environments; Collaborative learning*;

Keywords

Online learning; intelligent tutoring systems; educational technology; studio teaching

1. INTRODUCTION

The goal of this project is to support publishing of multi-format courses that are smart, social, and well-connected to external systems for authentic tasks such as studio courses. This draws on previous projects addressing each of these aspects: smart interaction on open-ended problems, interaction on social and authentic tasks, and multi-format course

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CSSPLICE '19 Minneapolis, Minnesota, USA

© 2019 ACM. ISBN 123-4567-24-567/08/06.

DOI: 10.475/123_4

publishing. As this describes an agenda for work I am currently undertaking, the bulk of this position paper describes the previous systems and how they relate to the problem. These systems are designed to be connectivist: in each case, they were designed to support connections to a variety of different models, technologies, or providers.

2. THE INTELLIGENT BOOK

We developed the Intelligent Book in the early 2000s as a suite of technologies for smart flexible online learning, inspired by John Seely Brown's idea of reactive learning environments [9] that could respond to students intelligently on open-ended problems.

Unlike traditional intelligent tutoring systems, the Intelligent Book was not specialised to a particular topic or style of question, but provided an architecture for including different domain models and interactive graphical UI components for different problems. We had observed that even within single subjects, such as Discrete Mathematics, teachers asked a wide variety of questions. So, for example although we supported formal mathematical proofs using one of the first blocks-programming languages for mathematics [3, 5], we also observed that many questions in a university mathematics course are asked in English prose and representing them formally would defeat the purpose of the question. Consequently, we also developed informally modelled questions [4]. In other topics, to prove the architecture, we also developed interactive questions for electronics [6] and high school geography (unpublished).

Recognising that students also spend time learning away from the computer (eg, when they are interacting person to person) it did not presuppose a fine-grained student model, but allowed each module to use a pedagogical script that could update its knowledge of a student in a document database that was then available to all other modules.

However, at the time of our work in this area, browser ecosystems were undergoing significant changes. At the start of the project, browsers did not natively have consistent support for interactive graphics (eg, SVG) or two-way communication with the server (eg, websockets). Now that browsers have become an application platform in their own right, many of the work-arounds we required are no longer necessary, and it also becomes more practical to write an Intelligent Book that use distributed domain models, rather than requiring the content to be brought together into a single server. Refreshing the Intelligent Book also provides an opportunity to consider how its techniques apply to social and authentic tasks.

3. ASSESSORY

Assessory was introduced in the early 2010s to facilitate studio teaching, first at the University of Queensland [7] and then with video support at the University of New England [2, 8].

Its original mission was to facilitate studio critiques for large classes. Students would present their progress in a project, either in-person or via video, and would then be allocated other students or groups to critique. The recipients of those critiques then assess the usefulness of the critique. Assessory is used in three subjects at the University of New England (UNE), each of which uses different systems for the students' project work.

As studio courses also tend to use authentic development tools, it has also become a natural locus for connectivity between the Learning Management System and the external tools. Initially, this was for the simple matter of linking a student's GitHub account to their student identity. However, with that link in place, it becomes attractive to gather data on students' and groups' work into Assessory. While currently students post links to the issues, code branches, and tests they have worked on, this is information that can be gathered automatically.

As well as being useful for student-to-student critique, data from these systems would also be useful for automated critique - adopting the techniques of the Intelligent Book to allow artificial intelligence to advise students on their work in authentic tasks.

4. TWEAKED.INFO

Tweaked.info began as a proposal to an entrepreneurial hackathon in Sydney in 2014. I had been offered a full-time academic position at the University of New England (UNE) after teaching for some years as an adjunct at the University of Queensland (UQ) while working professionally as a research engineer. My colleague at UQ and I were interested in continuing to collaborate on our studio software engineering course, but our institutions had different term lengths and different cohorts (UNE being predominantly online and UQ entirely on-campus). We would, therefore, find ourselves managing different versions of the same course that would need to be able to accept updates from each other.

After arriving at the University of New England (UNE), I discovered the university is beginning to encounter the same problem internally. As well as teaching trimester-long courses, it would like also to offer intensive and slower-paced versions of the same content, to support students whose employment workload causes their time for study to be variable. As the variety of student backgrounds, locations, degrees, modes, and cadences of interaction grows, the class undergoes fragmentation [1]. Course design begins to be shaped by the need for flexibility, and this creates new technical needs that current Learning Management Systems do not support well.

While teaching the course at UQ, I had developed a interactive learning system (Impressory) that attempted to bridge live in-class conversations with an out-of-class newsfeed. The intent of this system was that the two-thirds of on-campus students who do not physically attend their lectures would still feel brought into the conversation. The content model was similarly connectivist, allowing web content to be sequenced alongside local content even within the

same lecture. This had led to a design based around a semi-structured document database, allowing it to grow with new kinds of content item and to share content entries (and sequences) between courses.

Tweaked takes the content model from Impressory, and directs it towards course self-publishing. As a proof-of-concept, this has allowed course components to be shared between courses, dynamic modelling of course prerequisite pathways and skills progressions for accreditation, and flexible pre-viewing of course content.

5. CONCLUSION

This project seeks to produce a set of small utilities that allow scriptable, smart, social, and republishable courses. It is a case of a technologist solving their own problem. I can so I must. It is also connectivist, and seeks to use simple semi-structured documents to enable easy extraction of information from disparate systems. A course definition contains different "kinds" of content entry, which causes it to extract salient information from systems using different connectors.

6. REFERENCES

- [1] W. Billingsley. The case of the fragmented classroom. In *International Conference on Software Engineering*, 2019. in press.
- [2] W. Billingsley, B. Ngu, H. Phan, N. Gromik, and P. Kwan. Using a video-based critique process to support studio pedagogies in distance education—a tool and pilot study. In S. Barker, S. Dawson, A. Pardo, and C. Colvin, editors, *Show Me The Learning. Proceedings ASCILITE 2016*, pages 43–48, 2016.
- [3] W. Billingsley and P. Robinson. Towards an intelligent textbook for discrete mathematics. In *Proceedings of the 2005 International Conference on Active Media Technology*, AMT2005, pages 291 – 296. IEEE, 2005.
- [4] W. Billingsley and P. Robinson. Searching questions, informal modelling, and massively multiple choice. In *ALT-C 2007 Research Proceedings*, pages 159–168. Association for Learning Technology, 2007.
- [5] W. Billingsley and P. Robinson. Student proof exercises using MathsTiles and Isabelle/HOL in an Intelligent Book. *Journal of Automated Reasoning*, 39(2):181–218, 2007.
- [6] W. Billingsley, P. Robinson, M. Ashdown, and C. Hanson. Intelligent tutoring and supervised problem solving in the browser. In *Proceedings of the IADIS International Conference WWW/Internet 2004, Madrid, Spain*, pages 806 – 811, 2004.
- [7] W. Billingsley and J. R. H. Steel. Towards a supercollaborative software engineering MOOC. *Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014*, pages 283–286, 2014.
- [8] W. Billingsley, R. Torbay, P. R. Fletcher, R. N. Thomas, J. R. H. Steel, and J. G. Süß. Taking a studio course in distributed software engineering from a large local cohort to a small global cohort. *ACM Trans. Comput. Educ.*, 19(2):13:1–13:27, Jan. 2019.
- [9] J. S. Brown, R. R. Burton, and A. G. Bell. Sophie: A step toward creating a reactive learning environment. *International Journal of Man-Machine Studies*, 7(5):675–696, 1975.