Problem

At this very moment, hundreds of professors in the United States are simultaneously preparing lessons on transposing a matrix. They are doing so largely without receiving feedback from one another or directly building on one another's experience [1]. In this way, professors spend an enormous amount of time duplicating curriculum development efforts already tackled by colleagues. What's worse is that these efforts are rarely, if ever, reviewed by, shared with, or extended upon by peers.

Open source software development provides an excellent example of a possible solution. These communities have mastered distributed expert collaboration, dynamic peer review, and de-duplication of effort. In open source software, developers share code revisions in online repositories, review one another's work in small chunks [2], and contribute back their own improvements to the main project.

So, why aren't professors sharing their lesson materials online, collaborating on canonical lesson sets, diffing and merging similar lessons, and reviewing one another's learning materials? **Our goal is to explore the possibility that curriculum development for university courses can operate like open source software development does.**

Attempts at this have been made before. Most notable among these is Software Carpentry (which with the PI is quite involved). Unfortunately, our experience showed that forks of the original material tend to diverge without any strong incentive for contribution back to the master material [3, 4]. We expect that fine-grained modularity of lesson components and a clear dependency graph of prerequisite modules for each lesson may assist in overcoming the challenges encountered by Software Carpentry and others.

Proposed Solution

We propose a small-scale proof-of-concept for collaborative, open source, curriculum development to improve the transfer of lessons learned between instructors of the same course (either at a single university or among different campuses). This prototype collaboration will provide a template which could be adopted for collaboration among faculty teaching courses with an inherently larger scale (e.g. CS101).

Faculty in Nuclear Engineering from four institutions[1] have agreed to be participants in this prototyping effort. We all currently use GitHub to store, revise, and collaborate on research, especially source code. Additionally, this select group already started to host our course curricula online as well, but these are typically single author repositories (e.g. [5]).

The participants will collaborate on a master set of learning modules for an upper-division course in nuclear engineering : The Nuclear Fuel Cycle. In the near term, we will develop fine grained lessons which may be mixed and matched to meet the learning objectives. The curriculum will be hosted on GitHub, tested by all of us, and improved continually.

A two-day kick-off workshop ($15K) at UIUC will allow the participants to sketch the initial framework of the collaboration and coordinate logistics. The workshop will deliver:

- learning objectives[6] and a concept map[7] for the course

[1] P. Wilson at (UW Madison),Steven Skutnik (UT Knoxville), J. Roberts (Kansas State University), Anthony Scopatz (University of South Carolina)

- a list of fine-grained lesson topics
- a directed acyclic graph describing dependencies among lessons
- individual assignments for curriculum development
- decisions to adopt specific raw formats for lesson content (markdown vs. LaTeX, Jupyter vs MATLAB, etc.)
- a repository structure for organizing the lesson material (at this stage, compatibility with frameworks like RELATE [8, 8] and tools like PrarieLearn [9] will be considered)

Between Workshops work will commence on developing lessons. These lessons will be submitted by pull request to the main repository and each lesson will include:

- associated learning objectives (identified previously)
- content (e.g. speaking notes, presentation material, derivations, worked examples, active learning exercises, external readings, videos, images)
- learning assessments (e.g. project descriptions, exam questions)

Interactions will commence primarily via issues, pull requests, and reviews on GitHub during this time. Meanwhile, monthly video conferences will help to spur high-level conversation onward.

A two-day retrospective workshop ($15K) at UIUC will wrap up the project. Discussion concerning the process will allow reflection as well as a collection of lessons learned. Experiences live testing the developed course material (for example in NPRE412 at UIUC in Fall 2017) will be shared and suggested improvements will be captured as feature requests on GitHub. The process will be captured in a template repository and a collaboration instruction manual will be produced for use by other groups of faculty seeking to collaborate in a similar way.

Potential Impact

This work will provide an important proof of concept for groups of instructors willing to collaborate on open source curriculum for:

- core courses with many subsections in a single university
- niche courses taught by a select group of professors across universities
- fundamental courses in small fields (e.g. nuclear engineering)

Departmental Support

The Department of Nuclear, Plasma, and Radiological Engineering will support this work in kind with release time for Kathryn Huff through her Start-Up funds. Additionally, NPRE will support workshop activities with administrative effort and by providing space. Finally, collaborating external participants will contribute in kind with summer time hours.

References

[1] Elizabeth Green. *Building a better teacher: How teaching works (and how to teach it to everyone)*. WW Norton & Company, 2014.

[2] Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, **Kathryn D. Huff**, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, and Paul Wilson. Best Practices for Scientific Computing. *PLoS Biol*, 12(1):e1001745, January 2014.

[3] Gregory V. Wilson. Software Carpentry: Lessons Learned, July 2014. https://www.youtube.com/watch?v=1e26rp6qPbA.

[4] Greg Wilson. Software Carpentry: lessons learned. *F1000Research*, February 2014.

[5] **Kathryn D. Huff**. NPRE412, 2017. https://github.com/katyhuff/npre412.

[6] B. S. Bloom, D. R. Krathwohl, and B. B. Masia. *Bloom taxonomy of educational objectives*. Allyn and Bacon, Boston, MA. Copyright (c) by Pearson Education.< http://www. coun. uvic. ca/learn/program/hndouts/bloom. html, 1984.

[7] Joseph D. Novak. Concept mapping: A useful tool for science education. *Journal of research in science teaching*, 27(10):937–949, 1990.

[8] Andreas Kloeckner. RELATE is an Environment for Learning And TEaching, 2017. https://github.com/inducer/relate.

[9] Matthew West. Prairielearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning. *age*, 26:1, 2015.