

Free and Open Source Software (FOSS) in Distance Education

Dr. Karl Stolley

Assistant Professor of Technical Communication
Illinois Institute of Technology
218 Siegel Hall
3301 S Dearborn
Chicago, IL 60616

karl.stolley@gmail.com
<http://karlstolley.com/>

Freddrick Logan

Graduate Student (PhD in Technical Communication)
Illinois Institute of Technology
218 Siegel Hall
3301 S Dearborn
Chicago, IL 60616

flogan1@iit.edu
<http://freddricklogan.com/>

Matthew Ephraim

Graduate Student (MS in Information Architecture)
Illinois Institute of Technology
218 Siegel Hall
3301 S Dearborn
Chicago, IL 60616

matthew.ephraim@gmail.com
<http://www.mattephraim.com/>

Brief Project Description

In this project, we will create the core of a publicly available and editable library of instructional modules that support and encourage the use of free and open source software (FOSS) in technical communication distance learning classes.

Students enrolling in online classes may do so in part because they lack the financial resources to attend classes in traditional classroom settings. Requiring these students to purchase expensive software in addition to their online tuition may represent a hardship, if not an outright deterrent, to enrolling in distance classes. Our library of modules for achieving technical communication goals through FOSS, like OpenOffice.org or NeoOffice, represents a significant step towards enabling students' equitable participation in online courses, regardless of the software a student can afford to purchase for her personal machine.

In addition to helping ensure equitable learning and participation for distance students, our modules will also aid teachers who may find their teaching negatively affected by wild variations in students' software access. Traditional classes that meet in a computer lab classroom ensure equal software access for all students (to a degree—students who can afford to purchase the same software for their personal machines enjoy the advantage of working outside the lab). Distance learners, in stark contrast, must rely on whatever technology is available to them on personal or family computers. Students' lack of common software access can result in all sorts of unnecessary pedagogical challenges for teachers, including teachers' being unable to open assignments that students submit in unusual file formats.

Our modules are not just another set of software tutorials, however; they are geared for using software to solve particular, contextualized technical communication challenges (as opposed to exploiting features of specific software). That is, the modules will help students to establish a rhetorically grounded (rather than software-specific) digital literacy for use in professional and in-class contexts.

Although rooted in courses offered at our home institution, we believe that these modules will have appeal and use at other academic institutions and in professional contexts.

All of our FOSS-based modules will:

- Feature fine granularity, allowing teachers and students to reference/link to individual modules or clusters of modules for supporting specific communication tasks
- Address a dual audience of instructors and distance students, who may be equally unfamiliar with FOSS
- Be prepared and distributed in a format that is publicly available and editable
- Include mixes of textual, visual (image/video/screen captures), and audio presentations to support different learning styles and physical abilities
- Promote and use free and open formats to enable universal, device-/ability-independent access
- Be issued under a Creative Commons Share-Alike license to allow and encourage derivative works and institution-specific development and modifications

Significance to Members of CPTSC

Software access is a crucial, if under-explored, programmatic issue in technical communication, for both physical classrooms and distance learning: simply stated, software has become an integral part of most, if not all, courses in technical communication (representing shifts in industry, to be sure). Whether programmatic decisions regarding software take the shape of budgeting for computer labs at the departmental or institutional level, or requiring the use of a specific software package for a particular course (e.g., InDesign for document design; Dreamweaver for Web design), software access is a thorny programmatic issue that our field can neither ignore nor leave to chance.

Our project will emphasize digital literacy across software of a given type (word processor, image editing, Web authoring) rather than literacy tied to a specific piece or brand of software. FOSS should only be seen as a vehicle toward establishing digital literacy, and not merely a free alternative to expensive software. In other words, we hope to challenge not just the *economic* dimensions of access, but the *rhetorical/literate*—and therefore *programmatic*—dimensions as well. That is why we intend to publicize and make available these modules to the entire field, instead of keeping them cloistered at our own institution.

It is essential that technical communication programs (and indeed the entire field) begin questioning and responding to how the digital literacies our students develop prepare them for an era when there are not only many different software packages oriented toward the same communication activities, but also a proliferation of versions (and the variations in tasks these versions introduce) across even the same piece of software.

The modules in our library will be designed in such a way as to enable students to work with Microsoft Word or Mac Pages (part of the Apple iWork suite) just as effectively as FOSS like OpenOffice: an important professional, rhetorical, and programmatic goal, as software packages and versions vary widely from institution to institution, organization, and workplace. We believe that these modules and their emphasis on FOSS will eventually help technical communication program administrators to better understand the complex dynamic of software as it operates in technical communication, and to make programmatic software decisions where necessary—while seeing that there is much that can be learned from *any* software.

Potential Publication Locations

Because distance learning and access problems are cogent and pressing, we expect that a number of journals would be viable venues for publishing about this project. *Technical Communication Quarterly*, the *Journal of Business and Technical Communication*, *Business Communication Quarterly*, the *Journal of Distance Education*, and *IEEE Transactions on Professional Communication* all have histories of publishing articles on distance education. Our particular emphasis on software technology and rhetorically framed digital literacy vis-à-vis technical communication and program administration would no doubt garner interest in this era of spiraling software costs and choices.

We also expect that the process and results of building these software modules would be of interest not only to academics, but also to working professionals creating distance or online learning programs in the workplace and other non-academic settings. Campuses often enjoy licensing agreements with companies like Microsoft and Adobe to keep software costs down, agreements that businesses rarely enjoy—perhaps making the selection of software and its use even more critical in corporate settings.

Project Tasks and Timeline

- **Late Spring 2008**
 - Determine and/or develop a Web-available platform for delivering and editing the software learning modules
 - Target specific distance courses (e.g., technical editing, online design) for guiding the development of initial modules
 - Examine syllabi and extant software instructional materials from targeted courses
 - Develop prototype modules for further refinement and revision

- **Summer 2008**
 - Build initial modules around technical communication tasks, rather than software features
 - Develop contribution and technical guidelines for module development
 - Publicize modules on relevant listservs (ATTW-L, TechRhet) and news outlets (Kairosnews.org) and invite use and feedback
 - Hold workshops at home institution regarding FOSS and modules

- **Fall 2008**
 - Test module use in distance and/or traditional classroom settings
 - Present initial modules and findings at CPTSC

Budget Statement

Graduate Student Support, $\$25/\text{hour} \times 60 \text{ hours} = \1500

Because our team members, including the graduate student collaborators, will be working with free and open source software, our primary need for this project is attaining funding to support graduate students as research assistants (rather than acquiring hardware or software).