RIT's new minor in free and open source software and free culture: A five year journey driven by student-centered, applied research

Article · February 2015

DOI: 10.1109/FIE.2014.7044196

3 authors:

Stephen Jacobs
Rochester Institute of Technology
22 PUBLICATIONS **30** CITATIONS
SEE PROFILE

Amit Ray
Rochester Institute of Technology
5 PUBLICATIONS **14** CITATIONS
SEE PROFILE

Jon Isaac Schull
Rochester Institute of Technology
50 PUBLICATIONS **570** CITATIONS
SEE PROFILE

RIT's New Minor in Free and Open Source Software and Free Culture

A Five Year Journey Driven By Student-Centered, Applied Research

Stephen Jacobs, Amit Ray, Jonathan Schull
School of Interactive Games and Media, College Of Liberal Arts, MAGIC Center
Rochester Institute of Technology
Rochester, NY

***Abstract*— This paper briefly describes the five year [1] history of Free and Open Source Software course offerings and projects at RIT and how the Humanitarian focus of the student work built the foundation for the minor. It will then discuss the design of the minor, how the required courses lead to advanced electives via multiple paths, and how it and prepares students to become contributors and potential project leaders within their own future Free and Open Source Software and Free Culture communities.**

Keywords— Curriculum Design; FOSS; Open Source; Academic Minor

I. INTRODUCTION

RIT's New Minor in Free and Open Source Software and Free Culture (FOSS & FC) brings together professors and students from the Colleges of Computing, Liberal Arts and RIT's MAGIC Center to study some of the most compelling issues of the digital age and to join others around the world to develop and expand, real-world projects.

FOSS & FC technologies and practices are re-reshaping markets, business models, work practices, economies, and arguably governments. They represent a cultural shift as well as a technological shift, and they reflect a new arrangement of technologies and attitudes that cut across multiple academic disciplines. Like many universities RIT is seeking to increase its support for multi-disciplinary collaboration and project-based learning because it is important career preparation and because it is increasingly recognized as a best hope for addressing the world's problems. A minor that offers opportunities for project-based learning, and contribution-making across disciplines, institutions and countries can both benefit from, and contribute to emerging trends in technology, education, and economics.

This is the first Academic Minor at RIT to be offered across colleges. As far as the authors know, it is also the only Minor of its kind, though we are aware of several European Masters degrees in Open Source.

The work of the FOSS@MAGIC program, whose success has driven the creation of the minor, is sponsored in part, by a major gift from Red Hat. , inc. Individual activities of FOSS@MAGIC have received support from AT&T, inc. Amazon , inc and others.

Since offering its first seminar in Humanitarian Free and Open Source Software Development five years ago, [1] the School of Interactive Games and Media has seen a steady rise in student, faculty and staff interest in FOSS & FC. Not just from students across majors in Computing and Information Sciences but from the colleges of Engineering, Liberal Arts and Fine Arts as well. Students have taken additional courses, pursued independent studies, completed Summer Undergraduate Research Fellowship projects, and presented their work at conferences across the country and at the White House. Outside of academics and research on campus, they have also completed local, regional and international co-ops within the FOSS industry and used FOSS technology to participate in humanitarian hackathons as well. For many of our students, the opportunity to engage in humanitarian work beyond the classroom and the home institution is of particularly appealing [2], [3]

II. THE BASE FROM WHICH THE MINOR GREW

Since 2009 the FOSS@MAGIC initiative (formerly FOSS@RIT) has engaged hundreds of students, faculty and staff and community members in hackathons, professional development opportunities. It has created and contributed to dozens of FOSS projects. [4] In the summer of 2012, inspired by student enthusiasm and interest in further opportunities, the authors decided to develop a formal minor and some new courses to support it.

Despite the preponderance of students in computing programs in FOSS@MAGIC programs and events, there had been a enough participation from students in other programs to ensure that the design of the minor should allow university-wide enrollment. This also mirrored the broader interests of the authors, none of whom have degrees in computer science or engineering.

III. AN OVERVIEW OF THE MINOR'S DESIGN

The minor gives students direct experience in the creation of FOSS & FC projects while providing a deeper understanding of the theoretical and practical underpinnings of

the process, including its social, commercial and legal roots and impacts.

The required courses introduce FOSS & FC Communities of Practice and their methodologies of development and production. They examine the history and theory of FOSS & FC from the earliest days of the printing press to contemporary discussions of Open Access Journals and the tragedy of Aaron Swartz.

In the constrained electives within the minor, students can move from consumer, to contributor, to leadership of FOSS and FC projects and communities. In their final course in the minor, students take a recommended elective, ideally from their own major field of study, that uses FOSS &FC tools or processes. This allows them to function as FOSS & FC experts in their own majors, helping for other students, and perhaps even faculty move from consumers to contributors themselves. The design of the minor allows both technical and non-technical students a path to completion, because FOSS & FC communities require a wide range of participants ranging from coders, to technical writers to graphic artists program managers.

IV. THE THREE REQUIRED COURSES

Three required core courses comprise the minor's base. They are *Humanitarian Free and Open Source Software Development, Theories of Free and Open Source Culture*, *Legal and Business Aspects of FOSS*. The overall requirements for acceptance into the minor are that the student have 3rd year standing, that they have completed the University's required writing core and taken a two course sequence in programming or some form of content creation (2D Art, Animation, etc)

Humanitarian Free and Open Source Software Development introduces students to processes, practices, tools and technologies of the FOSS community and is the first FOSS & FC course RIT offered. In the earliest iterations, students developed educational games for the One Laptop Per Child and Sugar platforms; it has subsequently broadened to allow students to pursue other projects, if approved by the instructor.

Students who take the course are introduced to the GitHub version control system and other software development and management tools. It requires that students create and/or participate in production blogs, IRC meetings and follow other FOSS & FC distributed development practices. This course is where they make their first contributions to existing FOSS projects and create their own first projects. These projects generally require multidisciplinary teams, so students who lack strong programming skills can still contribute.

Theories of Free and Open Source Culture examines the historical emergence of the individual author and of intellectual property. It examines the role of software in highlighting changing content creation practices, facilitating new business and economic models, and inspiring emerging practices of open source, open access, participatory, peer-to-peer and "Free (as in speech, not beer)" cultures. It looks at societal reactions to copyright including such movements as Copyleft and the Creative Commons licensing system.

Legal and Business Aspects of FOSS examines practical implementations of the patterns considered in the course described above. It covers the impacts of "free software" and the varieties of FOSS and Creative Commons Licenses, Crypto Currencies, and more. It explores challenges for businesses that enable customers to redistribute or modify their actual product, and how this empowers the customer, the community and, in some cases, the world.

V. . THE FOURTH COURSE: PICK ONE OF TWO

Depending on their abilities and interests, students take either *Software Development on Linux Systems* or *Technical Writing*. In the former, students learn how to package software for release, and engage in version maintenance within the FOSS community. Lectures, lab exercises and projects cover topics such as Linux package managers, version control systems, potential license conflicts, development vs. production releases, bug tracking, maintenance managing, forking, patching and future development. The material is considered both from a management and an end-user perspective. The *Technical Writing* elective meets the constant need within FOSS & FC communities for solid documentation, instructions and related materials..

VI. BRINGING IT BACK AND PAYING IT FORWARD

For their final course in the minor, students choose from a list of seven (or eight, since a student who takes the Linux Systems course could also opt to take *Technical Writing*) electives. To qualify as an elective, a course must use a FOSS & FC technology or process as part of the instruction. (e.g., Virtual Reality uses Open Sim, a reverse-engineered Open Source version of Second Life.)

Additionally, most of them are required courses or advanced electives in other major programs or sequences on campus. As mentioned above, this qualification criterion allows students to bring FOSS & FC expertise back to their home program so that if students, or the professor, express a need or desire for change or improvement to the tools used by the course, the FOSS & FC minor student can become the class's expert, facilitate engagement between the class and the FOSS & FC community, change or fork the project to create their own version of the technology, etc. While future additions to the list of electives are anticipated, at the time of this writing the elective courses were…

From the College of Liberal Arts…

- *Introduction to Natural Language Processing*
- *Language Technology*
- *Text & Code*

From the B. Thomas Golisano College of Computing and Information Sciences

- *Unix-based System Forensics*
- *Foundations of Mobile Design*
- *Applications in Virtual Reality*
- *Project in FOSS Development*

Four of these approved electives are also required or elective courses within programs of the B. Thomas Golisano College of Computing and Information Sciences while the others are in the College of Liberal Arts. Some have no prerequisites outside of the Minor, others have deep prerequisite chains in the course's home program.

The Introduction to Natural Language Processing and Language Technology courses are part of a larger sequence in the English Department's computational linguistics program... They cover modeling and implementing natural language processing and digital text solutions. Students in the courses program in Python and use the Natural Language Toolkit and related tools (such as Weka).

Text & Code from the English department looks at intersections of text and literature in interactive narrative, digital games, mobile communication, geospatial mapping, interactive and locative media, augmented reality, and interactive museum design. Students use a wide variety of FOSS and FC tools such as Renpy and ARIS for their projects.

Unix-based System Forensics, from Computer Security program uses FOSS tools and techniques as part of its instructional model. Student projects include contributions to FOSS projects

Foundations of Mobile Design from the Information Science and Technology is a client-side UI/UX course that develops in HTML 5 and uses FOSS tools.

Applications in Virtual Reality is a graduate course in the Computer Science program. It uses, and occasionally contributes to, the FOSS virtual reality tool Open Sim and its community.

Project in FOSS Development, from the School of Interactive Games and Media was mentioned previously. It is an advanced project course with a focus that changes from year-to-year. Recent foci include developing games in the cloud on Red Hat's Open Shift platform and on developing applications with and for the Raspberry Pi platform. An independent study for advanced students is also possible for the "5th course" slot. As the program matures, we expect to add additional courses from other programs to broaden the reach of the minor and make it even more accessible to students across campus.

VII. Start Your Engines

The minor opened for registration just as this paper was submitted to the conference. We anticipate a strong start, due to an "installed base" of students who have taken *Humanitarian Free and Open Source Software Development* and/or *Project in FOSS Development previously*. Students can complete the minor in as few as three semesters.

RIT will be keeping a close eye on the first two years of the program, due to its status as the first cross-college minor. The University is actively seeking opportunities to "de-silo" its programs and if this one is successful, the model will likely be used for additional minors down the road.

References

[1] Jacobs, S. "Work in Progress Rochester Institute of Technology, One Laptop Per Child and Open Source; A Unique Approach to Service Learning", *IEEE FIE* 2010, Washington, D.C. October, 2010

[2] Morelli,R,Tucker, A, Danner, N, De Lanerolle, T,R, Ellis, H, J,C, Izmirli, O, Krizanc, D, Parker, G, "Revitalizing Computing Education Through Free and Open Source Software for Humanity", Communications of the ACM, 52, No 8, August 2009, 67-75.

[3] Jacobs, S. "Building an Education Ecology on Serious Game Design and Development for the One Laptop Per Child and Sugar Platforms." *IEEE 2nd International Games Innovation Conference*, Kowloon, Hong Kong, China. December, 2010

[4] http://yacht.rit.edu/timeline/timeline.html.