Adaptive Collaborative Unix Meta-Tutorial for Computer Science Students

Abstract

Although the ability to use Unix is a necessary prerequisite, many computer science (CS) students have little or no prior experience in Unix before they enter college. I am developing an online Unix tutorial in the form of a *meta-tutorial* that will accomplish two goals: 1) help students understand Unix concepts and how to make use of online resources, and 2) remove some of the barriers that impede equity in computer science programs.

Problem and Motivation

Unix is one of the most important computer operating systems available. It is portable, flexible, adaptable and is an ideal platform for developing research and software. Its structure allows students to learn a variety of computer science concepts. Although the ability to use Unix is a necessary prerequisite, many computer science (CS) students have little or no prior experience in Unix before they enter college. Unfamiliarity with Unix is an especially acute problem for women and non-traditional students who have had little exposure to communities wherein they could learn or build computing skills. Such students frequently encounter difficulties early in their computer science courses, which in turn affects continued performance, interest, and retention in computer science programs. Because I am woman and a non-traditional student, I wondered if a remedy to the Unix problem could be created that would improve the graduate school environment for students like myself. I hypothesized that providing an organized set of online resource as a Unix tutorial that was adaptive and supported collaboration would accomplish two goals: 1) help students understand Unix concepts and how to make use of online resources, and 2) remove some of the barriers that impede equity in computer science programs.

Background and Related Work

Previous research in the field of computer-science education shows a significant gap between incoming students' perceived and actual ability. Schools such as Carnegie Mellon University, Griffith University, and California State University Hayward (CSUH) have been working on bridging this gap [1,2], and success has been demonstrated in better retention rates among non-traditional students [5]. These programs have also demonstrated the importance of teaching students primary computing skills. Although Unix is a major tool for CS students, prior research conducted by the author and Harjot Sahwney at CSUH's Adaptive Hypermedia and Assistive Technology (AHAT) laboratory has revealed that many colleges and universities do not offer pre-requisite Unix courses or workshops [6].

Approach and Uniqueness

I am developing an online Unix tutorial in the form of a *meta-tutorial.* The tutorial is an organized collection of existing resources that adapts to individual students' learning processes, with tools to support collaboration. Each learner can explore the tutorial in a self-defined way. To collaborate, as learners explore the tutorial they can

comment and make notes, which can be made available to other learners. Each section of the content will be ranked based on a ranking function, which is proportional with the density of the comments for that section. I contend that overall confidence and interest outcomes in computer science will be different in the following groups of students, depending on whether they:

- Utilizing both the adaptive and collaborative capabilities of the tutorial
- Utilize the tutorial's adaptive capability, but not its collaborative one
- Utilize the tutorial's collaborative capability, but not its adaptive one
- Use the tutorial, but do not use either its adaptative or collaborative features
- Do not use the tutorial at all

My research project employs both *iterative design* and *participatory design* principles. Iterative design involves a series of development, testing and revision cycles [3]. Participatory design gives users an opportunity to be involved in the system's design by collecting data from them during different phases of the project [7]. By using this method, trails of documentation are generated unconsciously. In other words, the design processes contribute to learning goals, and contribute to the documentation and assessment as well.

Results and Contribution

Currently, I am developing the first version of the tutorial. I am organizing and assessing existing online Unix materials, with a strong emphasis on how the materials interact with the students' learning styles.

After developing the first version of the tutorial, I will collect data by alpha testing the tutorial on ten students, and later by beta testing on a larger group of students. Data collection will be done by capturing the daily interaction of students with the tutorial through their unconscious traces and by collecting individual data through assessments, task-based and open-ended interviews, and surveys.

I will perform ongoing and preliminary analysis, which will help me during different phases of design and development of the project. I will perform the final data analysis in order to measure the effectiveness of the tutorial, and success of my objectives. If the tutorial motivates computer science students and helps them perform better, it will be a significant tool for preparing students for academic success and can be widely disseminated among institutions wishing to retain and promote equity in their computer science programs.

As a demonstration of this ongoing research, I will present the annotated materials in the form of the meta-tutorial and aggregated results of alpha testing and partial beta testing, which will have implications for presenting Unix materials to non-traditional students.

References

1. Blum, L. (2001). Women in Computer Science: The Carnegie Mellon Experience. Retrieved July 29, 2002, from http://www-2.cs.cmu.edu/~lblum/PAPERS/women_in_computer_science.pdf
2. Hellens, L., Nielsen S., Doyle R., Greenhill, A., Bridging the IT Skills Gap. "*A strategy to Improve the Recruitment and Success of IT students*", Proceeding of 10th Australasian Conference on Information Systems, 1999.
3. Kelly, E. A., Lesh A. R. (2002). "*Design Experiments in Mathematics Education*", Retrieved September 10, 2002 from http://gse.gmu.edu/research/de/Lesh_Design Exp in Math Ed 6.pdf
4. Kelly, E. A. and Lesh A. R. (2000). Handbook of Research Design In Mathematics and Science Education. Lawrence Erlbaum Associates, Inc.
5. National Science Foundation. Women, minorities, and Persons with Disabilities 2000. Available from: http://www.nsf.gov/sbe/srs/nsf00327/. Accessed 2 March 2002
6. Sawhney, H. and Farzan, R. (2002). "*Teaching Unix Skills to Undergraduate and Graduate Students in Computer Science.*"
7. Strauss, A. and Corbin, J. (1990). "*Basics of qualitative research: Grounded theory procedures and techniques*", Newbury Park: Sage
8. Wexelblat, A. and Maes, P., "*Footprints: History-Rich Tools for Information Foraging*," In Proceedings of CHI 99, Pittsburgh, PA., pp. 270-277, May 1999