Three Ways of Enriching Classical University Courses with Web 2.0 Elements

Niels Pinkwart, Hannes Olivier, Frank Loll, Sven Strickroth

Clausthal University of Technology, Germany

Introduction

The use of different kinds of technology has become more and more common in academic teaching in recent years. For instance, campus management systems like Stud.IP offer students options for downloading presentation slides and other learning material, for receiving news about their class, and for discussing problems (e.g., in web based forums). Even these basic, commonplace systems already have some "Web 2.0" functions, since they allow students to interact with each other online. In this position paper, we outline three Web 2.0 methods of enriching University courses that go beyond "simple" discussion support or learning management systems. These methods and the systems that implement them are geared towards a classical University setting: a larger lecture (with several hundred attending students) together with some (5-10) exercise groups in which smaller groups of students can ask questions and have to present their solutions to exercises that they have to solve on a weekly basis. This model can be found frequently in classes in German (and other) Universities, especially in the engineering and natural sciences. The three methods for enriching this setting that we present in the following have with different purposes: a) supporting interaction and collaboration among the students, b) allowing students to practice, and c) supporting tutors who have to review exercise solutions.

Web-based lecture support

A classical method of providing web-based support for University lectures is to record the lectures on video and put these recordings online, for later re-use and as a service for students who missed a lecture or who did not understand a certain part during the live lecture and would like to see and hear these parts again. One weakness with these video recordings is the missing feedback channel: the existing systems provide a rather unidirectional information flow (content delivered to students). In most systems, students cannot directly ask questions if they do not understand the online lecture (of course, they could send mails or use a discussion forum, but these options are not connected well with the video material). We are currently in the process or developing a system (based on ILIAS technology) that allows students to pose questions they have, expressed textual or youtube video format. These questions can be attached to "the class in general", to a specific lecture, or even to a specific part of a lecture. This way, students can ask questions about the material as they are watching it, adding a video note and allowing other students or the professor to respond to it (again, by video or text).

A second feature we implemented in the system is designed to support the emergence of local study groups. Especially for first year students who have not established a lot of connections to other peer students yet, it is important and sometimes difficult to form study groups that practice together. Practically, it is of course helpful if members of a study group live more or less close to each other, to facilitate presence meeting. To increase student awareness about "who's close by", we embedded a GoogleMaps feature in our system so that students can see where other students enrolled in a class are learning/living, get in touch with them easier and start local study groups. Figure 1 shows a screenshot of the current system prototype. It will be finished in July 2009 and will be tested in practice in the fall of 2009/10 with a class of approxinately 300 students.

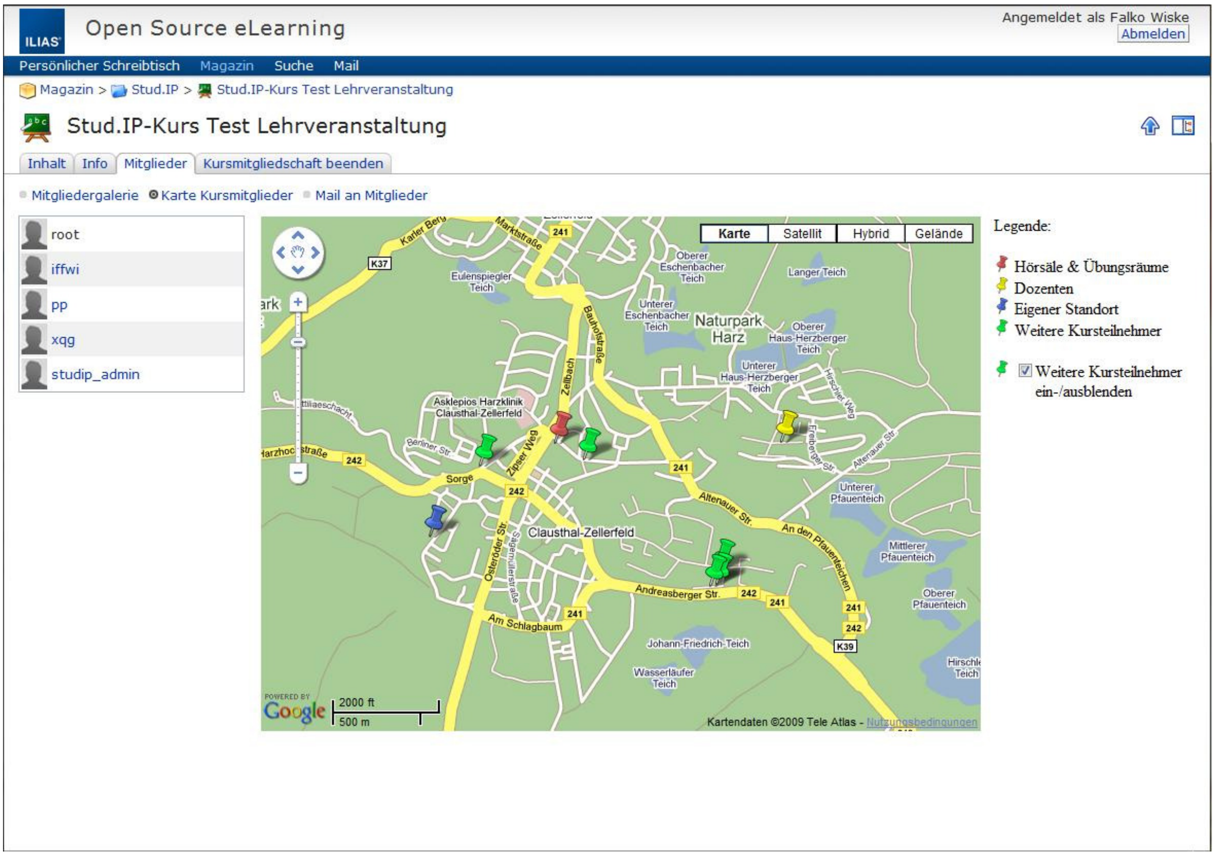

Figure 1. Using Google maps to locate peer students

Peer review as a practicing and learning method

Apart from being able to ask questions online and to build local study groups, students can also benefit from receiving *feedback* on the solutions they come up with as they work on exercises. Traditionally, student's exercise solutions are checked by the professor or someone from his staff. But this takes a lot of time, and students often have no chance to see alternate routes to the solution (since they only get feedback on their own). To address this problem, the system CITUC (*Collective Intelligence @ TU Clausthal*) was designed. The CITUC system (cf. figure 2) is a web-based eLearning system to disburden tutors from assessing student solutions by using peer reviews and collaborative filtering. While using the CITUC system, students work on exercises themselves before assessing and critiquing alternative solutions of other students. The system calculates a rating score for solutions (implemented as the average of received assessments weighted by the quality of the solution of the assessing student). Once a student has provided a solution and has assessed some alternative solutions, he is able to see all other alternative solutions for the corresponding task. The students also have the option of providing tasks on their own to clarify open questions, and they can comment on other solutions to highlight possible mistakes or potentials for improvements. Thus, the system promotes reviewing skills while at the same time allowing students to acknowledge and see different ways of solving a task. A controlled lab study and the use of the system in multiple University classes have shown that the underlying algorithm heuristic of the system leads to an acceptable quality (compared to human expert grading) once three to four peer-reviews are available, even in ill-defined tasks like text interpretations, which allow for more than one correct solution (Loll & Pinkwart, 2009a, 2009b). Furthermore, we have found evidence that the system is beneficial for students' learning activities.

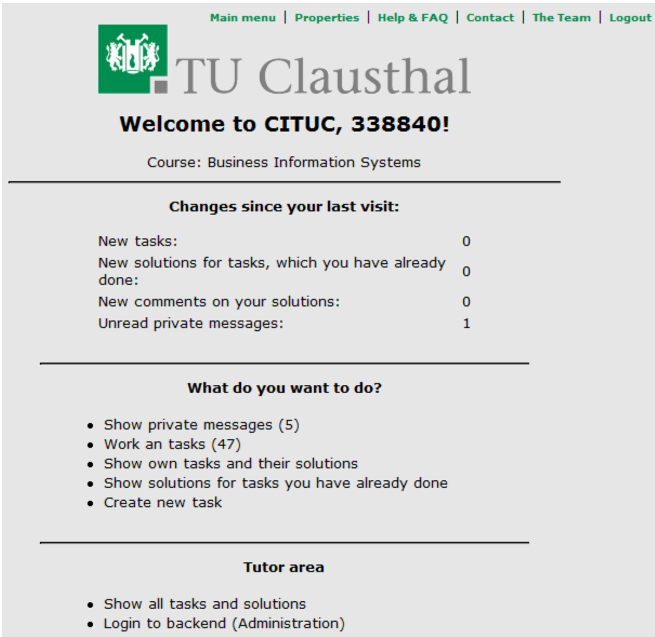

Figure 2. CITUC user interface with awareness information

Helping tutors to assess exercises

The two Web 2.0 techniques briefly shown above are primarily designed to help *students* through bringing them in contact with peers, allowing them to annotate and discuss learning material, and peer reviewing their solutions to exercise tasks. Especially for larger scenarios (such as in our case, with some hundred students and 10 exercise groups), also the *tutors* which run the exercise groups may benefit from computer support. A typical task for the tutors has two components: a) answering student questions and b) assessing specific homework tasks which students need to hand in as a prerequisite for the written exam. For b), an important aspect is to find out whether the students have actually solved the task themselves, or whether they just copied the solution of somebody else. For a smaller student group with a single tutor, this is a simple task and just means comparing the different solutions. With many groups in parallel and several hundred students, the situation gets more difficult: no single person will look at all solutions, and it is very time consuming to compare a solution to all others. Since students' submissions are graded by more than one tutor, this makes plagiarism detection very hard, because every tutor only knows the submissions sent to him for marking.

To support the detection of duplicates, which might go unnoticed in large groups of students where more than one person is checking solutions, we implemented a plagiarism detection system which is designed to facilitate interaction and discussion in the group of tutors by making them aware of possible cheating activities. The system is a modular web application designed for supporting the process of assessing homework. It has specific features for programming tasks (which is our main application field) and it supports the whole process of collecting, evaluating and marking solutions submitted by students.

In the system, participating students can be subdivided into groups with a tutor who is responsible for submissions of members of his group. Tutors can access the submissions via the web interface, review and grade them. To assist the tutors, three different functions are available:

a) A check if the submission is syntactically correct (makes sense for more formal tasks) – e.g., for submitted Java code, the system checks if the code compiles.
b) A functional check (if appropriate). For a programming task, the system can automatically detect if the if the output of the programs submitted by the students is correct for various inputs (black-box testing) or if JUnit-tests are successful (white-box testing).
c) Plagiarism tests which calculate the similarity of student submissions. Here, different checks can be performed automatically by the system after the submission deadline. In the current version, the system includes three different checks based on an adjustable normalization for plagiarism detection for simple tasks (where solutions tend to be similar anyway) and for more advanced tasks. The results of the plagiarism checks are presented to all tutors via the web interface as an overview.

All the test results are available for tutors online together with the submissions. For these reasons, there is no need for tutors to download all submissions in order to perform (plagiarism or function) tests locally. This is likely to save tutors a lot of time.

The system lets the tutor decide whether also the students can see the results of the tests. Indeed, this may have advantages (especially for the first two types of tests): If a student uploads a file as a solution to a programming exercise, the syntax of the file is checked automatically and the result is reported back to the submitter. This way a student can see whether his solution compiles and is complete (i.e. no required file is missing). If the function test is visible, students can also see if their program is working properly.

While the first and the second function (syntax and function tests) of the system are not primarily Web 2.0 related, the third one is: its main purpose is to provide the tutors with awareness about a situation worth looking at in detail, and stimulate discussion about this (i.e., let group of tutors decide how to handle a case of possible plagiarism).

We are currently finalizing the implementation of this online submission system. It will be available in August 2009 and will be tested in fall 2009/2010.

Literature

Loll, F., & Pinkwart, N. (2009a). *Using Collaborative Filtering Algorithms as eLearning Tools*. In R. H. Sprague (Ed.), Proceedings of the 42nd Hawaii International Conference on System Sciences. Los Alamitos (CA), IEEE Computer Society Press.

Loll, F., & Pinkwart, N. (2009b). *CITUC: Automatisierte Lösungsbewertung im E-Learning durch kollaboratives Filtern*. In In H. R. Hansen, D. Karagiannis, & H. G. Fill (Eds.), Tagungsband der 9. Internationalen Tagung Wirtschaftsinformatik Band 2 (p. 411-420). Wien (Austria), Österreichische Computer Gesellschaft.