Adapting Moodle to Better Support CS Education

Guido Rößling
Department of Computer Science
Technische Universität Darmstadt
D-64289 Darmstadt, Germany
roessling@acm.org

ABSTRACT

Many universities use a learning management system (LMS) or learning content management system (LCMS) to support the courses offered. One of the most popular LMSs today is Moodle (`http://www.moodle.org`).

However, as a general LMS, Moodle does not specifically support the requirements of CS Education.This Working Group proposal aims to address this need by researching how Moodle can be adapted to better support CS education.

1. BACKGROUND

Learning Management Systems (LMSs) aim to support the teaching and learning process. They are typically used to provide access to the learning materials, such as lecture slides, exercise sheets and solutions, and homework assignments. Additionally, they often incorporate elements to support interaction and cooperation, such as forums, wikis, or quizzes.

A large set of LMSs is currently available, many of which are commercial, such as WebCT or Blackboard. There are also a set of Open Source LMS, of which probably the most popular is *Moodle*. However, while essentially all existing LMSs can be used to support CS education, they are not specifically geared to support the teaching of CS. This especially concerns the special requirements and interests of CS educators, including:

- the ability to integrate dynamic visualizations and simulations of algorithms and data structures;
- the offering of programming homework assignments that ask students to submit code—optionally with an integrated automatic functional assessment;
- support for providing code examples for “copy and paste” into the students’ workspace;
- support for assigning students into groups or teams, for example for exercise groups in larger lectures or for lab teams;
- support for focussed discussion of learning or teaching materials, without requiring extensive references to the underlying text—ideally, discussing texts “in place”;
- making it easy to use source code in all entries, e.g., in forums or blogs, with appropriate syntax highlighting;
- providing the ability to submit an uploaded assignment to another server or process for grading. This includes running functional tests (e.g., using *JUnit*) or submitting VHDL code to a VHDL simulator to run the simulation and provide feedback on the results;
- support for specific subareas of CS, such as formal languages, modelling (e.g., UML), or simulations.

These and more features have also been described in a recent Working Group proposal to enhance LMSs to better support CS Education [**?**]. The previous report contained an overview of learning resource types, background about pedagogy, and technological guidelines. Additionally, it provided a set of (unrealized) example scenarios to motivate the use of such extended LMSs. In contrast, this Working Group will discuss additional goals that we will try to address with concrete recommendations and hints towards existing implementations. As such, this Working Group can be seen as a specialized continuation of the process started at ITiCSE 2008.

The large number of offered systems and possible extensions makes it difficult for a single educator to find the “best match” for the individual teaching situation. The goal of this Working Group is therefore to provide an overview of what makes CS education “special”, and how the requirements or wishes inferred from this can be met. We will provide concrete recommendations for modules to use, and also plan to provide additional modules specific for CS education.

It is beyond the scope of a Working Group to provide concrete guidance for the multitude of existing LMSs. Instead, we will focus on the LMS Moodle (`http://www.moodle.org`), which is probably the most popular LMs today. There are many reasons for this: Moodle is Open Source and thus does not bear any cost, it has a large development community and many additional modules, can easily be adapted to the target institution’s “look and feel” guidelines, is relatively easy to use for both students and teachers, and can be extended by additional modules as needed.

Example applications that already exist integrate algorithm visualizations in Moodle [3, 5], provide a study pack for CS1 [4], support a computer architecture course [1], or provide an extension for collaborative learning [2].

2. GOALS AND METHODOLOGY

The goals of this working group (WG) are threefold:

1. Explore in detail what features (most) CS educators may require, or most benefit from, from a LMS. Note that we cannot attempt to address all possible interests, as the complete field of CS is too wide.
2. Review the features provided by Moodle "as is", as well as by the large number of extensions (modules, blocks, filters, ...) available, for their use in CS education.
3. Recommend those modules available that can effectively support CS educators and students, and possibly provide or plan additional modules to address currently missing features.

A set of possible features that support CS education has already been listed in Section 1. Within the Working Group process, we will follow this up by researching relevant literature and discussing with other CS educators what features they use—and miss—in LMSs, and what their reasons for using—or not using—LMSs to support their teaching are.

In electronic communication before the working group convenes in Turkey, we will collect a variety of views from WG members and other educators on the questions above, and also encourage members to formulate other relevant questions that the group may want to address. We will also link our work to the Moodle community to benefit from their expertise.

Before ITiCSE 2010, a part of the Working Group will develop an overview of Moodle components and extensions. The other members will work on exploring the demands and wishes of CS educators for effective CS education support. During ITiCSE 2010, we will combine and refine these results, and develop recommendations for a "bundle" of components to support CS education. Hopefully, our report will become a "blueprint" for CS educators and Moodle administrators who wish to configure their system to better support CS education, and perhaps to consider switching to Moodle.

After ITiCSE 2010, we expect that many members of the working group will continues this process and develop "Moodle CS Teaching Bundles" which will then be made available to the public.

3. QUALIFICATIONS OF THE CHAIR

Guido Rößling received the Diploma in Computer Science from the Technische Universität Darmstadt, Germany, in 1996. From 1996 to 2001, he worked as a research assistant at the University of Siegen, Germany. He finished his Ph.D. thesis on AV system design in 2002. Since November 2001, he is researcher at the Technische Universität Darmstadt for e-learning applications and an educator for CS1 and other courses.

Since 1998, he has developed the extensible AV system Animal that is now also used in Naps' JHAVÉ system. He has published his research on e-learning applications since 2000. This includes several conference papers and journal articles on AV. He was a member of the program chair for the 2002 and 2004, and chair of the 2006 Program Visualization Workshop, held in conjunction with ITiCSE 2002, 2004, and 2006, respectively. One of this recent publications on extending Moodle to better support CS education has received a Best Paper award at the 2008 Program Visualization Workshop; an extended version also appeared in ACM Transactions on Computing Education [5].

4. POTENTIAL PARTICIPANTS

We have already contacted a small number of possible participants for their feedback to this proposal, and thus already have a few people who are likely to join us if their time schedule permits (and the Working Group is accepted).

In general, we look for participants who possess at least one of the following backgrounds:

- Experience in using Moodle as a (CS) educator;
- Experience in developing Moodle extensions, ideally for CS education support
- A good research background in CS Education and / or systems for supporting learning (LMSs, algorithm or program visualization, simulation, ...) that could or should be supported by Moodle.

5. REFERENCES

[1] Francisco Corbera, Eladio Gutiérrez, Julián Ramos, Sergio Romero, and María A. Trenas. Development of a new moodle module for a basic course on computer architecture. In *ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education*, pages 349–349, New York, NY, USA, 2008. ACM.

[2] Francesco Di Cerbo, Gabriella Dodero, and Giancarlo Succi. Extending moodle for collaborative learning. In *ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education*, pages 324–324, New York, NY, USA, 2008. ACM.

[3] Andrés Moreno. Program Animation Activities in Moodle. In *Proceedings of the* 13th *Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2008), Madrid, Spain*, page 361. ACM Press, New York, NY, 2008.

[4] Atanas Radenski. Digital cs1 study pack based on moodle and python. In *ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education*, pages 325–325, New York, NY, USA, 2008. ACM.

[5] Guido Rößling and Teena Vellaramkalayil. A Visualization-Based Computer Science Hypertextbook Prototype. *ACM Transactions on Computing Education*, 1(2):(accepted), June 2009.