Extending Moodle to Better Support Computing Education

Guido Rößling
roessling@acm.org

Andreas Kothe
andreas@oddbyte.de

Technische Universität Darmstadt
Dept. of Computer Science
64289 Darmstadt, Germany

ABSTRACT

Learning Management Systems such as Moodle are popular teaching tools with a broad spectrum of features. However, several aspects relevant for computing education are typically missing. This includes the annotation of materials, such as scripts or exercises, the inclusion of slides, and the integration of algorithm visualizations. We present Moodle activities that add these features to Moodle.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer & Information Science Education - *Computer Science Education*

General Terms

Management

Keywords

Moodle, activity, annotation, visualization, VizCoSH, CALMS

1. INTRODUCTION

Many Computer Science courses use a Learning Management System (LMS) to organize the course contents, support the students, and provide administrative and technical support for educators. One of the most popular and established Learning Management Systems today is *Moodle*, the *modular object-oriented distributed learning environment* [3].

Despite the advantages offered by Moodle and related systems, the existing LMSs could be far better suited to Computer Science education. They typically lack support for many aspects specific to Computer Science, such as the integration of simulators and visualizations, or the possibility to submit programming assignments and receive adequate automated feedback.

There have been at least two major efforts by international Computing educators to address this situation. In an ITiCSE 2006 Working Group report, the researchers proposed a Visualization-based Computer Science Hypertextbook (VizCoSH) [12]. Essentially, a VizCoSH provides the materials typically offered in a textbook as a hypertext, but enrichens them with static or dynamic visualizations. The report provides additional information on other aspects, such as the integration with an LMS to facilitate student profiles and adapting the contents to individual users.

An ITiCSE 2008 Working Group report proposed extending standard LMSs to better support Computer Science education in a so-called *CALMS* (Computing Augmented Learning Management System) [11]. In Section 2.1, we will take a closer look at the definition of a CALMS.

In this paper, we describe our efforts to bring these two concepts together. We have focused on three aspects that are important for our students: the ability to embed algorithm visualizations into the textual materials provided by Moodle, the ability to incorporate lecture slides, and the ability to annotate and discuss lecture materials.

The paper is structured as follows. Section 2 provides background on the underlying VizCoSH and CALMS concepts. Section 3 provides a brief overview of related work. In Section 4, we present an overview of our current prototype and discuss its strengths and weaknesses. Section 5 summarizes the paper and outlines areas of further development.

2. BACKGROUND

As stated in the introduction, our research builds upon the Moodle LMS, as this is one of the most refined and extensive Learning Management Systems available today. Moodle is Open Source and has been adopted by many universities as a campus LMS, as well as by research groups or individual educators. The popularity of the Moodle system is underscored by the large number of registered installations, as well as by the number of available extensions. At the time of writing, the Moodle web page listed more than 430 such extensions.

For the sake of simplicity, we will not distinguish between the Moodle terms *activities, blocks* and *modules*. Additionally, we will not go into details about the Moodle LMS in this paper, as this topic alone could fill more than one paper, and has already filled several books such as [3]. The reader is encouraged to take a look at the Moodle home page [4] and find out what Moodle can offer—especially since the features supported by Moodle are bound to have increased between the time we write this paper and when the paper appears in the proceedings.

ITiCSE'09, July 6–9, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-381-5/09/07 ...$5.00.

2.1 CALMS: A Computing Augmented LMS

Several aspects that are of interest to Computer Science education are missing even from extensive LMSs such as Moodle. According to an ITiCSE 2008 Working Group report, such missing features include the incorporation of programming exercises with automatic assessment and feedback, online exercises for abstract concepts such as data structures or formal languages, algorithm and program visualizations, interactive compiling and debugging systems, and program plagiarism detection support [11].

To separate the concepts needed for such an extended LMS that incorporates Computer Science education-specific support from a "standard" LMS such as Moodle, the Working Group referred to such an extended system as a *Computing Augmented Learning Management System* (CALMS). A survey conducted by the Working Group indicated that there seems to be "a growing need to integrate the activities fostered by the Computing-specific tools with the broader and more general capabilities of LMSs" [11, p. 144].

2.2 Visualization-based Computer Science Hypertextbooks

Another concept adopted from an ITiCSE Working Group report is the Visualization-based Computer Science Hypertextbook (VizCoSH) [12]. As mentioned in the introduction, a VizCoSH proposes merging the strengths of a textbook, such as having a clear structure, with the incorporation of dynamic visualizations of algorithms, data structures, or other dynamic systems. Additional concepts discussed by the report include the adaptation of content according to the user's exhibited level of knowledge, and the link to a Course Management System or LMS to better support the students and educators alike.

The report also discusses several scenarios how such a VizCoSH could be used to improve the teaching and learning in computer science. One example is to ask students to generate a visualization for a certain topic that they feel would be helpful for others when they try to understand the topic. This student contribution could then be peer-reviewed to ensure quality and provide feedback in a (relatively) risk-free environment, and foster communication skills for both peer reviewer and content creator. However, the report did not mention how such a peer review could be accomplished, and how the reviews would be linked to the actual visualization being reviewed.

In effect, a VizCoSH is a specialized instance of a CALMS, if limited in scope. As the chronology of the terms shows, the concept of a VizCoSH predates the definition of a CALMS by two years (2006 and 2008, respectively). Therefore, while the CALMS report mentions the integration of visualizations as one aspect, the VizCoSH report could not mention the more far-reaching concept of a CALMS. The close connection between these two concepts becomes less surprising when one notes that of the eleven and twelve authors of the two reports, six researchers were present in both Working Groups and thus made it easy to bring the ideas from the older VizCoSH report to the CALMS report. At the same time, it is interesting to see that these six authors come from universities in four European countries and the USA, illustrating that the ideas are not specific to a certain country or region.

3. GOALS AND RELATED WORK

We have decided to focus on three aspects that are of interest to Computer Science education, as well as to related fields. These areas are the ability to annotate and discuss lecture materials—slides on the one hand and textual materials, such as scripts or exercise texts, on the other hand—with other students, and the incorporation of visualizations and simulations of dynamic systems or data structures. All three aspects are also discussed in the CALMS report in the example scenarios "Integration with Visualization" and "Enriching Augmented Learning".

Annotations in the form of marginal notes have a long tradition in text-oriented scientific disciplines. They are commonly used to annotate a given text with notes, comments, additional explanations, cross-references to related or similar materials, and many other applications. However, traditional marginal notes also have disadvantages: they can often not be erased easily or cleanly, and are limited in length by the margin measurements. Additionally, the marginal notes placed in books or alongside slides cannot easily be shared with colleagues or searched for a given note, although both aspects can easily be offered for digital notes.

Many European Computer Science lectures focus on lecture slides rather than on textbooks, and slides copies are often available to students before a given lecture. Slide copies are usually provided in PDF format to allow easy printing, and prevent students from inadvertently changing the contents, as would be possible in "source" format, e.g., as a PowerPoint or OpenOffice.org Impress file. Our Moodle module should also support the import of such slides and be prepared to allow annotating these slides in the same manner as for the textual components.

Finally, the integration of dynamic visualizations or animations into Moodle is of great interest to us. This integration makes the learning materials much more dynamic and interesting, as students can directly attempt to apply their understanding of a given topic, and can verify if they have correctly understood the underlying concepts. Although the VizCoSH report that stresses the importance of such an integration is already two years old, the first prototypes for integrating dynamic visualizations with LMSs have only recently been reported [7, 13]. The annotation of such materials supports a limited (non-anonymous) peer review layer and may help students in their studies, as open questions about visualization contents can be addressed.

In the following, we will briefly outline the current state of art in related fields. Due to the restricted space, we can only discuss a few representatives of each category.

3.1 Support for Annotating Online Materials

Several current systems support the annotation of online materials. *Marginalia* [5] allows users to select areas and provide a marginal note, placed to the right of the text. Marginalia can be used by multiple users and can also indicate overlapping markings. A special version of Marginalia is available for Moodle, but not yet offered for the current Moodle release (1.9). Additionally, Marginalia modifies many core Moodle files, and thus makes updating Moodle difficult. While Marginalia works well with Mozilla Firefox, it does not work with either Opera or Internet Explorer, as the *W3C Range Object* used by Marginalia is currenty only supported by the Mozilla browser family.

The *Open Annotation and Tagging System* [2] also supports marginal notes, and accompanies them with annotations for collaborative tagging. Here, annotations do not take up space to the right of the text, but appear as a dynamic layer once a piece of marked text is activated. As OATS also uses the W3C Range Object, it is also limited to the Mozilla browser family. Additionally, it is not integrated into an LMS, but rather is a stand-alone servlet requiring a servlet engine such as Apache Tomcat.

Wizlite [6] uses JavaScript-based bookmarks which access the annotations stored on the Wizlite server. Wizlite also works with Opera and at least to a degree with Internet Explorer. When used on Internet Explorer and the annotated text contains HTML markup, this may affect the underlying HTML code and result in locally wrecked HTML code.

The *SlideAware* service at `http://www.slideaware.com` provided interesting features for annotating slides, such as adding text notes and other annotations. Alas, this service ended on May 15, 2008. A similar functionality is offered by the new *Apple iWork '09* suite, which allows sharing Apple iWork '09 documents. The documents can also be downloaded in Microsoft Office formats. However, it seems that one cannot share PowerPoint or OpenOffice.org Impress slides. Users who want to share materials require a cost-free *Apple ID.* While the current beta stage can be used without charge, the final service is likely to cost money.

The *eMargo* system [14] supports both the annotation of textual materials and slides in PowerPoint or OpenOffice.org format. However, there is no integration of this functionality into Moodle. Additionally, the code for the accompanying LMS *VOP* consists mostly of legacy code without a clean installer, and is not available as Open Source. Thus, getting a copy and installing it may prove difficult. The marking of text works on all common browsers, but has the drawback that all occurences of the given text in the selected paragraph are marked. If the user has only marked a single character or a short but common word such as *and*, this can lead to a cluttered display.

3.2 Support for Integrating Visualizations

As mentioned in Section 3, there are currently only two prototypes for integrating dynamic visualizations into an LMS—and both happen to use Moodle. Moreno [7] has provided a Moodle activity that allows running a *Jeliot 3* [8] program visualization from inside Moodle and also returns feedback about the learner's activities with Jeliot.

Rößling and Vellaramkalayil [13] have implemented a VizCoSH Moodle activity that is able to incorporate animations created in the Animal [10], *JAWAA 2* [1] and JHAVÉ [9] systems. Additionally, according to the authors, almost any system that can be started using Java Web Start can be integrated. For example, the authors also report a successful starting of Jeliot from the activity.

The VizCoSH activity allows adding a link to an animation, either using an image or a text, anywhere within the VizCoSH page. Thus, a page can contain as many links as the author wants. For example, a page describing a given sorting algorithm will typically show a screenshot illustrating an animation for a given input set, and provide links to sorting data that is in ascending, descending, alternating or "random" order. Additionally, a link to a generator allows the user to specify the data that he or she wants to use.

In contrast to the Jeliot 3 activity, the VizCoSH activity does not yet return feedback to Moodle.

4. OUR NEW MOODLE ACTIVITIES

Our new Moodle activities offer the following functionality: annotations as in *eMargo* for textual materials, integration of algorithm visualizations as in the VizCoSH activity [13], and support for lecture slides. We will now take a closer look at each of these features.

The *annotation* support follows the guidelines for improving augmented learning, as stated in the CALMS report [11]. The functionality is essentially the same as in the *eMargo* system. However, due to the structure of the legacy *eMargo* code, the Moodle activity is not simply a conversion, but a complete re-implementation. In the following, the materials to be annotated must be placed in a VizCoSH activity. They can represent any type of written document, such as scripts, exercise sheets, or student-authored materials. We have kept the name *eMargo* for the annotation facilities, as shown in Figure 1.

Users can use a text marker (using a different mouse cursor) to highlight text, and can see both their own and (after pressing a button) all users' markings. Each paragraph of the text can be discussed separately, as well as the complete document. This is shown in Figure 1 in the text box on the right. This text box floats as the user scrolls and thus always remains visible.

The annotation activity uses a part of the VizCoSH activity [13] that will be discussed in the next paragraphs. It is implemented using AJAX and the *Yahoo!* User Interface library available at `http://developer.yahoo.com/yui/`. As of Moodle 1.9.3, this library is also used in Moodle, and was therefore a natural choice for using AJAX in our activity.

As stated above, the *visualization integration* is the underlying activity for the annotation support. As this is essentially an update of the activity described in Section 3.2, it shares the characteristics described above regarding the supported algorithm visualization systems and the support for animation links [13].

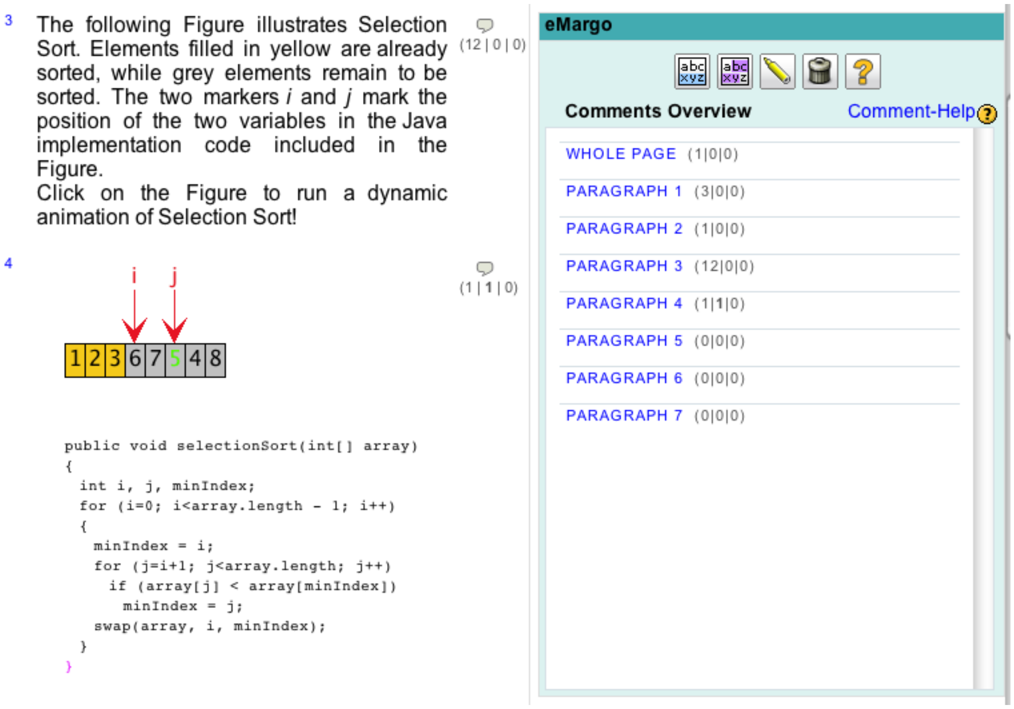
Two small icons shown at the top left of each page (not visible in Figure 1) offer the functionality to print a single chapter or the complete "book" [13]. The module was adapted to AJAX, and the chapter selection was moved from the left side (where it took up too much space) to a small drop-down box at the top. Two additional buttons can be used to step to the previous or next chapter.

Comments in the VizCoSH can be public or private. Private notes are only visible to the author, while public notes can be commented upon by other users and thus may spark a new discussion. Depending on which comment is replied to, a nested discussion is started, where each comment is placed next to the note it referred to, rather than simply at the end of the list as in *eMargo*.

Users can also set or remove a personal bookmark at an individual paragraph by clicking on a flag symbol. By clicking on a question mark symbol, the paragraph can be marked as hard to understand. The more users click on this question mark, the bigger it will become (in ten different degrees), and thus indicates how many users find the given paragraph hard to understand.

Text can be marked anywhere in the text, even across HTML tags. The problems with text marking in previous versions of the tool, and also described in the related work section, have mostly been resolved. Only Mozilla Firefox exhibits strange behavior if several list elements are marked.

Finally, users can also upload slides, either in the Open-



3 The following Figure illustrates Selection Sort. Elements filled in yellow are already sorted, while grey elements remain to be sorted. The two markers *i* and *j* mark the position of the two variables in the Java implementation code included in the Figure.
Click on the Figure to run a dynamic animation of Selection Sort!

(12 | 0 | 0)

4

(1 | 1 | 0)

```
public void selectionSort(int[] array)
{
  int i, j, minIndex;
  for (i=0; i<array.length - 1; i++)
  {
    minIndex = i;
    for (j=i+1; j<array.length; j++)
      if (array[j] < array[minIndex])
        minIndex = j;
    swap(array, i, minIndex);
  }
}
```

Comments Overview Comment-Help

WHOLE PAGE (1|0|0)
PARAGRAPH 1 (3|0|0)
PARAGRAPH 2 (1|0|0)
PARAGRAPH 3 (12|0|0)
PARAGRAPH 4 (1|1|0)
PARAGRAPH 5 (0|0|0)
PARAGRAPH 6 (0|0|0)
PARAGRAPH 7 (0|0|0)

Figure 1: Screenshot of the VizCoSH Moodle Activity showing numbered content paragraphs on the left and the eMargo-like annotation facilities on the right. The numbers inside the text box represent the number of comments, the number of unread comments, and the number of private comments of the current user, respectively. Clicking on the image in the screenshot runs an algorithm animation of Selection Sort.

Office.org XML format or in the Office Open XML format used by Microsoft Office 2007 and Office:mac 2008. To upload slides, a ZIP archive containing the presentation and a thumbnail of each slide in a subfolder *img/* is required. Once this file is uploaded, the contents are extracted and made available in the activity. As shown in Figure 2, each page contains a single slide, with a thumbnail preview of the complete presentation to the left. We also plan to extend this activity to support discussions about slides.

The Moodle activities were evaluated by several colleagues from the pedagogy department—from the same group that had developed *eMargo*, had won two “best teaching awards” for teaching with eMargo, and still uses the system on a daily basis. After evaluating the features of our system, they had a number of requests for extensions, most of which are already covered by our prototype. One element remaining on the wish list is an import facility for RTF text that recreates the structure of the text materials. The pedagogy group now considers using our Moodle activities for teaching in the upcoming summer term, instead of its own eMargo system.

5. SUMMARY AND OUTLOOK

We have discussed aspects for a Computing Augmented Learning Management System (CALMS), as presented in a recent Working Group report [11]. Based on the definitions in this report and in a previous report on a Visualization-Based Computer Science Hypertextbook [12], we have developed a set of Moodle activities.

The text-based activities support discussing text materials, adding private notes, bookmarking text for later review, marking text segments that contain relevant terms or definitions, and printing single chapters or the complete “book”. For lecture slides, the user can view one slide at a time and see a preview of the slides. At the moment, the annotation of slides has not been completed. Except for a bug caused by Mozilla Firefox, the text annotation facilities work with all browsers, and thus avoid the platform-specific problems common to related tools, as described in Section 3.1.

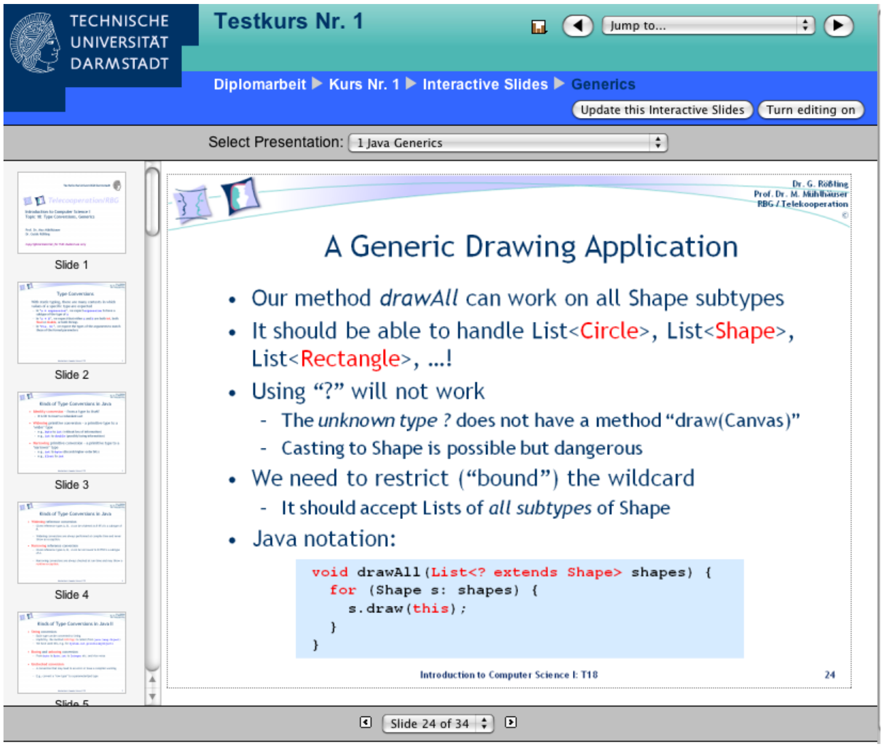
Algorithm or program visualizations can be added into a VizCoSH at any arbitrary position. As they are part of the paragraph structure of a chapter, they can be discussed and bookmarked as any other text element, and thus also allow a measure of (non-anonymous) peer reviewing.

We believe that these activities can significantly enhance learning when using Moodle. As they follow the Moodle programming guidelines, they are easy to extend or adapt to other purposes. They will be made available at the Moodle home page [4], once we have addressed some small issues, such as the discussion of slides.

6. REFERENCES

[1] Ayonike Akingbade, Thomas Finley, Diana Jackson, Pretesh Patel, and Susan H. Rodger. JAWAA: Easy Web-Based Animation from CS 0 to Advanced CS Courses. In *Proceedings of the 34th ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2003), Reno, Nevada*, pages 162–166. ACM Press, New York, 2003.

[2] Scott Bateman, Rosta Farzan, Peter Brusilovsky, and Gord McCalla. OATS: The Open Annotation and


TECHNISCHE UNIVERSITÄT DARMSTADT
Testkurs Nr. 1
Jump to...
Diplomarbeit ▶ Kurs Nr. 1 ▶ Interactive Slides ▶ Generics
Update this Interactive Slides
Turn editing on
Select Presentation: 1 Java Generics
Slide 1
Slide 2
Slide 3
Slide 4
Slide 5
Dr. G. Rößling
Prof. Dr. M. Mühlhäuser
RBG / Telekooperation
Slide 24 of 34

A Generic Drawing Application

- Our method *drawAll* can work on all Shape subtypes
- It should be able to handle List<Circle>, List<Shape>, List<Rectangle>, ...!
- Using "?" will not work
  - The *unknown type ?* does not have a method "draw(Canvas)"
  - Casting to Shape is possible but dangerous
- We need to restrict ("bound") the wildcard
  - It should accept Lists of *all subtypes* of Shape
- Java notation:

```
void drawAll(List<? extends Shape> shapes) {
  for (Shape s: shapes) {
    s.draw(this);
  }
}
```

Introduction to Computer Science I: T18 24

Figure 2: Slide presentation with chapter and page navigation drop-down boxes to the top and bottom, respectively. The slide thumbnails are shown on the left and can be clicked to jump to the associated slide.

Tagging System. In *Proc. 3rd Annual International Scientific Conference of the Learning Object Repository Research Network*, 2006.

[3] Jason Cole and Helen Foster. *Using Moodle: Teaching with the Popular Open Source Course Management System*. O'Reilly, 2007.

[4] Martin Dougiamas et al. Moodle - the modular object-oriented distributed learning environment. WWW: `http://www.moodle.org`, accessed March 21, 2009.

[5] Geof Glass. Marginalia web annotation. WWW: `http://www.geof.net/code/annotation`, accessed March 21, 2009.

[6] Alexander Kirk. Wizlite homepage. WWW: `http://www.wizlite.com`, accessed March 21, 2009.

[7] 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.

[8] Andrés Moreno, Niko Myller, Erkki Sutinen, and Mordechai Ben-Ari. Visualizing Programs with Jeliot 3. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2004), Gallipoli (Lecce), Italy*, pages 373–380. ACM Press, New York, May 2004.

[9] Thomas Naps. JHAVÉ – Addressing the Need to Support Algorithm Visualization with Tools for Active Engagement. *IEEE Computer Graphics and Applications*, 25(6):49–55, December 2005.

[10] Guido Rößling and Bernd Freisleben. ANIMAL: A System for Supporting Multiple Roles in Algorithm Animation. *Journal of Visual Languages and Computing*, 13(2):341–354, 2002.

[11] Guido Rößling, Lauri Malmi, Michael Clancy, Mike Joy, Andreas Kerren, Ari Korhonen, Andrés Moreno, Thomas Naps, Rainer Oechsle, Atanas Radenski, Rockford J. Ross, and J. Ángel Velázquez Iturbide. Enhancing Learning Management Systems to Better Suppoert Computer Science Education. *SIGCSE Bulletin inroads*, 40.4:142–166, December 2008.

[12] Guido Rößling, Thomas Naps, Mark S. Hall, Ville Karavirta, Andreas Kerren, Charles Leska, Andrés Moreno, Rainer Oechsle, Susan H. Rodger, Jaime Urquiza-Fuentes, and J. Ángel Velázquez-Iturbide. Merging Interactive Visualizations with Hypertextbooks and Course Management. *SIGCSE Bulletin inroads*, 38(4):166–181, December 2006.

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

[14] Werner Sesink, Sven Göller, Guido Rößling, and Daniel Hofmann. emargo: Eine Digitale Randspalte zum Selbststudium (nicht nur) der Informatik. In *Proceedings der Pre-Conference Workshops der 5. e-Learning Fachtagung Informatik (DeLFI 2007)*, pages 101–108. Logos Verlag Berlin, 2007.