

Collaborative Authoring, Use and Reuse of Learning Material in a Computer-integrated Classroom

Nelson Baloian, José A. Pino, Olivier Motelet

Universidad de Chile, Departamento de Ciencias de la Computacion, Blanco Encalada 2120,
Santiago, Chile
{nbaloian, jpino, omotelet}@dcc.uchile.cl

Abstract. Reuse of computer based learning material is a key requirement for any practical approach to support adaptive and dynamic teaching/learning inside and outside the classroom. Unlike hardware technology innovations, the software support for reuse inside the classroom is still not satisfactory. We present an approach to reuse material based in the Computer-integrated Classroom (CiC) environment and a special type of active documents used for containing learning material defined by the user to fit the particular needs of a lecture. This approach allows collaboration at two levels. The first one involves learning unit authors, who can jointly develop a course and teach it independently. The second level of collaboration refers to teacher and students exchanging and sharing documents. This framework does not impose a certain teaching/learning style. It rather supports the seamless transition from one teaching/learning mode to another one, using the same material.

1. Introduction

There has been much work done to support the creation and management of multimedia-based learning material and courses for e-learning (see [10] for a sampling). However, most of it is intended to be used only in a distance-learning scenario. Despite this, many current teaching/learning activities still take place inside the classroom. Also, the new tendencies are to redefine the role of the teacher as someone who catalyzes and moderates collaborative learning activities or someone who manages distributed activities and a variety of (learning) resources. We conceive a face-to-face lecture supported by computer technology as a succession of various teaching and learning activities like presentations by the teacher, individual or collaborative problem solving, discussion, etc., where computers and computer-based learning material play a key role enriching and supporting these activities [9]. Computer technology should support each activity with the proper tool and learning material. However, the costs of producing good quality computer-based learning material are frequently high and their range of use limited. For these reasons, many authors have pointed out the convenience of reusing already existing learning material for other teaching/learning contexts. The works of [14], [4] and initiatives like LOM [18] are directed to achieve this goal.

In this work we present a framework for supporting teachers in the collaborative authoring of a learning unit. On a second phase, it supports the flexible and collaborative use of this material in a Computer-integrated Classroom, considering the different backgrounds the students may have, preferred teaching/learning style of

teacher and students. By creating a learning unit we mean here selecting and sequencing some learning materials for a defined learning goal in such a way it can be adapted to a particular case of teaching/learning situation [8]. The structure of the learning unit syllabus plays a key role for achieving this goal [11].

The final users of this system are the teachers. It is designed to facilitate the management of the learning material (retrieval, distribution, share, and collection) supporting the seamless transition from one learning activity type to another. For example, the teacher may present a document previously stored in the repository of the lecture or create it on an electronic blackboard. Then, she can distribute it to the students and ask them to modify or complete it in an individual or group working session. Then, she can collect the results and present them on the e-board. Another teacher may prefer to collaboratively solve the problem with the whole class by sharing the same document. This framework allows to maintain the dynamics of the lecture by supporting all these actions with a minimum of required steps.

In a first part this article relates the framework to the existing approaches. Then it describes the tools that implement our ideas in the context of the CiC. Finally, it discusses their collaboration and reuse properties through the particular example of a current Java programming course at our university.

2. Related work

Numerous electronic supports for the teacher's task are studied in the field of distance e-learning. However our work goes out of the scope of distance learning since it deals with the face-to-face classroom situation. In this context the requirements are different and involve some visualization and usage issues.

Making a lesson for both distant and face-to-face electronic learning environments involves management, distribution, and collection of didactic activities. Syberwork [15], LearnITy [16] or WebCT [17] provide administration support for those tasks. Moreover, those tools enable the cooperative building of the lesson with asynchronous or synchronous facilities. LearnITy offers indeed the possibility of predefining some strategies that could distinguish the author's personality or adapt to different student levels or learning habits.

In a face-to-face context, we pretend to enable dynamic adaptation of the lessons structure depending on direct feedback from the students. In a distant learning context, building lessons structure and strategies is done before the actual learning period. Then, the student himself chooses the structure he thinks is best suited to his level. He could sometimes be assisted in his choice by an expert system computing the results of tests he previously fulfilled. By contrast, in an interactive lesson, the teacher should do this choice herself based on the analysis of the reactions and feedback from the students. The existing face-to-face learning environments like Hypercourseware [12], and E-Class [1] focus on supporting the teacher through distribution and collection of activities, on monitoring the lesson status or capturing the teacher live-contribution on the activities. Nevertheless, they do not aim to let the teacher dynamically adapt the lesson to the audience.

Adapting the lesson dynamically assumes the teacher has previously built a non deterministic graph of activities from which to extract the actual structure of the course. A teacher support should allow her to build, organize and use this graph during lesson delivery.

Concept maps [13] are web diagrams where ideas or concepts are visually related in an intuitively organized, non-deterministic structure. They present an interesting principle of organizing ideas in a visual and inductive manner. Each node maps one concept. However, in our case, we want to map activities that not necessarily match the scope of one specific concept. On the same visual principles, web cartographers like Nestor [7] and Webmap [6] allow to reference, position and link web documents for enabling easy and traced navigation. Those tools offer relevant visualization and collaborative building facilities but do not provide support for didactic activities.

3. Using Active Documents in a Computer-integrated Classroom

A Computer-integrated classroom (CiC) [3] defines a hardware environment and a specially designed software system trying to integrate computers into the everyday classroom rather than to use laboratories. The CiC system has been designed considering the room will generally be equipped with an electronic blackboard being operated by the teacher to present, create and/or manipulate learning material. Additionally, the teacher and some or all of the students have personal computers or PDAs, which are connected to the system via a network. A server acts as a repository for the learning material and administrative information. This system provides tools to enable teacher and students to retrieve, save, exchange, (asynchronously) distribute, and (synchronously) share documents of any type in a classroom setting.

3.1 Learning material as Freestyler Active Documents

Freestyler [9b] was incorporated to the CiC environment as a visual tool for generating, presenting and manipulating learning material. It manages and displays both visual objects and handwriting add-on in a layered manner. It benefits from the CiC support and enables synchronous collaborative work on shared documents.

A powerful feature of Freestyler is its usage of ad-hoc pluggable modules called *palettes* to define new functionalities. Each palette contains a domain-oriented visual object that can be placed in the documents. For instance, a palette for supporting the teaching and learning of programming with Java provides some specific objects: *java code*, *program input*, *program output*. By linking them, they become dependant. They accept also some specific actions like *compile*, *edit*, and *execute* (Fig 1).

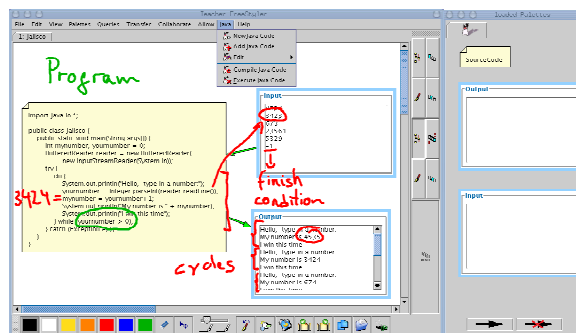


Figure 1. : A FreeStyler document using the Java Palette

3.2 Creating the Syllabus of a learning unit with Freestyler

As presented above, Freestyler is used to create, present and manipulate learning material. We extended it to define the structure and contents of a learning unit. This version is called *SurfStyler*.

McCalla [11] presents a number of self-adapting tutoring systems for supporting individual learners and he considers the graph as a key structure for the learning unit syllabus in order to achieve flexibility. This is because a graph allows more than one way to sequence concepts while maintaining a coherent order between predecessor and successor nodes. This structure is better than considering a syllabus as a sequential list of concepts to be taught. Baloiian et al. [2] present a graph structure for representing the syllabus of a learning unit with the aim of having several versions of a lecture according to different teaching styles, length and/or depth of the course. This was achieved by including a special palette and adding other functions to Freestyler.

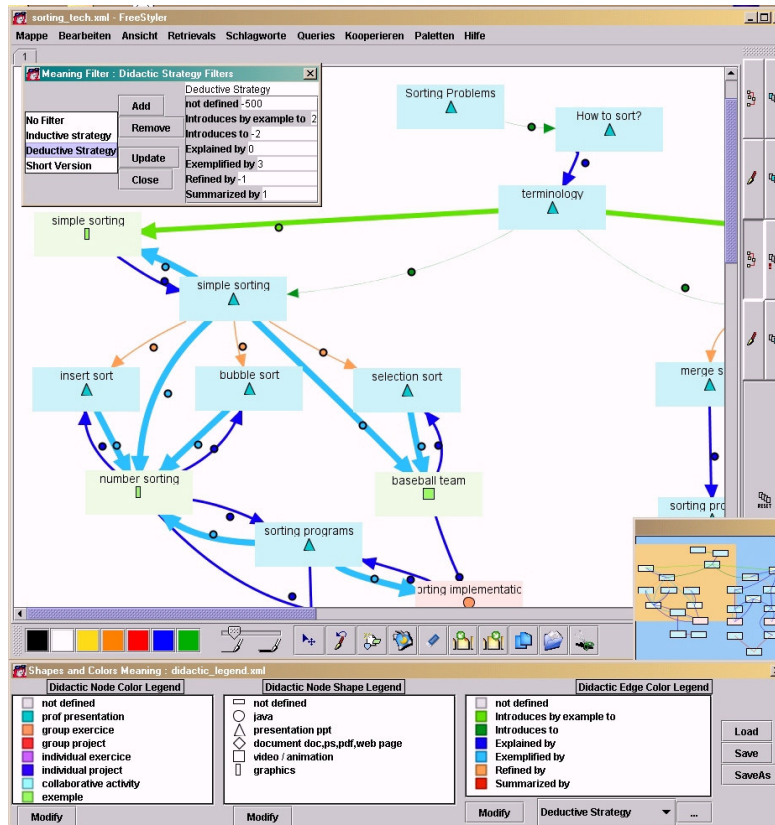


Figure 2: A lesson built with Surfstyler

Fig. 2 shows a lesson graph built with Surfstyler. Each node references one didactic material. Dragging the material in Surfstyler creates the node. Double clicking on the node opens this material. Because of their features, we use mainly FreeStyler active documents as learning material. However, Surfstyler accepts any type of material that can be opened by local applications installed on the operating system like Slide show, Web browser, Media player. Following Colazzo & Molinari's suggestion in order not to increase the students' disorientation while using educational software [5], the use of Surfstyler should be limited to the teacher. It allows the teacher to decide which materials to display on the electronic board and which documents or collaborative tasks to distribute to the students.

3.2 Legends in Surfstyler

Surfstyler provides a way to specify its contents to provide a consistent navigation and use of didactic material. First, each node is described with some keywords. Second, a *legend* is attached to the graph. Legends give meaning to colors and shapes of the nodes and colors of the arrows in a way similar to legends used in cartography. Each node has an array of 2-tuples that associates both colors and shapes with legends. Similarly, each edge has an array of 2-tuples that associates colors with legends. Therefore, when the user loads a legend, the graph takes the pre-specified appearance for this legend. The legends are XML files built with Surfstyler. The teacher herself can design and customize the legend for the specific purpose of the lesson. She can also share it and let it be improved by other teachers.

3.4 Didactic Legends and Didactic Networks

We designed a legend for didactic purpose, the Didactic Legend, aware of the fact that the different teachers' experiences and personalities will make it evolve. The Didactic Legend specifies the type of didactic activities with the color of the nodes and the type of documents with the shape. For the arrows, we implement the concept of Didactic Networks [2] and specify their color as different rhetoric relations between material types: *introduced by example to*, *introduces to*, *explained by*, *exemplified by*, *refined by*, *summarized by* (Fig. 2). This type of relations can help to define specific teaching strategies as *inductive learning*, *deductive learning*, or "*short version*" of the lesson. For instance, an inductive learning strategy means that examples and facts will induce the understanding of general rules and laws. This is why in such a learning mode the examples have higher priority than the explanations. It implies the links *explained by*, *refined by* will more likely be used instead of the *explained by* or *introduced by example to* links. We compute this notion by attributing different weights to the meaning given to the arrows. The weights are recorded as filters defined in the legend and applied to the arrow appearance.

4. Collaboration and reuse in the CiC Environment

Teachers can benefit from sharing the learning material and the syllabus of the learning unit, but they must be able to adapt it to their preferred teaching styles or

they can select appropriate examples or explanations during the lectures according to the students needs and feedback. Thus, the system does not impose a certain teaching/ learning style or activity. It rather gives teachers and students the opportunity to choose the learning activity they prefer to use at any time, enabling the use of the existing learning material, the creation of new one, and the support of a seamless transition between the different activities. We will explain how the system does it.

4.1 Collaboration and reuse during the authoring phase

In order to support collaboration during the authoring phase, teachers should be able to easily add, delete and/or modify elements without having to alter much of the original structure. The system should also support versioning and annotation as well as different roles like principal author, consumer, reviewer or coordinator (e.g., in the context of having a common learning material repository for the whole organization).

In our system, the syllabus of a lecture is created as a Surfstyler document using the didactic legend. It allows also the annotation of the changes being introduced by the different authors and the automatic versioning of a document. Surfstyler offers the granting and taking of read/write permissions, which allows the definition of various roles in this process. Reuse of learning material is strongly supported by the graph structure because the contents of a certain node (learning material) can be borrowed from the node of another learning unit. Asynchronous collaboration work during authoring is enabled by the CiC repository and the Surfstyler versioning feature. Synchronous collaboration is achieved because Surfstyler allows the co-editing of a single document in real time.

4.2 Collaborative use during a lesson

We want to enable teachers to decide when, how, and with which material they prefer to perform presentations, individual or collaborative problem solving sessions, discussions, etc. The system should support the teacher to perform these activities using the prepared learning material or developing new one during the lecture and especially, to allow seamless transitions between collaborative and non-collaborative activities using the same material. To support this we first tried to identify the most recurrent management activities performed by teachers with the learning material in a CiC environment and try to automate it. Figure 3 shows the teacher's tool for administrating the learning material. In the snapshot the teacher is selecting files and students for sending/sharing learning material. We can classify actions in two groups:

Distributing/gathering documents. During a classroom session the teacher can distribute and gather documents of any type to and from the students. Documents can be located in the Document Manager or in a local directory, either at the teacher's electronic blackboard or on the students' computers.

Sharing documents. By sharing a document we understand many people viewing and modifying the same document at the same time. This can be used when the teacher asks a student to present his work to the rest of the class. Allowing other students or the teacher to use the document being shown and adding notes and drawings can support useful contextual discussions.

5. Discussion

This paper presents Surfstyler, a tool for lesson-management in the CiC. It enables synchronous and asynchronous co-authoring, share and reuse of learning material, seamless distribution and collection of collaborative or non-collaborative learning activities and dynamic adaptation of the course based on direct evaluation of students' feedback.

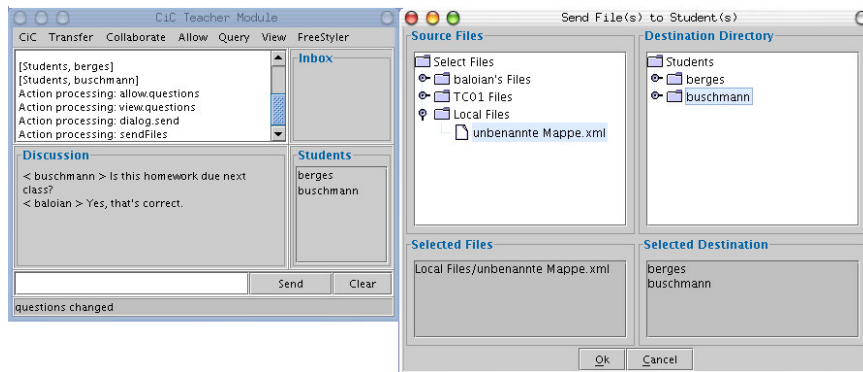


Figure 3. The teacher's CiC administration tool

CiC, Freestyler and SurfStyler are all written in Java and take benefits of its multi-platform portability. The data level is also portable in the sense that the lesson graphs built with Surfstyler, as any Freestyler documents, are stored as XML files. As we already discussed, a system of customizable legends enables the flexibility of Surfstyler. Those legends can be adapted to the application domain, the culture of the users or/and the specific needs of a lesson. They are independent modules stored in XML files and can be reused with any other lesson graph.

While most research in the fields of e-learning and computer-supported classrooms mainly focus on lesson scheduling and monitoring or lesson capturing, our approach deals mainly with the issue of supporting adaptable teaching in the classroom. At a practical level, we claim the adaptability of the course is a factor that depends on the granularity and diversity of the materials. The other key factor is the structuring of the material in a graph form.

The fact is that educational institutions will continue accumulating teaching material. Some of it will become obsolete, the rest will continue to be useful. A tool such as Surfstyler will allow to profit from this collection, since it makes possible to incorporate portions of it in several courses and in different ways.

We have used SurfStyler and CiC in a real case involving a course on Java programming, lectured in two occasions. Two teachers have used these new tools incorporating previous material they had. Before this experience, they had used a Web browser, MS Powerpoint, and Studio One (Java Programming Environment) with the same hardware. A total of 43 students have followed these courses and the new environment was used in a total of 10 sessions. The first informal evaluations are positive, showing added value to the tools use. We expect to have more results soon on usability and worth of the approach. Of course, further studies and experiments on

the visualization and navigation aspects in the graph of the lesson syllabus are definitely required.

References

1. Abwod, G. *Classroom2000: An Experiment with the Instrumentation of a Living Educational Environment*. IBM Systems Journal, Special issue on Pervasive Computing, vol. 38, Nr. 4, pp. 508-530, October 1999.
2. Baloian, N., Pino, J.A., Hoppe, H.U. *Intelligent Navigation Support for Lecturing in An Electronic Classroom*. In: Artificial Intelligence in Education, Lajoie and Vivet (Eds) IOS Press, 1999, Amsterdam, pp. 606-610.
3. Baloian, N. Berges, A. Buschmann, S. Gassner, K. Hardings, J. Hoppe, H.U. Luther, W. *Document Management in a Computer-Integrated Classroom*. In Haake & Pino (Eds): *Groupware: Design, Implementation and Use*. LNCS 2440, pp. 35-44, 2002.
4. Brown, A., Short, K. *On Components and Objects: Foundations of Component-Based Development*. Proceedings of the 5th International Symposium on Assessment of Software Tools. IEEE Computer Society Press. Jun. 1997.
5. Colazzo, L. & Molinari. *To see or not to see: tools for teaching with hypertext slides*. Proceedings of the ED-Media 95. Graz, Austria; June 17-21, 1995. Pp. 157-162.
6. Domel, P. (1994). *WebMap - A Graphical Hypertext Navigation Tool*. Proceedings of Second International WWW Conference, 1994
7. Eklund J, Sawers J & Zeiliger R *NESTOR Navigator: A tool for the collaborative construction of knowledge through constructive navigation*. In R. Debreceeny & A. Ellis (eds.) proceedings of *Ausweb99*, Southern Cross University Press, Lismore. 1999.
8. Halfff, H.M. *Curriculum and instruction in automated tutors*. In: Polson, M.C.; Richardson, J.J. (eds.). *Foundations of Intelligent Tutoring Systems*. Hillsdale NJ: Lawrence Erlbaum.
9. Hoppe, H.U.; Baloian, N.; Zhao, J.; *Computer support for teacher-centered classroom interaction*. Proceedings of the International Conf. on Computers in Education. Taipei (Taiwan), Dec. 1993. pp. 211-217.
- 9b. Hoppe, H.U., Gaßner, K.: *Integrating Collaborative Concept Mapping Tools with Group Memory and Retrieval Functions*. In: Stahl, G. (eds.): *Computer support for collaborative learning: Foundations for a CSCL Community*. Lawrence Erlbaum, Hillsdale, New Jersey, USA (2002) 716-725
10. Landon, B. *OnLine Educational Delivery Applications: A Web Tool for Comparative Analysis*. Standing Committee on Educational Technology; Center for Curriculum, Transfer and Technology. URL: <http://www.ctt.bc.ca/landonline/>
11. McCalla, G. *The search for adaptability, flexibility, and individualization: approaches to curriculum in intelligent tutoring systems*. In: Jones, M.; Winnie, P. (eds.). *Adaptive Learning Environments*. Springer-Verlag, NATO ASI Series. 1992 pp. 123-143.
12. Norman, K.: *Hypercourseware for assisting teachers in the interactive electronic classroom*. In: J. Willis, B. Robin, & D. A. Willis (eds.): *Proceedings of Fifth Annual Conference of the Society for Technology and Teacher Education*, Washington, D.C., USA, 1994 pp. 473-477
13. Novak, J. *Learning, Creating, and Using Knowledge: Concept Maps As Facilitative Tools in Schools and Corporations*. Lawrence Erlbaum Assoc. 1998.
14. Roschelle, J., DiGiano, C., Koutlis, M., Repenning, A., Jackiw, N., Suthers, D. (). *Developing Educational Software Components*. Computer, Vol.32, N° 9, 1999 pp. 50-58.
15. *Syberworks*: e-learning course authoring http://www.syberworks.com/product_SA.htm
16. *LearnITy™*, Aunwasha integrated suite, e-learning suite: <http://www.aunwasha.com>

17. *Webct*, e-learning suite: <http://www.webct.com>
18. IEEE P1484.12: *Learning Object Metadata Working Group*. Draft Standard for Learning Object Metadata. An unapproved draft of a proposed IEEE Standard, 2000. (<http://ltsc.ieee.org/wg12/>)