

# A Document-Centered Architecture for Classroom Collaboration

Jens HARDINGS(\*), Nelson BALOIAN(\*), H. Ulrich HOPPE(\*\*)

(\*) *DCC - Universidad de Chile* {jharding,nbaloian}@dcc.uchile.cl

(\*\*) *Gerhard-Mercator-Universität Duisburg* hoppe@informatik.uni-duisburg.de

## 1 Introduction

Computer technology in the classroom as a non-intrusive technology enriching the situation is not aimed to define new settings but promoting a change based on small steps: new roles, activities, material complement the traditional ones. In this paper, we present a framework system built to provide basic capabilities to teachers in order to improve their ability to manage information within and across classroom sessions which complement human abilities such as decision-making with the powerful potential of information management systems, being able to move issues from the background to the foreground and back again as described by Weiser in [1]. The fields of use can vary, as well as the pedagogical methodologies used, because we do not constrain our system to a particular view of how the sessions have to take place. Thus the CiC does not force a modification on the curricula or the way teaching is done within a classroom, but opens up new possibilities that can be exploited.

## 2 The System

A CiC defines both a hardware environment and a specially designed software system, trying to integrate computers into the everyday classroom rather than to using laboratories. The CiC software system [2] has been designed considering that the room will generally be equipped with an electronic blackboard being operated by the teacher to present and manipulate learning material as well as document management operations like distributing or collecting learning material, homework or assignments. Additionally, the teacher and some or all of the students have personal computers or PDAs which are connected to a (probably wireless) Local Area Network. The network is either connected to the outside in order to have access to a server providing multimedia learning material and administrative information (like student enrolled in courses) or the server has to be located on the local network. The system provides tools to enable teacher and students to retrieve, save, exchange, distribute, and share multimedia documents in a classroom setting. These actions should be carried out swiftly in order not to interrupt the normal flow of the lecture. For achieving this goal, we defined a set of general propose semantic actions for teachers and students who are likely to be performed recurrently, in order to implement them in a way it would require from students and especially the teacher the minimum amount of time and action. The actions can be classified into two types: **Document exchange**: Several of the common tasks within a classroom require the exchange

of documents. This is the case when assigning tasks to work on, delivering homeworks or distributing complementary learning material. These actions can be performed for documents of any format and an example can be seen on figure 1. **Document Sharing:** Synchronizing the use of a certain document by two or more users is achieved by coupling its content. This is only possible while working with special documents adapted for this environment.

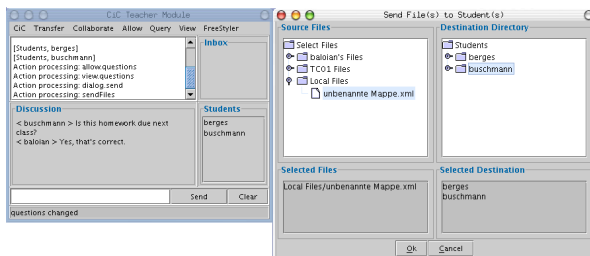


Figure 1: Classroom scenario: teacher sending a file to students

The documents used within the CiC can be of any type, but we have also integrated the FreeStyler [3] into the architecture as a primary tool to manage a special kind of documents since FreeStyler allows the manipulation of documents in a way more according to what we think should be the structure of the learning material in a Computer enhanced learning environment.

FreeStyler can be used during multiple working phases as preparation, creative meetings, presentation, postprocessing or wrapping up information. Therefore, it serves also as an interface between face-to-face discussions and the documentation process. The content type and functionalities of a FreeStyler document can be extended defining "palettes" with new node and link types for a specific learning domain (see 3). The documents may include a workflow description about how and when they should be distributed to the students, collected, where they should be stored, etc. This allows the system to offer useful default alternatives to the user, minimizing the input required to perform each task and keeping the focus on the session. FreeStyler documents are stored in XML format. The system provides an interface to query the XML documents, both in their current state while being edited and off-line in the repository. This way, the teacher can monitor the work of the students over the document as well as grade the work or analyze the result of past sessions. Another aspect of the system is that it should provide feedback information about the classroom sessions in order to allow reflection and keep track of the state of the course. The information gathered should include the file used (opened, modified, created), coupling sessions, file transfers among others.

### 3 Adding new functionalities through palettes: an example with Java

A very powerful feature of FreeStyler is the usage of ad-hoc pluggable modules called palettes [4] to define new functionalities. Each palette contains elements called nodes that can be placed in the documents, and also can contain different types of edges that connect different nodes. In this way, different models can be used for different learning subjects or complexity levels. As an example, a palette for supporting the teaching and learning of programming in Java was developed. The Java palette contains three types of different nodes and one link type. The node types are: **Java Code Node** which is a node which contains and shows a Java program, **Program Input Node** which is a node which implements the standard input for a Java program, **Program Output Node** which is a node which shows the standard output of a program. Figure 2 shows a FreeStyler document with the Java palette. The palette is shown at the right hand side of the screen and contains the 3 kind of nodes which can be added to the document by drag-and-drop operations. There is an icon for creating links and another one

for deleting them. The document of the example shows a Program node connected to both an Input and an output node.

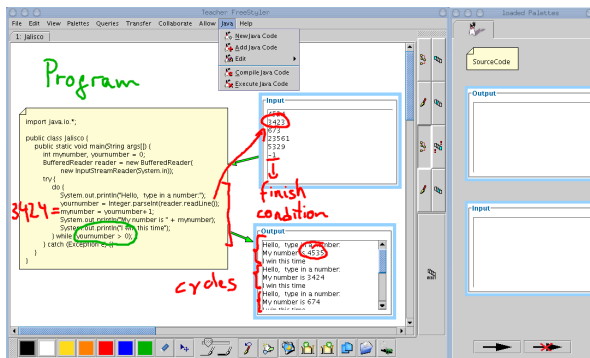


Figure 2: a FreeStyler document showing a Java example

as homework or assignment. By establishing a coupling session and synchronising the documents of a certain student with the one on the electronic board the teacher can enable a student to show her solution to a certain problem proposed earlier.

## 4 Conclusions

The framework presented here is designed to aid teachers in managing the documents to be used in classroom and related activities in a non-intrusive way, with a clear focus on face to face classroom work as opposed to distant learning scenarios. At the same time it enables processing of digital media to improve the ability and quality of data management that a teacher has to overcome nowadays. The focus is centered on internal course issues rather than a broader view over several courses, and the system is flexible enough so that it does not force any teaching/learning style but rather accomodates to the teacher's needs. It can therefore be used within an existing administrative system without much change, but as a complement that operates within the classroom, even accessing external resources as needed.

## References

- [1] Mark Weiser and John Seely Brown. The Coming Age of Calm Technology. In Peter J. Denning and Robert M. Metcalfe, editors, *Beyond Calculation - The Next Fifty Years of Computing*, pages 74–85. Copernicus - Springer Verlag, 1997.
- [2] Nelson Baloian, Alexander Berges, Stephan Buschmann, Katrin Gaßner, Jens Hardings, H. Ulrich Hoppe, and Wolfram Luther. Document management in a computer integrated classroom. In Joerg M. Haake and Jose A. Pino, editors, *Proceedings of CRIWG 2002, 8th International Workshop on Groupware*, LNCS, pages 35–46. Springer, September 2002.
- [3] H. U. Hoppe and K. Gaßner. Integrating collaborative concept mapping tools with group memory and retrieval functions. In Gerry Stahl, editor, *Proceedings of CSCL 2002*, pages 716–725. Lawrence Erlbaum Associates, January 2002.
- [4] Niels Pinkwart, H. Ulrich Hoppe, Lars Bollen, and Eva Fuhlrott. Group-oriented Modelling Tools with Heterogeneous Semantics. In Stefano A. Cerri, Guy Gouardères, and Fábio Paraguaçu, editors, *Intelligent tutoring systems*, volume 2363 of *Lecture Notes in Computer Science*, pages 21–30. Springer, June 2002.