A FRAMEWORK FOR INTERACTIVE COURSES AND VIRTUAL LABORATORIES

Laurent Balmelli, Serge Ayer, Yves Cheneval and Martin Vetterli
Laboratory for Audio-Visual Communications
Swiss Federal Institute of Technology
1015 Lausanne (Switzerland)
tel ++ 41 21 693 56 34 fax ++41 21 693 43 12

Abstract - Theory and experimentation are both complementary in sciences. Where the former builds up a formal framework, the latter helps humans to develop their intuition. Hard-printed books are great supports for theorems and formulas repositories, but stay desperately static when providing examples. This paper presents some directions towards a framework for digital publishing, distance learning and computer-aided teaching. Preliminary results were presented in [2]. The *Interactive Digital Signal Processing (DSP) Book* **is built in this framework and provides a concrete example to the reader. The book, reachable at** *http://lcavwww.epfl.ch/DSPBook*, **is used as a digital support for a course taught by our lab to a third year undergraduate class in Electrical Engineering.**

INTRODUCTION

Traditionally, both in teaching and research, theory and experimentation were the two complementary approaches to achieve progress. Currently, computer experimentation plays an increasingly important role. Therefore, in many areas of science and engineering, there is a threefold approach to problems where theory and experiments are complemented by computational procedures. The aim of this project is to explore how this view influences teaching both in class and on-line, as well as textbook writing and electronic publication. Note that our framework is well suited for the problem related in [1], and could be used to produce on-line results. We identified three related basic problems:

(i) **Distribution and Portability:** The distribution format and delivery mechanism have to cope with the platforms and supports (CD-ROM, network) diversity. The portability of the format facilitates its distribution and promotes its usage. Moreover, writing digital publications should require minimal knowledge of programming and edition. Thus, the format has to remain simple, or maybe provide automated features like automatic index generation, table of content, included search engine, and so on.

(ii) **Electronic Illustration:** Book illustrations are essential since it often help to develop intuition. Graphical representations and images are common illustrations of scientific books. A digital support can be exploited more efficiently if it is linked to a processing unit, thus, electronic illustrations should improve common illustrations by adding computational ability. For example, imagine an illustration that can be recomputed with different parameters. Moreover, the development of an electronic illustration should not be more time-consuming than preparing a nice set of electronic slides.

(iii) **Exercises, Projects and Laboratories:** Another important aspect of teaching are the exercises session and practical experimentation. Consequently, there is a need for an environment allowing to practice exercises interactively. An example would be a panel where one can construct with the mouse a block-diagram by mixing basic computational modules. This could be used in teaching to build up algorithms by wiring elementary building blocks. Once all modules wired, the student could see the output of his construction. This environment could be seen as a virtual laboratory. One could also imagine supervised exercises sessions, where the environment would help the user to pursue his goals. It would be also well suited for the achievement of projects.

APPROACH AND SOLUTIONS

To address problem (i), we need to solve portability issues and propose an approach to digital edition. First, We choose the Web as distribution media and Java as the programming language, for obvious portability reasons. We could have chosen Matlab for our particular brand of material, but it would not be suited for other areas (e.g. networking). Also, Matlab is a commercial product, and so it is not suited for on-line publishing. Regarding the edition of scientific texts: there are various ways to edit formulas and equations in HTML pages. A simple way is to use *latex* and an appropriate converter for HTML. The final edition can be easily carried out with an efficient interactive HTML editor.

For problems (ii) and (iii), we decided to develop a library of high-level Java classes allowing to easily build up electronic illustrations. Figure 1 shows an example of electronic illustration.

The library contains high-level GUI (Graphical User Interface) classes for displaying graphs, arrays, matrices and images. It also adds new data types, utilities and, since our example book is DSP-oriented, a set of classes dedicated to DSP. All these classes are used to model electronic illustrations with *Java applets.* This library can either be used through common programming or through an interface for visual programming. Often in the DSP community, Matlab is used to develop demos for teaching purpose. To facilitate the

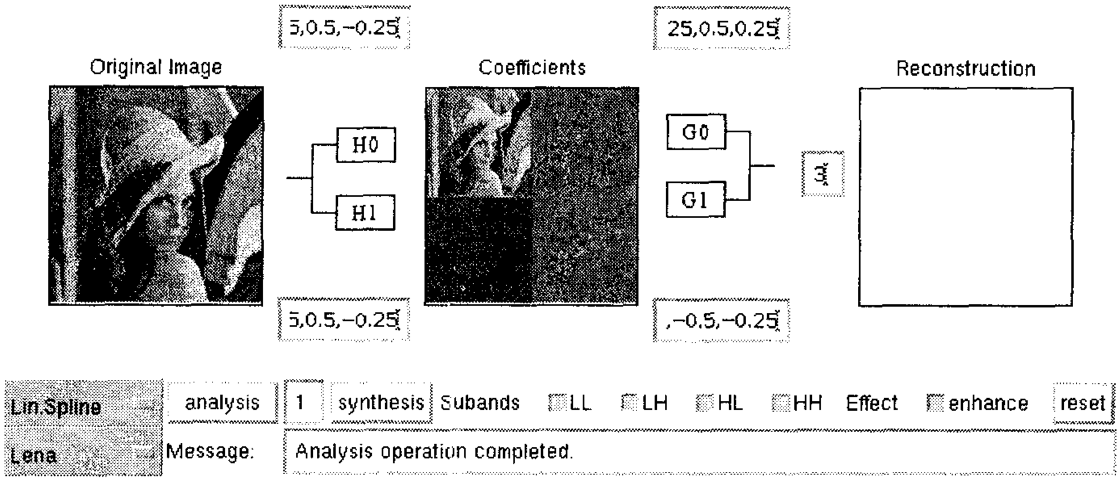

Figure 1: Example of an Electronic Illustration: Interactive Wavelet Coding of Images

utilization of the library, we built it so that its nomenclature and semantic is as close as possible to the one used in Matlab. This feature provides a mean to translate a m-file (Matlab function) to a Java applet. For this purpose, a special applet template is available with the library. The whole library content is exposed in Section .

The electronic illustration in Figure 1 is an example of what can be produced with the library. It shows an interactive wavelets coder for images. The user can choose an image and an initial set of filters (representing a complete filter bank configuration). He can also enter its own filters using the input fields and obtain a completely customized filter bank. The middle image shows the current wavelet coefficients of the decomposition tree. The synthesis (image reconstruction) can be performed only with part of the subbands, so that any of them can be visualized independently. The applet is available at *lcavwww.epfl.ch/DSPBook/Complements/wavelets/wavelet.html.*

Finally, to solve specifically problem (iii), we developed a graphical environment called *the Numerical Workbench.* This environment allows to construct interactively algorithms using building blocks. These ones are then linked together using the mouse and the user can visualize the output. He can also probe any subpart of the wired construction to see intermediate results. The Numerical Workbench is very appropriate for organizing virtual laboratories, allowing to set up exercises sessions in digital books, or in classrooms. Figure 4 shows an illustration of an interactive exercises session. The Numerical Workbench can be in fact also used as a visual programming front-end for the library. With its help, users can design interactively applets and then don't need any knowledge of the underlying language (Java) to produce electronic illustrations.

DISTANCE LEARNING FRAMEWORK

Introduction

In this section, we provide a global picture of our distance learning framework. This framework is adequate for digital publishing, virtual labs and interactive exercises sessions. It is composed of a components library (see Section), a delivery mechanism (simply the WWW), a solution for edition of hyper-linked scientific texts with interactive illustrations and a virtual environment for exercises sessions and projects. Figure 2 depicts our framework. The next sections describe its main parts, namely the components library and the Numerical Workbench.

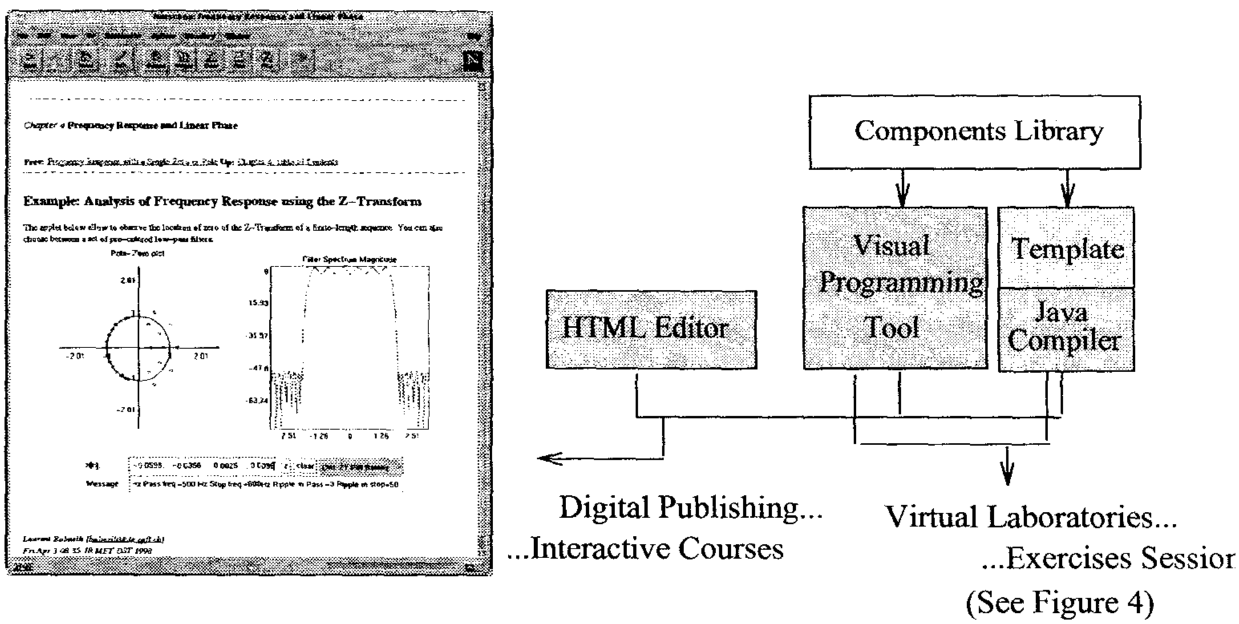

Figure 2: The Distance Learning Framework

Library

We first describe the main features of the components library and then present its actual contents. The library is organized as a hierarchical set of classes. One can distinguish three different groups of classes: the GUI classes, the types/utilities classes and the DSP classes.

A set of high-level GUI classes allows easily to build visual scientific applications. Any type of 1D, 2D functions, graphs, images can be displayed. Special visualizers for matrices are also included. The types/utilities group contains new data types for signals, complex numbers and polynomials, routines for conversions (for ex. images into matrices of values), matrix operations, quantizers and several numerical solvers. Finally, the DSP group contains classes for different operations such as FFT, Wavelet transform, DCT,

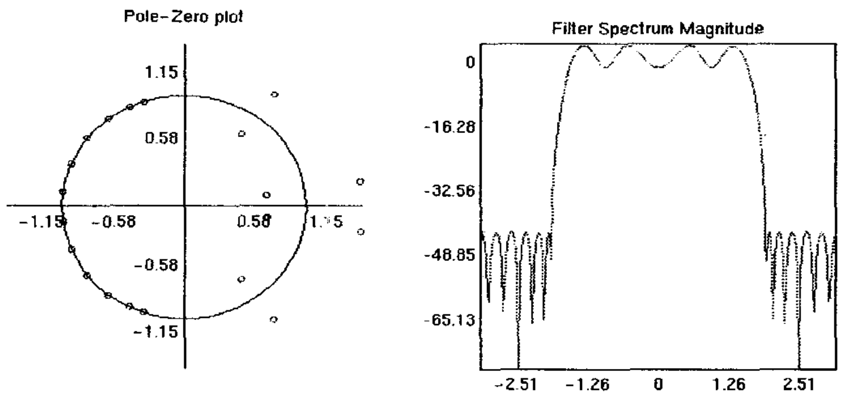

Figure 3: Example of High-Level DSP Classes: Pole-Zero Plot of a Low-Pass filter of Order 20

Quadrature Mirror Filters...and so on. Figure 3 shows the output of a simple code available at *lcavwww.epfl.ch/DSPBook/4/Applet/ZDemo.html*

A generic applet structure is proposed with the library to ease the development of a electronic illustration, especially when translated from a Matlab m-file. The model helps the user to organize efficiently his code (interface, processing, events management).

The Numerical Workbench

The Numerical Workbench (NW) can be used as a graphical front-end for the library. Every class in the library is available in the NW environment (through the JavaBeans [3] mechanism) and can be assembled with others. This interface allows users to build up interactively electronic illustrations using a visual programming paradigm. On the other hand, the Numerical Workbench is a very efficient tool to organize virtual laboratories. Students can build up algorithms by linking components and probe any part of the construction in order to visualize intermediate results. Figure 4 shows a session of the Numerical Workbench.

CONCLUSION AND FUTURE WORK

Our framework for digital publishing showed itself efficient to produce the DSP book. Obviously, the set of available classes is not as rich as Matlab's functions, however, an already fairly large set of teaching and distance learning applications in DSP can be implemented. The library is completely free and is available at our Web-site (*lcavwww.epfl.ch*), and we hope that our work will encourage other contributions. The Numerical Workbench should be improved by including the possibility of implementing exercise scenarios. One could think of a computer-assisted teaching within the environment, allowing

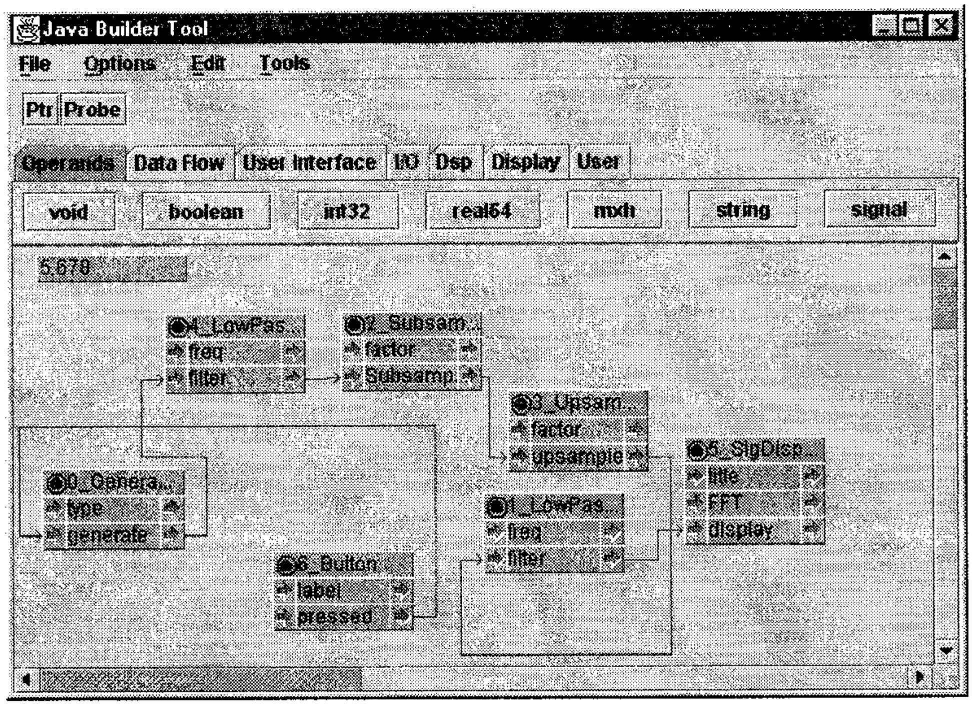

Figure 4: Example of an Interactive Exercise Session in the *Numerical Workbench.* The Numerical workbench can also be used as a visual programming tools.

the students to be guided to the solution.

References

[1] J. Buckheit and D. Donoho. Wavelab and reproducible research. *Wavelets and Statistics Lecture Notes*, 1995.

[2] Y. Cheneval, L. Balmelli, P.Prandoni, and M. Vetterli. Interactive dsp education using java. *Proceeding of ICASSP, invited paper*, 1998.

[3] R. Englander. Java beans programmer guide. *O'Reilly and Associates*, 1997.