AN INTERACTIVE DSP TUTORIAL ON THE WEB

Martti Rahkila and Matti Karjalainen

Helsinki University of Technology
Laboratory of Acoustics and Audio Signal Processing
Otakaari 5 A
FIN-02150 Espoo, Finland
Martti.Rahkila@hut.fi, Matti.Karjalainen@hut.fi

ABSTRACT

This paper describes an interactive, self-study DSP tutorial package built for the World Wide Web (WWW, Web). The package includes concept explanation, graphical presentations, listening-ready audio examples and interactive exercises and demonstrations of basic concepts of signal processing. The WWW is a convenient and easy-to-use hypermedia tool. From the point of view of DSP education, the main problem is how to implement interactivity and signal processing. Our solution is to apply CGI (Common Gateway Interface) and Java in tasks that they are best suited for. A Common Lisp based WWW server CL-HTTP together with the QuickSig environment is used for providing signal processing routines in a CGI-like manner. Java applets are used for taking care of the user interface at the client end.

1. BACKGROUND AND MOTIVATION

At the ICASSP '95 [1] we described a Computer Based Education (CBE) system that was built on top of the object-oriented, Common Lisp based, QuickSig DSP programming environment [2]. The CBE application "Introduction to Signal Processing", an interactive, self-study tutorial covering the basic concepts of signal processing, was also presented. The application was then used within acoustics course "Fundamentals of Acoustics 1" by approximately 100 students and the results were encouraging: nearly all the students felt that DSP concepts are easy to learn with CBE methods [3]. "Introduction to Signal Processing" was designed to be a stand-alone Macintosh application exploring multi- and hypermedia features of new computers. Still, audio and signal processing features stated some extra demands for the computer, so the usage of the application was restricted to only one computer located at the laboratory. Therefore, a need for multi-user capabilities arose. Natural tool for such development is the WWW.

The main goal for the application is to give undergraduate students a practical but yet general enough view on signal processing especially from the viewpoint of audio. The application has been integrated into one self-study multimedia package with concept explanation, graphical and audio presentations and interactive experiments. Special attention was paid in order to design an easy-to-use user interface. This project aims for a WWW version of the original CBE application sharing the same goals and without compromizing the flexibility and efficiency provided by the QuickSig system.

2. COURSE OVERVIEW

"Introduction to Signal Processing" is a CBE application that is designed to give 2nd-3rd year university students a practical and yet general enough view on signal processing especially from the point of view of audio and speech signals. No previous knowledge of DSP or electric circuits is required, but a certain familiarity with technology is needed in order to get the most out of the application. Some mathematics have been included, but the "inside theory" of signal processing is left for more specific courses. The approach is practical, emphasizing examples and exercises, which have been proven to be useful in undergraduate education, see for instance [4]. Also CBE principles and design methods (mainly the Lifländer design method [5]) were applied in the design phase of the application in order to guarantee the educational contents. The main difference with other similar projects [4] is that this CBE application is an integrated, self-study package without any accompanying books, lectures, lab assignments etc.

"Introduction to Signal Processing" covers the following areas:

- Signals: properties and representations (graphical,

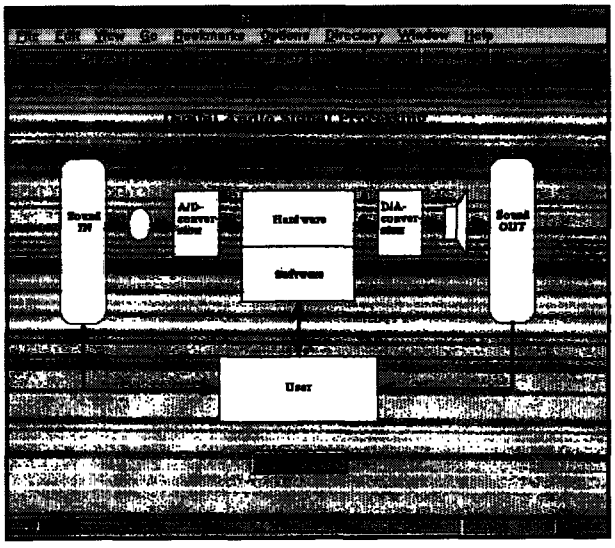

Figure 1: *The system model ("Home Page") for the DSP tutorial "Introduction to Signal Processing"*

numerical, function representations)
- Spectrum Analysis: spectrum and its properties, calculating spectrum and windowing
- A/D & D/A-conversion: sampling, quantizing and coding
- Filters: impulse response and transfer function, digital filters, basic filter types
- Signal Processing: general principles (linear/nonlinear, analog/digital etc.)

The key to this CBE application is the model illustrated in Fig. 1. It serves as a base for the application, being simultaneously a system model of (audio) DSP and the "Home Page" for the application. Each block is a link that leads to an independent session of that subtopic. Students can study these subtopics in any order they wish, but also a default path has been built that guides through all the topics.

3. WWW AND CBE

The WWW is without a doubt one of the most convenient hypermedia tools around: it is based on a client-server architecture and provides means of distibuted computing as well, it is thoroughly documented (*http://www.w3.org*) and it is simple to use. The Web is also a very natural tool for CBE purposes, because hypermedia is the key approach in modern CBE. From the CBE point of view, WWW offers many possibilities: mass education with freedom of time and place, the use of existing resources, individual teaching with authentication systems, ease of updating and so on. Perhaps the most important feature is interactivity, another key aspect of modern CBE.

Unfortunately, the HTML (Hyper Text Markup Language), language behind the web-pages does not include any active components, which makes the development of interactive systems rather difficult. Several work-arounds have been developed in order to overcome this problem. The most common ones today are Java (*http://java.sun.com/*) [6], JavaScript [7] (*http://home.netscape.com/eng/mozilla/Gold/handbook/javascript/*), CGI-scripting (Common Gateway Interface,*http://hoohoo.ncsa.uiuc.edu/cgi/*) and plugin technology. All of the above have their pros and cons. Java allows computing to be done in the browser end, but it is best suited for small tasks rather than heavy computing like signal processing. CGI scripts can be used for practically any operations, but the computing is restricted to the server end and the results must be sent over the Internet. JavaScript, the "little brother" of Java, provides some Java features plus others with an easy, HTML-oriented code. It is somewhat browser-oriented in a sense that for example the Netscape 3.0 browser provides enhancements of JavaScript usage with its LiveConnect architecture (*http://home.netscape.com/comprod/products/navigator/version_3.0/building_blocks/liveconnect/*). Java, on the other hand, is the same for every browser, as long as the browser supports it. For security reasons, neither Java nor CGI-scripts can utilize client-end filesystems or applications. The plugin modules can do that, but since they are platform- and browser-dependent, it would mean plenty of work for the developer to manage several environments.

The Web is, by default, a multiuser environment. This means that CBE developers must design and implement their applications in a multithreaded way. Multithreaded means that programs are divided into threads that operate independently, which guarantees smooth performance even when several students are studying simultaneously. A very interesting multiuser feature of the WWW is that the servers can record user actions in log files. Together with user authentication, this information can be used to provide individual teaching by taking previous actions into account. Also developers can benefit from this feature by performing analysis on the logs and using the results for evaluating and further developing the CBE applications.

4. WWW IMPLEMENTATION

Today, implementing interactive CBE with the WWW is a compromise of resource allocation and independency of environments. Our solution is to apply Java and CGI-scripts in tasks that they are best suited for.

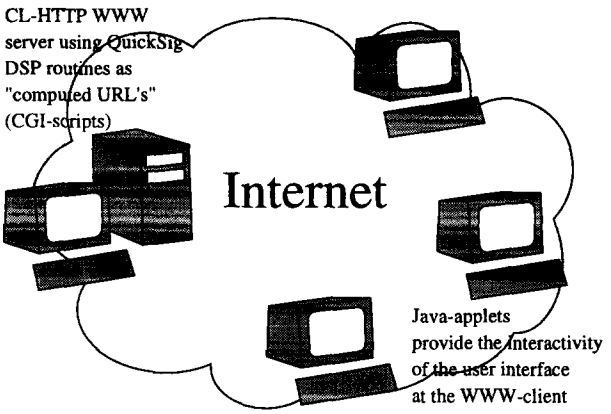

Figure 2: *The main structure of the Web-based DSP tutorial.*

That is, let Java take care of the browser-end user interface and CGI-scripts of the signal processing. This is possible with a Common Lisp based Web server CL-HTTP (*http://www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html*) allowing QuickSig signal processing routines to be applied as so called "computed URLs". Naturally this requires a lot of computing power at the server side. However, "Introduction to Signal Processing" includes only quite simple processing. Figure 2 illustrates the idea. The main benefit of this approach is that only a Java capable browser (and of course, necessary audio equipment) is required at the user end.

Figure 3 shows an example of the interactivity of the application. The student is presented with three magnitude spectra, one of the input signal, one of a filter and one of the filtered output. The task is to choose a filter from a popup menu and discover the changes at the output signal by studying the spectrum and listening to the output. The whole thing is a Java Applet communicating with the server that performs the filtering and calculates the spectra and returns them as audio and image files.

5. FUTURE

Several issues must be taken into consideration when planning DSP education with the Web. For instance, in audio signal processing, the amount of data can be large. Thus, if the processing is done at the server, the results (i.e., audio signals) must be transferred over the Internet and causes undesirably long response times for the user. This could be solved for example with "real-time" Internet audio tools ("streaming" audio or audio/video-conferencing tools) many of them already available as plugin modules for common browsers [8]. These tools provide efficient compression and play-while-

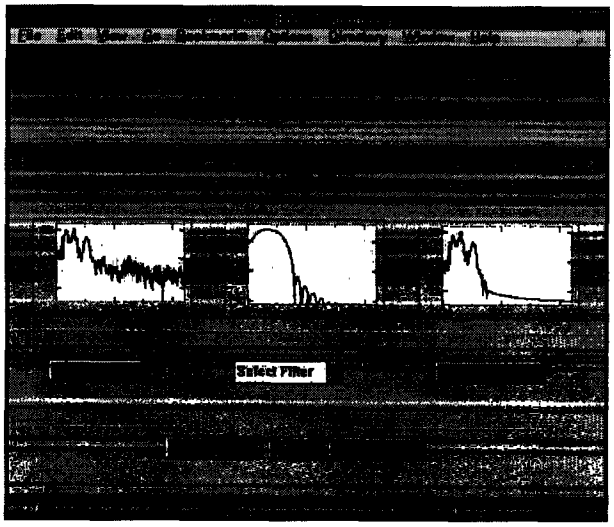

Figure 3: *An example of a simple, interactive excercise.*

loading capabilities.

Another interesting question is the computing power needed for DSP. There are too basic problems: 1) How much computing power is required and 2) how to allocate those resources. Modern computers can easily manage many signal processing tasks with adequately short response times but for certain time-critical applications like for instance in audio signal processing, special hardware may be needed. Of course, the fast development of computer technology will offer more computing power and advanced support for signal processing tasks as well.

The allocation of resources is a more difficult question. In our solution, the signal processing is concentrated on the server. This results as a need for a very powerful computer to be used as the server. Especially in a situation of many simultaneous users, the server load can become remarkably high. Basically, the server power can be increased by adding hardware or by using several servers in a shared manner. Another approach would be to use client-end processing. In principle, this approach is elegant because it saves server and network capacity and takes advantage of local system resources more efficiently. Currently, Java-based signal processing is very much out of the question because it is a slow and restricted language compared to CPU native code. Also, in a Web environment, neither Java nor Javascript can access local filesystems or other resources outside the scope of the browser due to security reasons.

One of the big questions in Web-based DSP education is how to build the user interface intelligently. The interface should be of generic nature and provide both

easy use and easy development of CBE applications. The standard graphical elements should be easily controllable but the interface should be capable of providing also complex DSP related functions like scaling of signal/spectrum views or selecting parts and playing of audio signals. A smart implementation of such an interface requires advanced WWW technology, hopefully to appear in the near future.

6. CONCLUSIONS

WWW provides interesting possibilities in signal processing education: a standard, easy to use hypermedia environment, multiuser capabilities, ease of updating and so on. Most of all, it provides interactivity, a key aspect in modern Computer Based Education. The possibilities, however, are also challengies for CBE developers.

This paper describes an interactive, self-study DSP tutorial package "Introduction to Signal Processing" that is based on the Web. It introduces the fundamental concepts of signal processing especially from the point of view of digital audio signal processing. The emphasis is on practical examples and excercises.

The WWW implementation is based on distributed processing between client and server. The user interface is built with Java applets running on the client browser, the signal processing takes place on a CL-HTTP Macintosh WWW server applying QuickSig signal processing routines. This approach demands only a Java-capable browser and audio equipment at the user end, but efficiency and computing power at the server end. At this point, "Introduction to Signal Processing" is a pilot project for us to experiment with Web-based DSP education. In the future, we are aiming for more general principles and systems.

The WWW is an extremely fast growing media under constant development. Major changes and improvements are likely to occur in the next few years. Therefore, further research of CBE on the Web has to be done in order to find longer lasting techniques and principles.

7. REFERENCES

[1] Karjalainen M., Rahkila M., Learning signal processing concepts and psychoacoustics in the QuickSig DSP environment. In Proc. 1995 IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP'95), Detroit, Michigan, USA, vol 2, pp. 1125-1128, May 9-12, 1995.

[2] Karjalainen M., DSP software integration by object-oriented programming: a case study of QuickSig. IEEE ASSP Magazine, pp. 21-31, April, 1990.

[3] Rahkila M., A Computer Based Education System for Signal Processing. Helsinki University of Technology, Department of Electrical Engineering, Master's Thesis, 47 p. 1996. *http://www.hut.fi/HUT/Acoustics/ publications.html*

[4] McClellan J.H., Schafer R.W., Schodorf J.B., Yoder M.A. Multi-media and World Wide Web resources for teaching DSP. In Proc. 1996 IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP '96), Atlanta, Georgia, USA, vol. 2, pp. 1101-1105, May 7-9, 1996.

[5] Lifländer V-P. A design method for a computer based learning environment. In ADCIS 30th conference proceedings, Philadelphia, Pennsylvania, USA, pp. 259-264, 1988.

[6] Lemay L., Perkins C.L., Webster T. 1996. Teach Yourself Java for Macintosh in 21 Days. Hayden Books. USA. 567 p.

[7] Manger J.J. 1996. Essential Java. McGraw-Hill. UK. 364 p.

[8] Rahkila M., Huopaniemi J. 1997. Internet Realtime Audio - Problems and Solutions. To be presented at AES 102th convention, Munich, Germany, March 22-25,1997.