An example of generating Internet-based course material

Igor Fischer, Simon Wiest, Andreas Zell
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen, Köstlinstr. 6, 72074 Tübingen, Germany
{fischer, wiest, zell}@informatik.uni-tuebingen.de

Abstract

With universities being at the same time research and educational institutions, one would expect that WWW, which started as an exchange medium for scientific documents, quickly finds its application in higher education. Although many different concepts for building Web-supported courses have been developed, none could establish itself as the dominating standard. We have developed a "lightweight" platform for Internet-supported education and implemented courses on neural networks and evolutinonary algorithms on it. The platform consists of a site-management software and corresponding content and configuration data. The content encompasses on-line textbooks, spiced with interactive components, and simulator.

Keywords

WWW; Education; Visualization; Neural networks; Evolutionary algorithms.

I. Introduction

Starting as a scientific document exchange medium, the World Wide Web has become today's most rapidly growing, biggest and most popular service on the Internet, and established itself as the *de facto* standard for on-line information interchange. Moreover, in perception of most people, it *is* the Internet. The reasons behind its surprising success are certainly numerous, but in our opinion the most important are:

- Simplicity: anybody could learn enough HTML in an afternoon to build his own home page.
- Affordability: free browsers, no expensive development tools needed and cheap access.
- Timing: as the WWW appeared, computers were already popular and widely used, in offices as well as at home.

Universities, being research as well as educational institutions, were among the first to actively use the WWW. From using it only as an exchange medium for scientific documents, over publishing general information like lecture timetables and application procedures, they now increasingly regard the Web also as an educational medium. Many different concepts for Web-based or Web-supported education are being developed and tested but, so far, no standard model has emerged. Although there is no shortage of corresponding software, none seems to be convincingly superior from a didactical, technical or marketing point of view.

For our part, at the Computer Science Department in Tübingen we have developed Web-supported courses on neural networks and evolutionary algorithms, among others. They use mostly common, open and freely available technology, instead of commercial proprietary tools. In a situation with no authoring tool being the de facto standard (and considering that Microsoft Internet Explorer and Netscape Communicator succeeded as client platforms mainly for their compatibility with open standards), we tried not to depend on a single vendor-specific technology. Although there are established tools for developing on-line courses, for example Macromedia Authorware, the content is encapsulated proprietarily and hard to access and edit outside the authoring environment. Another concern was availability for different platforms, especially for Linux, which is widely used at our University, but still often neglected by software companies.

The courses developed are of medium size (300 WWW pages of roughly double A4 size and 150 pages, resp.). Their contents follow closely the classical lectures on neural networks and on evolutionary algorithms held at our university.

II. Related work

Educational on-line services have been regarded as a special case of electronic document management systems. Thus intensive reseach is done on the exchange and availability of on-line information (BMBF, 1997) and existing electronic content managment systems are adapted to cater the needs of educators (Dietinger and Maurer, 1998). Large scale, database-driven learning environments are currently developed in serveral disciplines like chemistry (Schubert and Deplanque, 1999) or medicine (Uesbeck et al., 1999).

While these systems can offer complex features like versioning, personalized data views or support for colaborative authoring Kerres warns that technology driven projects tend to invest most efforts in the features of the learning environment, leaving only a small amount of resources available to content creation itself and proposes a low tech approach for educational media, using only open standards and field-tested tools (Kerres, 1998). As learning environments represent a very dynamic field, authors are looking for ways to make their laboriously created content future-proof. General markup languages like XML and new metadata standards for learning objects like LOM (IEEE, 2000) address this need.

III. The course topics

Artificial neural networks are computing models, that have been developed since the mid-fifties (Rosenblatt, 1958), following theoretical foundations laid already in the early forties. Inspired by their biological counterparts, artificial neural networks consist of a large number of highly interconnected processing units ("neurons"), whose connections resemble synapses in biological networks. The development of artificial neural networks has been motivated by some attractive properties observed in biological networks, most importantly the ability to learn from examples, ability to generalize knowledge, ability to extract prototypes from noisy data, high fault tolerance and inherently parallel processing. A disadvatage is that it is extremely hard to analyze the knowledge stored in neural networks. High interconnectivity results in high dimensionality of parameter space, and compounded with nonlinearities involved in each neuron, it makes mathematical analysis very difficult. This problem is further accentuated in education. Therefore, our efforts have been concentrated on visualizing neural networks.

Evolutionary algorithms (Holland, 1975) try to solve complex, multimodal problems using concepts borrowed from nature. Starting with a randomly chosen group ("population") of solution parameter sets ("individuals"), each set is scored based on its ability to solve a given problem ("fitness"). Ideally the specific parameters contributing to a good solution get combined by "crossing over" the components of several individuals or by mutation of good individuals to still better ones while bad ones may not reproduce. As time progresses the average fitness increases from generation to generation. Being a heuristic method, the algorithm depends on a number of user-defined strategy parameters that have to be chosen right to receive good results. It is therefore important for students to be able to test the impact of strategy parameters in several simulator experiments to grasp the concept of evolutionary algorithms.

IV. Educational concept

Our Web-supported courses enclose classical lectures, exercises on simulators for neural networks or evolutionary algorithms and an on-line book containing the necessary theory on the corresponding subject.

The lectures are held once a year and the main target audience are computer science and recently bioinformatics graduate students. The course on neural networks started out as a HTML version of the textbook “Simulation Neuronaler Netze” (Zell, 1994) that was made available to students for the first time in late 1998. Every HTML page in the on-line book corresponds to a section in the printed book and is, in most cases, not longer than two screen pages. All pages have a common layout, with navigation components on the top, left and bottom. Some pages contain interactive elements, implemented as Java applets, for visualization of neural networks. They are used instead or in addition to static images at places where interactivity is preferred. Internally, they consist of full-featured neural network simulation routines and a graphical user interface, and are intended to give inexperienced users a better insight into abstract processes, interdependencies and dynamics of neural networks. The book is especially intended for self-education and as a reference, but also for continuing education and professional training in industry. An example page is shown in figure 1.

For managing the book, which consists of some 300 HTML pages, we have developed a book handling tool, implemented in Perl. On the server side we run a user tracking process for evaluation purposes. A local search engine simplifies orientation in the book.

The exercises on the simulator are themselves also Internet-supported: the current exercise is available for download, and the students can use e-mail to communicate with the supervisor, if they do not prefer personal communication. In the exercises we mostly use the *SNNS* (Zell et al., 1994), (Zell, 1995), an industrial-strength neural network simulator. Furthermore we work on linking the book with the simulator, intending to use the book as an on-line help or reference for the simulator and, the other way round, the simulator as an environment for practical demonstration of theory from the book.

We follow the same concept in our course on evolutionary algorithms.

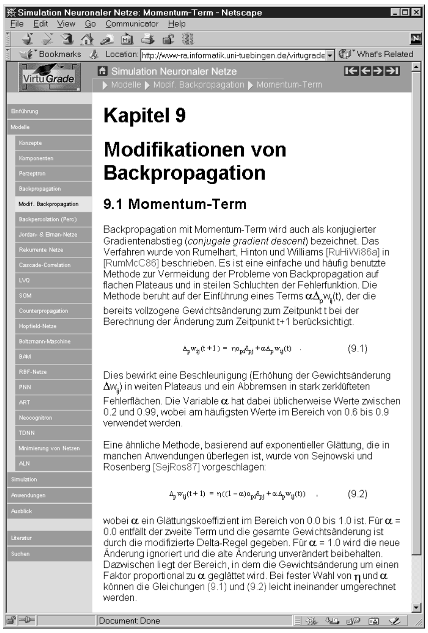

Fig. 1. A page from the on-line book.

V. TECHNICAL IMPLEMENTATION

The Web-based part of the course consists of static HTML pages, with exception of authentication and search engine generated pages. They are hosted on an Apache server, which has been configured for user tracking. The implementation of the course included following steps:

- converting the book into HTML
- linking the HTML pages into desired structure
- indexing the pages by the search engine
- developing interactive components
- implementing user authentication and tracking

The book master, from which the paper version is printed, was written in Adobe FrameMaker. Although FrameMaker contains a HTML export converter, the resulting pages are not satisfactory and have to be post-processed. Simply put, the converter tries to produce HTML pages that resemble the printed book as close as possible. One could say that it is more focused on appearance than on structure. We implemented the post-processor in Perl, which is very appealing for such tasks for its simplicity and and powerful regular expression processing capability.

```
<!--VARIABLES
  $FILTER *.html
  $COPY_FILES logo/*.jpg logo/*.gif
  $TEMPLATE TemplateStart.html
  $SOURCE source
  $DEST dest
  $PAGE a
-->

<H2>Inhalt</H2>
<DL>
  <DL>
    <DL>
      <DT><A HREF="index.html">Startseite</A>
      <DT TEMPLATE="Template.html"><A HREF="book.00.html">Vorwort</A>
      <DT><A HREF="book.01.html">Danksagung</A>
    </DL>
  </DL>
  <DT PAGE=1><H3>Einf&uuml;hrung</H3>
  <DL>
    <DT><H3>Einleitung und Motivation</H3>
    <DL>
      <DT TEMPLATE="TemplateChapter1.html"><A HREF="chapter1/book.1.html">Was sind Neuronale Netze?</A>
      <DT><A HREF="chapter1/book.2.html">Geschwindigkeitsvergleich</A>
      <DT><A HREF="chapter1/book.3.html">100-Schritt Regel</A>
      <DT END><A HREF="chapter1/book.7.html">Bemerkungen</A>
    </DL>
```

Fig. 2. An excerpt from a site description file.

```
<TD VALIGN="TOP" WIDTH=128 BGCOLOR="#99ccff">
  <FONT SIZE=-1><B><!--INDEX-->
  <!--L1 TYPE="H3.current"--><P><!--TITLE--><BR>
  <TABLE CELLSPACING=2 CELLPADDING=0 BORDER=0>
    <!--L2 TYPE="H3.current"-->
    <TR><TD VALIGN="TOP"><IMG src="logo/tag2.gif" WIDTH=8 HEIGHT=12></TD><TD><FONT COLOR="#003366" SIZE=-1
    ><!--TITLE--></TD></TR></TR>
    <!--/L2-->
    <!--L2 TYPE="H3.other"-->
    <TR><TD></TD><TD><SMALL><A HREF="<!--HREF-->"><!--TITLE--></A></TD></TR>
    <!--/L2-->
  </TABLE>
  <BR>
  <!--/L1-->
  <!--L1 TYPE="H3.other"--><A HREF="<!--HREF-->"><!--TITLE--></A><BR>
  <!--/L1-->
  <!--/INDEX-->
  <FONT SIZE=1><BR> </FONT>
  </B></FONT>
</TD>

<TD WIDTH=8> </TD>
<TD COLSPAN=3 VALIGN=TOP>
<!--BODY CONTENT-->
  <P> </P>
```

Fig. 3. An excerpt from a template file.

Another problem is that the display of formulae has not been satisfactory solved in HTML. For HTML export, FrameMaker converts them into graphics. However, when exporting graphics for HTML, FrameMaker generates images that, when printed at a resolution of 72 dpi, have the same absolute size as in the printed book. These images, although still legible, are relatively small and do not look very good on screen. To solve this problem we are waiting for the proposed MathML support of FrameMaker and appropriate browser plugins.

Finally, conversion of a FrameMaker document into HTML forces the author to be very consequent when writing the original document, since different items may look the same in FrameMaker, but have different appearance in HTML.

The HTML pages, as converted from FrameMaker and post-processed, still do not have the appropriate appearance and do not provide for comfortable navigation. We pass them through our simple site management tool, which links them in desired order and gives them a uniform look-and-feel. The tool takes two additional files as input: a site description file, which determines the logical structure of the site, and a template file, that controls the appearance of the pages. Both files are themselves written in extended HTML, what makes the tool particularly easy and attractive to use. The description file is structured similarly to Netscape bookmarks file, and the template file is an ordinary HTML page with some wildcards that get substituted during processing.

The resulting pages get a navigation bar for simple forward/reverse navigation on the top and a more complex navigation bar with book index on the left edge. For keyword searching through the book, we use ht://Dig, a freely available, simply configurable and, for purposes of a single site, sufficiently powerful search engine.

The most time consuming part was development of interactive components. They have been written in Java, which is not as suitable for multimedia applications as, for example, Macromedia Director, but has other significant advantages: as a powerful, object-oriented language, it still includes all features needed, from mathematics to graphics; it is available for all major operating systems and most users have it already installed, as soon as they install a web browser. We now have some 20 visualization applets and are considering adding some more. One of them is shown in figure 4.

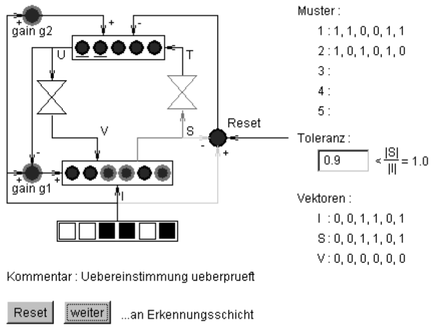

Fig. 4. Adaptive resonance theory applet.

Since our Web-supported course is a pilot version, we implemented a user tracking system, in order to find out how students use the on-line book. We also wanted to be able to analyze the behavior of each student individually. That meant that we could not simply track IP addresses, because students mostly use computers in our computer pool and are likely to use a different computer for each session. On the other hand, we did not want students to feel observed, so tracking the user name was ruled out. As a compromise, we have decided to let each student choose his or her ID code, with which he or she would be authorized to access the course. The students would then enter the code at the login-page, and the server would generate a cookie with that code. Finally, the server would log all page requests together with the user's cookie.

Evaluation of log files reveals that the students were particularly interested in the pages with interactive components, whereas they usually did not spend much time on static pages that were simply converted from the book.

One lesson we learned converting documents from a text processing package downstream to HTML was that this conversion process lacks a detailed, flexible control. The obvious remedy would be creating and editing course content directly in HTML. While this works fine with text and several multimedia formats like Flash, Shockwave, Quicktime etc. -even in WYSIWYG editors-, support for scientific notations (especially equations and chemical structures) is still in its infancy.

We therefore took a modified approach when we started creating new course material for the project Bioinform@tik which focuses on interactive learning modules for students in the field of Bioinformatics:

Using an iteratively refined document type definition (DTD) which catalogues the constituents of our course material semantically (e.g. section, quiz, example, proof) we began to markup new course content in XML using FrameMaker+SGML as WYSIWYG editor. This combines FrameMaker's ease of use concerning the handling of complex equations with the possibility of exporting the course material in an non-proprietary format. The XML output gets then batch converted into a desired target format like HTML by a Java application. Our DTD also allows us to assign additional educational, technical and legal metadata right to the document source and makes way for future features like versioning, distributed authoring, educational search engines or component reuse.

Two of our students' most requested features were smart update mechanisms for locally installed offline versions of our courses and higher readability of mathematical equations. Using globally unique identifiers assigned to all document elements we can now compare locally installed courses with the most recent version on our web server and generate incremental update packages. It also enables us to provide alternative versions of the same element, e.g. a low resolution, high resolution and MathML representation of an equation. While GIF images of equations constitute the best practice standard of mathematical elements embedded in HTML today, the course content itself is prepared for MathML-savvy browsers of the next generation.

VI. Discussion

Lack of de facto standard tools and concepts for producing Web-based training courses is one of the major problems in implementing a virtual university. We have presented our experiences from building Web-supported courses on neural networks and evolutionary algorithms. Except for the tools that we developed ourselves, we use only standard software, mostly free or open source.

Although technical implementation of a Web content may also have its pitfalls, the main problem is the content itself. As we have seen, it is not enough to simply convert a classical content into HTML. WWW is a specific medium with rules different from print media or TV. Producing web content that is perceived interesting by users can easily become time consuming and laborious task. That should not surprise us. After all, filming a theatre play will hardly result in a good movie, and reading newspapers in a TV studio is not likely to make a good news program.

As can be concluded from our course, interactive graphics and simulation suits the Web well, at least when used consciously. What kind and under exactly what circumstances, as well as which other media types are appropriate, is still subject to research.

VII. Acknowledgments

The site management tool is based on an initial concept by Uwe Oestermeier, and has been developed by Igor Fischer with contributions from Simon Wiest. The applet presented in figure 4 has been programmed by Fabian Hennecke, who, together with Igor Fischer, programmed most applets in the book. SNNS was developed by a team around Andreas Zell and is being maintained in Tübingen. The VirtuGrade project is funded by the state of Baden-Württemberg. The Bioinform@tik project is funded by the state of Baden-Württemberg and the Deutsche Telekom AG.

References

BMBF (1997). Neues Förderkonzept GLOBAL-INFO. Bundesministerium für Bildung, Wissenschaft Forschung und Technologie, Forschungsinfo 14/97, Bonn, Germany.

Dietinger, T. and Maurer, H. (Jun. 1998). Gentle general networked training and learning environment. In *Proc. ED-MEDIA '98*, Freiburg i. Br., Germany.

Holland, J. (1975). *Adaption in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor.

IEEE (2000). *Draft Standard for Learning Object Metadata*. IEEE Standards Department, Learning Technology Standards Committee, Piscataway. (IEEE P1484.12/D4.0).

Kerres, M. (1998). *Multimediale und telemediale Lernumgebungen : Konzeption und Entwicklung*. Oldenbourg, Munich.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**, 386–408.

Schubert, V. and Deplanque, R. (1999). The changing technology environment for scientific and technical information: A network for chemistry education. In *11th International Chemical Information Conference*, Annecy, France.

Uesbeck, M., Skalej, M., Kortmann, B., Strayle-Batra, M., and Wursthorn, B. (1999). New approaches for a world-wide usage and preparation of virtual patients. In *Proc. 20th Annual Conference of the Int. Society for Clinical Biostatitics*.

Zell, A. (1994). *Simulation Neuronaler Netze*. Addison-Wesley (Germany).

Zell, A. (1995). Snns user manual, version 4.1. Technical report, University of Stuttgart, IPVR. Report Nr. 6/95.

Zell, A., Mache, N., Hübner, R., Mamier, G., Vogt, M., Schmalzl, M., and Herrmann, K.-U. (1994). SNNS (Stuttgart Neural Network Simulator). In J. Skrzypek, editor, *Neural Network Simulation Environments*. Kluwer Academic Publishers.