POLITECNICO DI TORINO
Repository ISTITUZIONALE

An Alternative Approach to E-Teaching

Original
An Alternative Approach to E-Teaching / Risso, FULVIO GIOVANNI OTTAVIO. - STAMPA. - (2002), pp. 248-253. (Intervento presentato al convegno 5th IASTED International Conference on Computers and Advanced Technology in Education (CATE 02) tenutosi a Cancun, Mexico nel May 2002).

Availability:
This version is available at: 11583/1417042 since: 2020-12-14T10:15:02Z

Publisher:
Acta press

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2024

AN ALTERNATIVE APPROACH TO E-TEACHING

Fulvio Risso[1]
Dipartimento di Automatica e Informatica, Politecnico di Torino
Corso Duca degli Abruzzi, 24, 10129 Torino (Italy)

Abstract-- Documents prepared for e-learning projects are usually enjoyable only through CD-ROMs or the web. However several people prefer books or classroom-based learning. This paper presents *NetLibrary*, a project that developed a new approach in creating e-learning topics. Material is managed automatically (through the use of a custom tool, *WebLibrary*) in order to support several media such as Web, CD-ROM, books, and it includes slides that can be used by instructors in classroom. However the real advantage is the cheap, fast and simple content creation, which makes this approach suitable for fast changing topics.

Index Terms— Web-based education, fast content update, fast book creation, NetLibrary, WebLibrary.

I. Introduction

Although the entire Information Technology market is experimenting an unforeseen growth, the most impressive demand goes for network infrastructures, network applications, networked devices. Internet, the most popular networking infrastructure, has become widely available and most of us already have a full Internet connection 24 hours a day.

At present, one of the most urgent problems we have to solve is the lack of technical people able to properly operate with these technologies. From an educational perspective, this includes both improving the number (and the quality) of students in networking and helping the re-conversion of workers that stop previous job and start working on networks. In other words, there is a high pressure on the Universities to fill this knowledge gap; however the problem is even more serious because of the lack of specialized people in the University as well.

In order to mitigate this problem, a new project started at Politecnico di Torino that aims at becoming the key reference for teaching networking topics. Perhaps the most appropriate approach to present this project, called *NetLibrary*[2], is to say what NetLibrary is not. NetLibrary is not a virtual classroom like BlackBoard [4] and others. NetLibrary is also completely different from other systems such as VNET [3], which proposes a different paradigm to teach networking topics: the way of simulations. VNET includes an environment in which students can see what happens by launching simulations, but this requires that the user already knows what he wants to see. It follows that this approach cannot be used alone and it requires for example some form of classroom teaching before using the simulator.

NetLibrary looks like a set of online tutorials instead of an integrated learning system. Like with a book, a student using NetLibrary can start reading whatever topic (i.e. chapter) without any constraint; obviously a list of requirements is presented in order to inform the reader about the degree of knowledge required.

NetLibrary has two main objectives. First, it wants to provide support for traditional classroom teaching, which is the main way of teaching used in our University. Second, it wants to create learning material that can be used as unique source by external students (for example workers) that need to acquire networking knowledge or increase their skills. Both objectives must be obtained using the same set of material, since this is the only way to keep everything up to date (e.g. two distinct set of documentation will bring synchronization problems).

Although the creation of a new set of documentation that supports classroom teaching and offline students at the same time is not easy, we added other constraints. First, we want to be able to use several media, hence the documentation should be web-browsable, should be saved on CD-ROMs, and should be printed in order to create a traditional book. Second, our targeted users are mainly adults, so the *learn by example* method seems to be the most appropriate. For networking topics, this can be accomplished by showing real traffic, captured on real networks, that presents what really happens in practice. Third, documentation must be easily upgradeable to speed up the development process and to allow fast updates.

No other existing product has these same goals: tools exist that solve in an excellent way one or two of our requirements, but (at our knowledge) nothing exists that satisfies our objectives. Before starting our project we identified a list of requirements, our 'wish list'. Then, we identified a method that was able to satisfy the higher number of requirements. The result has been a simple procedure that allows the creation of documentation that is easily upgradeable, that requires limited authoring knowledge, and that is able to produce high quality output on different media.

---

[1] Contact information: Fulvio Risso, tel. +39 011 564.7008, fax +39 011 564.7099, e-mail risso@polito.it.

[2] The NetLibrary project at Politecnico di Torino has nothing to do with the NetLibrary company (i.e. the *library on the network*), available online at http://www.netlibrary.com/.

The following of the paper presents this project, the lessons learned from it, and is organized as follow. Section II shows the objectives that are behind NetLibrary, while Section III presents the technical choices and the final result of the project. Finally, Section IV discusses the obtained results and Section V gives some conclusive remarks.

II. The objectives

A. Choosing the appropriate media for the target audience

The main objective is to produce documentation that can be used for classroom teaching, by students studying at home, by workers that need to improve their knowledge because of personal interest or job needs. Slides are one of the most appropriate solutions for classroom teaching: they provide an outline for the speaker and help students to identify the key points. However slides are not appropriate for people who cannot attend the lessons. They need a documentation that is self-explanatory.

Moreover, if people are far from the University, we need a system that brings documentation to them. Among the most common alternatives we can see the web, CD-ROMs, or printed books. Although there are several tools that allow the creation of web-based (and CD-compatible) systems, we believe that workers do not like this media. First, they often study at home or during the day travel to their job, so a pure web-based approach is not appropriate[3]. Second, a traditional web-based reading is annoying: people tend to read short articles on the screen, but they prefer printed material in case of long readings or when the topic requires more concentration. Possibly, they want a learning method based on reading printed text and doing exercises.

The solution we adopted consisted of the usage of a text with slides inside; obviously the text can be printed in order to create a book. The text is self-explanatory and the slides can either contain figures that help to understand the subject or an outline that helps to identify the key points inside the text. Meanwhile, the teacher can use slides during classroom lessons. However, web or CD-ROM documentation is really handy as a reference and for fast searches. Therefore, the documentation should be able to reside on these media as well.

B. Fast creation, fast updating

The content creation process is now considerably faster than several years ago when the writer was working on the paper. Then everything moved to the typographer, then back to the author for corrections and so forth. However the creation of a book is still a quite complicated issue even if word processors speeded up the content publication. Moreover, the author cannot change anything after the book has been published until the next reprint.

These facts are among the reasons behind the success of the web: publishing web sites is very simple unless the author wants to produce high-quality graphics and multimedia. If we want to be able to create contents easily (and without third-parties intermediation) and we want to add material incrementally, we can use a web-compatible creation process. NetLibrary will thus be able to grow day after day by accepting new contributions as soon as the authors have enough time to add a new topic.

Moreover, a simple authoring process will allow fast content upgrade for the existing topics as well, and this makes NetLibrary constantly up-to-date from both technical and educational perspective. Technically, errors can be fixed immediately and new topics can be added easily. From the educational perspective (the most important one), topics can be reordered to make learning easier, rewritten if they appear too complex, examples can be added that better clarify the content. From the author's experience, it is really difficult for a teacher to understand precisely the points in which students will have difficulties. A really useful method is to evaluate the classroom response during the teaching process: a plethora of astonished faces means that the subject has not been understood properly. In this case, it is likely that the topic has not been explained properly; maybe more examples are needed or the order chosen to present topics is not appropriate. Fast updates means that the teacher can change the way topics are presented almost *on the fly*, tuning the documentation day by day in order to become even more effective.

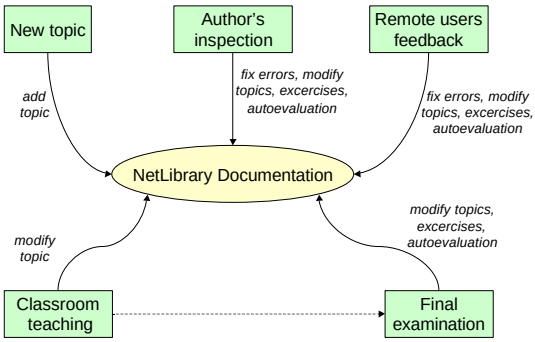

Figure 1. The update process.

For instance (Figure 1), the NetLibrary documentation is easily modified according to several inputs: the author's inspection, of course, but also the remote user's feedback, the classroom teaching feedback, and to the results of the students in their end-of-course exams.

As far as the content creation is concerned, the Author believes that the widely adopted method (i.e. defining a content expert, an instructional designer, a graphical designer), is not worthy from the costs/benefits point of views at least for highly specific technical documentation. Of course, this method allows the creation of high-quality

[3] Workers often tend to improve their knowledge without any input from their company, because they hope to be able to move to a new job or to improve their position.

documentation (particularly from the graphical point of view), but this requires time and a huge coordination effort. It is really common indeed that the creator of the documentation (starting from the storyboard) is an educational expert, which has not knowledge about the subject (especially if it is really specific), it does not know the most difficult points, and so on. He knows *how* to teach, but it does not know *what*. The result will be excellent only at the expense of a huge amount of feedback from the content expert, and it will be hardly upgradeable. The author believes that this method is not compatible with the goals behind NetLibrary, whose topics are highly specific and should be updated very often (networking is a really fast-changing world).

C. Standard and well-known tools

E-learning authoring environments (such as WebCT [6] and TopClass [5]) usually require the knowledge of advanced educational concepts. Unfortunately content experts are not aware (and they do not want to be aware) of them because they are not educational experts. Therefore content experts want to produce documentation according to the paradigms they are familiar with, i.e. papers, books and slides. For the same reason, content expert want to use known tools such as the ones already used for standard productivity (word processors, slide presenters, HTML editors). For example, a recent survey cited in [1] reports that 85% of the e-learning content is being created making use of standard tools such as Word and PowerPoint. For instance, the content authors want to create the material easily with the minimum effort and (possibly) the maximum result.

Standard tools are less powerful (e.g. you cannot create full-motion videos) but they are simpler, faster, and the resulting documentation can be easily integrated with other material not included in NetLibrary. It is quite common to use the 'cut and paste' method to create a new classroom course (addressed maybe to beginners) starting from existing documentation and using slides only.

For this reason several powerful authoring tools (like the Macromedia or the Click2Learn ones) are not appropriate. They are really powerful and support excellent multimedia authoring, but they require specialized people in order to produce anything and cannot be easily integrated with standard productivity tools. Moreover a recent study [2] shows that standard tools are cheaper, easier to learn, faster, and the result is usually equally effective compared to e-learning authoring environments and powerful authoring tools.

D. Embedded objects

Due to the recent advances in the multimedia technology, a standard output with text and graphics seems to be a bit poor. At least animation and sounds are now a must even in standard products; therefore NetLibrary must support them. Especially animations are of great help for showing complex phenomena that require a temporal evolution. For example, the trip of a mail message from a user to another can be rendered with a simple animation that shows graphically the various steps required for this message to be delivered. Of course, animations and sounds cannot be seen on a printed text; however they can be seen on screen.

Another useful feature is the use of real network traces that will show the network packets exactly as they travel on the network (for the *learn by examples* paradigm). For example, a student will take advantage from seeing a real IP packet exchanged by two stations when he is studying the IP protocol. This requires having a network sniffer that can export its captured packets somehow, and a way to include these traces into the documentation.

Finally, since a certain degree of interactivity is required, the chosen technology should support some kind of embedded code, for example under the form of scripting languages such as JavaScript.

III. The choices behind NetLibrary

This section describes the most important choices that are the fundaments of NetLibrary; it presents the technical topics, the methods used to create the documentation and the obtained result.

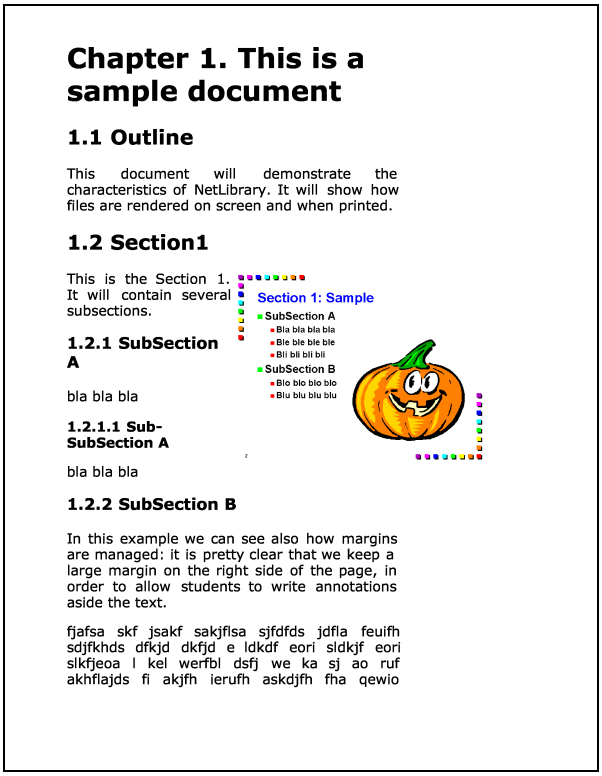

Figure 2. A printed sample page from NetLibrary.

A. Basic technology

From the layout designer standpoint, NetLibrary is completely web-based and it makes use of the widely accepted standards of the web. This assures that the

documentation can be easily put on a web site, stored on a CD, and it can even be printed with excellent results. In fact, the latest Cascading Style Sheet version 2 standard [8] allows different formatting on different media with high quality results. So, for instance, the page can use different fonts, margins, etc., when displayed or printed. In our case the text on screen uses the entire browser window and it maintains a clickable outline on the right side of the page. Vice versa, the printed text has larger margins (shown in Figure 2) in order to allow students to write annotations aside the text and it does not contain the index on the right.

Despite the huge amount of interest in XML, we choose HTML because of the more stable support in browsers and because NetLibrary does not need the advanced features of XML. NetLibrary needs just to render the text in different ways according to the selected media, but it does not associate any semantic to it, which is one of the key points of XML.

B. Production process

Since NetLibrary is made up of slides and text files, we use the well-known Microsoft PowerPoint 2000 for slides and an HTML editor for text. Users do not need advanced authoring knowledge: slides are created and modified in PowerPoint as usual, and animations and sounds can be added in the same way as before. Of course, animations are quite simple and they are not even comparable with other products (for example Macromedia Flash) that, on the other side, are targeted to expert users.

Using PowerPoint there is also another advantage: new presentations (that have nothing to do with NetLibrary) can borrow some slides from NetLibrary and, vice versa, existing presentations can be the foundation for a new topic. This makes the usage of the content of NetLibrary very flexible.

Microsoft PowerPoint can export the entire presentation in HTML, including animations and sounds, although it makes use of proprietary JavaScript extensions so that these advanced features can be seen only using Microsoft Internet Explorer. However PowerPoint is able to export all the slides as GIF images, and this is compatible with every browser. A custom macro, defined into the PowerPoint file, automates the export process.

The text file, edited with standard HTML tools (for example Microsoft FrontPage), is very simple and it makes use only of standard HTML tags: headings, titles, numbered and unnumbered lists, and more. In addiction, there is a special tag in the form `<SLIDE+n>` that the author inserts into the text in the place where the slide should appear. In other words, whenever the text recall for a new slide (for example a new section is starting), the author will insert the special tag, where `n` is the difference between the wanted slide (in the PowerPoint file) and the last inserted one. It follow that usually the tag is `<SLIDE+1>`; this makes easier the insertion of a new section (and a new slide) into an existing file because the following `<SLIDE+n>` tags do not require any modification thanks to their relative offsets.

At this time, the problem is the merging of the text file with the HTML output coming from the PowerPoint presentation. A custom tool, WebLibrary, has been created to overcome this problem. This simple tool is able to mix the HTML text file (interpreting the `<SLIDE+n>` tags) with the PowerPoint output and to create a new set of files that will appear such as in Figure 3. The index file (in the left frame) is created automatically, slides replace the `<SLIDE+n>` tags, and even the section numbering is done automatically be means of special commands included in the style sheet.

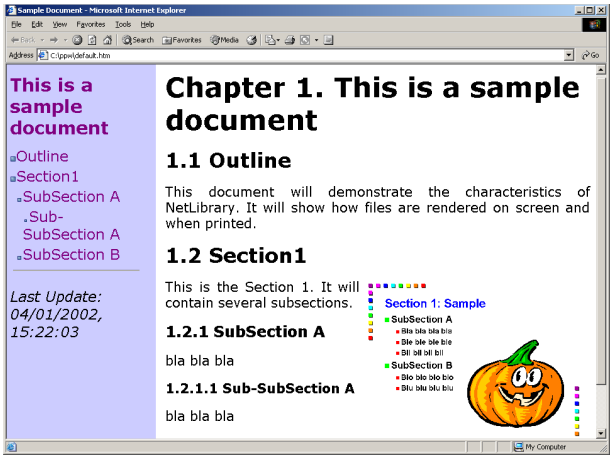

Figure 3. A sample page in the browser.

C. Advanced features

In order to create printable-friendly animations we carefully designed the slides in a way that the components of the animation are not overlapped. Since an animated slide, when printed, contains all the components present in the slide, everything looks well even when printed. Furthermore a special image has been inserted on the top-right corner of the slide so that the readers will recognize immediately which slides contain animations. In this case, readers can switch temporarily on the online (or CD-ROM) version of NetLibrary. Obviously, animated slides have been inserted only where really needed and we avoided animations in which the text appears step-by-step.

A similar trade-off existed for network traces: these have been obtained by modifying a freeware network sniffer, Analyzer [7], developed by our group. Analyzer is now able to export dynamic HTML files (i.e. the packet can be dynamically explored in order to visualize all the contained protocols), which can be seen only through a browser. A special text in the printed file will warn the reader about the existence of some captured packets, so that he can switch temporarily to the electronic version.

Finally, also sounds are active only on screen; however NetLibrary does not make use of sounds.

D. Final output

The final output can be seen in Figure 2 (printed), Figure 3 (online), Figure 4 (slide) and Figure 5 (captured trace), in which the font size has been enlarged in order to make the text visible.

While the classroom teaching can make use either of the original PowerPoint file or the generated HTML files, the online and the CD-ROM version appear like in Figure 3. This view uses standard HTML, thus it can be seen by any browser. The text file has been processed, a hypertext index has been inserted on a left frame, and the appropriate style sheets have been added.

The slide shown in the main window (Figure 3) is not animated; however the user can click on the image and a new window will appear that contains only the selected slide (Figure 4). Since this new window originates directly from the PowerPoint output, it makes active all the embedded objects including animations and sounds. For instance, this window is dynamically resizable (and the size of the content changes accordingly) and it is able to show the image at full screen as well. Unfortunately, this window appears correctly only in the Microsoft Internet Explorer browser.

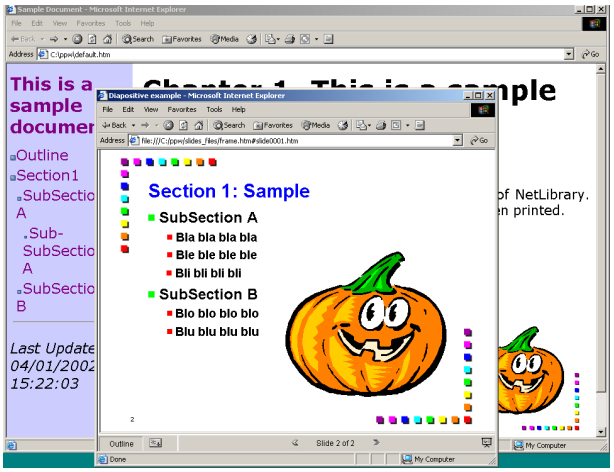

Figure 4. The sample page on the background and the window containing the slide on the foreground.

Network traces are linked in the text file, and a hyperlink is inserted in the HTML final product. Selecting this hyperlink, a new window will appear that looks like in Figure 5. The content of this window is similar to the output generated by a network analyzer (the index of the captured packets on the top frame, the details of the currently selected packet on the left, the hexadecimal packet dump on the right), and it contains dynamic code in order to explore the packet (left frame) field by field. Unfortunately, this window cannot be printed due to its large size (it does not fit into a sheet size) and its dynamic content.

The printed version, shown in Figure 2, is maybe the most interesting output. The quality of the layout is excellent and it can compete with printed books. Basically, the printed version corresponds to the right frame of Figure 3 (the text file with static slide images), with a different layout. Margins have been modified in order to leave enough blank space on the right, some layout parameters have been changed and the index file on the left has been taken out.

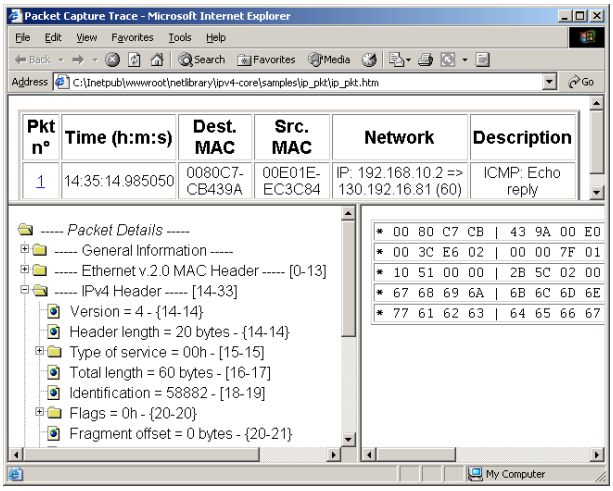

Figure 5. Example of a packet in the browser.

Moreover, the WebLibrary tool is able to collate several text files into a single document. So, for example, the online version will have several files containing each a single topic (IP, TCP, and so on). A book, obviously, must contain a set of chapters: WebLibrary allows the creation of a single file, ready for printing, which includes a given set of text files. This is useful especially for the automatic numbering of sections, which will be done automatically by the style sheet if all the content reside in a single file. The separation between different topics is done automatically by the style sheet, which inserts a page break when a new topic begins.

IV. Discussion

A. Technical problems

Although HTML can be considered a stable technology, there are several differences among the browsers used for display data. For instance, the best browser is Opera 6.0 [9] because of its (almost) complete support for the CSS-2 standard and for its excellent printing capabilities. Netscape 6.0 [10], Mozilla 0.9 [11] and Internet Explorer 6.0 [12] are far behind Opera: automatic numbering does not work, and especially the printing process is annoying. Some browsers are not able to wrap text around the images, others are not able to detect if an image fits in the end of a page (so the image will be printed half on a page, half on the next), others even are not able to detect if a single line of text fits in the end of a page (so even a line of text is printed half on a page ad half on another).

Although Opera is the best browser from a technical point of view, it cannot be used to display animations due

to the proprietary JavaScript used by PowerPoint. Summarizing, the best choice will be Internet Explorer on screen (automatic numbering is not important there) and Opera for printing. In any case, NetLibrary includes also a PDF file (obtained by printing the text file with Opera) in order to avoid the printing process to the readers.

B. Supporting different media devices

Perhaps the most critical point using NetLibrary is that it does not exploit all the possibilities granted by each media. For instance, animations are quite poor compared to other products, the printed version looks like a book but it cannot have the final quality of a traditional book, and so on. In other words, NetLibrary (i.e. every documentation that is created using WebLibrary) supports several media but none of them is exploited in depth.

This is the fee to be paid for the multi-target support. A parallel can be found in software programs that support multiple operating systems: they run everywhere, but they can make limited use of platform-specific features (for example searches on the online and CD-Rom version).

While a small post-processing phase is able to transform the printed version of NetLibrary in a book (for example inserting the page numbering and the table of content with page references), the online version seems to be more limited. Interactivity, for instance, can be used only at the expense of platform compatibility.

WebLibrary supports animations and sounds as well as other interactive features (for example tests and exercises with interactive answer) although they cannot be printed. However NetLibrary makes a limited use of these features because the most important media is the printed version. In our experience, readers and students prefer the printed version even in presence of advanced multimedia capabilities. Thus, the addiction of further advanced interactivity features into NetLibrary does not make sense. First, interactivity requires efforts that content experts alone cannot afford; moreover the NetLibrary project does not want external professionals like instructional designers and graphic experts. Second, interactivity is not important if readers are mostly using a non-interactive media.

V. Conclusions

NetLibrary is not the final answer to the problem of teaching technical topics[4]. However, it proposes some novelties: a very short production cycle because only the content expert is involved, the usage of simple and well-known tools, a fast-update capability, a multi-media support, a limited (but sufficient to our objectives) support for interactive components. This has been obtained by means a new method for fast document creation/update, by creating the custom program (WebLibrary) which merges slides and text file, by adding the “export as HTML” to our network analyzer, and by defining the appropriate HTML style sheets.

The most important limitation is the poorness of interactive objects (although supported by WebLibrary); however, since users prefer the printed version, the advantage of interactive components vanishes. Moreover, advanced graphical capabilities are expensive to obtain and do not match with the NetLibrary objectives.

The overall result is interesting: the NetLibrary project has been online since several months, updates are done quite often and feedback from users is enthusiastic, particularly for the printed version. NetLibrary demonstrates that this simple project can be used as a model to create high quality documentation.

NetLibrary topics have been engineered keeping in mind the most recent advances in the educational science. Thus, every document has a common look and feel, the subjects are presented in the same order (requirements, objectives, document structure, abstract, historical perspective, and so on) and a set of exercises and tests will allow the student to monitoring its progress. However, this is not a novelty in the educational science; therefore we do not want to show these characteristics in this paper.

The NetLibrary project is completely free. The online version is freely browseable and downloadable and slides are publicly available in both the PowerPoint and HTML format. For instance, anyone can download the slides, modify and use them into another course.

VI. Bibliography

[1] Clive Shepherd, Content builders - tools for e-learning authors, *IT Training*, June 2001. Available online at http://www.fastrak-consulting.co.uk/tactix/features/content.htm.

[2] Howard Kaplan, Building Your Own Web Course: The Case for Off-the-Shelf Component Software, *Cause Effect*, Volume 21 Number 4, 1998. Available online at http://www.educause.edu/ir/library/html/cem9849.html.

[3] Paul Tymann,, VNET: A Tool for Teaching Computer Networking to Undergraduates, *32nd SIGCSE Technical Symposium on Computer Science Education*, 1991.

[4] BlackBoard Inc., The Course Service. Available online at http://coursesites.blackboard.com/.

[5] WBT Systems, TopClass e-Learning Suite. Home page at http://www.wbtsystems.com/.

[6] WebCT Inc., WebCT Software. Home page at http://www.webct.com/.

[7] The NetGroup at Politecnico di Torino, Analyzer, a public domain network sniffer. Available for downloads at http://netgroup.polito.it/analyzer/.

[8] W3C Consortium, Cascading Style Sheets version 2, May 1998. Available at http://www.w3c.org/TR/REC-CSS2/.

[9] Opera Software, Opera Browser v. 6.0, Dec 2001. Available at http://www.opera.com.

[10] Netscape, Netscape Navigator v. 6.2, Dec 2001. Available at http://home.netscape.com/browsers/6/index.html.

[11] Mozilla.org, Mozilla Browser v. 0.9.7, Dec 2001. Available at http://www.mozilla.org.

[12] Microsoft Corporation, Internet Explorer. v. 6.0, Aug 2001. Available at http://www.microsoft.com/ie/.

[4] Although the method behind NetLibrary can be used to teach everything (not only technical topics), we do not have any experience in that. Therefore we prefer to say only that NetLibrary can be used to teach technical topics in an excellent way.