Distributed Digital Book

M. Žagar[1], I. Bosnić[1] and M. Žagar[1]
[1] University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia

***Abstract*—Concept of distributed digital book (DDB) based on the XML is proposed. The author's side as well as the reader's side are analyzed. Different book modules and their frameworks are defined. Module design tools are proposed. Practical solution with appropriate examples is shown.**

***Index Terms*—digital book, distributed computing, e-learning, xml**

I. Introduction

E-learning, distance learning, distributed learning tools, digital books are just some of the keywords spread all over the Internet world. In such an environment it is normal to talk about the book as an educational material – prepared and written in a digital form, very often in HTML (PDF) format and visible in different browsers (Acrobat Reader). There are numerous other formats, but they are not widely accepted by a variety of users. All these formats allow text searching, but they cannot be adapted to user demands regarding visualization. Usage of XML (eXtensible Markup Language) facilitates application of different styles.

User – reader can accommodate some characteristics (window size, zoom, font...) or the way of using/reading the material (on-line or download/off-line), but digital books as we know them today are mainly just replacements for their analogue/classical paper counterparts. New technologies provide various new means of creating the educational materials, directed at the better quality of student learning. Total distribution of resources, user mobility and context-aware computing deserve some changes in the way we think, act and behave. Projected to the book writing and reading, this means many changes invisible in the classical process of writing, distributing and reading paper books.

In this paper, the basic concepts are given and a new approach to writing Distributed Digital Book (DDB) is presented. The author's side as well as the reader's side of DDB are analyzed. Proposals of modules of design tools with an emphasis on flexibility and easy transformation are shown. The environment on the author's side is defined based on the fact that the author does not need to be a technical guru, but it is important that the author is aware of the capabilities of technologies that are applied in a DDB (XML, XML-RPC – eXtensible Markup Language-Remote Procedure Call, XSLT – eXtensible Stylesheet Language Transformation…). The reader is a regular Internet user with little or no knowledge about anything except using a Web browser, and expecting maximum flexibility and a book which accommodates their needs. This accommodation could be done on both sides and it is possible that one reader can adjust the same book in different ways.

II. Basic Concepts

Core technologies helping to achieve the abovementioned goals are defined and the whole model is designed and proposed. The usage of XML (and associated technologies) as the basic production tool for Distributed Digital Book is also described. It enables the adjustment and visualization of the DDB on different viewing platforms (personal computers, Pocket PC's, mobile phones…) without the need to change the book content. It also enables quick searching, selective access and reading by chapters (depending on the field of interest), easy and quick linking with list of references and examples which can be very useful. The cover sheet of DDB is shown in Fig. 1.

Examples can be viewed but also, depending on the reader's wish, executed on-line. Executed examples can be either static – prepared in advance, or dynamic, which allows a student to quickly change the given example the way they like, eliminating the need to install different development tools for simple practicing. Connection with the on-line virtual laboratory is provided. It is also possible to develop text to speech and multimedia presentation of materials.

A classical book is printed on luxurious or recycled paper, with defined style text and graphics that do not change within the same edition. The book is easy to measure (size, number of pages, weight and price). Digital book has the same purpose as the classical one and this is where the similarities cease. All other parameters are virtual. Switching from a classical to a digital book provides a lot of benefits, but we should not stop half way by emulating the classical approach. If we analyze classical and digital books, they have similar building blocks.

XML book is a dynamic structure which allows different versions of the book to be published during the development, with easy-to-use updating or adding of the materials. Therefore, the basic concepts of writing an XML book, modules of the book, the visualization and changing methods, depending on a reader's preferences,

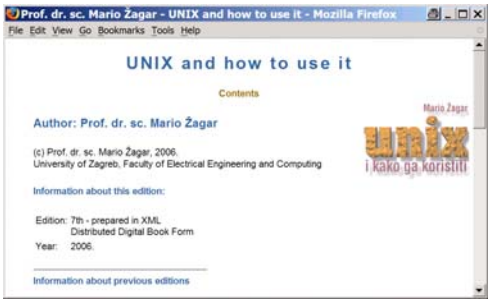

Figure 1. Cover sheet of DDB

will be described. This paper will also include a proposal of a solution of the problems with adaptation of text, graphics, examples and interactivity. This solution presents interrelations and suggestions on how to read such a book. This comprises different visualization definitions and different editions including network versions and off-line versions.

By properly designing and deploying the abovementioned modules, with new technologies in mind and forgetting the paper-book way of thinking, we can end up with a new generation book. The basic organization of the DDB is shown in Fig. 2. The author produces Rich Text Formatted (RTF) document/book, with their editor of choice. For MS Word oriented users – a special style template is developed to help authors with little or no knowledge about XML to write their books. The RTF document is then automatically uploaded to the server (with the appropriate plug-in) and transformed into an XML format. This plug-in can also be used to upload different documents on the Web (automatic Web publishing). Another possibility is using XML editor customized for the purpose of writing digital books. If preferred, author can also use some other XML/text editors.

The XML book (uploaded manually or automatically) is merged with related DTD (Document Type Definition), XSL and one of dedicated CSS-s (Cascading Style Sheet) that define the book's visual identity which is alterable by the publisher, as well as the reader. From now on, examples, interactive or static, are linked with the book. To achieve true mobility, Virtual Laboratory (VLAB) and Digital Video Broadcasting - Multimedia Home Platform (DVB-MHP) can also be encapsulated. For better understanding of some parts of the book, it is possible to connect DDB to another XML – SMIL (Synchronized Multimedia Integration Language) lectures – which incorporate audio, video and text. If used for educational purposes, DDB can also be connected to the desired Learning Management System (LMS) to help students test their newly acquired knowledge. The entire solution is designed with context awareness and the simplicity of the Client side (any type of browser) in mind.

A. Book modules and solution proposals

Let us provide a step-by-step explanation of the directions in which investigation of new frontiers is possible. The book consists of several modules:

- body (text - words, words, words...)
- graphics - pictures, tables, diagrams
- examples (static or interactive)
- audio support
- content, literature, impressum
- related materials
- interrelations or suggestions on how to read it (linearly, following specified topic, independent chapters...)
- different visualization definitions/different editions (large/small screen version, high/low resolution, LAP/Pocket PC/mobile version...).

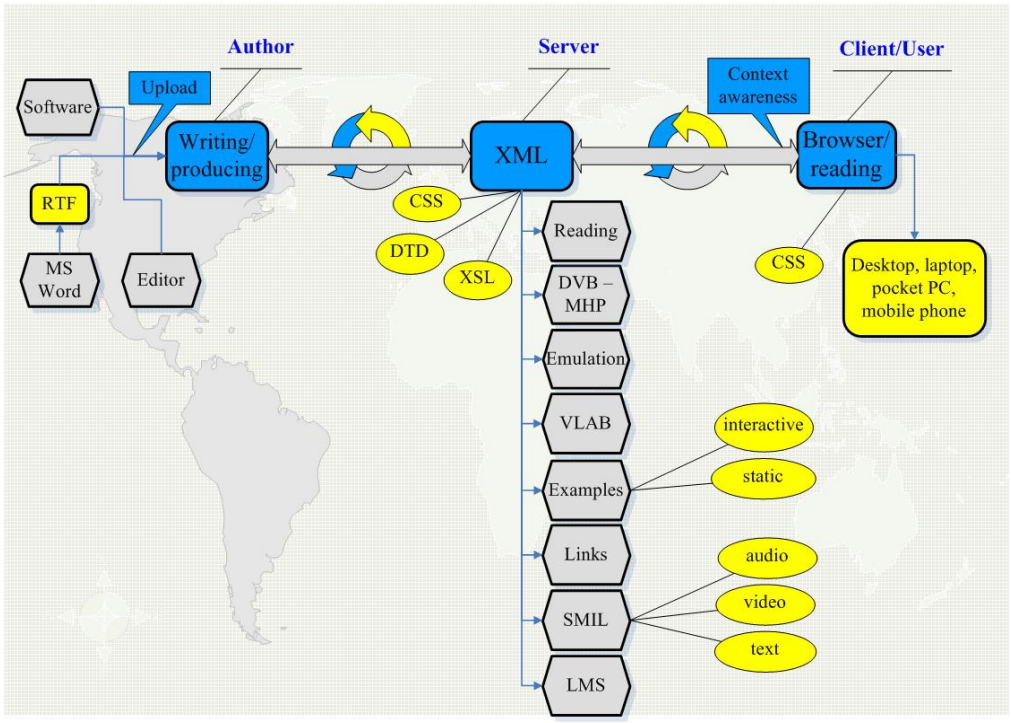

Figure 2. Basic organization of the Distributed Digital Book

B. *Text*

The text is written in XML format and therefore very flexible as well as easy to transform and process. As a structural language, XML does not support definitions of elements responsible for its visualization (creating appropriate style). Therefore XSL (eXtensible Style Language) exists, the task of which is to define the look of an XML document (like CSS for HTML). XSL can transform XML format to HTML format, but it also allows data to be filtered, sorted, joined, mixed etc.

Another thing that can be achieved with XML Linking Mechanisms is creating complex links between documents in order to improve linkage and location of data on the Internet. These mechanisms comprise XML Linking Language (XLink), XML Path Language (Xpath) and XML Pointer Language (Xpointer). Xlink is used for establishing simple or complex connections between Internet documents.

Each XML element can be a link to another document which makes use much easier (user-friendly). Xpath is available for addressing some parts of the documents. Xpointer provides, in this context, an improvement in the sense of addressing the inner structure of an XML document (an instance of some element). With the usage of XML Namespaces we can achieve unique tags which are required for the defining of different tags for an XML book, XML teaching or a virtual laboratory. XML DTD defines a schema that consists of rules for the development of structure and content of an XML document. An example of the text is shown in Fig. 3.

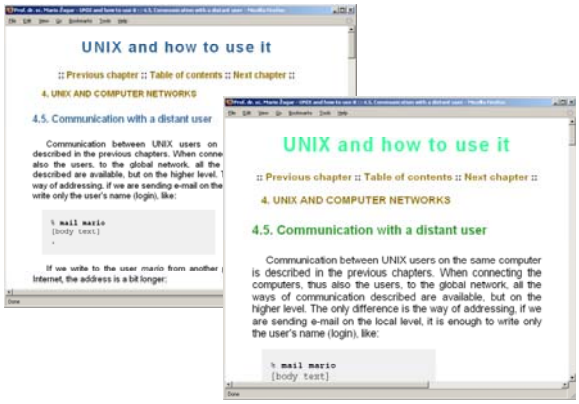

Figure 3. Example of text in DDB - different styles (font size, colour) of the same book

C. *Graphics*

Pictures are scalable and can be resized according to the output media possibilities. The concept of graphics is based on the fact that distance learning requires visualization of materials on the part of the reader. However, because of the limited communication throughput, it is necessary to reduce the amount of information for transfer to as little as possible. Therefore, we use resampling of static pictures on the server. Through communication with the client it is possible to find out the size of the client's browser. A batch file that runs on server and uses a resampling method (for instance, cubic convolution) exists for this purpose. This is shown in Fig. 4. Thus, demanding components, such as pictures, can be merged to the appropriate output parameter (laptop, Pocket PC) and be visualized faster.

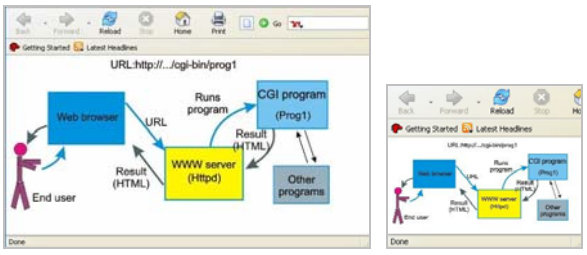

Figure 4. Resampling of pictures for visualization

D. *Examples*

Apart from static text and graphic, Distributed Digital Book enables students to get better acquainted with the educational material by learning through examples, which is a very valuable method in Computer Science. There is a variety of ways to achieve connectivity to examples - static or dynamic.

Some examples are executed in advance, which enables the users to quickly see the results and get some idea about the provided output. This is the quickest way to view examples and never poses additional requirements on the client's side. Still, users feel more comfortable when they are able to complete examples and see the current, real-time output, which can vary, depending on the example.

At the moment of accessing the example by clicking on the example link, the server-side PHP script is invoked. Depending on the type of example (basic UNIX commands or programming languages – HTML, XML, PHP, Java), various ways of execution are possible. Examples of UNIX commands usage are accessed by sending the desired UNIX command to the PHP (PHP Hypertext Preprocessor) script, which executes the command on the UNIX computer set up for learning purposes. The execution of a UNIX command, initiated from within the digital book, is shown in Fig. 5. To achieve better security, only selected commands, approved by the script, can be executed. To broaden the acquired knowledge, UNIX manual pages can also be accessed remotely, so the students can learn more about the command without the need to install the UNIX operating system.

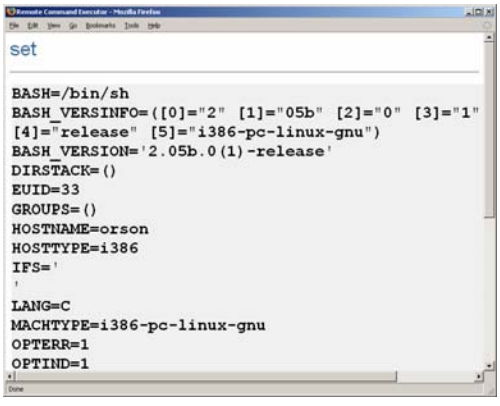

Figure 5. Remote execution of basic UNIX commands

If the examples show the application of programming languages, a different approach is taken. A database with HTML, XML, PHP and Java static examples is already created; students can access those examples through a Web application, which will enable them to view or download the source code, comments for easier understanding, and the current output. The students are given an opportunity to dynamically change the examples already made, but also to create new ones, and try them online. This is a significant educational value, as students learn by doing the most. In addition, this way the need for installing big and complicated tools only to get hands-on experience of some technology is reduced. The usage of the dynamic examples system, where showing the changes in code is just a click away, is shown in Fig. 6. The opportunity to comment on the examples (which are either created originally by the teaching staff, or added later by the students themselves) can be an interesting learning tool, as comments will contain useful, concrete information which students give to each other concerning the example (usage of the functions, better ways to solve the problem, difficulties observed etc.).

Another interesting approach to this problem would be allowing students to choose the system which best suits their needs, and to define it, so the examples would be executed on the desired platform (different Linux and UNIX distributions, programming language versions). That way, examples could be shown on the exact system which the students use; that would reduce confusion when differences appear. For those who want to know more, differences and similarities between the systems could be perceived.

E. Audio support

Reading alone is sometimes not enough or not suitable. New technologies allow users to listen to text-to-speech translation of the book. This possibility can be used for various purposes. The most obvious one is helping people with visual impairment to benefit from the book also. Another purpose could be listening at the moments when it is not possible to read the book. People who are traveling could listen to the book content while driving, some User/Technical Manuals written as a Distributed Digital Book could be easily listened to while fixing that particular device etc.

Audio support for the Distributed Digital Book in our case was developed using MBROLA freeware software **[3]**. It takes a list of phonemes as input, together with prosodic information (duration of phonemes and a piecewise linear description of pitch), and produces speech samples on 16 bits (linear), at the sampling frequency of the diaphone database that is being used. As it was hard to find a text-to-speech synthesizer which would support Croatian language, this unordinary solution was used. As the processing power is limited and storage capacities of the server are more than enough for the Distributed Digital Book library, a decision was made to asynchronously convert text to speech using MBROLA software. Audio is exported and presented to the reader/listener in Ogg Vorbis format **[4]**, an open source high-quality audio encoding technology. As the player is a Java Applet, there is no need to install specialized software, except the Java Runtime Environment (which is already widely spread).

III. DDB - AUTHOR'S VIEW

Different approaches to writing the Distributed Digital Book are possible, depending on the author's preferences and familiarity with technology. Author's awareness of technologies used in the process of creating such a book are recommended, as it leads to better understanding of

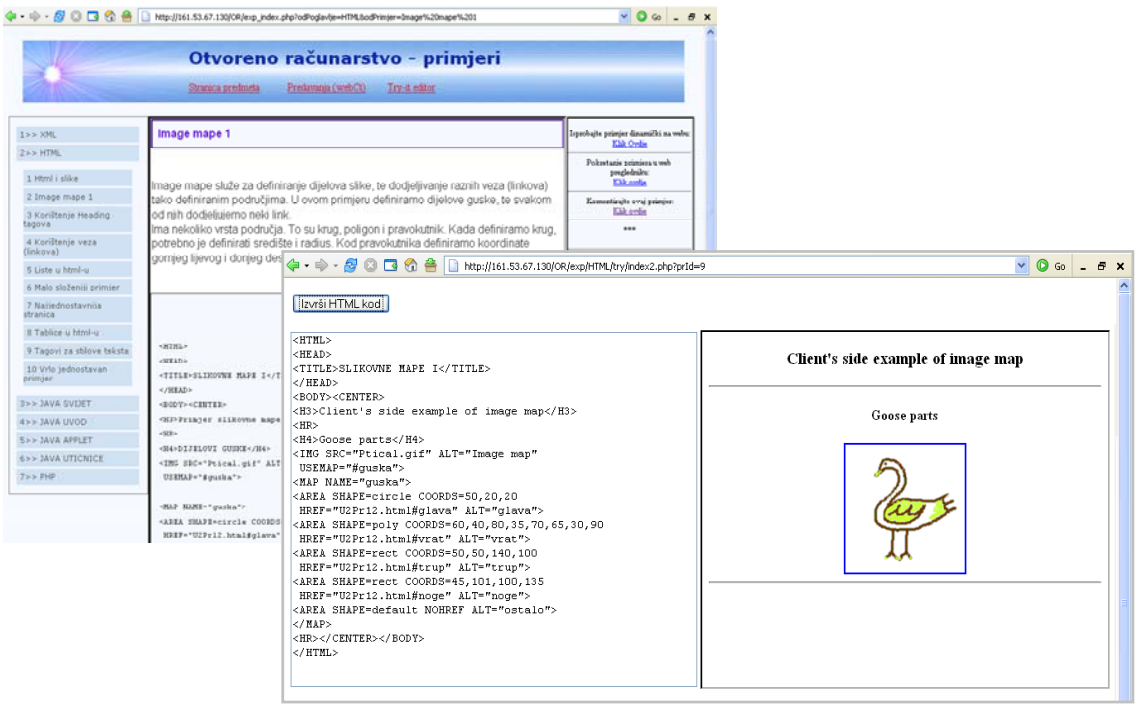

Figure 6. Dynamic changing of examples

what can be done and why things are done that way. Nevertheless, as writing a Digital Distributed Book should be accessible to all, different ways to create materials are offered. The process of creating and publishing the DDB materials from the author's view is shown in Fig. 7.

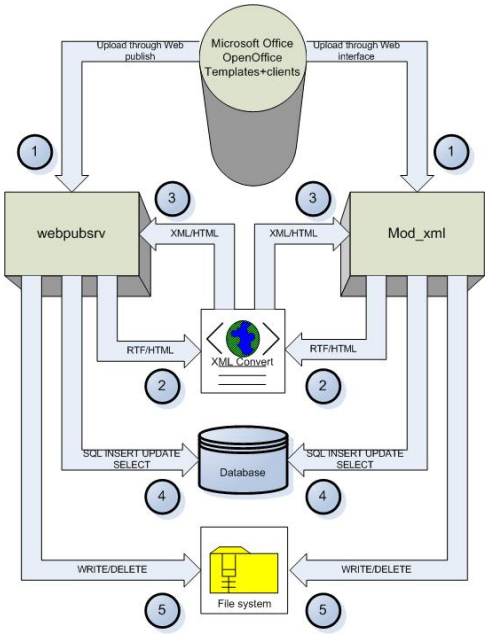

Figure 7. Scheme of author's side of generation process

The most convenient way is creating a Rich Text Format (RTF) document, based on a specially designed template. Currently, only Microsoft Word Document Template is available, as Microsoft Word is the most frequently used text processor in our environment, but, with the increased usage of other applications (such as OpenOffice.org Writer) new templates could be easily created, also. The purpose of this template is not to define the document design (which is done afterwards, using XSLT and CSS technologies), but to associate the content with the type, which will later be transformed to XML tags and attributes. In order to enable the correct transformation to XML, some rules should be observed when formatting text.

File can be transformed to XML using the plug-in created in Visual Basic for Applications and PHP. It takes the document as an input, transforms it to XML using an external, open-source program called MajiX **[5]**. As WMF (Windows Meta File) format, in which the images are saved in RTF documents, is not suitable for the Web, images are automatically converted to JPEG (Joint Photographic Experts Group) or PNG (Portable Network Graphics) formats.

In order to provide easier content development, automated Web publishing of newly created XML documents/books is available in the already mentioned transformation plug-in. This publishing process is in direct connection to FERWeb Content Management System [6]. A special FERWeb module is developed in PHP and uploaded into the CMS. The content is automatically transformed to XML and uploaded – published – on the desired Web page. The module enables administration of the documents so the content archiving becomes much easier. In addition, all the documents have the same structure, which can be quite important when handling a lot of documents, student papers for example. If the author does not want to automatically publish the created material, it can be done afterwards, in any way preferred.

Another way of creating the content is direct work in XML. For this purpose, an XML editor was developed, which enables easier tag handling. It uses the supplied DTD to create the list of all the tags available in the book, to check if all the XML DTD rules are observed, but also to help the author to quickly choose the tag needed, based on the current position in the XML tree. If preferred, other XML/text editors can also be used. Creating achievable examples is usually done separately, depending on the example type. Those can be inserted into the existing example database, or, if the example is not usual, the external link to it can be provided.

IV. DDB – READER'S VIEW

The reader expects maximum flexibility and the book to accommodate their needs. As XML alone does not contain any design information, different designs can be applied to the same content. Changes could be either in style/color sets, to achieve better appearance, or device-dependant, shown in Fig. 3. The style used should be designed separately for eac h type of output device (desktop PC's, notebooks, handheld devices, mobile phones, TV, etc.). Apart from the user choosing the desired design, another option is that the server-side script detects the type of the device and matches the design style automatically.

Distributed Digital Book solves the problem of searching for references at the end of the book by showing the full text of the reference on mouse over when a reference is found in the text. If a reference is a Web page, or is accessible on the Internet, it will be a link which can be followed from the text (shown in Fig. 8).

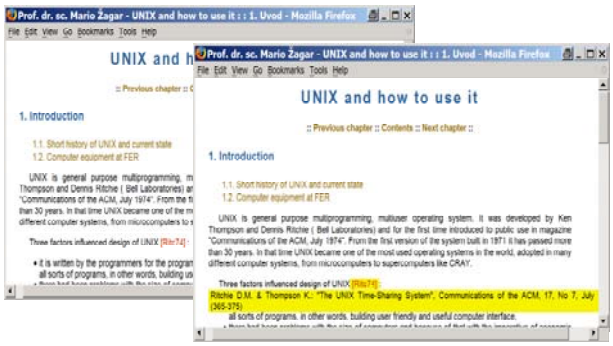

Figure 8. References in the text

On the client's side, information about the already/last visited chapters could be stored, to help the reader continue what they were reading the next time they access the book. Depending on the reader's wish, different reading paths can be proposed, for instance, some chapters of greater importance or a reading sequence for an already experienced user etc.

In case the reader prefers other languages, the content could be translated in advance without the need to change anything in the framework. The links, examples and design remain the same, only the text content is changed.

Another opportunity available to the students is the link to parts of lectures (in different courses currently taught) in which a specific topic is elaborated in more detail or in a way which better suits the user's needs. Lectures are made in Synchronized Multimedia Integration Language (SMIL) format [7], a World Wide Web Consortium recommendation for choreographing multimedia presentations where audio, video, text and graphics are combined in the real time. Lectures are placed in the Learning Management System, which allows the student to quickly find the lecture needed, test their knowledge and communicate with the teaching staff or other students, shown in Fig. 9.

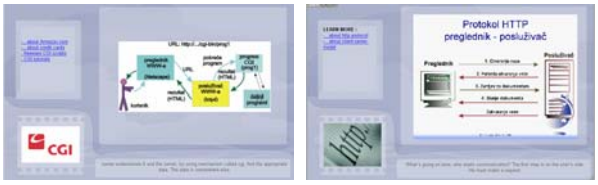

Figure 9. SMIL lectures

In the educational environment it is particularly important to test the student's newly acquired knowledge, not only to grade the student, but also to recognize which parts of the lectures are more demanding, to help students better understand the material. Therefore, in the quizzes, feedback to each proposed answer can be provided/ written, including linking it to the Distributed Digital Book, where more information about a desired topic can be found (shown in Fig. 10).

Figure 10. Link to the DDB in quiz question feedback

When talking in terms of distributed and distance learning in computer science, virtual laboratories are another important way of learning and practicing. Various systems can be controlled remotely, allowing students to practice on real systems that may be at a distance from anywhere and at any time they prefer. A reservation of a real, hardware resource, a microcontroller of some kind, has to be made in advance in order to prevent collisions between different users. After making a reservation, the students control the resource by programming it in a

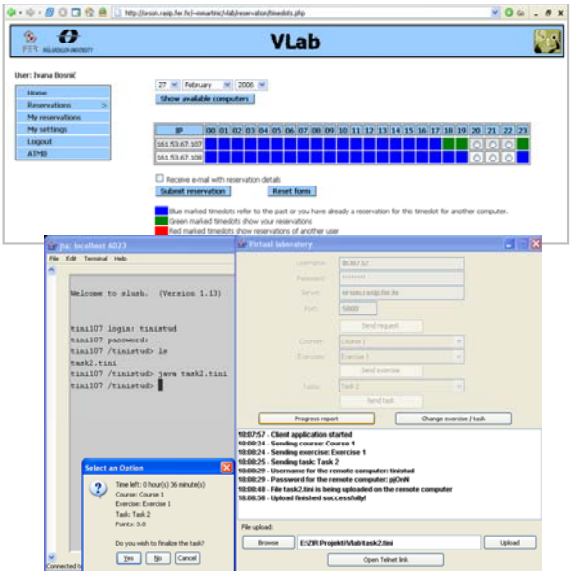

Figure 11. Virtual laboratory reservation and remote control system

supported language. The exercise, if successful, can be graded automatically, thus reducing the workload of the teaching staff. The resource reservation system and the client application for remote control of the microcontroller are shown in Fig. 11.

Examples in the book do not have to be limited to execution of commands or running source code. Flash animations, shown in Fig. 12, can also be used in order to provide more comprehensive, graphical representation of the written text [18]. Flash examples can also allow different levels of interaction with the students.

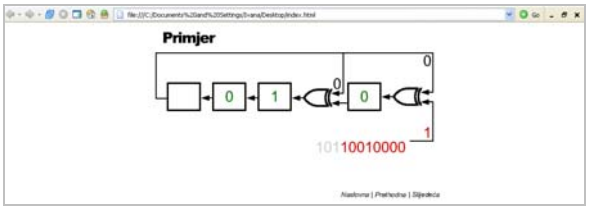

Figure 12. Flash animation of the CRC technique

There is also the possibility to connect Digital Video Broadcasting – Multimedia Home Platform (DVB-MHP) [19]. In the field of digital television, this is a standard that specifies the platform for interactive and multimedia utilities. It enables portability of content for iTV (Interactive Television), creating universal services for DDB readers, and combining them in the sense of interoperability between applications and different implementations of MHP with generic interfaces.

There are some standard components that are encapsulated in each MHP device such as standard Java packages, Java TV 1.0, Media Player, secure sockets and specific API. It is possible to use different profiles for defining different elements of the platform. In the case of DDB, the most interesting is *Internet Access Profile*. This profile is extended from *Enhanced Broadcast Profile* and *Interactive Broadcast Profile*, as shown in Fig. 13.

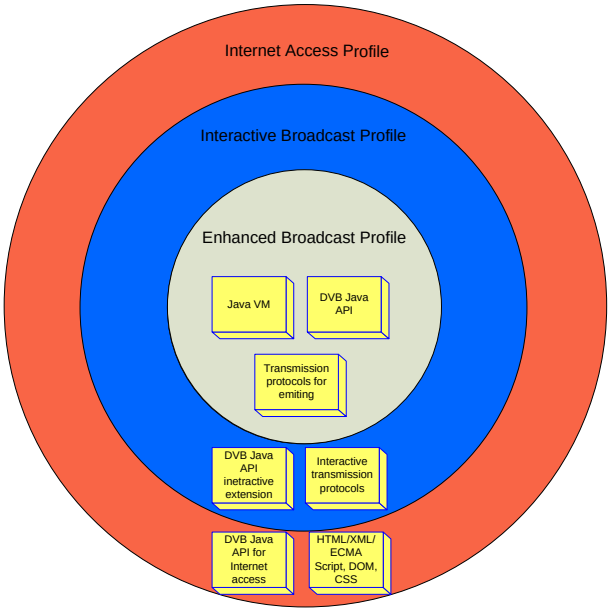

Figure 13. MHP profiles

V. DDB – CASE EXAMPLE

Let's learn distributed! In the case in question, the main part is reserved for two Distributed Digital Books, *UNIX and how to use it* and *UNIX and how to utilize it*, which were originally written as "paper" versions, but with the development of technology, transformed to XML. Both books can not only be read, but also listened to.

If the reader wants to broaden their knowledge about a specified topic, the references (which are in different places on the World Wide Web) can be followed as links. Book examples concerning UNIX commands and programming in HTML can be accessed, tried out and changed online.

To provide support for students enrolled in two different courses, *Open Computing* and *Computers and Processes*, different possibilities are proposed to them. Besides UNIX and HTML, online examples of XML, PHP and Java are available, with the possibility to modify the examples by themselves and try the changes online. Flash animations are used for explanation of the more difficult parts of the courses. The virtual laboratory enables students to work on TINI (Tiny InterNet Interface), a microcontroller-based development platform, which runs Java Virtual Machine to execute code for embedded web servers [21]. In the virtual laboratory, students learn how to program microcontrollers remotely, by using Java language. All the parts are distributed, and form a kind of a knowledge database, which is dynamically updated, depending on the student needs.

WebCT, a Learning Management System, hosts the SMIL course lectures, a number of assessment tasks and also fosters course communication.

VI. CONCLUSION

In this paper, a new approach to writing and publishing digital content is presented. An overview of the author's and reader's side is provided, keeping in mind the simplicity of creating and using the Distributed Digital Book.

Publishing of digital book can be considered the author's contribution to the *World Digital Library* which can make the distance and life-long learning easier. DDB represents the technical development in the sense of content management logic. For example, you can hop from one chapter to another or skip some part of the text. Also, there is a possibility of defining guidelines of a concept by reading the book following some terms and key-words. DDB can be visualized by the reader's demands, so relationship between the book and the reader acquires personal desires. This contributes to the improvement of learning quality.

The book content is based on the XML, which enables separation of content and presentation components. Authors can create the content directly from the Office Application Software, and publish it automatically to our Content Management System, or use preferred XML/text editor. Visualization is done by using XSL language family (XSLT, XPath, XPointer), with the help of CSS and server-side scripts, to achieve the most appropriate presentation for different types of media (desktop PC's, handheld computers, mobile phones, TV…). Book content can also be listened to, in an open-source audio format. Different features, like connection to dynamic examples, virtual laboratory, quizzes and additional SMIL lectures in the Learning Management System broaden educational usability of the Distributed Digital Book.

REFERENCES

[1] B. Goertzel, "Creating Internet Intelligence: Wild Computing, Distributed Digital Consciousness, and the Emerging Global Brain" Kluwer Academic Publishers, Dordrecht, 2001.

[2] K. Pittman: The Digital Talking Book, 2002., URL: http://www.xml.com/pub/a/2002/10/16/dtb.html

[3] T. Dutoit, V. Pagel, N. Pierret, F. Bataille and O. Van der Vreken, "The MBROLA Project: Towards a Set of High-Quality Speech Synthesizers Free of Use for Non-Commercial Purposes", *Proc. ICSLP'96*, Philadelphia, vol. 3, pp. 1393-1396 .

[4] -: "Vorbis", Web page, URL: http://www.vorbis.com/

[5] -: MajiX project, Web page, URL: http://sourceforge.net/projects/majix

[6] -: FERWeb, Content Management System, URL: http://www.fer.hr

[7] -: "W3C Synchronized Multimedia Home page", Web page, URL: http://www.w3.org/AudioVideo/

[8] R. Jimenez, "Rapid e-Learning Design and Development: Alternative Architectures (Part 3)", Learning Solutions e-Magazine, 11/14/2005

[9] B. Hurley, "Interface and Repository Considerations For a System Architecture To Support A Distributed Digital Library" UC Berkeley Library, URL: http://sunsite.berkeley.edu/moa2/papers/braindump2-03.pdf

[10] K. Abell, "Templates, Reusability, Future-Proofing, and the Technology Side of Rapid e-Learning", Learning Solutions e-Magazine, 01/23/2006

[11] W. Brunner, J. Heinstein, "E-Learning As You Like It: From Print to Desktop to PDA Using One Set of Content", Learning Solutions e-Magazine, 12/12/2005

[12] -: "The Whole Picture of Elearning", Web page, URL: http://www.elearnspace.org/Articles/wholepicture.htm

[13] -: "834 Tips for Successful Online Instruction", Web page, http://www.elearningguild.com/pdf/4/guildtipsbook-final.pdf

[14] F. Martinelli Duarte, "Automate Courseware Development with XML and Microsoft Excel™", Learning Solutions e-Magazine, 09/26/2005

[15] P. Clothier, "An Introduction to m-Learning: An Interview with Ellen Wagner", The eLearning Developers' Journal, 07/18/2005

[16] M. Žagar, "Unix and how to use it", DDB in XML, URL: http://staticweb.rasip.fer.hr/library/unix1/

[17] Wrox Author Team, "Early Adopter VoiceXML, VoiceXML with XSLT (HTML and WML)", on-line tutorial, URL: http://www.tutorialized.com/tutorial/VoiceXML-with-XSLT-HTML-and-WML/3703

[18] J. Kaye, D. Castillo, E. Castillo, "How to Build Product Simulations with Flash MX", Web page, URL: http://www.flashsim.com/pubDown/hbps/index.html

[19] T. Ćurin, "Multitier Application Model In Interactive Television System", Master thesis, FER, Zagreb, 2005.

[20] M. Matejiček, "Integration of Office Application Software with Content Management Systems", Diploma thesis, FER, Zagreb, 2005

[21] -: TINI -Tiny InterNet Interfaces, Web page, URL: http://www.maxim-ic.com/TINIplatform.cfm

AUTHORS

Mario Žagar is with the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Unska 3, 10000 Zagreb, Croatia (e-mail: mario.zagar@fer.hr).

Ivana Bosnić is with the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Unska 3, 10000 Zagreb, Croatia (e-mail: ivana.bosnic@fer.hr).

Martin Žagar is with the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Unska 3, 10000 Zagreb, Croatia (e-mail: martin.zagar@fer.hr).

Manuscript received 28 April 2006.