

Integrated Environment for Developing E-Learning Lectures

Vladimir Bondarev¹, Alexander Ossyka² and Afif Mghawish³

¹Department of computer software, Kharkov National University of Radio Electronics,
Kharkov, Ukraine

bond-19@mail.ru

²Department of Computer Science, Al-Zaytoonah Private University, Amman, Jordan

ossyka@yahoo.com

³Department of Computer Science, Al-Zaytoonah Private University, Amman, Jordan

drafif@yahoo.com

Abstract

Commercial packages for developing E-Learning lectures are intended for all kinds of subjects. They lack all necessary or convenient tools for presenting learning content in some specific university courses. The task of the present research project was to design a computer-aided system for developing E-Learning lectures in computer programming. The designed system presents a lecture in several formats. The internal presentation format is used at the stage of the lecture development. This development format is automatically transformed into other formats depending on how and where the lecture is used. Components and the architecture of a software system were designed to implement this approach to creating online lectures. The first version of a computer system with this architecture is successfully applied at the computer software department of the Kharkov National University of Radio Electronics.

Keywords

E-Learning, computer-aided lecture design, computer programming, system architecture, lecture development environment.

1. Introduction

Education creates future. In the foreseeable future any successful human activity will be based on information technologies. Introducing such technologies in education is an urgent task of the present time. E-Learning is a fast growing trend of the modern university education. Experts study its aspects, problems and implications, such as [1, 2, 3, 7, 8, 9, 10]:

- Changing philosophies, theories, strategies of learning/teaching, integration of online teaching with campus-based lectures, tutorials, etc;
- Legal, ethical and psychological issues in online learning;

- Technical aspects (software and hardware for E-learning, methods of presenting E-learning content, support of teacher-student interaction, etc.);
- Practice of online learning, experiences of successful implementation of E-Learning technologies, “de facto standards” in E-Learning, current trends and changes in the field;
- Economic aspects of E-Learning, facts, tendencies of the world E-learning market, etc.

The diversity of learning contents leads to creation of numerous E-Learning tools. Major software manufacturers have their own products for this purpose. For example, IBM Corporation sells its packages IBM Content Producer 3.3 and IBM Simulation Producer 1.0 [11]. Adobe Systems Incorporated offers its products Adobe Captivate 4.0 [12] and Adobe Presenter 7.0 [13]. Microsoft presents its tool Microsoft Learning Essentials 2.0 [14]. Oracle Corporation markets its UPK package [15]. Each of the above packages provides a large number of services for creating E-Learning content. Online university course lectures are the main concern of this paper. For example, the IBM Content Producer 3.3 package performs more than 200 functions. The package enables an online lecture designer:

- to create E-Learning content in a standard form;
- to design new lecture patterns;
- to include text fields, images, data sheets, buttons, animation into lectures;
- to use a special data base of content components for a parallel team work on several E-Learning projects;
- to create E-Learning content for handicapped people;
- to combine an existing online course with various learning objects from different sources and projects;
- to include java-script functions written by outside programmers into a lecture.

The Adobe Captivate 4.0 package can transform text into speech. It can manipulate Microsoft PowerPoint and Photoshop files. The package has many other functions that are useful for creating E-Learning content.

Smaller software manufacturers also have their own tools for developing E-Learning content: Courselab [16], WBTEExpress [17], REDCLASS Course [18], QuizForce [19], eAuthor [20], etc. Typically, teachers and students require substantial training to use such software packages.

Despite the existence of a large number of E-Learning tools, the computer-based education is not universally accepted by teachers and students even at universities. Some reasons for teachers’ reluctance to bring E-Learning into play are as follows:

- There is a lack of information about existing E-Learning solutions that are suitable for presentation of a particular learning content;
- It takes some time and effort to master a large-scale universal tool for developing E-Learning content. An instructor may need cooperation of the tool designers. These factors leave less time for the instructor to concentrate on a new lecture itself, its accuracy, brevity, and clarity.
- A universal package for developing E-Learning content may lack means convenient for performing specific operations in a particular course, like analysis of mathematical

functions, presenting chemical structures, experimenting with a code of a computer program, etc.

The current research aims to address some of these problems in respect of a limited set of university courses.

2. Research objectives

The task of the current research was to create a computer system for designing E-Learning lectures in computer programming. Studying computer programming necessarily includes such important activities as studying syntax of a programming language, solving problems in program design, practicing in algorithms design, developing skills in reading and writing programs, etc. E-learning lectures in programming should provide material and means for these activities. The objectives of this research are as follows:

- to study lectures at their model level and to find out the components of such models.
- to enable an instructor to develop an online lecture at its model level. Such a lecture should be comparable to a traditional campus-based lecture by its educational effect.
- to find out the structure of a system for developing E-Learning lectures. The system should present study content, support interaction between a student and a teacher (a lecture on a computer screen), enable a student to assess the quality of his/her knowledge and skills acquired from the lecture.
- to implement the system in order to use it for teaching students via the internet.

3. Special features of a computer-presented lecture

Learning is an active process. That is why the work on a lecture content should permanently require some activity from students (solving problems, searching contradictions, making brief notes of the studied material, etc). There should be a wide variety of teaching assignments, tasks, questions, and problems. The more different teaching methods are used the better is a lecture.

A lecture is not a reference material. It is studied gradually, section by section. That is why it should consist of a sequence of elements. Sequential arrangement of lecture elements corresponds to their study sequence. A lecture is not viewed as a general hypertext with links in it.

A lecture is assumed to be moderate in size. A student should be aware of the lecture plan and its part discussed at a given moment. A student has to assess his/her knowledge and skills acquired from the lecture. So any lecture is supposed to contain problems with solutions and tests.

The design of a computer-presented lecture is expected to take the amount of time and effort comparable to the preparation of a traditional lecture presented by a lecturer. Visual presentation of a lecture should not be a hard task. A lecturer is supposed to concentrate on the lecture content only.

4. Lecture component model

A lecture model is the internal representation of a lecture. It is used for automatic generation of the lecture in different formats: in the dhtml format for a web-browser, in the format for printing, in the xaml format for offline applications, etc.

The lecture model consists of separate loosely connected components with various properties. It is similar to the component model of a user program interface. The structure of a lecture model is linear, not hierarchical, since a lecture consists of a sequence of components.

4.1. Component set

The set of components is stored in the library of components. In computer programming a program component (operator, function call, etc.) can be presented in two formats: the format for the program development period and the format for its execution. In contrast, a lecture component has one format for the development purpose and many “execution” formats by the number of presentations of a lecture model. The development format does not have to resemble any execution formats of a lecture. It has to precisely identify a given component in the set of all components and to make it possible to set or change the component’s properties.

The model component set is open and expandable. Here are some core components of the set:

- section heading
- plain text
- program code example
- problem with solution hints
- test with predefined questions
- historical information
- list
- table
- comment.

4.2. Lecture development environment

The lecture development environment makes it possible to construct a lecture model from standard components. See Figure 1. Theoretically speaking, it can be done manually by creating a xml-document. From a practical point of view, it is better to have a visual lecture builder similar to the visual builder of a graphical interface for computer programs.

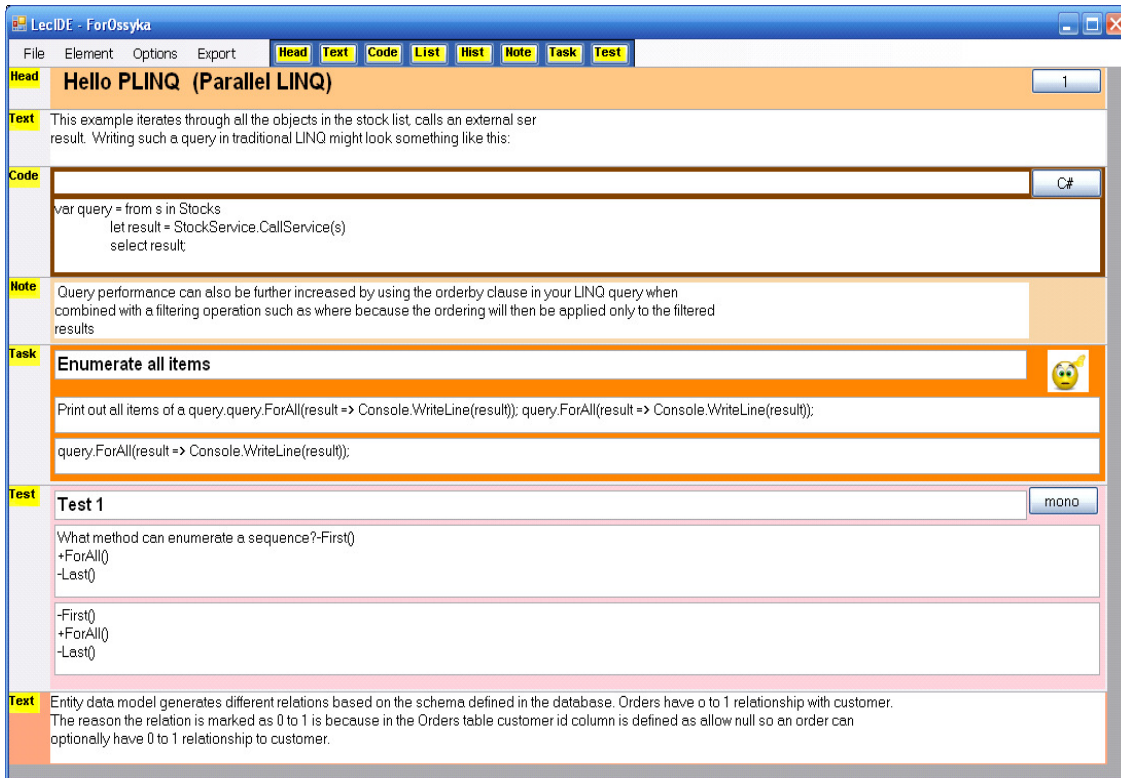


Figure 1. Lecture development environment

4.3. Lecture export

Lecture export means automatic transformation of a lecture model into some generally accepted format, like dynamic html for studying a lecture by means of a web-browser, a format for publication in some social network, etc. See Figure 2. The set of export types is open. It can be expanded without the necessity to remake the previously created lecture models.

5. Architecture of the lecture development environment

The program implementation of the lecture development environment consists of three subsystems (See Figure 3): a model (It is marked by red in Figure 3), mapping (marked by green), and export (marked by blue).

5.1. Model subsystem

The class LectureModel is the base class of any lecture model though it inherits a visual control methods of the class FlowLayoutPanel. Another part of a model is a sequence of objects which inherit from the class BaseElement . This sequence of objects is denoted by the name XxxElement in Figure 3. BaseElement is derived from the class UserControl which provides means of control like FlowLayoutPanel.

A question might arise: “Why have we deviated from the rule of separating models from their implementation?” The answer is as follows. This system aims to support lecture editing operations , such as add, delete, transfer, highlight for manipulating lecture elements. These operations have been already implemented to a certain degree in the standard library of control components. To separate a model from this standard library means the necessity to implement the control components once again. It should be also mentioned that in this system, the visual representation of a model is obtained in result of the model export. It takes us back to the traditional separation of a content and its representation.

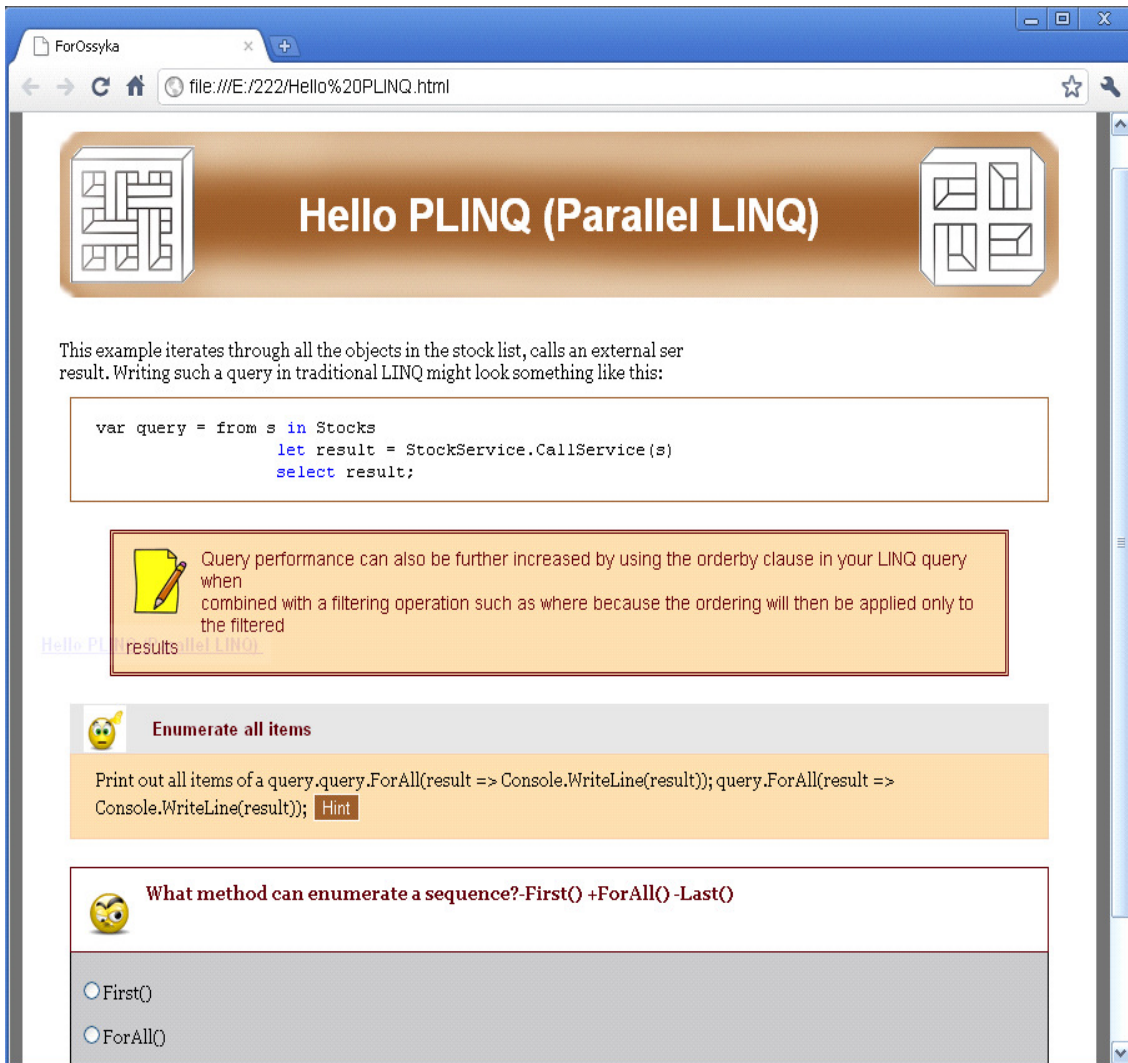


Figure 2. Lecture export in the DHTML format

5.2. Mapping subsystem

Mapping and editing of a lecture are performed to a great degree by means of the lecture model. For lecture display and editing, the means of a lecture model are used substantially. The class LectureModel inherits FlowLayoutPanel which is based on the form and is capable to be a container for visual representation of lecture elements.

The event system is a necessary part of control of editing operations. The system monitors issuing commands applied to lecture “documents”: *create new*, *open*, *save*, and also *start* and *end* execution of a lecture. These events are used to perform the required operations and to inform a user about a possible loss of data in case of a wrong operation.

This section of standard functionality is provided by the class HiBye. This class controls events associated with menu commands and events of the type *Load* and *Closing*. Execution of commands, like loading and saving data in a file, is performed by a client of the class HiBye. The client is responsible for the

implementation of the interface IFileHolder. HiBye improves the structure of this application. It can be used repeatedly.

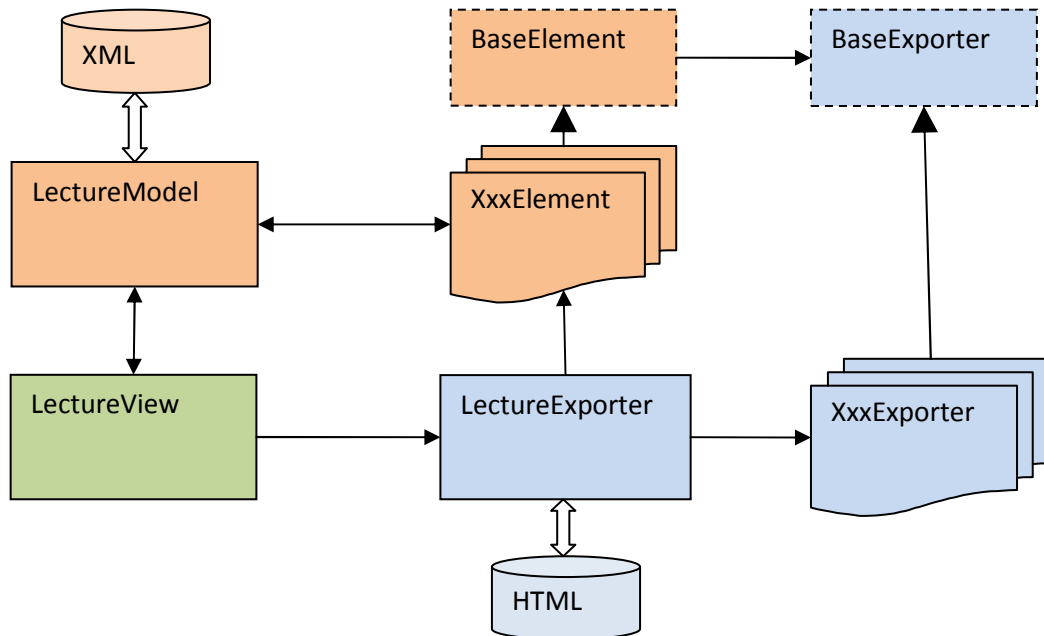


Figure 3. Architecture of the lecture development environment

5.3. Export subsystem

LectureExporter is the base class of the export subsystem. See Figure 3. A LectureExporter object receives a sequence of lecture elements stored in a lecture model.

The method ToHtml () performs export of the lecture model into the html format. This method traverses the sequence of lecture elements and transforms each element into a html document

fragment. The sequence of such fragments compose a complete html document, the export result. HTML fragments are generated by objects of partial exporters.

These objects are created when required while an input list of the lecture elements is processed. Lecture elements are mapped into partial exporters by names of classes. The ending “Element” in an element class name is replaced by the ending “Exporter” to obtain the class name of the partial exporter: CodeElement → CodeExporter, TextElement → TextExporter, etc. It ensures a complete independence of a lecture model from the export system and a minimal dependence of the export system from a lecture model.

6. Related work

E-Learning will become a universal practice only if students are willing to study online. Here are some reasons why some students are not inclined to do so:

- Typically, students study online on an individual basis. It may create problems with study motivation and peer learning. It becomes more difficult to build relations with teachers and other students. There are other psychological problems related to modern E-Learning [2].
- Tutorials are essential for learning. At present tutorials delivered by online tutorial systems are less attractive for students compared to face-to-face tutorials. The situation may be improved by building Intelligent Tutorial Systems. Such systems will provide customized instruction and feedback to students simulating the behavior of human tutors. It might be helpful to employ computer game techniques in computer tutorial systems, introducing elements of competition with the system and/or peers [4, 5, 6].

The above outlined student-related problems of E-Learning will be the subject of research and practical work for many years to come.

7. Conclusions

Online learning is a promising trend of modern and future education. At present E-Learning is not as effective and popular as it could be. Some instructors are not inclined to use advantages of the Internet in teaching their courses. To a certain degree it is caused by the considerable amount of work required for mastering some universal E-Learning package and for developing the required learning content by means of such a package. It looks appropriate to create simple-in-use specialized E-Learning tools along with universal packages for developing online study content. The task of this research was to find out the essential components of campus-based lectures in programming languages and to design a computer system (environment) for developing online quality lectures which is easy to learn and use. At present, the first version of the system has been created which starts the design of a lecture with the construction a generic lecture model. This version includes a library of ten standard lecture components, two styles of presenting a learning content. It performs lecture model export into three formats: DHTML, HTML, and MS Word document. The available component library is oriented on developing online lectures in programming courses. It can be expanded by including new components into the library. At present the system is used at the Kharkov National University of Radio

Electronics. It is a tool of choice for developing pilot courses of online lectures, especially under the pressure of time. At the stage of course modification an instructor may find the system functionality insufficient: absence of animation, sound, etc. When and if the instructor becomes unsatisfied with the environment he/she is already prepared to switch to advanced commercial packages.

References:

- [1] Haythornthwaite, Caroline & Andrews, Richard (2011) *E-Learning Theory and Practice*, SAGE Publications Ltd.
- [2] Rudestam, Kjell & Schoenholtz-Read, Judith (2010) *Handbook of Online Learning*, 2nd ed., SAGE Publications Ltd.
- [3] Anderson, Terry (2008) *The Theory and Practice of Online Learning*, 2nd ed., Athabaska University Press.
- [4] Nkambou, Roger; Bourdeau, Jacqueline & Mizoguchi, Riichiro (2010) *Advances in Intelligent Tutoring Systems*. Springer.
- [5] Woolf, Beverly Park (2009) *Building Intelligent Interactive Tutors*. Morgan Kaufmann.
- [6] Evens, Martha & Michael, Joel (2005) *One-on-one Tutoring by Humans and Computers*. Routledge.
- [7] Bates, Tony (2005) *Technology, E-Learning and Distance Education*, Routledge.
- [8] Fernandez-Manjon, Baltasar & Sanchez-Perez, Juan (2007) *Computers and Education: E-Learning from Theory to Practice*, Springer.
- [9] Horton, William & Horton, Katherine (2003) *E-Learning Tools and Technologies: a Consumer's Guide for Trainers, Teachers, Educators, and Instructional Designers*, John Wiley & Sons.
- [10] Horton, William (2000) *How to Teach Anyone Anything Anywhere Anytime*, John Wiley & Sons.
- [11] www.ibm.com/ru/educ
- [12] www.adobe.com/products/captivate/
- [13] www.adobe.com/products/presenter/
- [14] www.microsoft.com/learningessentials/default.msp
- [15] www.oracle.com/global/ru/infex.html
- [16] www.websoft.ru
- [17] www.arima.com.ua
- [18] www.redlab.ru
- [19] www.elearningsoft.ru
- [20] www.learnware.ru

Authors

Dr. Vladimir M. Bondarev graduated from the Kharkov National University of Radio Electronics in 1974 in Ukraine. He got his Ph.D degree in Computer Science from the same university in 1981. At the present time he is working as a full professor at the department of computer software at the same university. He is also a part-time consultant-instructor for the joint American-Ukrainian computer software company Global Logistics. He has 30 years of experience in research and teaching university courses like computer programming, Internet programming, data structures and algorithms.

Dr. Alexander F. Ossyka graduated from the Kharkov State University in 1969 in Ukraine. In 1979 he got his Ph.D. degree in Computer Science from the Kharkov National University of Radio Electronics. He worked as an associate professor at the department of computer software at the same university. Since 2004 he has been working at the Al-Zaytoonah private University in Amman, Jordan. He has more than 30 years of experience of research and teaching courses like discrete mathematics, computer programming, Internet programming, data structures and algorithms at the university level.

Dr. Afif J. Mghawish graduated from the Kharkov Polytechnic University in 1986 in Ukraine. In 1991 he received his Ph.D. degree in System Analysis from the Leningrad State University. In 1997 he became a member of the faculty staff at the Al-Zaytoonah private University in Amman, Jordan where he is working now as an associate professor, a deputy dean of the Science & Information Technology faculty. He has 19 years of experience in research and teaching various courses in Computer Science.