PLANNING, DESIGNING AND EVALUATING CBL HYPERMEDIA APPLICATIONS

M.Vladoiu

ABSTRACT
This paper presents some criteria about how to plan, design, evaluate and use a CBL hypermedia application. In our university, four years ago, we started a new program called "Multimedia Integrated Model for Active Learning" (MIMAL). During the development of this program we have found out some basic guidelines for building sound hypermedia CBL applications and for using them properly. These include: planning carefully the project, developing wisely the content, designing a good layout, proper evaluation of the final application, and distributing/managing the application.

We present here also a sample application: an instructional package for teaching and learning Operating Systems built and distributed following these guidelines. The project team was split into four sub-teams – each one of them used a different hypermedia authoring technology (GLpro/Gmedia, FrontPage/Microsoft, Java/SUN, and TOOLBOOKII/Asymetrix). The application presented here was achieved using the second sub-team's choice, the Asymetrix solution.

KEYWORDS
Guidelines for sound hypermedia Computer Based Learning application, online learning, hypermedia authoring technology, interactive learning system, virtual university environment

INTRODUCTION

Perhaps more than any other type of instructional environment, a computer-based learning (CBL) application represents a combination of three equally important elements: content, form and functionality. So developing an application that works the way it was intended to and that successfully delivers its content to its users requires the combination of several capabilities: careful planning and project management, mastery of the material to be presented, the understanding of the technical development issues, a proper aesthetic sense, and a thoughtful approach to the use of design elements and special effects. So, to create educationally sound and complete instructional materials is a difficult task that should be tackled in a careful way.

In this paper we present some criteria on how to plan, design, evaluate and use wisely a CBL application. Four years ago at "Petroleum -Gas" University of Ploiesti we started a new program called "Multimedia Integrated Model for Active Learning" (MIMAL). The program goal was the creation of multimedia software tools that offer every teacher the possibility to build and present their CBL applications (courses, seminars, laboratories and exams) in a hypermedia way. During the development of this program we have found out some basic guidelines for building sound CBL applications and using them properly. We present these guidelines further on:

- *planning carefuly the project*
  - evaluate resources and identify requirements
  - consider distribution options
  - develop a user profile
  - create a project plan

- *developing wisely the content*
  - distinguish between concepts and tasks
  - develop teaching and learning objectives
  - create application outlines
  - choose an instructional approach
- *designing a good layout*
  - understand the elements for a good design
  - use visual metaphors, text, multimedia, navigational paradigms and templates
- *proper evaluation of the final application*
  - make sure the application works properly
  - test effectiveness
- *distributing and managing the aplication*
  - make the multimedia classroom ready
  - implement the application
  - test the application running
  - get user feedback
  - manage and maintain the application

We present here a sample application: an instructional package for teaching and learning Operating Systems built and distributed following these guidelines. This project team has split in four sub-teams – each one of them has been using and experimenting with a different hypermedia authoring technology. The first one has chosen the GLpro multimedia engine owned by Gmedia. The second has preferred the Asymetrix TOOLBOOK II solution [2,14]. The third one has taken as their choice a programming language, the well-known Java from SUN. The last one has tried an HTML editor, FrontPage. The application presented here was achieved using the second sub-team's choice, the Asymetrix solution.

PROJECT PLANNING

Project planning is a preliminary step in which we evaluate the available resources, decide a distribution strategy, develop an end-user profile and create a project plan. Even though this process is time consuming this is a worthwhile effort. Its outcome is a road map for our project that can help us to measure expectations, activities and results [2].

1. *Resources vs. requirements*

First we are supposed to identify the people who will work to build the application, the usable tools and the available time and money to spend for this project.

- *Personal*: A CBL application could be constructed by only one person or by a team of teachers, programmers, graphic artists, animators, audio/video specialists and, of course, subject-matter experts;
- *Tools*: Beyond the requirements for authoring and management tools or for programming environments with built-in facilities for manipulating multimedia information, generally there are necessary some other software packages (audio- and video-editing applications, three-dimensional graphic programs), and computers capable of running these packages as well;

- *Copyright fees* – beyond such fees for the above software packages, we must consider also acquiring the copyright for every piece of information we use (including artwork that seems to be free, but it is not!);
- *Classrooms*: When we are starting a CBL program we must carefully think at the spaces we will be able to use as multimedia classrooms. The old classroom will need improvements in order to be used as a multimedia classroom. We also should think what changes are necessary for our labs to make them ready for these modern requirements. So, the old instruction spaces have to be modernized for the right implementation of the CBL applications. We might also need some new places (we must keep in mind the virtual reality is quickly emerging nowadays);
- *Costs*: In order to get a cost-effective teaching and learning solution (vs. traditional instructor-led training), the development and deployment of a CBL application involve thoughtful budget planning. The salaries of team members, consulting fees for domain experts, equipment investments, and charges for stock photos, audio and video-clips are some important components of the application cost;
- *Time budget*: Although scheduling a project can seem like guesswork, there are some formulas that can help us in evaluating the necessary time for developing a CBL application. For example, if the development involves a substantial programming effort, than it has estimated two and a half hours effort per minute of completed application [2]. We can significantly reduce this development time (to 16 minutes per minute of completed course) if we use extensively predefined objects stored in the libraries of working environment. Finally, the amount of time needed to develop a CBL application depends primordially on the scope of the application, on its role within the overall teaching and learning strategy. We are therefore required to answer questions like: is the application just one of the ways the information will be presented, or is it the primary means of instruction? Is this application a part of a series or is it stand-alone? What should be the quality of the final product? Will it likely to be updated or expanded in the future?

2. *Distribution options*

An important part of planning a CBL application is deciding how to deliver it to the end-user. Such an application could be distributed over the Internet or some intranet, over a LAN, or over stand-alone computers (running the application from diskettes or from CD-ROMs). It can be used also a hybrid method combining network advantages with load-drive speed, some information being distributed using the network, while other information is distributed locally. Choosing a deployment method depends primarily on the environment in which the application will be run, on the available equipment and on application technological demands.

- *Distribution using Internet/intranet*: It is a well-known fact the Internet enables the delivery of CBL applications to anyone, anywhere, at any time. The main advantage of Internet use is that any change we make to the application is fast, easy and inexpensive to implement. However there are still some problems – e.g. bandwidth limitation can make the transfer of media files slow and tedious for the user;
- *LAN Delivery*. Within a LAN a CBL application can reside on a server. The users can retrieve and install a copy of it on their hard drive. Nevertheless, some large media files can still reside on the server until they are requested by the application;

- *Deployment using CD-ROMs or floppy disks*. Delivering applications in this way has a serious shortcoming: changes become more difficult to make. They can require rewriting of materials on diskettes or re-manufacturing CD-ROMs;
- *A hybrid distribution method*. In order to overcome the bandwidth bottleneck, we can combine the above delivery methods by having some information distributed via a network while some other files (especially large media files, that may download slowly from a network) are stored locally.

3. *User profile*

Understanding end-users is crucial for developing an efficient CBL application. So we should build a user profile. This profile is very helpful in estimating the appropriateness of the application content and of the instructional methods used. In order to get to know the end-users we can use both formal and informal procedures. We can interview them, monitor their work, observe them reading documentation materials etc. While gathering this information we need to find the answers the following questions: What are the educational and cultural groups in which our users fit in? What age ranges do they represent? Are they somehow disabled? How computer-literate are they? What are their studying styles? In addition to this information we should also evaluate their instructional environment. If our application is meant to be used in some complex environments, we should answer some extra questions: do the users sit, stand or move around? Is it quiet or noisy? Is it bright or dark?
All these factors will have an effect on application content, instructional approach and on layout.

4. *Project plan*

We get as a result of the planning process a project plan. This will assist us during the whole developing process and will also help us to evaluate the progress of that process. This plan will be as a map from application demands to its completed state. A project plan should usually contain the following entries:

- *Need for the CBL application*: We should include here some explanations about the need to use such an instructional alternative. The general idea here is that CBL is usually cost-effective compared to traditional print- and classroom-based training;
- *Resource list*: We are supposed to describe here the resources (human, technical, classrooms, financial, time) needed for the project (that we have or not);
- *User-profile*: Here we have all the analyzed information about the target users of our CBL application;
- *Teaching and learning objectives* should not only be detailed for the application as a whole, but also for the significant parts of it;
- *Content outline*: We have to say what are the main topics and sub-topics, to describe quizzes, overviews, summaries and feedback links;
- *Instructional approach*: We must determine the most suitable teaching and learning model. We also have to determine the right pedagogical methods to implement that model;
- *Layout design*: We must write here about the chosen layout approach and explain why this is suitable. In addition, it is useful to include here some sample pages in order to make clear to the learners how everything in a CBL application will act (namely menus, navigation links, buttons etc);

- *Evaluation strategy*: In this part we must list the testing, debugging and evaluation methods for our CBL application;
- *Developing schedule*: Keeping in mind our ancestors' proverb, *divide et impera*, we must divide the task of the CBL application development into small sub-tasks and develop a detailed timeline for solving each of them. Such sub-tasks could be project planning, content deployment, design, reviews, testing, and production.

CONTENT DEVELOPING

In this second step we have to model the application content according to users' needs and instructional goals.

1. *Concepts and tasks*

Two main types of information have to be contained in a CBL application [2]: concepts and tasks. Conceptual information answers the question *Why do I do that*, while tasks are a reply to *How do I do that*? Concepts have a declarative nature – they consist of background information, terminology, theories etc. The most suitable way to present them seems to be text enriched with a few multimedia elements. Tasks, on the other hand, require users to follow a particular sequence of steps in order to achieve their goal. This comes from the task procedural nature. For example we can think of assembling a computer as a task. It is easier to deal with tasks if we apply again that proverb and we *divide* the task into smaller parts that are easier to handle (*impera*). In order to make the task information come alive to the learner, it is more appropriate if we use prevalently media information to represent them.

2. *Teaching and learning objectives*

We should fix some benchmarks to achieve for each CBL application and for the significant parts of it. These must be concrete in order to be effective [2].

3. *Application outlines*

During the outlining of the CBL application we must consider the following issues:

- Providing basic instructions on using the application - they should be very few if we did the application right!
- Overall and progress view – people like to know what they have to do and how the things are going on;
- The application should be divided into small clear and concise units (modules), every one of them ending with some tests;
- Being favorable to user interaction, taking into account that users learn better if they interact to the full with the application.

4. *Instructional approach*

There are many ways to deliver information to learners [2,7] and we need to choose the appropriate style in the light of our plans. We list below some of them:

- *Presentation* is the classic way to deliver information. This is a linear flow of concepts and non-interactive tests, that do not involve the learner enough to master the subject or to develop a skill;
- *Tutorial* is more interactive than a presentation. It uses menus and hyperlinks to move through the content. There are several kinds of tutorial: presentation-and-practice, interactive-demonstration, skill-modeling, guided-exploring;
- *Simulation* presents a concept/process by mimic its real-world characteristics and behavior. This approach offers a good level of knowledge because it is close to the real thing, having the feedback built in;
- *Quizzes* are a good way to clarify the knowledge achieved by the user. When we design a quiz we better include a feedback in response to each question. This shows users whether they have answered right and can also provide other helpful information like further instruction or hyperlinks to related topics. Quizzes must not be intimidating in order to be effective. They should allow the user to try again or to skip - go back later;
- *Virtual reality* provides the highest level of mastery because of the immersion into the 3D virtual environment that users can manipulate interactively. This immersion makes the computer-generated world feel real.

It is obvious that using an interactive way to convey information to the users we can get better results. A “live” educational environment is built this way.

LAYOUT DESIGN

The use of any computer-based system involves a range of cognitive constructs and processes including perception, short and long-term memory, reasoning and problem solving. Attentional capabilities are very relevant for developing a good hypermedia CBL application. The common view of attention is that of a process. The attention process involves the use of mental effort in selectively processing information from a range of different sources, both internal (including our own thoughts and dreams) and external (including auditory and visual stimuli). Although psychologists have been studying the features of human attentional mechanisms since the beginning of the previous century, the nature of attention is still not completely understood. The literature presents a number of competing theories of attention, each of which appear to explain a range of empirical findings gathered to date. The findings themselves are of various kinds and have varying degrees of relevance to multimedia design. However, the following phenomena appear to be of general agreement in the literature [7, 11]:

- People are able to focus their attention on particular stimuli;
- People are able to attend to more than one stimulus at any one time;
- Attending to information requires mental effort;
- People are able to direct attention to groups of stimuli sharing sensory characteristics.

The consequences of existence of these phenomena for designing hypermedia interfaces are summarized below:

General guidelines:

- More than one piece of information may be displayed on the screen at any one time using static media (text or graphics) as long as the user is given control over the appearance and disappearance of the information;
- Users are likely to have little difficulty in attending to a particular passage of speech, sound or music while other auditory information is also being presented;

- Information about substantially different subjects or topic areas should be removed from the display (and from auditory channels) once a user begins to study a presentation about a new subject;
- When more than one form of information is presented concurrently using dynamic media, the combination will tend to be more usable if:

    - the different pieces of information are presented using different sense modalities, or require different forms of cognitive processing;
    - the medium of presentation is familiar to the user;
    - the extraction of relevant information is easy;

- Users should be given the flexibility to "switch off" particular forms of output where they are not essential to the task in hand;
- Groups of information which users are intended to see as related should be given common perceptual features;

Design guidelines for still images:

- Scenes are not attended to uniformly. So objects which are larger, brighter, shown in more detail, away from other objects, or shown away from rest will be focused on in preference to other objects;
- Using of a highlighting technique (change colour, add symbol or label) draws attention to important objects. By default only the scene level will be focused upon unless the user is motivated for closer inspection;
- Allowing viewing time is essential. We must avoid showing an object in motion or using a highlighting technique when the user is extracting information from an image. We should delay with at least a second the changing of an image;
- We must be aware of using too many highlighting techniques within an image at once. Sequencing highlights to move attention from one object to another is a good solution;
- Gradually revealing of objects and symbols is a way to control viewing order. The reveal effectively attracts attention;

Design guidelines for moving images:

- Onset of motion will attract attention; but it will also shift focus away from static objects in the scene;
- Animation requires attention. If attention is already focused (such as reading a text/label or attending to another animation) motion onset may not automatically shift attention;
- Animation should be used with care. Tracking an object motion will maintain focus but may prevent focus shifting elsewhere;

Design guidelines for text:

- Generally an image will be focused on before text. If focus is required on the text prior to the image, the text should be displayed before the image or the text area should be larger than the image;
- Revealing captions and labels is particularly effective and can be used to direct the user's reading sequence;
- Ensuring that a label and an object appear together improves the identification process. Displaying a label and an object produces fixation shifts between the object and the label;

- Reading time is necessary for text. When users are reading text, focus will not be able to shift until a break is encountered (end of a sentence or paragraph). The more complex the text is, the longer the reading time required.

Design guidelines for speech:

- Multiple strands of speech or sound will interfere with each other and distract focus. Speech can be focused upon concurrently with visuals, but no more than one strand of speech should be presented at once;
- Revealing of objects and labels when cued in the speech track is essential. Cueing labels within the speech track will produce a shift of attention to the object and its label;
- Reading time after cueing a text is also needed. Reveals or animation for the duration of speech segment, which cues a label, must be avoided. If the label is complex, reading speed will be similar to that for speech track to pronounce it.

We can use predefined templates (only if we use a specialized tool for CBL) or we can develop our own patterns. Using visual metaphors (e.g. the application as a book or as a building), navigational paradigms and multimedia elements can increase the quality of our design. "Can" because we must follow some guiding rules in order to get the best of these. The Asymetrix people give us some tips to do that [2]:

- Unify each page around one dominant element;
- Keep the overall design of the application consistent and simple – do not change background layouts or typefaces from page to page;
- Do not design every page to look exactly the same in order to keep user's attention;
- Keep a balance between the topics and the way they are presented, according to the audience and purpose;
- Experiment! Many good designs emerge from lots of revision and input from users.

Concerning navigational paradigms we should use familiar or intuitive models that help users to go through the material according to their needs in light of educational aim. Here are some tips again [2]:

- Make the application sequence and the progress evident – displaying the application/module content on each page when possible and highlighting the current module or section makes progress evident. This is also a good motivational technique;
- Simplify the informational paths – every page should offer to users the option to move forward/backward to the next/previous page or to the home page;
- Use visual anchors – windows, graphics, or text that remain consistent from page to page give users a sense that the pages are arranged in an orderly sequence.

Here we should consider also the issues related to consistent combination of several hypermedia CBL applications that are hyper-linked according to user request.

"Pure" multimedia elements (graphics, sound, video, animation) can add attractiveness and effectiveness to our CBL application if we use them properly. Multimedia is great, but it can become distracting or overwhelming if it is used inadequately. Learners should focus on application content, not on its special effects. In addition to that, it is true that a picture may be worth a lot of words, but we must help our students to know exactly what are those words. However, we must not forget the user profile!

EVALUATING THE APPLICATION

During the last few years a number of research studies of the problems involved in evaluating multimedia and hypermedia courseware have been made. A very useful survey of some of the different approaches that can be used is made by Barker in [3, I1].
The author presents also a questionnaire-based assessment methodology for interactive learning products that have been published on CD-ROM. The factors that form the basis of this methodology are: engagement, interactivity, tailorability, appropriateness of multimedia mix, mode and style of interaction, quality of interaction, quality of end-user interfaces, learning styles, monitoring and assessment techniques, built-in intelligence, adequacy of ancillary learning support tools, suitability for single-user, group and/or distributed use, availability in terms of cost and delivery platforms, outstanding strengths and attractive features and outstanding limitations and weaknesses.

The entries in the questionnaire used in the case study "What makes a good hypermedia package" versus results are presented below (the respondents have been able to use the following options: strongly agree, disagree, no opinion, agree, strongly agree):

The question	*The response*
The spoken word must replace written text wherever possible;	43% - disagree, 2% - strongly disagree
The learner must be able to print textual and graphical information	51% - agree, 23% - strongly agree
The learner must be able to add personal notes and bookmarks	58% - agree, 17% - strongly agree
The learner must be able to copy information to other applications	51% - agree, 7% - strongly agree
Hypertext links distract the learner from the main issues	52% - disagree, 19% - strongly disagree
Hypertext must be available in all multimedia packages	45% - agree, 15% - strongly agree
The facility to locate information by keywords is essential	65% - agree, 30% - strongly agree
Simple animation is more effective than video clips	40%- disagree, 43% - no opinion
Buttons should have labels instead of graphics	48% - disagree, 5% strongly disagree
Graphics must occupy more space on the screen than written text	38% - agree, 10% - strongly agree
Simulation or game play must always be present in an educational package	53% - agree, 17% strongly agree
A touch screen must be used instead of mouse wherever possible	42% - disagree, 36% - no opinion

The results of this test are relevant for what a good hypermedia CBL application should contain. The author concludes that increasing use of the Internet is a good opportunity to conduct modern surveys and evaluative studies. These studies can get full benefits from the fact that an experiment can be "continuously ongoing" and can address to a much higher audience (pertaining to user taxonomy) than ever before. These new advantages are in addition to the usual benefits of electronic evaluation: automation of result collecting, processing, storage and presentation.

Because of their non-linear nature, hypermedia educational materials are often more difficult to evaluate than simple linear multimedia courseware. In [13] the authors suggest that any evaluation of such materials should take into account six basic models: the classical experimental model, the research and development (industrial) model, the illuminative model, the briefing decision-makers (political) model, the teacher and researcher model, and the case-study model. They conclude that there is no single path for the potential evaluator of hypermedia to follow. Therefore, the evaluator must be clear about the goals and objectives of assessment, the available resources, the audience of reports and the decisions that will result from the reports. They add that all evaluations are limited in some way and it is up to the evaluators to take into consideration these limitations.

In order to get good results during the process of evaluation of the quality of a CBL application we must develop it keeping in mind all the above ideas.

The process of evaluation starts with testing whether the application works the way it was intended to or not, both in content and functionality. The testing process starts with checking every object (drag-and-drop objects, hyperlinks, hotwords etc) if they act right in their contexts. It continues with trying every quiz and exercise, verifying the scores for accuracy. We have also to test our multimedia inserts (by playing them in the location they are used) for good comprehensibility and proper consistency with their context. The process ends with some preliminary installing tests.

After testing that the application works the way it is supposed to, we have to make some preliminary tests on application effectiveness. We do so by evaluating a sample group of users taking the application and then providing feedback. This should be applied to several aspects of the application like the instructional approach, clarity, motivation, adequacy, mastery of the content, and multimedia. It is also important to see if users can properly use the achieved knowledge and skills into the real world.

DISTRIBUTING, MANAGING AND MAINTAINING THE APPLICATION

In this final step we start by reviewing our planned delivery options and do what is necessary to implement the ones we choose. After that we have to prepare the “classroom” (it can be a lab or some other place where the educational process take place). We could just improve an existing classroom, by adding the multimedia extensions to it, or we can start from scratch to build such a place. We continue by implementing the application where it will be used (this is not always a trivial step!). It follows the testing of the application in the field.

If everything above goes well, the hard work just starts! It begins now a long-term process of getting user feedback and managing (including maintaining) the CBL application. Because this task is quite difficult we can use some help from a learning management system (like Asymetrix Librarian) that can assist us to manage our CBL applications and to capture the information generated from learners’ day-to-day activity (learner progress, test results, answers to specific questions, the amount of time spent on a particular question, feedback etc.). Using such a tool we can build a structure around the applications that allows the teachers to track the student participation and guide them through the teaching and learning process [2,14]. Systems of this kind also allow the creation of an interactive classroom environment by supporting industry-standard applications for e-mail, threaded discussion groups, and interactive chat. We can conclude that a learning management system provides for building a complete interactive learning environment. And these environments can be locally situated, stand-alone or integrated systems, or globally distributed systems that are located at many different geographical locations. They can be used both as a foundation for individual self-study programs and as a mechanism for the realization of group learning activities.

OPERATING SYSTEMS PACKAGE

From the very beginning of our project we agreed upon the general templates to be used for each class of CBL application. For instance, for our course (the Operating Systems case) the template should contain the following items:

- A first page containing the course title and its authors;
- A table of contents;
- Each chapter must include:
    - Its title;
    - The chapter's objectives;
    - The chapter's content (its modules);
    - The content itself (module after module);
    - The chapter's summary;
    - Final exercises and discussion topics;
- The bibliography.

We can see first page from chapter one of this application in the Figure 1:

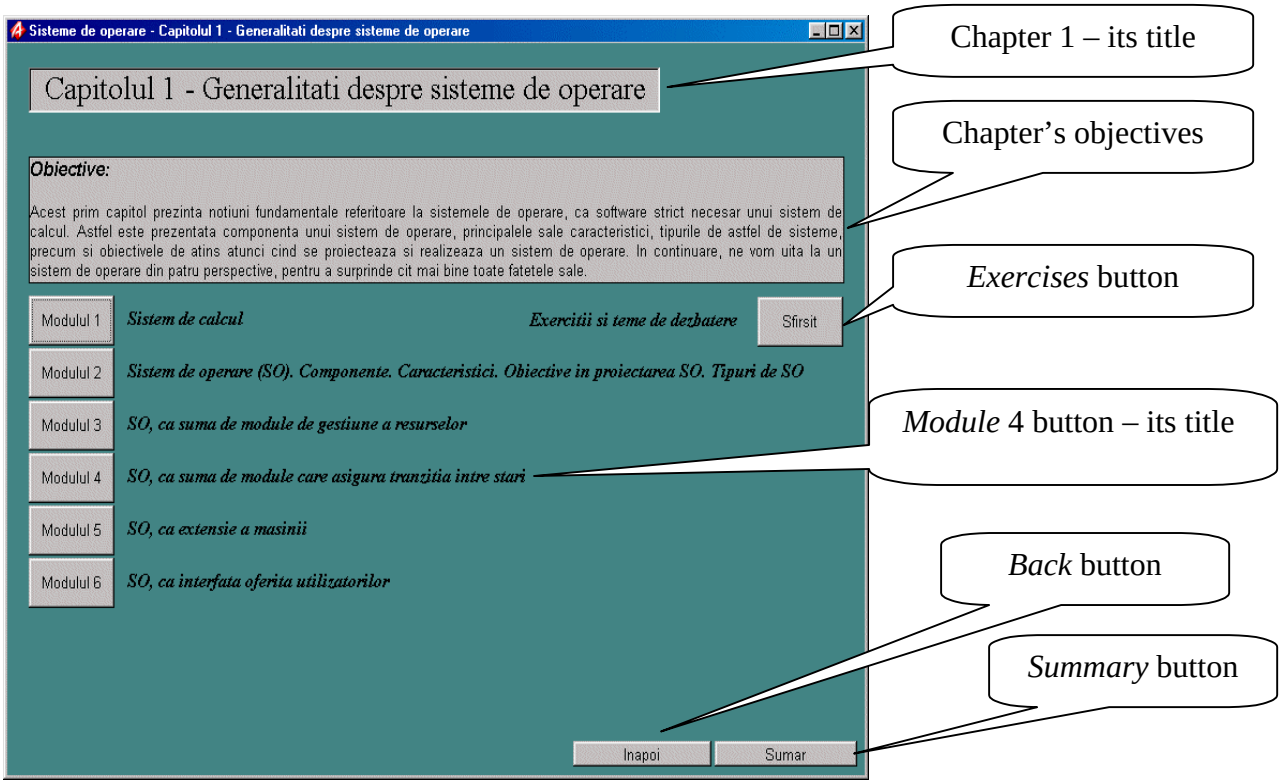

Fig. 1 First page of chapter 1 (with Romanian text)

CONCLUSIONS

The first conclusion is that the use of a methodology for building and distributing our CBL applications is always a wise decision to make. However, no matter how close to that we keep our work, the personal touch of the team working on the project will weigh heavily in the final product. Because building CBL applications is difficult, creative work that cannot be automated completely. So, such a methodology can only assist the team during its work.

We must notice also that it takes time to get the know-how for developing sound CBL applications using all the capabilities of today's technology. But, because we must start from somewhere, we should start with a very simple skeleton and enrich it with interactive elements, step by step. The process is cyclic taking into consideration user feedback. During the process we have to keep the focus on the message of our application and not to forget the technical limitations.

The MIMAL project is still under way in our university. We are now at the beginning of a new stage, the second important stage of our program. In this stage we need to improve the applications we have built in the first step, to develop new applications and to extend their use within our university. In the next step we want to consolidate our learning environment by using a learning management system to manage our CBL applications and the students using them. We also intend to use the courseware we create as a starting point for developing a distance-learning program. This can be a first step through the development of a virtual university environment.

BIBLIOGRAPHY

1. Angeles M.C., Dustdar S., *Multimedia Information Systems*, Kluwer Academic Publishers, 1997
2. Asymetrix Learning Systems, Inc., *The Asymetrix Guide to Interactive Online Learning*, 1999
3. Brusilovsky P., Kommers P., Streits N., Lectures Notes in Computer Science: *Multimedia, Hypermedia and Virtual Reality - Models, Systems and Applications*, Springer - Verlag, 1996
4. Chapman G.M., *Computer Based Learning in Science - CBLIS'97, Proceedings, The Third CBLIS Conference*, The Montfort University, Leicester, UK, 1997
5. Chapman G. M., *Computer Based Learning in Science - CBLIS'99, Proceedings, The Fourth CBLIS Conference*, University of Twente, Enschede, The Netherlands, 1999
6. Ferrari D., Ed., *ACM Multimedia 94 - Proceedings- Second ACM International Conference on Multimedia* - San Francisco, California, ACM Press, 15-20 Oct 1994
7. Furht B., *Multimedia Technologies and Applications for the 21st Century - Visions of World Experts*, Kluwer Academic Publishers, 1998
8. Hollan J.D., Foley J.D., Ed., *ACM Multimedia 97 - Proceedings - The Fifth ACM International Multimedia Conference* -San Francisco, California, ACM Press, 9-13 Nov 1997
9. Fuhr N., Dittrich G., Tochtermann K., *Hypertext - Information Retrieval- Multimedia 1997*, Proceedings HIM '97, UVK UniversitatsVerlag Konstanz, Konstanz, 1997
10. Maybury Mark T., *Intelligent Multimedia Information Retrieval*, AAAI Press, The MIT Press, 1997
11. Nwosu K., Thuraisingham B., Bruce Berra P., *Multimedia Database Systems, Design and Implementation Strategies*, Kluwer Academic Publishers, 1996
12. Sheth A., Klas W., *Multimedia Data Management*, McGraw Hill, 1998
13. TOOLBOOK II Assistant Asymetrix Learning Systems, Inc., *User Guide*, 1999
14. Zelleweger Polle, Ed., *ACM Multimedia 95* - Proceedings- The Third ACM International Multimedia Conference and Exhibition - San Francisco, California, ACM Press, 5-9 Nov., 1995

Monica Vladoiu
PG University of Ploiesti, Department of Computer Science
39 Bucuresti Blvd, 2000 Ploiesti - Prahova, ROMANIA
Email: mvladoiu@yahoo.com