An Object Oriented Data Model for Web-Based Courseware Design

S. RETALIS, V. VESCOUKIS, E. SKORDALAKIS
National Technical University of Athens
Department of Electrical and Computer Engineering
15780 Zographou, Athens-Greece
Tel: ++ 301 7722486-7
Fax: ++ 301 7722519

Abstract : The use of the World Wide Web in education is evident. A lot of web-based courseware (webware) products are being developed. These products are special-purpose software applications that run over the internet. Although the established software engineering disciplines apply to this kind of products as well, the specific domain characteristics and requirements may narrow the design choices of developers and let them focus to more important aspects of the software, such as the learning effectiveness and the pedagogical efficiency. Today there is lack of design methods and guidelines targeted to webware. Such methods could help the developers increase the learning effectiveness of the end product, the efficiency of the design and implementation processes. The object oriented design methods have been proven to be very useful for general-purpose hypermedia applications. However, in the special case of webware, the generic guidelines and freedom of the object-oriented paradigm can put developers into a chaos of choices where valuable effort is easily wasted. The need for domain-targeted object models rises. Such models will emphasise on the characteristics of webware providing an initial architecture of this application under development, allowing developers to put their effort in the most efficient implementation. Such an object model is presented in this paper followed by an example of its use.

Keywords: Object-oriented hypermedia data models, hypermedia in education, web-based courseware development
CSCC'99 Proceedings:-Pages 6011-6017

1 Introduction

One of the most significant characteristics of the new technologies and especially the Internet and the World Wide Web, is the capacity they provide to store, edit, present and retrieve information in an ever increasing variety of ways and forms [4]. Because of this capacity it is now possible to design sophisticated web-based courseware which brings up a new learning paradigm. Learners can access huge amounts of learning resources and information, as well as interact with each other and their teachers over whatever distance or time. The term “Web-based courseware”, or webware, means software that is created for educational purposes and is run over the Internet via the WWW and related technologies (server and client-side scripting, Java applets, COM objects, etc). Webware, is in this context a mosaic of software-based learning resources, such as study guide, didactic book, case studies, self-assessment questionnaires etc, usually written in mark-up languages that incorporate active elements (like HTML with scripting, XML or Java).

The learning resources can be classified into two broad categories:

1. *instructional resources* which are expositive or active hypermedia documents which contain informative and learning material. The expositive hypermedia documents are being typically viewed by a learner without entailing any degree of interaction, other than navigation. On the contrary the active hypermedia resources offer a variety of ways for human-computer interaction for instructional purposes.
2. *communication resources* which refer to hypermedia learning material that triggers the computer mediated synchronous or asynchronous communication among human agents involved in the learning process.

The development of webware is labour intensive and costly. Suitable methodologies should be followed in order to assure the development of effective end-products within efficient development process. Such methodologies will be domain-oriented specifications/extensions to existing software engineering frameworks, that will offer developers guidance in how to analyse, design, implement and evaluate webware.

In this paper we concentrate only in the design process. We present an object model which can serve as a guide or a pattern for constructing a webware detailed design. This model aims to help the designers in structuring the webware learning resources and representing the subject matter in various hypermedia modalities, rendering it clear and accessible to the learning audience.

This work follows the principles of the object oriented hypermedia design methods by [7, 8]. These methods have provided systematic ways to design general purpose hypermedia application but not especially educational ones. However, in the specialised case of the webware applications, the conceptual modeling of the webware application should be built upon a specialised object oriented data model. In order to create the model, the UML notation [2] has been adopted, supported by the utilisation of the Rational Rose case tool [http://www.rational.com].

The structure of the paper is as follows. In section 2, the principles of the object data model are given. In section 3, an example of the model's application is illustrated. This example refers to a webware which has been developed for enriching the traditional classroom-based teaching with open and distance learning methods of the course «An introduction to Compilers». It should be noted that this webware has not been reengineered so as to fit to the data model. On the contrary it was been designed for scratch utilising the data model. Finally, in section 4 we make some concluding remarks about the applicability and usefulness of the model.

2 An Object-Oriented Data Model

An object oriented data model should play the role of a guide in constructing a detailed conceptual schema of the webware providing also some details of navigational aspects. The principles of this model are the following:

- A webware consists of two different types of resources: a) the instructional resources with specific objectives, which include both expositive and active hypermedia documents with either informative or learning material; b) the communication resources which are hypermedia documents specifically created for communication purposes (open questions, videos with annotations, etc.). A webware contains at least one of the above resources, as shown in Fig. 1.

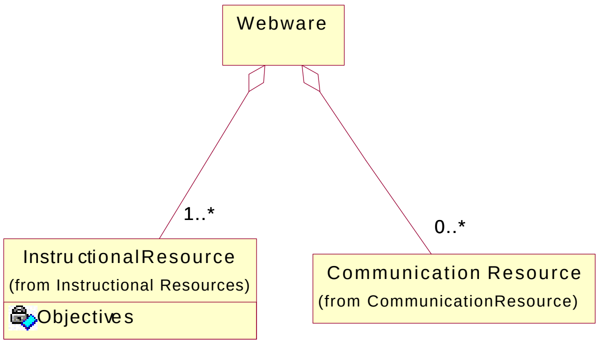


Fig. 1. Webware resources

- Each one of the webware resources contains a number of web pages, as shown in Fig. 2. These webpages include multimedia elements, active elements (scripts, applets, COM/DCOM objects), navigational links (e.g. forward, backward, back to the contents, etc.) and code in a markup language (HTML, XML, etc.). It should be mentioned that the navigational links are subject matter independent. For each link, its name and the linked webpage should be noted.
- The webpages of the instructional resources are divided into the following types, as illustrated in Fig. 2:
  - ♦ *Accesspages*, whose learning content is minimal, used for navigational purposes since they provide access to webware's webpages with learning material. From one access page the user can also navigate to another accesspage (as often happen with the contents of the hypermedia book).
  - ♦ *Contentpages* which include informative or learning material
  - ♦ *Questionnaires* which are webpages that have a unique structure and functionality in the learning process. They contain open or close type of questions.

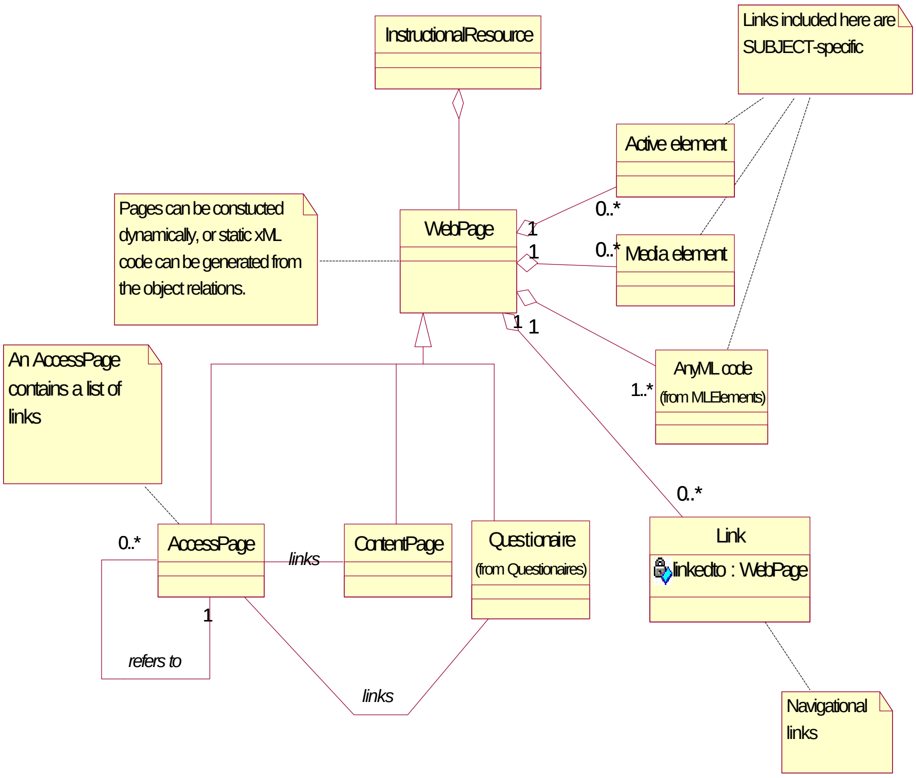


Fig. 2. Detailed conceptual schema of the webware's instructional resources

- The communication resources are divided into synchronous and asynchronous ones as presented in Fig. 3. For each kind of communication resource, the designer must specify the learning material that will be included as well as issues about the nature of the communication such as: a) the *communication system* like video/teleconferencing, web-based mailing list, etc., which is a re-usable software subsystem that differs among implementations of webware systems, and b) details about the *communication event*, which is responsible for handling data about the communication itself such as participants, time and place, etc. The communication system depends on the technological infrastructure of learning environment of which the webware will make use.

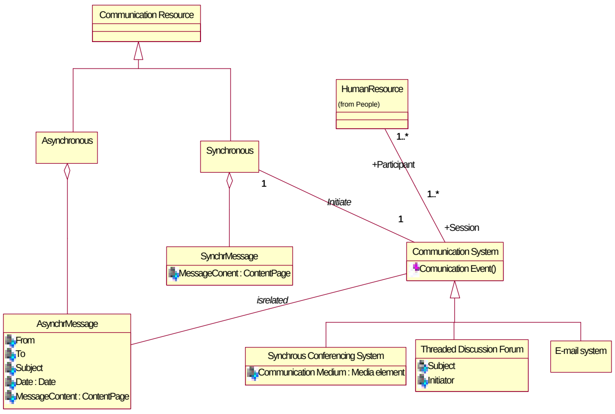


Fig. 3. Structure of the communication resources

- The webpages of the questionnaires are divided into two types: *open questionnaires*, which are contentpages and the *closed type questionnaires* (such as multiple choice, cross references, "fill-in the blanks" exercises, etc.) as shown in Fig. 4. The later ones are used as self-assessment exercises, since there is only one correct answer from a collection of many. For these questionnaires the content provider must write down the question, the right answer and the wrong ones. A class called QuestionRun contains details of the evaluate method (e.g. weights for some questions, negative grade for wrong answers, the appearance or not of the correct answer after the completion of the questionnaire, etc.) as well as data about the assessment procedure, i.e. time, date, etc. The open type of questionnaires contain free-text fields, completed by learners and evaluated by instructors. They implement the classic “homework assignment” notion of traditional learning environments in a more time and place flexible way.

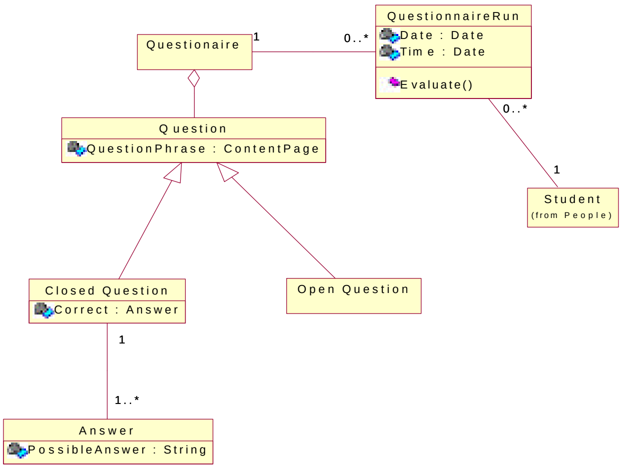


Fig. 4. Structure of questionnaires

- A webpage includes multimedia elements, active elements (like Javascripts), links and code in a markup language (e.g. HTML). For each kind of multimedia element, the designer must specify some characteristics such as author, size, format, description of its content etc., which is valuable for reusability [1]. Moreover, each one of these elements contains embedded Links which are subject specific and do not depend on the general webware navigational schema. For some kinds of multimedia elements, the designer must specify the software system that will handle its presentation, such as RealAudio or RealVideo, or a special plug-in. In Fig. 5, the abstract structure of multimedia elements are illustrated using a class diagram.

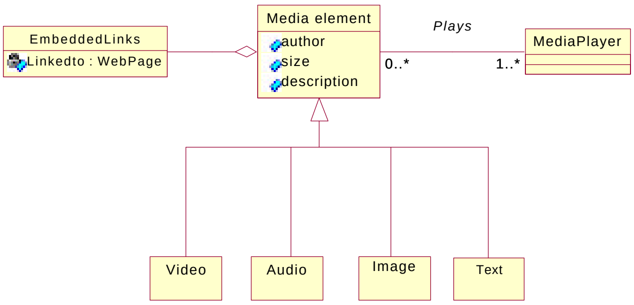


Fig. 5. Structure of the multimedia elements of a webware’s webpage

- Finally, for specifying the nature and behavior of the active elements which are contained in a webpage, the designer should give details of the data model of language that will be used for its development as well as the behavior of the element. For example for a Java applet the Java data model should be used. For a simple CGI script a transition diagram will be enough to show its behavior and guide the developers in its implementation.

It is obvious that a webware application might not entail all the different types of resources, webpages or elements mentioned above. However, this model was conceived to assist designers in their effort in order that a detailed design model of their application can easily be constructed. So, the designer’s main task is to reduce the abstraction level of the generic model presented so far by completely specifying the objects of the various classes referred to in the object model presented so far.

3 An Example Application

In this section we present some parts of the detailed design of the webware that was developed for the course “An Introduction to Compilers” offered by the Software Engineering Lab of the National Technical University of Athens. This webware includes a variety of learning resources such as: a hypermedia didactic book, two case studies, self assessment questionnaires, a collection of past exam papers, a course description, and a study guide. Because of space limitation the illustration of the utilisation of the OO data model will be concentrated in one of the case studies. It contains learning material for giving the students a complete example on how to create a compiler for an educational language called “Russel”. It has the structure of a hypermedia didactic book. It also contains active elements that are being used by learners to test the validity of already prepared examples as well as their own intermediate products during the development process of a compiler (e.g. lexical analysis). The webware is developed using HTML and CGI scripts and it is stored on a webserver that hosts the integrated networked learning environment WebCT [http://www.webCT.com]. The language of the learning material in the current implementation is Greek.

Fig. 6 is an instance of the class diagram of Fig. 1 and it shows the resources of the webware of the case study, including the case study-1 to which we will refer as resourse-CS1 in the sequence. The webware is comprised of 6 instructional resources and three communication ones, which are one close type questionnaire and two discussion topics (by means of electronic conference) for a threaded Web-based forum that was constructed for the purpose of the course).

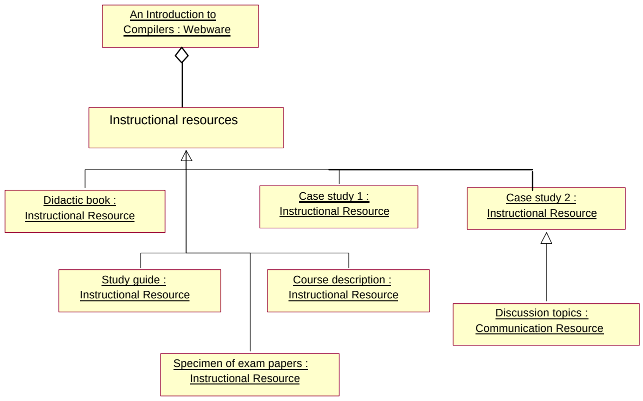

An Introduction to Compilers : Webware
Instructional resources
Didactic book : Instructional Resource
Case study 1 : Instructional Resource
Case study 2 : Instructional Resource
Study guide : Instructional Resource
Course description : Instructional Resource
Discussion topics : Communication Resource
Specimen of exam papers : Instructional Resource

Fig. 6. Structure of the webware “An Introduction to Compilers”

The rest of this section will concentrate on showing the detailed design of the resourse-CS1. As shown in Fig. 7, the resourse-CS1 consists of one accesspage and 27 contentpages. The structure of the accesspage is illustrated in Fig. 8. It contains two images (arrowleft.jpg & arrowdown.jpg), text with links (contents.html) and one active elements that handles the presentation of the subsections of a section. This webpage is written in the HTML language. The active element is a CGI script whose behavior is shown in the state transition diagram of Fig. 9. This script was already provided by the WebCT and the developers just made a reference link to it. The implemented webpage of this accesspage is shown in Fig. 10.

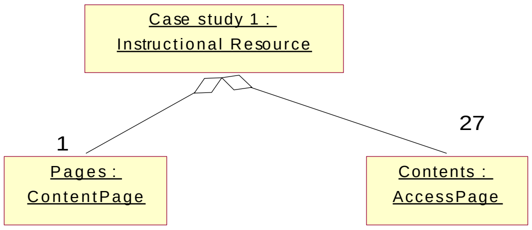


Fig. 7. The structure of the resourse-CS1

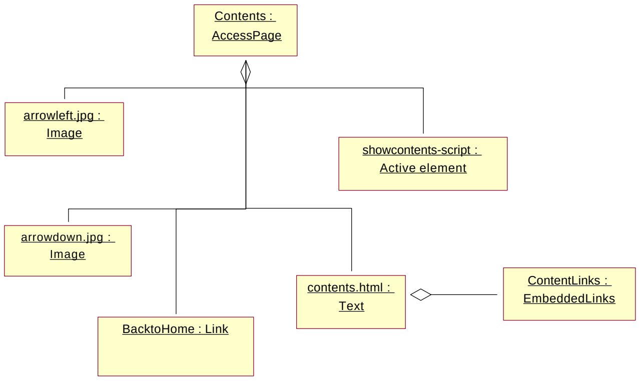


Fig. 8. The structure of the accesspage of the resourse-CS1

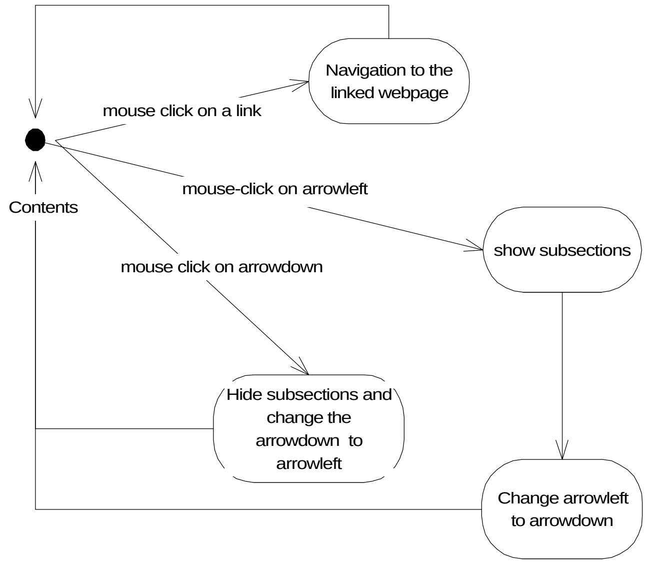
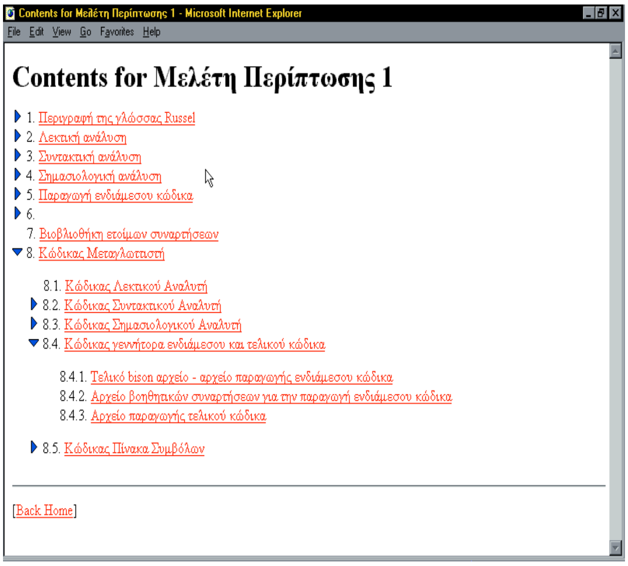

Navigation to the linked webpage
mouse click on a link
Contents
mouse-click on arrowleft
show subsections
mouse click on arrowdown
Hide subsections and change the arrowdown to arrowleft
Change arrowleft to arrowdown

Fig. 9. The state transition diagram which shows the behavior of the active element of the accesspage of the resourse-CS1


Contents for Μελέτη Περίπτωσης 1 - Microsoft Internet Explorer
File Edit View Go Favorites Help

Contents for Μελέτη Περίπτωσης 1

1. Περιγραφή της γλώσσας Russel
2. Λεκτική ανάλυση
3. Συντακτική ανάλυση
4. Σημασιολογική ανάλυση
5. Παραγωγή ενδιάμεσου κώδικα
6.
7. Βιβλιοθήκη ετοίμων συναρτήσεων
8. Κώδικας Μεταγλωττιστή

8.1. Κώδικας Λεκτικού Αναλυτή
8.2. Κώδικας Συντακτικού Αναλυτή
8.3. Κώδικας Σημασιολογικού Αναλυτή
8.4. Κώδικας γεννήτορα ενδιάμεσου και τελικού κώδικα

8.4.1. Τελικό bison αρχείο - αρχείο παραγωγής ενδιάμεσου κώδικα
8.4.2. Αρχείο βοηθητικών συναρτήσεων για την παραγωγή ενδιάμεσου κώδικα
8.4.3. Αρχείο παραγωγής τελικού κώδικα

8.5. Κώδικας Πίνακα Συμβόλων

[Back Home]

Fig. 10. The implemented accesspage of the resourse-CS1

Some of the contentpages of the resourse-CS1 deal with the trials of the intermediate products of a compilers development process. The structure of one of these pages is shown in Fig. 11. This contentpage is also written in HTML. It contains a form which includes a text box where the learner can type his/her programming code and three buttons, as shown in Fig. 12. These button are the following:

- **Submit** which takes the code of the text box, validates it and presents the compiled outcome in a separate webpage. For example in the case of final code generation, the outcome of the validation will be either compilation error messages or the final code in assembly 8086.

- **Reset** which «clears» the data of the text box
- **Example** which will enter in the text box an example of a correct example.

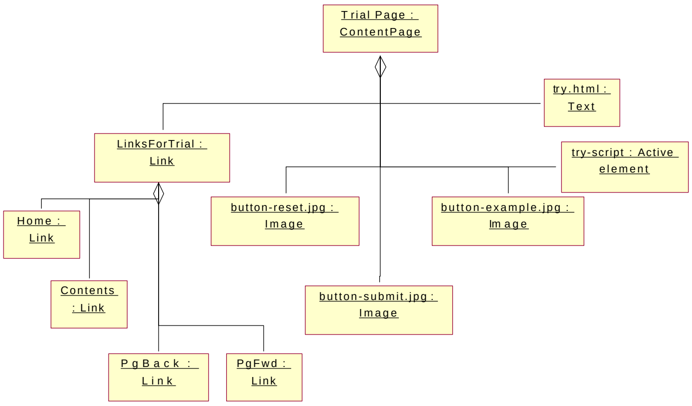
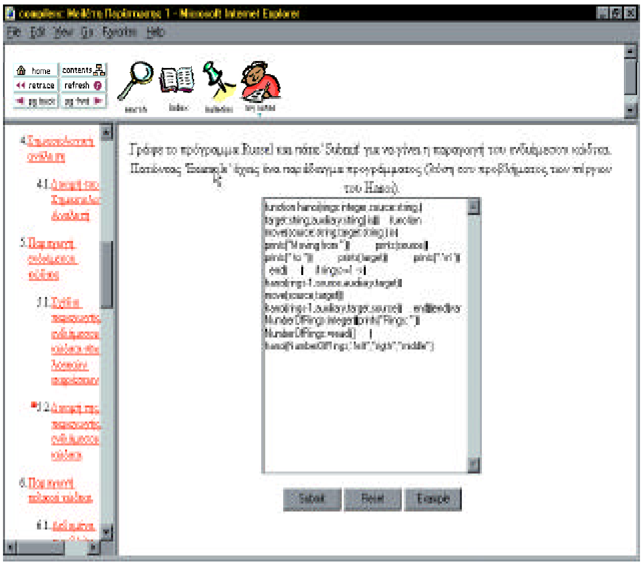


Fig. 11. The structure of a content page of the resourse-CS1 which includes an active element.

The behavior of the CGI script has also been designed using a state transition diagram.


compiler: Μελέτη Περίπτωσης 1 - Microsoft Internet Explorer
File Edit View Go Favorites Help
home
contents
retrace
refresh
pg back
pg fwd
search
index
bulletin
4 Ενδιάμεση γλώσσα
5 Παραγωγή ενδιάμεσου κώδικα
6 Παραγωγή τελικού κώδικα
Submit
Reset
Example

Γράψε το πρόγραμμα Pascal και πάτα 'Submit' για να γίνει η παραγωγή του ενδιάμεσου κώδικα. Πατώντας 'Example' έχεις ένα παράδειγμα προγράμματος (λύση του προβλήματος των πύργων του Hanoi).

Fig. 12. The implementation of the content page designed in Fig. 11.

For the rest of the content pages as well as the webware's resources the same philosophy in applying the OO data model has been used.

4 Concluding remarks

In the last few years there has been a growing interest in hypermedia design methods. A variety of them have appeared [5]. However, none of them is concentrated in the design of web-based courseware applications which is a special type of hypermedia application that becomes more and more popular as the use of www in education is spread. In this paper we presented an object data model which helps designers in structuring the subject domain in a hypermedia fashion.

This model is based on the page metaphor [3] and retains as much simplicity as possible. Effort was made to construct a model that will enhance the reusability and maintainability of the webware components. This model holds such potential but further experimentation and formalism needs to be made. This model was successfully utilised in a webware development project with great success. It has been incorporated into a webware design method, called CADMOS-D, which is one of the methods of CADMOS instructional systems development methodology [7].

Some of the future research plans are:

- further experimentation and validation of the model. At present time, the model is used at the development of a webware application to support on-the-job training in secondary-level technical education [9].
- the construction of a tool to automate the translation of the webware detailed design into a WWW runtime environment.
- the collaboration with other design groups and initiatives such as the IEEE LTSC Learning Object Metadata (p1484) [http://www.manta.ieee.org/p1484/], the consortium of the Ariadne project, in order to formalise the attributes and methods of the abstract classes of the OO data model.

Acknowledgements

Thanks to the Ministry of Education in Greece for funding the development of the web-based courseware which has been used as a case study of this research (Program EPEAEK, project «Undergraduate studies from Distance»). Many thanks to Dr. G. Rossi for providing valuable material about the OOHDM.

References

[1] E. Duval, K. Hendrikx, H. Olivie' (1998). Building Hypermedia with Objects and Sets, *Journal of Universal Computer Science*, Vol. 4 (4).

[2] H. E. Eriksson & M. Penker (1998). *UML Toolkit*, Wiley Computer Publishing.

S. Fraisse', J. Nanard, M. Nanard (1996). Generating Hypermedia from Specifications by Sketching Multimedia Templates, *Proceedings of the ACM Multimedia 1996*, Boston USA.

[4] V. Hodgson (1997). New technologies and learning: Accepting the challenge, *Management Learning*, J. Burgoyne & M. Reynolds (eds), SAGE Publications, London.

[5] T. Isakowitz & M. Thuring (1994). *Methodologies for designing and developing hypermedia applications*, Working paper series, Stern New York University.

[6] D. Lange (1994). An Object Oriented design method for hypermedia information systems, *Proceedings of the 27th Annual Hawaii International Conference on System Science*, January, Hawaii.

[7] S. Retalis (1998). CADMOS: An Instructional Systems development methodology with emphasis on the construction of web-based learning resources, *PhD thesis*, National Technical University of Athens (in Greek).

[8] D. Scwhabe & G. Rossi (1995). The Object-Oriented Hypermedia Design Model (OOHDM), *Communications of the ACM*, Vol. 35 (8), August.

[9] V.C. Vescoukis and S. Relalis. *A hybrid network learning approach for technical education built around on-the-job training*. Proceedings of Networked Lifelong Learning conference, Sheffield, UK, 1998.