Journal of Universal Computer Science, vol. 4, no. 11 (1998), 825-838
submitted: 1/9/98, accepted: 26/10/98, appeared: 28/11/98 © Springer Pub. Co.

Supporting Structured Teaching Material in the WWW[1]

Maria da Graça Campos Pimentel
(Universidade de São Paulo, Brasil
mgp@icmc.sc.usp.br)

João Benedito dos Santos Jr.
(Universidade de São Paulo, Brasil
joao@icmc.sc.usp.br)

Renata Pontin de Mattos Fortes
(Universidade de São Paulo, Brasil
renata@icmc.sc.usp.br)

Abstract: This paper presents aspects of modelling, authoring and presenting structured documents corresponding to teaching material presented in the *World Wide Web*. In this context, it is discussed the importance of providing the formalization of the structure of the documents using SGML. Specifications for structured documents corresponding to didactic texts and questionnaires in SGML are described. Software tools to the authoring and presentation of those documents are also presented in order to manipulate and delivery them in the Web, showing an effective support for structured web teaching material.

Key Words*:* Structured Teaching Material, SGML, RMM, WWW, CSCL, Java.

1 Introduction

The increasing interest in providing teaching material available in the WWW has lead to the development of many HTML documents by teachers and authors of teaching material. Because most tools for generating such documents were built with a generic document model in mind, authors have little support to implement any necessary structure. Important work has been done in terms of providing support to the authoring process of teaching material by casual novice users, a case in point is the *Web Course Tools (WebCT)* developed at University of British Columbia [WebCT 1997]. *WebCT* supports the authoring and delivery of course-material in the WWW providing tools to support, among others, the creation and manipulation of the structure and the index of a course, student progress tracking and student self-evaluation, grade maintenance and distribution, navigation tools and timed quizzes. However, their system is proprietary not only in terms of controlling the access of

[1] This is an extended version of the paper "*Tools for Authoring and Presenting Structured Teaching Material in the WWW*" presented at the WebNet'98 Conference in Orlando, Florida. The paper has received a "Top Full Paper Award".

authors and students, but also in terms of the contents and structure of teaching material information.

Another issue in this area is related to static nature of HTML documents: every student is exposed to the same information, regardless of their experience or background. In order to provide customised content information to students,. a tool for generating adaptive hyperbooks from annotated *RTF (rich text format)* documents is presented by [Brusilovsky et al. 1996].

As the *WebCT* case, however, the format of the documents is proprietary such that only their tools are able to built and explore the document structure and control the presentation. As far as the provision for the interchange of documents containing teaching material is concerned, [Tinoco et al. 1996] report work on modelling quizzes using SGML (*Standard Generalised Markup Language*) [see ISO 1986]. This means that the documents containing a quis is marked so as to conform with a formally defined structure, and as such can be used by any software tool able to process the SGML markup. In their work, quizzes are stored in a server so that they can be accessed and solved by students: the answers are sent back to the server and processed, the achieved score being stored for further processing [see Tinoco et al. 1996]. The advantage of using such a standard in the definition of the structure of documents is that they can be used in any environment supporting that technology, whether they are related to generation, storage or presentation of the information.

In this context, this paper presents a set of tools supporting the authoring of teaching material structured using SGML, as well as the presentation of this material in the WWW. In the next section the modelling of the application domain using a well-known method from the hypermedia literature is shown.

Although the modelling has been made for teaching information domain, particularly didactic material, we could apply the hypermedia design methodology (RMM) for another kind of documents which could be structured for several types of applications. We think that, for example, the domain of Software Engineering documentation produced during the software development process, having several types of structured inter-related documents, could be a good experience for applying our approach. Further investigations will be made, concerning about the requirements which can be considered in order to enable an integration of a DTD construction phase into a hypermedia design method.

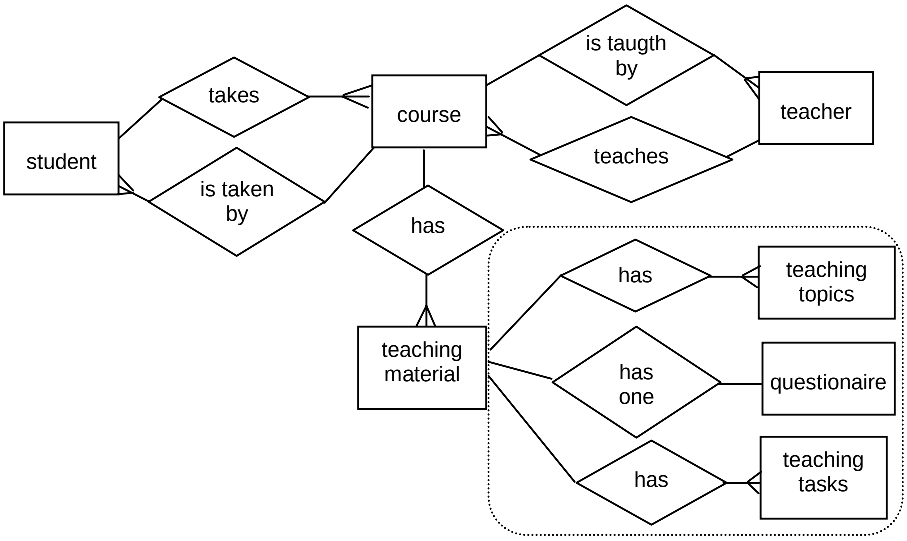
After discussing about the modelling process of the teaching material presented in [Section 2], using simplified SGML notation, [Section 3] formalizes the structure of documents corresponding to generic teaching material and questionnaires, whereas [Section 4] describes tools for authoring and presenting the corresponding documents. [Section 5] presents final remarks and future work.

2 A Model for the Educational Domain

The *Relationship Management Methodology (RMM)* was used to define the initial model for the domain where the teaching tools and associated material is to be applied. The methodology has been built to be used in domains where classes of objects have relationships among them, and where multiple instances of objects are related to each class. As it has already been demonstrated by the proponents of the methodology, the educational domain meets both requirements [see Isakowitz et al 1995].

The first step to model a domain is to identify its components in terms of entities and relationships, using a Entity-Relationship (E-R) diagram. [Fig. 1] presents, outside de doted area, the part of the model having the entities *student, teacher, course* and *teaching material*, related by *(student) takes (course), (teacher) teaches (course)* and *(course) has (teaching material).*

Figure 1: E-R diagram for the target environment



The doted area in [Fig. 1] corresponds to the components of the entity *teaching material*, where the composition is given by the relationship with the entities *teaching topics, questionnaire* and *teaching tasks,* related by *(teaching material) has (teaching topics), (teaching material) has one (questionnaire)* and *(teaching material) has (teaching tasks).*This model is generic so as to be applied to many educational domains, and is in accordance, for instance, to the Hypermedia-Based Learning Environment [Nykäne and Ala-Rantala 1997] and Framework for Hyperbook Design [Fröhlich and Nejdl 1997].

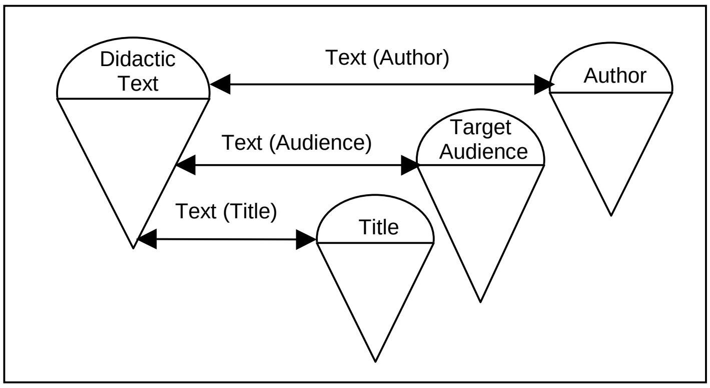
In fact, it is not our purpose to propose a new educational strategy, we aim to help teachers and students, during the use of WWW, to manipulate teaching material. Regarding this information domain, we provide mechanisms to simplify the authoring activity of teaching material performed by teachers, and other mechanisms to guide the users (students) during the navigation activity throughout those available material.

The model in [Fig. 1] was used in two complementary ways. First, the set of entities identified as components of the *teaching material* was associated to software tools to be built in order to facilitate the authoring and presentation of the correspondent material. Second, the whole model was used to provide the storage of the underlying information in a database accessed by users and teachers through any WWW browser.

The next step in the RMM design methodology was performed in order to detail the modelling of each entity that is related to the presentation of information so that the user can navigate it: the presentation of the entities can be done in separate but has the identified relationships as the underlying structure that allows user navigation.

Moreover, the information that an entity *contains* can be divided into meaningful units that can be presented as separated but interrelated wholes [Isakowitz et al. 1995]. In RMM, this is done by defining *slice diagrams*, which present the components of an entity that can be used as navigational elements in terms of the hyperdocuments to be supported. As an example, in [Fig. 2], it is shown the slice diagram of a subset of information available in the *teaching material* entity.

Figure 2: Slice Diagram for Teaching Material entity



The work reported here took a different approach: instead of providing only a graphical slice diagram for each entity to be presented to the user, a SGML-based DTD was built. In the DTD, both the information contained within the entity and the relationships with other entities defined in the E-R diagram are specified, with the further advantage that the information is defined in a well-defined structured language. This is also an advantage over less formal directions as those presented by [Bevirt 1996].

As a result of this approach, DTDs for the entities *teaching material, class topics, questionnaire* where built; two of them are detailed in the next section.

3 Structured Teaching Material and Questionnaire

The SGML standard (*Standard Generalized Markup Language*) was proposed to allow the formalization of the structure of documents stored in an electronic medium. SGML demands that definition of the document structure be performed independently of the presentation form of the document. The advantage of using such a standard is that the documents, stored in some electronic medium, can be processed by any compliant environment, whether it is related to authoring, storage or presentation of such documents.

Each SGML document is associated with a DTD (*Document Type Definition*) file, which formally defines the logical structure of a whole class of documents. A DTD defines the logical structure of a class of documents, and declares: the types of elements that the document can hold; what is the hierarchy relating those elements; the attributes that can be associated with each element. It is an important point that the application that processes a SGML document is in charge of defining how the document instances are to be processed: for instance in order to be presented in a WWW page, printed in an specific format, or stored in a given database.

A DTD defines a class of documents establishing definitions for elements, attributes and entities. To *markup* a SGML document implies that its content is specified in terms of those definitions using SGML initial and end *tags* corresponding to the elements, attributes and entities. Three DTDs were defined in the context of the educational domain presented in the previous section: a DTD for teaching material, *Teaching Material Markup Language - TMML;* a DTD for questionnaires, *Questionnaire Markup Language - QuestML;* and a DTD for teaching tasks, *Teaching Tasks Markup Language – TTML.*

Such definitions not only allow the construction of tools that support those documents, for instance in terms of authoring and presenting, but also guarantees that both documents and underlying structure can be exploited by many other tools such as those in charge of storing and retrieving the documents from a database associated to the whole application (as defined by the E-R diagram in [Fig. 1]). As a consequence document contents can be interchanged with ease.

3.1 The Teaching Material Markup Language - TTML

[Fig. 3] presents a simplified version of a DTD for teaching material, the TTML. A TTML document has elements in three levels, corresponding to the elements *head, body* and *tail.* A *head* element contains a set of mandatory elements: *title, subject, author, level, target audience, abstract* and *author*.

The *body* element is composed by any number of nested *items* and *sub-items* elements, each one having its own *title* part. The *tail* element allows the specification of a list of optional *tasks*, followed by a mandatory specification of the associated *questionnaire*.

Figure 3: Simplified DTD for Teaching Material: TTML

```
<! Simplified DTD for Teaching Material: TTML -- >
<! Elements min content -- >
<!ELEMENT TTML - - (HEAD, BODY, TAIL) >
<!ELEMENT HEAD - - (TITLE, SUBJECT, LEVEL, AUDIENCE, ABSTRACT,
AUTHOR) >
<!ELEMENT BODY - - (ITEM)+ >
<!ELEMENT TAIL - - (TASK*, QUESTIONNAIRE) >
<!ELEMENT TITLE - O (#PCDATA) >
<!ELEMENT SUBJECT - O (#PCDATA) >
<!ELEMENT LEVEL - O (#PCDATA) >
<!ELEMENT AUDIENCE - O (#PCDATA) >
<!ELEMENT ABSTRACT - O (#PCDATA) >
<!ELEMENT AUTHOR - O (#PCDATA) >
<!ELEMENT ITEM - - (TITEM,SUBITEM)+ >
<!ELEMENT TITEM - - (#PCDATA) >
<!ELEMENT SUBITEM - - (#PCDATA|(TSUBITEM,P)+) >
<!ELEMENT TSUBITEM - - (#PCDATA) >
<!ELEMENT P - - (#PCDATA)+ >
<!ELEMENT TASK - - (#PCDATA) >
<!ELEMENT QUESTIONNAIRE - - (#PCDATA) >
<!-- element name attribute name value default -->
<!ATTLIST TTML STATUS (OK,DRAFT) DRAFT
ID ID #IMPLIED >
<!ATTLIST ITEM ID ID #IMPLIED
IDREF IDREF #CONREF >
<!ATTLIST SUBITEM ID ID #IMPLIED
REFID IDREF #CONREF >
```

A set of attributes is also defined. A TTML document must contain attributes for its identification *(id)* and *status* (*ok* or *draft*), whereas the *items* and *sub-items* in the body of the document may contain an identification (*id*) attribute to be used for hypertext links, or refer to other elements identifiers *(idrefs)* in order to establish hypertext links.

3.2 The Questionnaire Markup Language - QuestML

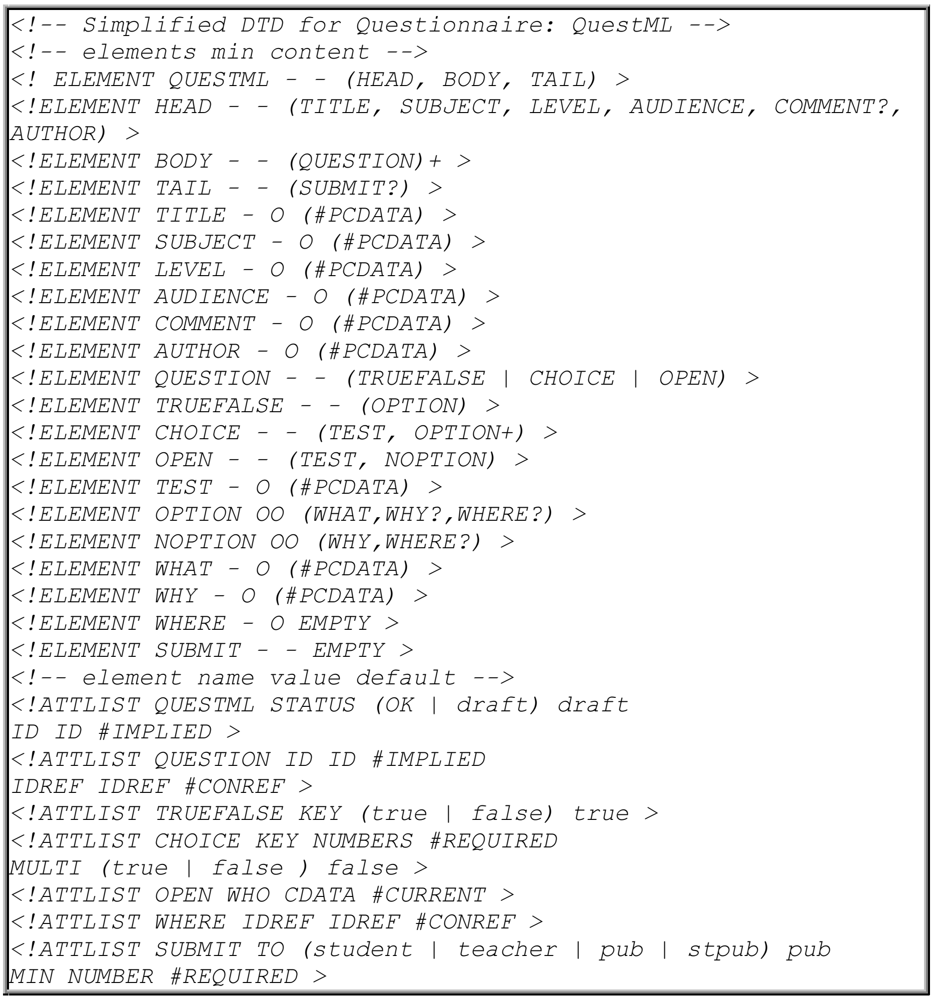
[Fig. 4] presents a simplified version of a DTD for a questionnaire, the QuestML. Similarly to TTML, a QuestML document has elements in three levels: *head, body* and *tail.* A *head* element has all the elements of the TTML *head* but the *abstract* element; an optional *comment* element is allowed instead.

The *body* element is composed by any number of nested *questions.* Three types of *questions* are defined: *truefalse, choice* and *open*:

- a *truefalse* question has a single *option* element, which corresponds to the concept a student has to agree or disagree with.

- a *choice* question has a *test* part and an *option* part. The *test* correspond to what concept is being evaluated, and the *options* part correspond to a set of concepts the user can *select*. Each *option* is composed of a *what* element (the question itself) and optional *why* and *where* elements. The *why* element is supposed to state why the select option is or not correct, whereas the *where* element can indicate, via a hypertext link, a point where that particular subject is discussed.

Figure 4: Simplified DTD for Questionnaire: QuestML



```
<!-- Simplified DTD for Questionnaire: QuestML -->
<!-- elements min content -->
<! ELEMENT QUESTML - - (HEAD, BODY, TAIL) >
<!ELEMENT HEAD - - (TITLE, SUBJECT, LEVEL, AUDIENCE, COMMENT?,
AUTHOR) >
<!ELEMENT BODY - - (QUESTION)+ >
<!ELEMENT TAIL - - (SUBMIT?) >
<!ELEMENT TITLE - O (#PCDATA) >
<!ELEMENT SUBJECT - O (#PCDATA) >
<!ELEMENT LEVEL - O (#PCDATA) >
<!ELEMENT AUDIENCE - O (#PCDATA) >
<!ELEMENT COMMENT - O (#PCDATA) >
<!ELEMENT AUTHOR - O (#PCDATA) >
<!ELEMENT QUESTION - - (TRUEFALSE | CHOICE | OPEN) >
<!ELEMENT TRUEFALSE - - (OPTION) >
<!ELEMENT CHOICE - - (TEST, OPTION+) >
<!ELEMENT OPEN - - (TEST, NOPTION) >
<!ELEMENT TEST - O (#PCDATA) >
<!ELEMENT OPTION OO (WHAT,WHY?,WHERE?) >
<!ELEMENT NOPTION OO (WHY,WHERE?) >
<!ELEMENT WHAT - O (#PCDATA) >
<!ELEMENT WHY - O (#PCDATA) >
<!ELEMENT WHERE - O EMPTY >
<!ELEMENT SUBMIT - - EMPTY >
<!-- element name value default -->
<!ATTLIST QUESTML STATUS (OK | draft) draft
ID ID #IMPLIED >
<!ATTLIST QUESTION ID ID #IMPLIED
IDREF IDREF #CONREF >
<!ATTLIST TRUEFALSE KEY (true | false) true >
<!ATTLIST CHOICE KEY NUMBERS #REQUIRED
MULTI (true | false ) false >
<!ATTLIST OPEN WHO CDATA #CURRENT >
<!ATTLIST WHERE IDREF IDREF #CONREF >
<!ATTLIST SUBMIT TO (student | teacher | pub | stpub) pub
MIN NUMBER #REQUIRED >
```

- an *open* question has a *test* part and a *noption* part. The *noption* element is equivalent to the *option* element without the *what* part, since that the user is supposed to freely reply to the question proposed by the *test* part.

The set of attributes is as follows:

- the whole questionnaire QuestML must specify its identification *(id)* and *status* (*ok* or *draft*).
- each question may contain attributes may contain an identification (*id*) attribute to be used for hypertext links, or refer to other elements identifiers *(idrefs)* in order to establish hypertext links.
- for a *truefalse* question, the attribute *key* can state whether the question is *true* or *false, true* being the default.
- for a *choice* question, the attribute *multi* states whether the question has one or several correct options, and the attribute *numbers* allow the specification of the correct option(s).
- the *where* element has no content because its attribute *idref* is used for defining the associated location via a hypertext link
- the *submit* element is also empty; its attribute *to* indicates the destination of the results of the questionnaire (a teacher, the student himself, or both); whereas the *min* attribute allows the specification of the minimum score to be considered before the result is sent *to* the specified destination.

After the presentation of the DTDs, the next section discusses aspects of using such structures from the point of view of tools for authoring and presenting the related information.

4 Authoring and Presenting TTML and QuestML Documents in the WWW

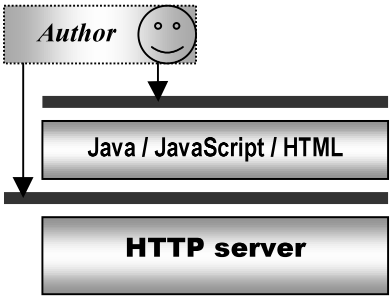
Although a set of document structures as those defined by the TTML and QuestML has many uses in an educational domain, general SGML authoring tools are of little for the authors of the correspondingly documents, both in terms of functionality and, probably, cost. Conversely, pure HTML-based tools are not indicated as the simplicity of the HTML model limits the both content and link structure that can be associated with the underlying document collection.

The aim of the work here reported is not only to facilitate the creation of the structures documents, but also to allow them to be presented in the WWW. Moreover, for the QuestML documents, a user (student) should be allowed to answer the questions while using a WWW browser and receive the associated score.

[Fig. 5] presents a general architecture of the components used in the WWW to present interactive documents: besides preparing the HTML document itself, an author can create any desired JavaScript & Java extension before publishing the document(s) in a WWW server.

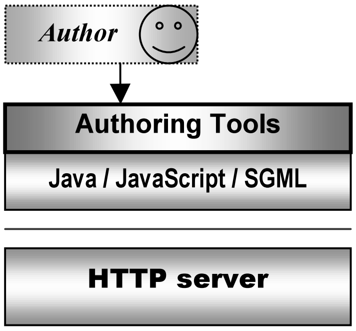
The general approach adopted in this work was to build tools that would help the authoring of the structure of the documents in order that they could, then, be extended with HTML contents using and presented in the WWW.

Figure 5: General architecture of the components used in the WWW. The arrows represent the Interfaces between an author and the languages used in the WWW.



The approach adopted in this work is illustrated in [Fig. 6], where a new level corresponding to the *Authoring Tools* isolates the author from the details of the authoring and publishing activities.

Figure 6: Architecture extended with tools that allow the authoring, publishing and presenting of SGML-based documents extended with Java and Javascript.



4.1 Authoring and Presenting Structured Teaching Material

Hyper*Builder* is a tool that guides the authoring of the elements corresponding to the TTML DTD. The approach is to present to the author the several elements allowed in the document; as the author completes the contents for those elements, a document reflecting the TTML structure is built.

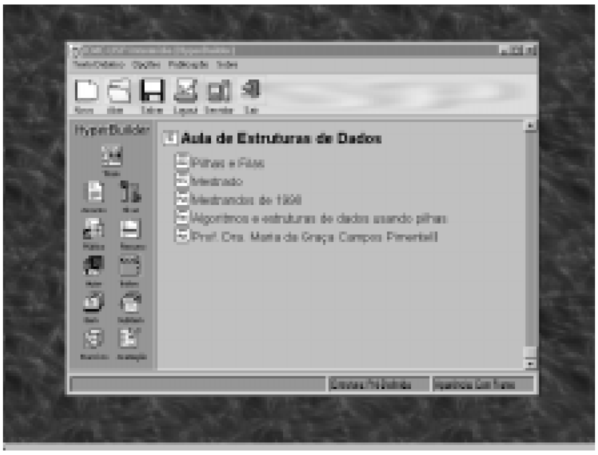
HyperBuilder, illustrated in [Fig. 7], allows an author to include the elements the document, according to the structure defined in the TTML DTD.

First-level elements, such as *title* and *abstract*, have their contents inserted in the main document, while *items* and *subitems* are associated to new files corresponding to lower levels in the hierarchy of the document. At any time, HTML existing contents can be imported to any of the documents in the hierarchy.

When a file is saved, the main TTML document is saved along with a set of HTML related files. The author has an option to presenting the hierarchy using frames or and an index embedded in the main document. Another option allows the HTML files to be published in a WWW server, from where they can be accessed from any web browser, as shown in [Fig. 8].

The HyperBuilder tool has been evaluated in undergraduated courses, through its use by teachers (authors) with different backgrounds: Computer Science and Chemistry. The initial results of the experience have shown that the tool does not inhibit the authoring process, indeed it gives an effective aid to teachers who have their didactic notes previously structured. On the other hand, if the didactic material is not previously structured, the tool helps the task of making structured the teaching material, in the same way didactic material is presented in conventional classes.

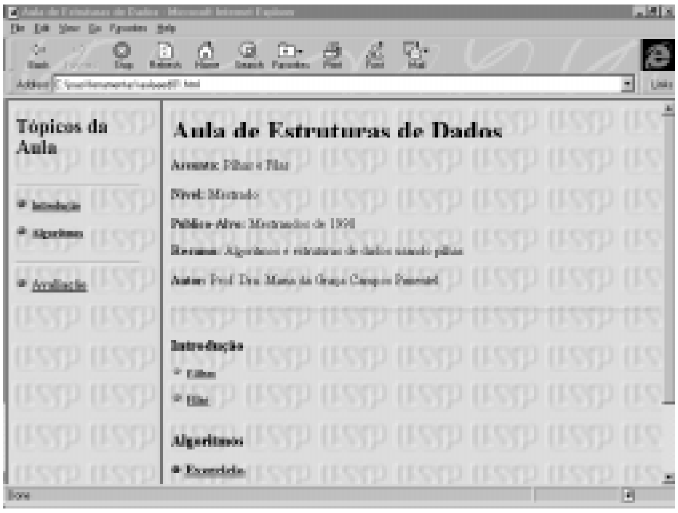
Figure 7: HyperBuilder interface



Aula de Estruturas de Dados

- Pilhas e Filas
- Mestrado
- Mestrandos de 1998
- Algoritmos e estruturas de dados usando pilhas
- Prof. Dra. Maria da Graça Campos Pimentel

Figure 8: A TTML-based document presented in the Web



Tópicos da Aula

- Introdução
- Algoritmos
- Avaliação

Aula de Estruturas de Dados

Assunto: Pilhas e Filas

Nível: Mestrado

Público Alvo: Mestrandos de 1998

Resumo: Algoritmos e estruturas de dados usando pilhas

Autor: Prof. Dra. Maria da Graça Campos Pimentel

Introdução

- Pilhas
- Filas

Algoritmos

- Exercícios

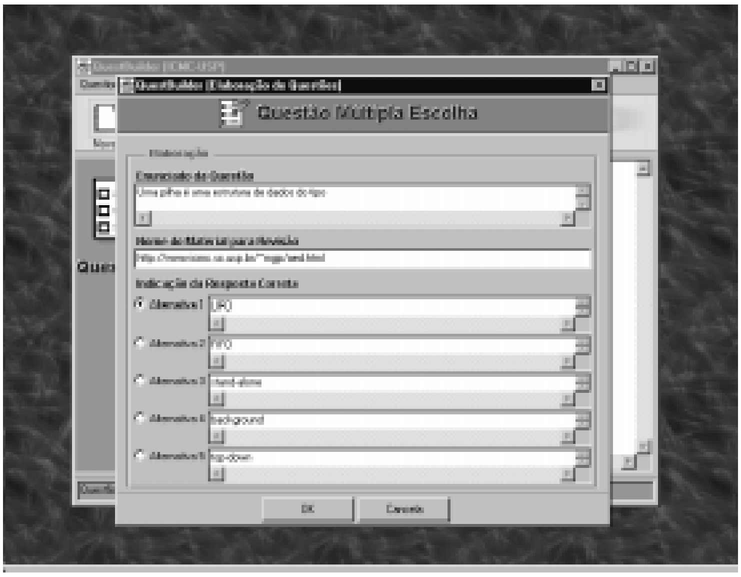
4.2 Authoring, Presenting and Evaluating Structured Questionnaires

[Fig. 9] illustrates the use of the QuestBuilder tool, which allows the authoring of a QuestML document. Similarly to TTML documents, QuestML questionnaire can also be published directly a the WWW server. Because a questionnaire is one of the elements of a TTML document, a user accesses a questionnaire when the link ***test*** is activated in the corresponding TTML document [Fig. 10].

The answers are only shown when a minimum score is achieved, as specified by the MIN element. When open questions are used, the answer is mailed to the person specified by the element WHO in the question. The element SUBMIT will be used when the tools are integrated within the database, in order to keep track of the evolution of the student.

The QuestBuilder tool has also been evaluated in undergraduated courses, by teachers with different backgrounds. Although the task regarding the use of QuestBuilder is very different from that one required in HyperBuilder, the initial results of the experience is similar. The tool does help this authoring process of questionnaires, allowing an effective aid to the teachers who want to structure previous material.

Figure 9: QuestBuilder interface



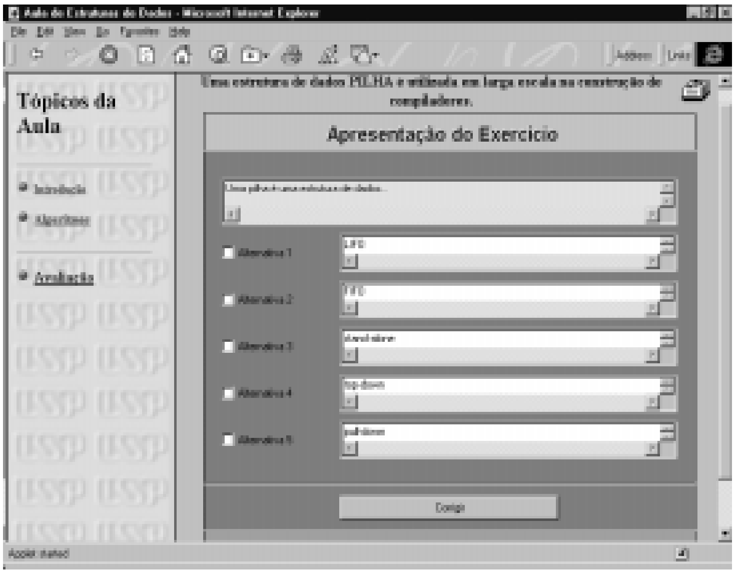
Enunciado da Questão: Uma pilha é uma estrutura de dados do tipo

Nome do Material para Revisão: http://[illegible].icmc.sc.usp.br/~[illegible]/aed.html

Indicação da Resposta Correta

- ● Alternativa 1: LIFO
- ○ Alternativa 2: FIFO
- ○ Alternativa 3: stand-alone
- ○ Alternativa 4: background
- ○ Alternativa 5: top-down

Figure 10: A QuestML-based document presented in the Web


Aula de Estruturas de Dados - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address
Links
Tópicos da Aula
Introdução
Algoritmos
Avaliação
Corrige
Applet started

Uma estrutura de dados PILHA é utilizada em larga escala na construção de compiladores.

Apresentação do Exercício

Uma pilha é uma estrutura de dados...

- □ Alternativa 1: LIFO
- □ Alternativa 2: FIFO
- □ Alternativa 3: stand-alone
- □ Alternativa 4: top-down
- □ Alternativa 5: pull-down

5 Final Remarks

Gaines remarks that the growth of the Internet and the WWW, and the evolution of their underlying technologies, may contribute to the foundations of the *knowledge science* [Gaines 1996]: there is no doubt that exploiting such an environment in the educational domain may bring many contributions. Important work has been done in terms of providing environments where teachers and students can interact, produce and navigate through course-related documents.

This paper has presented undergoing work aimed at building an environment that supports teaching and learning activities in the WWW. The highest level of the entities and their relationship have been presented. An important step was taken in terms of creating document type definitions (DTDs) from the entities and relationships identified. Such mapping, which has guided both the construction of the related authoring tools and defined presentation and navigation structures, has not yet been reported in the literature.

The fact that the Java language allows fine interaction and computation in the browser has been exploited in this tool. Since the correct answer is part of the content of test-based questions, these can be automatically evaluated when a student finishes a questionnaire.

Specifications for structured documents containing teaching material and questionnaire have been discussed, tools associated to the authoring of those documents presented, and the approaches for presenting the documents indicated. At the time of this writing, the environment that will integrate the documents stored in the WWW server with the remaining information related to the courses [Fig. 1] is under construction.

The work reported has advantages over other reported in the literature because: (a) the documents supported are structured according to a standardized language; (b) the clients and servers are freeware and platform independent; (c) the tools for authoring and presentation built are also platform independent since Java has been used in all implementations.

The next steps of this work include: (a) concluding the tools allowing the storage and retrieval of the documents and other related information from a database; (b) experimenting the tools with novice and casual teachers and students; (c) exploring alternative linking structure within the hyperdocuments, as proposed in [Pimentel and Buford 1996]; and (d) further investigating the integration of a DTD construction phase in hypermedia design models.

References

[Brusilovsky et al. 1996] Brusilovsky, P.; Schwarz, E. & Weber, G.: "*A tool for developing adaptive electronic textbooks on WWW*"; Proc. WEBNET'96.

[Bevirt 1996] Bevirt,B.: "*Designing hypertext navigation tools*"; Proc. WEBNET'96.

[Fröhlich and Nejdl 1997] Fröhlich, P.; Nejdl, W.: "*A database-oriented approach to the design of educational hyperbooks*" Work. Intelligent Education Systems on the WWW, 1997.

[Gaines 1996] Gaines, B.R.: "*Convergence to the Information Superhighway*" Proc. WEBNET'96.

[Isakowitz et al. 1995] Isakowitz, T. et al.: "*RMM: a methodology for structured hypermedia design*" CACM v.38, n.8, 1995.

[ISO 1986] ISO/IEC IS 8879. Information processing - text and office systems - "S*tandard Generalized Markup Language (SGML)*", 1986.

[Nykäne and Ala-Rantala 1997] Nykäne, O.: Ala-Rantala M. "*A design for hypermedia-based learning environment*" Http://www.tut.fi/~onykane/papers/hci-et/hble.html.

[Pimentel and Buford 1996] Pimentel, M.G.C. & Buford, J. K.: "*Non-invasive improvements to hyperlinking with HTML*" Proc. WEBNET'96.

[Tinoco et al. 1996] Tinoco,L.C. et al.: "*Quizit: an interactive quiz system for www-based instruction*" Proc. VII Simpósio Brasileiro de Informática na Educação, Belo Horizonte, MG, 1996.

[WebCT 1997] "*World wide web course tools*" Http://www.webct.com/webct, University of British Columbia, 1997.

Acknowledgements

This work is partially supported in Brazil by CNPq/ProTeM-CC/SMmD grant #680077/94-4. João Benedito dos Santos Jr. is partially supported by CAPES.