DOCUMENT RESUME

ED 476 979 IR 021 707

AUTHOR	Buendia, F.; Diaz, P.; Benlloch, J. V.
TITLE	A Framework for the Instructional Design of Multi-Structured Educational Applications.
PUB DATE	2002-06-00
NOTE	7p.; In: ED-MEDIA 2002 World Conference on Educational Multimedia, Hypermedia & Telecommunications. Proceedings (14th, Denver, Colorado, June 24-29, 2002); see IR 021 687.
AVAILABLE FROM	Association for the Advancement of Computing in Education (AACE), P.O. Box 3728, Norfolk, VA 23514. Tel: 757-623-7588; e-mail: info@aace.org; Web site: http://www.aace.org/DL/.
PUB TYPE	Reports - Research (143) -- Speeches/Meeting Papers (150)
EDRS PRICE	EDRS Price MF01/PC01 Plus Postage.
DESCRIPTORS	Computer Assisted Instruction; Computer Software Development; Courseware; Educational Media; *Educational Technology; *Instructional Design; Instructional Development; *Instructional Materials; *Material Development; Models; *Programmed Instructional Materials

ABSTRACT

An instructional application consists of a set of resources and activities that implement interacting, interrelated and structured experiences oriented towards achieving specific educational objectives. Computer-based instructional applications have to be looked at as any other development activity following a well defined process. With this purpose in mind, some design methods for computer-based instructional applications have been proposed. However most of them are focused on on-line courseware structures, which are quite rigid in that they have serious shortcomings in dealing with courses structured in multiple ways. Moreover, these methods usually lack a specific mechanism to model instructional concepts and strategies. This paper proposes a design framework to develop multi-structured instructional applications combining a didactic model with a software engineering approach to deal with educational and technical requirements. The underlying model extends knowledge structures, such as those involved in the Merrill's Transaction Instruction Theory, adding to them didactic information. It also considers the functional aspects of these structures. An XML-based notation is proposed to represent such structures and their management. Includes five figures. (Contains 11 references.) (Author)

A framework for the instructional design of multi-structured educational applications

F.Buendía, P. Díaz*, J.V Benlloch
Escuela Universitaria de Informática Universidad Politécnica de Valencia.
46022-Valencia (Spain)
mailto:{fbuendia, jbenlloc}@disca.upv.es
*Departamento de Informática.Universidad Carlos III de Madrid
28911 Leganes (Spain)
mailto: pdp@inf.uc3m.es

Abstract:

An instructional application consists of a set of resources and activities that implement interacting, interrelated and structured experiences oriented towards achieving specific educational objectives. Computer-based instructional applications have to be faced as any other development activity following a well defined process. With this purpose some design methods for computer-based instructional applications have been proposed. However most of them are focused on "on-line" courseware structures which are quite rigid as far as they have serious shortcomings to deal with courses structured in multiple ways. Moreover, these methods usually lack a specific mechanism to model instructional concepts and strategies. This work proposes a design framework to develop multi-structured instructional applications combining a didactic model with a software engineering approach to deal with educational and technical requirements. The underlying model extends knowledge structures, such as those involved in the Merrill's Transaction Instruction Theory, adding them didactic information. It also considers the functional aspects of these structures. An XML-based notation is proposed to represent such structures and their management.

1. Introduction

An instructional application can be defined as a set of resources and activities which implement interacting, interrelated, structured experiences that are designed to achieve specific educational objectives. Instructional applications are usually structured by means of static patterns which are based on a sequence of book-like electronic pages. They are designed using courseware methodologies that focus on presentation and navigation issues but lack a didactic basis. The resultant products are mostly "pretty-printing" on-line courses but didactic aspects are hardly considered. The current work proposes a design framework which is independent from computer-based delivery technologies and it allows the representation of multiple didactic structures from an instructional point of view. Such a framework requires a model to deal, on one hand, with instructional design topics and, on the other hand, with the design of computer-based applications.

Instructional design can be defined as the discipline that connects descriptive theories with instructional practice. Among the different instructional theories proposed in the literature, we will assume the Instructional Transaction Theory (Merrill 1996) since it makes possible to figure out the relationships between educational and technical components. Merrill's mental models consist of two major components: knowledge structures (schema) and processes for using this knowledge (mental operations). The Merrill's hypothesis states that adequate instruction would require multiple types of knowledge structures to be identified and made explicit to the learner. Thus, the Instructional Transaction theory provides a powerful tool to structure the knowledge about a given topic and to define the procedures for accessing it. In the framework proposed in this paper, Merril's model is extended to add didactic information to knowledge objects and structures. A Didactic model is built using the previous entities as the basis and assigning them attributes whose values suggest how they can be used or even adapted in a specific learning or teaching scenario.

On the other hand, the proposed framework deals with the design of educational computer-based applications. Most of the existing methodologies are addressed to author courseware using formats such as HTML documents. They seem adequate for presenting and accessing information in a open environment like the Web but lack a data structuring capability. Another option is using proprietary multimedia authoring tools like Authorware™ or

Tooolbook™ that provide a more powerful data model but they bound the reuse and exchange of content module (Wiest, 2001). In this sense, an important effort has been made to organize and to manage educational data using metadata (Duval 2001). However, standard metadata proposals mainly aim at the reuse and exchange of learning material and they are not directly involved in the context of instructional application design. In the current work, notational systems based on metadata are used to represent instructional and didactic entities. The proposal consists of defining an XML-based notation which specifies the Merrill's Instructional Transaction Theory entities and their extension introduced in our Didactic model. Such notation will allow an instructor to define his own didactic structures which can be built from a common repository of instructional objects.

The remainder of the paper is organized as follows. Second section revises some related works. Third section introduces the design framework proposed in the current work. Such framework is based on a didactic model whose structural components are presented in fourth section. Fifth section complements this model with its functional view. An application example is described in section 6. Finally, section 7 presents some remarking conclusions.

2. Related work

There are several technology-based educational initiatives that propose a separation between didactic aspects and content related issues. In this context, the Multibook project (Steinacker et al. 1999) considers two domain spaces: the Concept Space and the Media Brick Space. The first one contains a network of knowledge topics which are connected via semantic relations. The second one contains information units of various multimedia formats. Media Brick elements are linked to the Concept entities and instructional mechanisms such as "example", "deepen" or "explain" are setup between these elements. A similar approach is proposed in LMML (Süß et al 2000) which differentiates Pedagogical and Instructional properties from the Module and Content objects. However, only strategy attributes with possible constant values "beh" (behaviouristic) or "con" (constructivistic) can be specified. Other proposals such as the Targeteam project (Teege 2000) or the Palo language (Rodriguez et al 1999) are also constrained to relations or mechanisms such as motivation, illustration, exercise or explanation without an additional didactic value.

In metadata contexts, there are Educational Modeling Languages (EML) which allow the specification of many kinds of educational data. These data can be implemented using notations which take advantage of standard proposals on one hand, or particular and specific proposals on the other hand. The first option is used in the Multibook project to define the Media Brick elements from IEEE Learning Objects or the Chameleon project and its TeachML (Wehner 2001) notation that is based on IMS standards. These standard formats are useful for exchanging them in different learning contexts but their didactic attributes are very restrictive. In the second category, there are formats such as LMML that defines instructional ContentObjects which contain media units such as tables, lists, images or text. The problem is that these units are strongly coupled with instructional objects and it prevents to assign them with different media objects. Similar problems are found on other EML proposals such as Targeteam contents, EML learning objects (Koper 2001) or Palo elements.

3. Design of Instructional Applications

Figure 1 shows the global architecture in the proposed instructional application design framework. The upper level deals with Instructional Design Theory which manages entities such as "knowledge objects", "knowledge structures" and "transaction shells". According to Merrill, a knowledge object is defined as "a precise way to describe the content to be taught". Knowledge objects can be combined into knowledge structures. Knowledge structures are external representations of knowledge that are parallel with mental models that in turn are internal (cognitive) representations of models. Transaction shells consist of rules for selecting and sequencing knowledge objects. The entities defined in the Merrill's model are the basis for the next level (Instructional Design Modeling). This central level is characterized by a Didactic Model whose components extend the previous entities with didactic information. They are divided in two categories that represent the structural and functional model, respectively. Structural model is composed by instructional objects, derived from knowledge objects, and didactic structures which extend knowledge structures. Functional model is based on instructional tasks and learning scenarios. An EML notation is being designed to specify both structural and functional components. Next section will describe them in depth. The lower two levels deal with the computational implementation of instructional applications. The first one is based on a hypermedia model to represent the instructional entities and their relationships in an formal and abstract notation (Buendia et al 2001). The second level is related to the technology involved in the delivery of instructional applications using an e-learning environment.

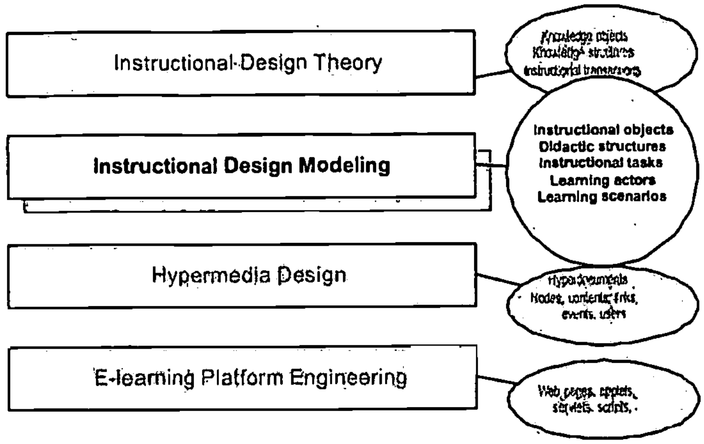

Figure 1. Instructional Application Design Architecture.

4. Structural model

This section focuses on the structural components of the Didactic Model proposed in this paper. The simplest element of the current model is the *Instructional Object* (IO) which is used to manage Merrill's knowledge objects from a learning point of view. Instructional Transaction Theory describes knowledge in terms of three types of knowledge objects: entities, activities and processes. When an entity, activity or process is addressed in a learning context, there are multiple aspects that differ depending on the learning conditions around the knowledge object. The main goal of IO elements is to represent such aspects. If the learning target is an entity, IOs such as definitions, statements or examples can be used to describe it. Activity knowledge items can be characterized using IOs such as exercises, questions, analogies, hints an so on, and finally, processes are also qualified with simulations, feedback elements, or animations.

IOs are not only isolated information units such as definitions or exercises but they are also concerned with the way these units are used in a learning context. For instance, an entity definition is assigned with a difficulty level, a statement declares a property value assigning it a relevance index and an example displays the entity with a specific portrayal (text, audio, image, and so on). Therefore, IOs must consider the multiple possibilities for a knowledge object to be learned in order to deal with different learning styles and needs.

The model presented in this paper provides a flexible framework to incorporate multiple kinds of didactic information which can be added to IOs. The instructor has to decide the different difficulty or abstraction levels assigned to each object, its relevance in a certain learning context or the portrayal that requires a given user profile. This design process is highly related to the features of a subject domain and it becomes a manual and laborious process. Tools as metadata notations have been used to support this process.

In this work, there is a simple XML notation proposal and the definition of IOs is adapted according to the subject domain. This means that they can be defined on top of other entities such as Merrill's knowledge objects or IEEE learning objects, extending them with didactic elements as those referred in Figure 2. In this example, a *Description* IO is used to teach the "magnetic disk structure" (knowledge object). The "ObjectDescription" shows that an image representation has been selected in the current teaching context. Didactic parameters as the portrayal configuration or the abstraction type indicate how this image is displayed and the level of abstraction it represents. EML notations usually also include information about the organization of the educational contents in different structures. In some cases like the Multibook project this information is mixed with the own instructional objects. In our model, there is a strong separation between both types of information.

Instructional objects are organized using *Didactic Structures* (DS) which can be managed as independent entities. These entities are addressed to capture the didactic relationships between those objects. There are two kinds of didactic relationships in the current model: explicit and implicit. Explicit relationships link IOs using a specific action, for instance, "an example *illustrates* a concept definition" or "a question *evaluates* an explanation". Implicit relationships are derived from the way knowledge items are organized.

```
<ComponentValue DidacticRef="Image reference">
        <ObjectRef>Magnetic Disk Structure reference (storage area).</ObjectRef>
        <InstructionalObject ObjectID="IO-404" ObjectName="Description" ObjectDescription="Bi-dimensional image">
                <Portrayal Dimension="2" Background="false" Position="center"/>
                <Abstraction Type="structural" Depth="0"/>
        </InstructionalObject>
</ComponentValue>
```

Figure 2.- Instructional object example.

In this case, we are interested in knowledge structures coming from the Instructional Transaction Theory which are used in the current work to model DSs. Merrill mentions different types of knowledge structures such as lists, taxonomies, dependencies, algorithms and causal nets. A DS can be built on top of one or more knowledge structures. For instance, the description of a magnetic disk in a computer system can be based on identifying its components and assigning them a portrayal configuration. It can also include an algorithm representation to show the access to a specific component (e.g. a cylinder). The DS connections with other model entities are represented on Figure 3 It shows a simplified UML diagram which specifies that DSs are entities aggregated from instructional objects (IO) and knowledge structures (KS). Both IO and KS entities are based on knowledge objects (KO) which are composed by elements such as *Identif, Portrayal* and *Properties*. DS entities also extend the basic information coming from IO and KS, using didactic attributes such as the portrayal selection.

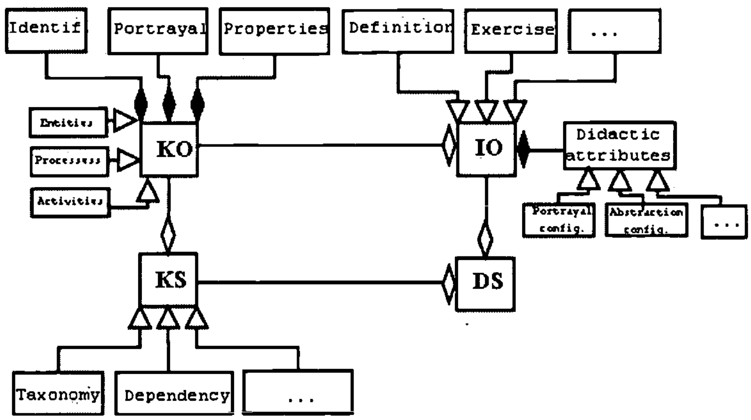

Figure 3.- Structural Design Model entities.

The current work is closer to the IMSDL proposal (Silberhorn & Gaede 1999) which defines instructional strategies, responsible for structuring the information units to be learned but independent from the subject domain. However, these structures are addressed mainly for courseware. Nevertheless, we do not know any proposal focused on generating structure templates which can be applied in specific didactic contexts in a reusable and modular way. We are developing XML Schemas to represent a wide range of didactic structures from basic knowledge structures.

5. Functional model

The functional model describes the entities that allow a user (student or instructor) to interact with DSs. DS entities are the nexus with the other structural elements. Figure 4 shows a simplified UML diagram that represents the main functional entities and their relationships with the remainder entities. These entities are learning scenarios and instructional tasks, respectively.

Learning scenarios (LS) are defined as the set of terms and conditions that characterize the user learning. Each individual user or group of them is assigned with one or more LSs. Instructors have to configure the LS entities, assigning features such as learning modes, timing schedules, instructional methods and learning goals. Each LS aggregates one or more *Instructional Tasks* (IT). They are defined as the operations a user has to perform to achieve a specific learning goal. IT entities model the functions associated to the didactic structure interface. They can access to one or more didactic structures and a certain DS can be attached to multiple ITs.

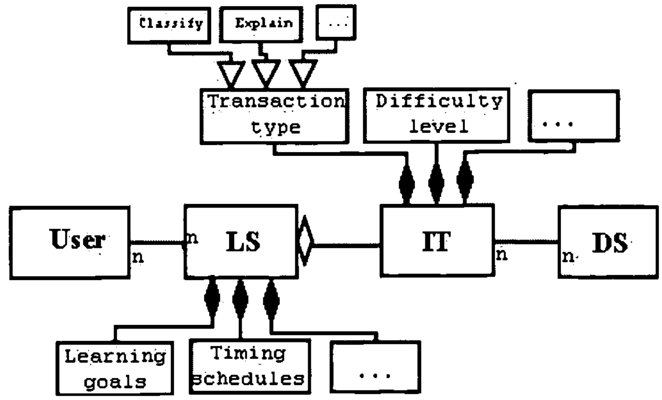

Figure 4.- Functional Design Model entities.

ITs are related to the Merrill's concept of "transaction shell" which consists of rules for selecting and sequencing knowledge objects. In the current model, IT entities do not access directly to IOs which are encapsulated into the didactic structures. This feature eases the design process because the instructor deals with IOs in a more abstract way. For instance, in a dependency DS, the IO components are all considered as chain elements without differentiating them. An IT like *Explain* specifies the navigational operations through the branches of the Dependency structure. A navigational operation can consist in the selection of a given branch and it could depend on learning modes such as “learning by examples”, “by doing” or by "exploration and experimentation". In the first case, the *Explain* task is based on revising *Example* IOs while the other cases involve the working with *Activity* IOs. One of the main instructor responsibilities is to define the ITs attached to each DS. This definition is also a manual and laborious task and an EML notation is being developed to assist the instructor in this process.

6. Application example.

In this section, an example is used to show the application of the proposed Didactic model. The example consists of a Didactic Guide called "Learning XML basics" which is based on a Dependency structure. Figure 5 shows a diagram of top-down dependency. The root node represents the main learning goal described as "Learning XML". The descendent node has assigned a *Definition* IO which intends to answer the "What is XML" question. The didactic relationship between the *Goal* and *Definition* IOs is an *IsBasedOn* relationship type and it means that the knowledge about the XML notion (current node) is required to meet the goal defined in the previous node. From the current node, there are several branches which represent extensions (*IsExtendedBy* relationships) of the current definition node. These branches store requirements in order to understand the previous node and each one has assigned a specific competency (difficulty) level. According to this assignment, a different IT can be attached to each branch. These competency levels are based on the learning modes cited previously. Learning by examples can involve identifying instances of a concept, e.g. an XML document (left branch) or a DTD document (central branch). Moreover, each XML document example can be assigned with a certain difficulty level which is used as parameter by the IT. It causes that the structure branches can be navigated-accessed in different ways. A *Predict* IT can check a specific process, e.g. the validation of an input XML document using a DTD syntax. This validation process is based on an *Activity* IO which is related to a previous *Example* object by means of an *IsCheckedBy* didactic relationship. In a higher level learning mode, the *Predict* task could include additional activities such as the DTD configuration.

7. Conclusions

A design framework has been proposed for developing instructional applications beyond the rigid courseware structures that underlie the typical Web-based courses. The proposed framework is based on a didactic model which provides a gateway between instructional theories and concepts, and hypermedia and Web application engineering. This model considers two issues: a structural view based on instructional objects and didactic structures, and a functional view that manages the previous entities using instructional tasks and learning scenarios. The didactic model is supported by an XML-based notation which eases its translation to computing environments.

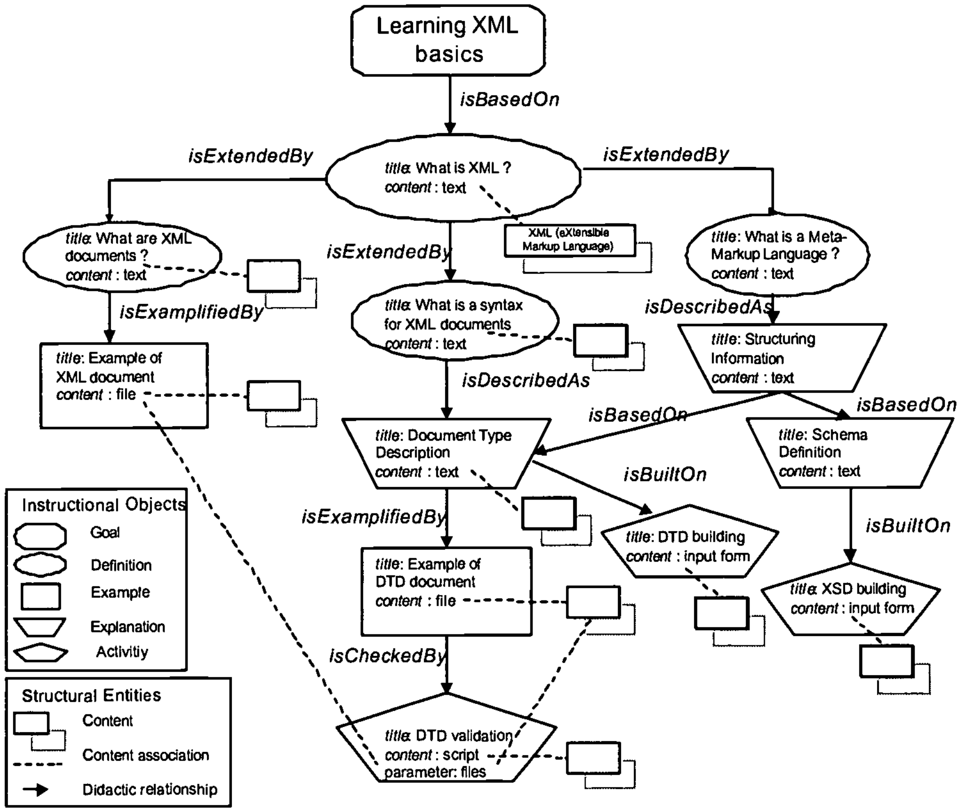

Figure 5.- Didactic Structure example.

This notation has been applied to represent taxonomy-like didactic structures. It provides the possibility to organize available educational resources in a way closer to the instructor teaching requirements. We have also planned to extend the XML-based specification to other didactic structures based on dependencies and algorithms. Further works include their usage in specific learning contexts such as Electric and Information Engineering areas. We are also developing a tool to allow the access to these structures in a Web-based environment.

References

Merrill M. D., & ID2 Research Team (1996). *Instructional Transaction Theory: Instructional Design based on Knowledge Objects.* Educational Technology, 36(3), 30-37.

Wiest, S, Zell A. (2001) *Improving Web-Based Training Using an XML Content Base.* Proceedings of ED-MEDIA 2001, Tampere, Finland, pp-2045-2050.

Duval, E (2001). *Standardized Metadata for Education: a Status Report.* Proceedings of ED-MEDIA 2001, Tampere, Finland, pp-458-463.

Steinacker,A., Seeberg, C., Reichenberger, K., Fischer, S., Steinmetz, R. (1999) *Dynamically Generated Tables of Contents as Guided Tours inAdaptive Hypermedia Systems.* Proceedings of ED-MEDIA June 1999.

Süß,C., Kammerl, R., Freitag, B.A (2000). *Teachware Management Framework for Multiple Teaching Strategies.* Proceedings of ED-MEDIA 2000, World Conference on Educational Multimedia, Hypermedia & Telecommunications, Montreal, Quebec.

Teege, G. (2000) *Targeteam, TArgeted Reuse and GEneration of TEAching Materials.* http://www11.informatik.tu muenchen.de/forschung/projekte/targeteam/index.html.en

Rodríguez-Artacho, M., Verdejo, M.F., Mayorga, J.I., Calero M.Y. (1999) *Using a High-Level Language to Describe and Create Web-Based Learning Scenarios* Frontiers In Education Conference FIE '99 IEEE Computer Society San Juan, Puerto Rico.

Wehner, F., Lorz, (2001) A. *Developing Modular and Adaptable Courseware Using TeachML* Proceedings of ED-MEDIA 2001, Tampere, Finland, pp-2013-2018.

Koper, R. (2001) *Modeling Units of Study from a Pedagogical Perspective: the pedagogical meta-model behind EML.* http://eml.ou.nl/introduction/docs/ped-metamodel.pdf

Buendía, F., Diaz, P., Sahuquillo, J., Benlloch, J., Gil, J.A., Agustí, M. (2001) *XEDU, a Framework for Developing XML-based Didactic Resources.* EuroMicro'01, pp427-434.

Silberhorn, H., & Gaede, B. (1999). *IMSDL: Instructional Material Structure Description Language*, 7 th BOBCATSSS Symposium on Learning Society, Learning Organisation, Lifelong Learning, Bratislava.

NOTICE

Reproduction Basis

- [X] This document is covered by a signed "Reproduction Release (Blanket)" form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

- [ ] This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").