A Framework for Adaptive Educational Hypermedia System

José M Parente de Oliveira
Clovis Torres Fernandes
Computer Science Division, Technological Institute of Aeronautics
Brazil
{parente, clovis}@comp.ita.br

Abstract

Adaptive Educational Hypermedia Systems (AEHS) have been used to support customized learning. The adaptation mechanisms provided usually try to define the better concept sequence to be presented and to select the materials and activities more appropriate for a given learner. Nevertheless, despite the primary purpose of AEHS in supporting learning, adaptation mechanisms in these systems have no compromise in using an instructional design theory as source of information. This paper addresses a way in which an instructional design or learning theory is used in conjunction with learner’s domain knowledge, background, preferences and learning style as source of information. To make that viable, a framework for AEHS was defined. The paper describes the framework’s features and its potential benefits.

1. Introduction

Adaptive Educational Hypermedia Systems (AEHS) are a kind of Web-Based Educational Systems that tries to provide a customized interaction for learners, in the form of content and navigation adaptations [1, 2]. To provide adaptation, AEHS normally use learner’s domain knowledge, background, and preferences as reference. Another source of information for adaptation found in the literature is learner’s cognitive or learning styles [3, 4, 5, 6].

Adaptation based on these kinds of information basically tries to define the better concept sequence to be presented and to select the materials and activities more appropriate for a given learner. That is, from a pre-defined curriculum the system defines the better concept sequence and the appropriate materials to be presented.

Despite the primary purpose of AEHS in supporting learning, adaptation in these systems has no compromise in using an instructional design theory. That means it is not common the use of instructional design or learning theories as source of information for adaptation in AEHS. Due to that, it was hypothesized that the use of instructional design or learning theories could be used in conjunction with the above mentioned source of information as driving forces for adaptation mechanisms.

To solve the problem of how to use an instructional design or learning theory in conjunction with learner’s domain knowledge, background, preferences and learning style as source of information for adaptation, a framework for AEHS was conceived.

This paper mainly presents a description of the proposed framework and exemplifies how it functions in order to provide adaptation, as well as presents the framework’s benefits.

The paper is organized in the following way. Section 2 presents the framework and its components. Section 3 describes how framework’s components communication takes place. Section 4 presents an implementation architecture for the framework. Section 5 presents some concluding remarks.

2. Framework for AEHS

To solve the problem of using an instructional design or learning theory as source of information for adaptation, a previous problem should be solved first. This problem was how to explicitly include an instructional design or learning theory in AEHS, given that AEHS did not contemplate an instructional design or learning theory in their architecture. But before this problem there was another one. This other problem was how to define a framework that was capable of facilitating the inclusion of such a theory.

To address these problems, a framework for AEHS [28] was conceived and has been evolving as the result of the analysis of AEHS, Intelligent Tutoring Systems, Web-Based Educational Systems and Adaptive Systems described in the literature [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 35, 36]. Figure 1 shows the framework for AEHS. It should be noted that the framework is a conceptual model that describes what is intended by each component and how the components are related to other components.
A description of the framework’s components is presented next.

2.1 Interaction Analyzer

The Interaction Analyzer is responsible for acquiring information on learner’s behavior. It has two main functions [8, 9]:

- To monitor learner-system interaction in order to get information on activated links, selected tools, etc.
- To infer relevant conclusions on the learner’s behavior, as for example if the learner entered into a specific learning unit, if he or she has finished a unit, if he or she has taken the initiative to change some parameters in the learner model etc.

2.2 Application Model

The Application Model represents the main features of the application in terms of a model of the domain, the instructional or learning theory used and how the domain concepts are grouped into learning units. The Application Model comprises three sub-models: Domain, Learning and Hyperbase Sub-models.

2.2.1 Domain Sub-model

For sure there are several ways to represent the Domain Sub-model. A simple way to represent it is by a list of topics and subtopics. But a more expressive way of representing it is by means of a concept map [32], where the semantic of concept relationships is specified. Additionally, the concept map can be enriched with some meta-information for the map’s concepts. Figure 2 shows a representation of the domain sub-model using Backus-Naur Form [23].

```plaintext
<Domain>:: <SingleDomain> | <CompositeDomain>
<CompositeDomain>:: <SingleDomain> | <CompositeDomain>
<SingleDomain>:: <Topic> <TopicsRelationship>
<Topic>
<Topic>:: TOPICID <TopicKnowledgeType>
<TopicProperty>
<TopicKnowledgeType>:: <SingleTopicKnowledgeType> | <MultipleTopicKnowledgeType>
<MultipleTopicKnowledgeType>:: <SingleTopicKnowledgeType> | <MultipleTopicKnowledgeType>
<SingleTopicKnowledgeType>:: CONCEPTUAL | PROCEDURAL | OPERATIONAL
<TopicProperty>:: <TopicImportance>
<TopicDifficulty>
<TopicImportance>:: LOW | INTERMEDIATE | HIGH
<TopicDifficulty>:: EASY | MODERATE | DIFFICULT
<TopicsRelationship>:: <TopicsRelationshipType>
<TopicsRelationshipType>:: PREREQUISITE | PART-OF | TYPE-OF
```
The representation of the domain sub-model using Backus-Naur Form has a twofold purpose. First, it is useful to express the syntactical aspects of the model, offering flexibility to incorporate new elements or to change some of them. Second, it is independent of the form of implementation. It could be implemented as a formal ontology or as a graph.

Figure 2 shows that a domain can consist of a single topic or several topics, and that each topic has an identifier (TOPICID), a single or multiple knowledge type and some other properties, importance and difficulty. Also the topics are connected by semantic relationships, which can be of types prerequisite, part-of, type-of etc.

2.2.2 Learning Sub-model

The Learning Sub-model represents the instructional design theory used in the application [24] and the learning units defined on the domain model. This sub-model represents the main organizing way the learning process is to be carried out.

An example of an instructional theory is Ausubel’s Meaningful Learning [25, 26, 32]. Meaningful learning is a process in which new information is related to an existing relevant aspect of an individual’s knowledge structure, usually occurring when more specific, less inclusive concepts are linked to more general existing concepts [32]. Thus according to this process, the most general, most inclusive concepts are introduced first and then these concepts are progressively differentiated in terms of detail and specificity [32]. The meaningful learning includes also the inverse of progressive differentiation, the integrative reconciliation. In the integrative reconciliation more general ideas are obtained from more specific ones.

Another important aspect of meaningful learning is the use of as an artifact advance organizers. Advance organizers aim at bridging the learner’s present knowledge and new one to be acquired. A common type of advance organizer is a concept map.

On the basis of meaningful learning and learning units, the following instructional strategies can be defined for the Learning Sub-model:

- At the beginning of a course, the system presents a course overview with a short description of the learning units.
- At the beginning of a learning unit, the system presents an advance organizer, a unit overview or the content of the most inclusive topic of the unit.
- In a given learning unit, the learner accesses the topics according to the restrictions imposed by the topic relationships defined in the domain model.
- Having visited every not known topic in a given unit, the learner is provided with an exercise, an integrative reconciliation and a test, respectively in this order.
- Topic contents are presented in accordance with the type of objective of a learning unit, which can be conceptual, procedural or operational.
- When a learner reaches the last topic in a giving unit, the system suggests links to content of the next type.
- If the current topic is conceptual and the subordinated topics have been visited, then present synthesis for conceptual content.
- If the current topic is conceptual and procedural and the subordinated topics have been visited, then present synthesis for procedural content.
- If the current topic is conceptual and operational and the conceptual subordinated topics have been visited, then present operational content.
It should be noted that the semantic for each individual instructional strategy is not defined. The idea here is to keep conditions and actions independent in order to allow the strategies be defined in a later moment by the instructional designer. This way conditions and actions can be combined in different ways.

### 2.2.3 Hyperbase Sub-model

The last sub-model of the Application Model, Hyperbase Sub-model, keeps the metadata library of the learning objects and the learning object properly. The learning objects include the concepts to be learnt, examples, exercises, learning evaluations, and specific contents related to the instructional design theory defined in the Learning Sub-model. For the instructional strategies defined in Figure 3, the following subset of metadata categories and the correspondent elements can be used [33, 34]:

1) General: Identifier of the Learning Object
2) Technical: URL of the Learning Object
3) Educational: Learning Resource Type:
   - Problem Statement
   - Example
   - Exercise
   - Topic Content Presentation:
     - Conceptual
     - Procedural
     - Operational
   - Test
   - Course Overview
   - Unit Overview
   - Advance Organizer
   - Integrative Reconciliation

4) Classification: Domain Topic Correspondent: TOPICID

It is worthwhile noting that the learning objects of type topic content presentation can include the three types of content, as defined in the Domain Sub-model, in the same object. Then each type is presented in accordance with the instructional strategies.

### 2.3 Learner Model

The Learner Model is the structure that contains the information on the learner’s characteristics that allow the AEHS to adapt to these characteristics [27].

For the purpose of an AEHS that uses a concept map for the domain and as instructional theory the meaningful learning, a learner model can take into account the domain knowledge, learning styles [29, 30] and learner’s media preferences. The learning styles considered are serialists and holists. The serialist learners prefer to study a limited number of issues in sequence, while holists tend to set a wider focus, opening up more topics in a learning episode and hence working with a more complex organizational scheme [27].

The learning styles have a profound influence on the navigation adaptation [4, 5]. For example, in a domain represented as a concept map in which the topics are progressively differentiated from top to base, a serialist learner would be provided with a depth-first navigation adaptation, with the system suggesting more specific topics, while a holist learner would be provided with a breadth-first navigation adaptation, with the system suggesting topics with the same level of abstraction of the current topic.

The learner’s domain knowledge has also influence on navigation adaptation. Once the learner knows a topic the system would not suggest it to be visited. The learner’s media preferences have influence on the content presentation. Based on this information, the system tries to present the contents accordingly.
2.4 Adaptation Decision Model

The Adaptation Decision Model is responsible for deciding what the system should do in terms of presentation and navigation adaptation on the basis of the conclusions drawn by the Interaction Analyzer, parameters from the Learner Model, and information from the Application Model.

The Adaptation Decision Model uses a “four-layer adaptation model”. At the highest level is the instructional design governing the main structure of the learning activities. At the second level is the learner’s domain knowledge. At the next levels are the learning style and the learner’s preferences. With this information, the Adaptation Decision Model decides what should be done in terms of presentation and navigation adaptations.

The Adaptation Decision Model is represented by a set of rules describing the decisions that are sent to the Presentation Generator Model. Some samples of high level rules, expressed as a conjunction of antecedents and a conjunction of consequents, are shown in Figure 4.

(Course beginning && There is partial order within course units && Learner does not know the units && Learning style is holistic) \(\Rightarrow\) (Present a course overview && Present a concept map of the course && Allow access to first unit in the order).

(Current topic type is conceptual and procedural && Children topics have been visited) \(\Rightarrow\) (Present procedural content).

(Current topic is conceptual and operational && Procedural children topics have been visited) \(\Rightarrow\) (Present operational content).

(All unit topics visited && Unit exercise done && Integrative reconciliation presented) \(\Rightarrow\) (Present a test for the unit).

Figure 4: Sample of High Level Rules of the Adaptation Decision Model.

2.5 Presentation Generator

The Presentation Generator is responsible for generating what will be presented to the learner as a result of processing the information received from the Adaptation Decision Model and Application Model. For example, it can present an annotated list of topics and subtopics of the current learning unit, the topics previously accessed and those being suggested, as well as present the content of a learning object informed by the Application Model.

Figure 5: Representation of framework’s components communication.
The Presentation Generator is the final state of the framework’s components communication. Figure 5 shows a schematic representation of this communication.

Having the learner clicked on a topic, the Interaction Analyzer identifies the topic and informs it to the Adaptation Decision Model. The Adaptation Decision Model checks the learner preferences on the Learner Model and then informs the just accessed topic to the Application Model, as well as asks this model the sub-topics of the present topic, the unit objective and the instructional action. The Application Model contacts the Learning Sub-model in order to inform the present topic and to ask the sub-topics, the unit objective and the instructional action. The Learning Sub-model asks the Domain Sub-model the sub-topics and returns the sub-topics, unit objective and the instructional action.

On the basis of the information received, the Adaptation Decision Model decides for the kind of adaptation to be carried out. In this case, it decides for a color-annotated list of topics, with the present topic in gray, the allowed topics in green and the ones not allowed in red. The decisions are sent to the Presentation Generator which asks the Application Model the URL of a Learning Object that corresponds to the information provided. Finally, the Presentation Generator presents on the user interface the information received from the Adaptation Decision Model and from the Application Model.

4 Implementation Architecture

This section briefly presents the way the proposed framework is being implemented. The implementation architecture chosen is based on Java technologies to accomplish the idea of framework’s component modularity, component modifications and component reuse. Figure 6 presents the implementation architecture for the framework.

As can be seen from Figure 6, every requisition coming from the learner’s web browser are interpreted by the servlet Interaction Analyzer that can access the Learner Model and Application Model data bases. After drawing conclusions on the requisition, the Interaction Analyzer contacts the Adaptation Decision Model, which is implemented by means of JavaBeans. The Adaptation Decision Model then accesses the Learner Model and Application Model data bases to get the information required. After getting this information, the Adaptation Decision Model sends its decisions to the Presentation Generator, which is also implemented by means of JavaBeans. After receiving a decision, the Presentation Generator accesses the Application Model data base to get the URL of a learning object to be presented. Finally, the Presentation Generator requests a composition of the

![Implementation Architecture Diagram](image-url)

Figure 6. Implementation architecture for the AEHS framework.
presentation to the JavaServer Pages.

As JavaBeans are components of an application in the context of JavaServer Pages or Servlets, they are suitable to implement the Presentation Generator and the Adaptation Decision Model. Additionally, the JavaBeans offer advantages of separating the programming logics from presentation, as for the Presentation Generator and the final page composition in Figure 6.

5 Conclusion

The framework for AEHS has been a solution for the two main problems raised. First, the framework was a solution to the problem of how to use an instructional design or learning theory in conjunction with learner’s domain knowledge, background, preferences and learning style as source of information for adaptation in AEHS. Second, it was also a solution to the problem on how to explicitly represent an instructional design theory in an AEHS, instead of having it diluted over other system’s components. Additionally, as a core concern in the framework’s conception was the modularity of the components, the result was a framework with well defined and separated roles for the components.

With the separation of the components’ roles it can be speculated that be possible the reuse of materials from one application to another. For example, the domain model and some learning objects could be reused in an application with a different instructional design theory.

Even being based on a single case of use of some aspects of an instructional design theory, it is possible to envisage the use of other theories in the framework.

The possibility of reuse of material and the use of different instructional design theories raises another possibility. Supposing for example that an application would be defined using two different instructional design theories, then they could be put together to form a single application. With that, besides the presentation and navigation adaptations provided, this new application could also chose the instructional theory more suited to the learner’s learning style.

Another speculation that requires further studies is related to the quality of the support provided by the adaptation obtained with the framework. Would the use of an instructional/learning theory in conjunction with knowledge domain, learning styles and learner preferences make adaptation strategies more supportive in terms of learning? It is not an easy question to be answered.

The framework can also serve as a basis for creating authoring tools, as long as several tools would be required for authoring an application. Among the authoring tools required are domain, instructional design and learning object metadata.

By not committing to any specific technology, in contrast to the ongoing implementation, the framework can be adapted in order to be implemented with more intelligent technologies, like formal ontologies and intelligent agents.

References


