THE ANNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2000 ISSN 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS

A HYBRID APPROACH TO DYNAMIC COURSE GENERATION ON THE WWW

Diana STEFANESCU, Emilia PECHEANU, Luminita DUMITRIU

Department of Computer Science
University "Dunarea de Jos" of Galati
Str. Domneasca nr. 111, Galati 6200, Romania
phone (+40)/036/460106
E-mail: Diana.Stefanescu@ugal.ro, Emilia.Pecheanu@ugal.ro,
Luminita.Dumitriu@ugal.ro

Abstract: In recent years, hypermedia has been grown in interest in advanced research on teaching and learning. Futhermore, the extension of hypermedia toward intelligent hypermedia comes more than obviously to meet specific requirements of learning, such as the student modeling, diagnosis and guiding the learner in his investigation.

In this paper, we present an architecture for a system based on integrating hypermedia system and intelligent knowledge based system. The object formalism is chosen to represent static and predefined knowledge, whereas the rule formalism is chosen for dynamic knowledge and reasoning processes modeling. The knowledge base can be described by three models of expertise: the domain model, the student model and the didactic model (including the communication model). This architecture allows automatic generation of dynamically individualized courses according to the learner's goal, his or her previous knowledge and the learner's success in acquiring knowlede.

Keywords: Intelligent Tutoring Systems (ITS), hypermedia system, object formalism, rule formalism, WWW, dynamic course generation.

1. INTRODUCTION

The authors of this paperwork describe the project of an instructional system using both modern communication technologies and teaching methods.

The designed instructional system is a WWW-based tutorial that it will be used to teach the curricula disciplines of the Computer Science field of study. (Bumbaru, 1996). The proposed architecture takes advantages of integrating multimedia/hypermedia and intelligent knowledge-based system.

The tutorial will be a colection of related lessons consisting of theoretical parts (the instructional environment) that can include different simulations (animated images, Java applets) and a collection of testing exercises (the testing and assessing environment). Lessons and exercises will be presented to the user as a sequence of Web pages.

The Web pages forming lessons can include text, examples, images/animated images, diagrams and links to related information. The applicative (testing) environment will permit the assessment of students' knowledge and can turn into a tool to enhance learning (Razmerita, 1997).

While designing the tutorial architecture, we used the knowledge-based approach.

This paper was recomended for publication by S. Bumbaru

THE ANNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2000 ISSN 1221-454X

2. BENEFITS OF THE KNOWLEDGE BASED APPROACH

The knowledge based approach includes the following design principles :

- Represent instructional content and instructional strategies separately.
- Explicitly represent abstract pedagogical entities (such as "topics").
- Design at the pedagogical level, as opposed to the media level, when possible.
- Modularize the instructional content for multiple use and re-use.
- Create generic teaching strategies that can be used with different instructional content.

Designing tutoring systems in this way has many advantages over the traditional CAI (Computer Assisted Instruction) design paradigm (Murray, 1988).

3. THE NEED FOR DOMAIN CONCEPT STRUCTURE

One of the problems with Hypermedia for educational applications is that the learner needs to have a good conceptual map of the domain being taught, in order to effectively use the system.

In our framework the creator of the hypermedia courseware needs to carry out the following steps:

- Create a concept hierarchy for the domain being taught. This would involve analysis of the domain to determine these concepts.
- Develop hypermedia material corresponding to that concept. The hypermedia architecture uses typed links such as those found in semantic networks. Types include, is-a, which represents class definition; a-kind-of, which represents membership of and inheritance from superclasses; has-a, relating an object to an attribute or property and part-of which is used to show how an object is composed from smaller parts. A hypermedia system built using such link types provides a simple knowledge base about the domain and such a knowledge base may be used to reason new links automatically by utilising automatic reasoning algorithms.

Implementing various domains in the proposed system would require some effort from the domain author in that it is necessary to think about the domain carefully in order to produce the correct class nodes and correct links types.

4. THE COURSEWARE ARCHITECTURE

As a first level of decomposition, the architecture purposed by Elhani and Gouarderes (Elhani and Gouarderes,1992) may be split in three logical levels: Presentation, Hypermedia and Knowledge Base (Stefanescu, 1996).

The Presentation level is devoted to describe the content of multimedia interactive objects as they will be shown to the learner on his workstation, and how each object will interact with the learner.

The hypermedia level permits sequencing and navigation through these interactions taking into account the learner’s input.

The knowledge base level uses Artificial Intelligence skills to meet the requirements of intelligent tutoring systems.

As shown in figure 1, it is a hierarchical architecture built around four layers: Presentation objects (i.e., layer 1), Hyper objects (i.e., layer 2), Knowledge Base (i.e., layer 3).

Layers 1 and 2 are a collection of autonomous and active objects with related services; each object is an instance of a related class. All classes and meta-classes are predefined and independent from a specific learning application.

For layer 3, three main knowledge-based components are highlighted: domain model, student model and didactic model. Each component, as a collection of rules and facts conforming to the declarative approach, can do specific services without any ad hoc functions.

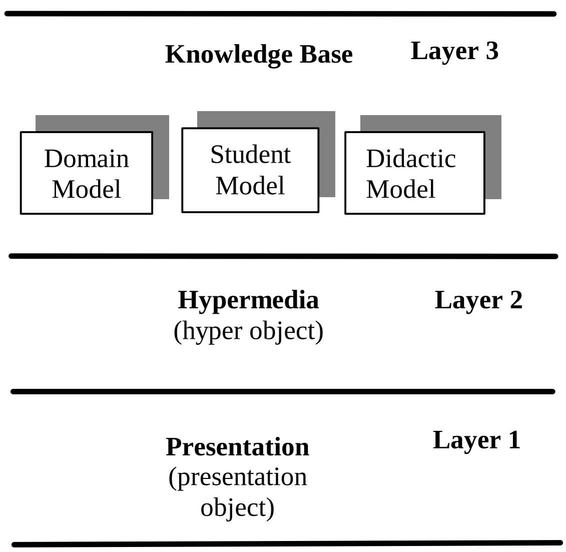


Figure 1. Layers of courseware architecture

During the courseware execution, layer components interact between them by means of message sending, for services invocation. A layer component (i.e., object of the three bottom layers or a rule of layer 3) has to send a specific message, to an object for a services activation. Then the receiver replies with a report witch carries the relevant results of the service activated. Message are generally propagated from higher layers to lower layers and reports in the reverse way.

3.1. Layer Services

A *presentation object* is an elementary information to be used through the learner-computer interface. Services: In reply to an invocation from higher layers, it restitutes the presentation contents back to the Hyper object for further analysis.

The purpose of an *Hyper object* is mainly deciding what will be the next learner-computer interaction taking into account the current results of interaction and the pedagogical goal. Like in hypermedia systems, hyper objects are structured as nodes of a network.

Three kinds of Hyper objects can be distinguished:
- Learning-unit: Object of an output kind, giving information about what is to be taught. It may be an example, an hint or an explanation about a domain concept.
- Solicitation: Object of an input kind, devoted to assess the learner's comprehension about what is already taught: it may be a simple question, an exercise, or a problem to solve. Author's expected responses, witch are objects too in our architecture, are attached to a solicitation object. A solicitation object refers either to presentation object from lower layers.
- Module: As a collection of relevant and organized Learning-units and/or Solicitations, in layer 3, a module is an object specialized in teaching a specific domain concept.

Services: Hyper objects are entities witch are involved when results of input data come back from the presentation objects. Two main services Hyper objects support:
- Analysis of input data, made by the current solicitation object, in comparison with expected response objects. These latter may be either stored beforehand by the author or calculated dynamically by layer 3.
- Two levels of decision making should be highlighted:
- Local decision: What will be the next Learning Unit or Solicitation following the current one in the current module. This kind of decision is made by the current solicitation object as looking for the link object which activation conditions are true.
- Global decision: When a module is finished a next module is chosen. Which one is chosen depends upon the learner's behavior to all solicitation in the current module. This decision is made by the current module object as looking for the link object which activation conditions are verified.

Intelligent Knowledge Base has the ability to supervise the functioning of the bottom layers, to make more dynamic the learner-computer interactions, and to adapt them to the individual learner.

Layer 2 and 3 have some common services such as the analysis of input data and making local or global decision, but implemented in different ways. An hyper object provides these services only if the learner's response is expected and the static link is foreseen by the author. In such a case, the current Hyper object has the ability to make decision without involving the Knowledge base which is only informed of the student's response to update the student model, and of such local decisions.

Otherwise layer 3 is the only means to deal with dynamic and unexpected responses (i.e., dynamic link). Thus, when the Hyper object discovers, after analysis, that the response is unexpected, it will not make decision itself, but it will forward the learner response to the Knowledge base for diagnosis and decision making.

Services:
- Domain model: To present domain knowledge to the learner such as examples or problems, to solve problems and explain solutions, to give hints and to evaluate dynamically the responses and paths the learner gives for the problems he has to solve.
- Student model: Modeling knowledge of the student (i.e., what is known, not known or badly known) to adapt the learning application to the individual student, diagnosing the student's problem solving process especially when his response is unexpected.
- Didactic model: Sequencing concepts of the domain to be taught, making dynamic decisions when layer 2 is unable to do it.

3.2. Module and pedagogical unit

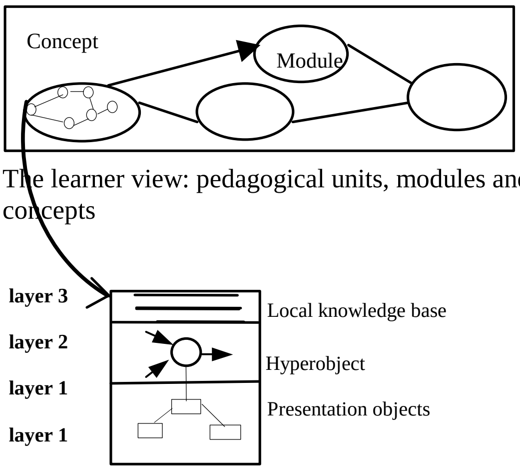
A domain concept is either factual (i.e., factual knowledge) or procedural (i.e., exercise, problem to solve). The section of courseware which is centered around a domain concept is called *module*. A module is represented by an object in layer 2 entailing both statistics about the learner's behavior toward the module and a method to make global decision. A module is also represented by some knowledge in layer 3.

From the learner's point of view, a courseware may be seen as a sequence of modules; a module is seen as a sequence of pedagogical units. A *pedagogical unit* is a learner-computer interaction, which may be either an explanation (i.e., Learning-page), an example shown to a learner, an exercise to solve (i.e., Solicitation) about the related concept. From the system's point of view, a PU is seen as a vertical section of the architecture (figure 2).

3.3. The knowledge base

Layer 3 may be divided in two parts: local knowledge (LKB*)*, describing the behavior of the learning during a transaction (answers, timing, learner's choices) and global knowledge (GKB) defining the transmission of information, that is to say the author's model. Several knowledge bases can be associated with a pedagogical unit, each of them can define a new behavior for the system.

Local knowledge base it should include knowledge which are specific to a pedagogical unit (domain facts pre-stored by the author or set dynamically by the domain expertise when the exercise is made up, they are used to select rules from the global domain knowledge base and to activate them in order to solve the specific exercise or problem tackled in this pedagogical unit. LKB may include results of interactions such as the student's response, delay of time of response, type of response. These latter facts are instantiated when the response comes back from the Hyper object. LKB may include rules too. It may be represented as a temporary memory.



The system view: pedagogical unit structure

Figure 2 - The learner and the system views of courseware

When the pedagogical unit is deactivated some of its knowledge (e.g., student's response) are included in the GKB, in the sense to be exploited by the following pedagogical unit for diagnosing and making decision.

Global knowledge base includes rules and facts, about the domain knowledge, the student model and the didactic strategies, witch are shared by all pedagogical units.

GKB contains:

- Predefined learning paths which define both synchronized stages in the presentation of knowledge and control of these stage which adapts the system according to the student's behavior;
- Dynamic paths, that provide a remedy for learning problems which cannot be resolved by predefined learning paths.

4. CONCLUSION

The framework for instructional system will join the information spread technology WWW-specific with the object oriented capabilities in storing and managing structured data. This can lead to multiple advantages in exploiting, maintaining and updating the computer-aided instructional system, offering a higher degree of generality approach in implementing and using such a system. We have implemented this framework in C++ programming language and Prolog. The framework has the advantage of simplicity and domain independence.

REFERENCES

Bumbaru, S.; Stinga, O.; Pecheanu, E.; Dumitriu, L.. and Tudorie, C. (1996), *A CBTS built upon Oracle RDBMS*. Proc. The 3-rd East-West Congress on Engineering Education. Poland

ElHani O. and Gouarderes G. (1992). Standardized Architecture for Integrated Open Courseware, *Lecture Notes in Computer Science*, 602, pp. 198-211

Razmerita, L.; Stefanescu, D.; Bumbaru, S. and Istrate, A. (1997), Student testing and assessment in Computer Based Training Systems, Proceedings of 4-th International Conference Computer Aided Engineering Education, Krakov, Poland, pg. 223-231

Stefanescu, D.; Bumbaru, S. and Ariton, V. (1996), Hybrid Systems for Assisted Learning, The 9-th Symposium on Modelling, Simulation and Identification Systems, SIMSIS '96, pg. 372-381.