Chapter 5

PERSONALIZED INTELLIGENT TRAINING ON THE WEB: A MULTI-AGENT APPROACH

Nicola Capuano[1,2] Massimo De Santo[3] Marco Marsella[1,2] Mario Molinara[3] Saverio Salerno[1,2]
[1] *CRMPA, Centro di Ricerca in Matematica Pura ed Applicata c/o DIIMA, Università degli studi di Salerno, Italy.*
[2] *DIIMA, Dipartimento di Ingegneria dell'Informazione e Matematica Applicata, Università degli studi di Salerno, Italy.*
[3] *DIIE, Dipartimento di Ingegneria dell'Informazione e Ingegneria Elettronica, Università degli studi di Salerno, Italy.*
niccap@crmpa.unisa.it; desanto@unisa.it; marmar@crmpa.unisa.it; molinara@unisa.it; salerno@ponza.dia.unisa.it

Abstract One of the most interesting realm among those ones brought up to success by the development of the Internet is distance learning and training. For this reason, the investigation for adequate architectures and platforms supporting flexible and tailored training solutions is nowadays of great interests in the scientific community. This paper is concerned with the presentation of an original architecture for intelligent distance tutoring which make use of software agents. The way in which the knowledge is represented and stored is discussed together with the ability of our system to manage individual learning paths for different users. The rationale for using Agents is presented and the implementation of the system is discussed.

1 INTRODUCTION

The great amount of information available across the Internet brought to the development of new sophisticated information-based technologies; interests in knowledge management, in information retrieval and information filtering are becoming hot topics in several areas for different applications across Internet.

Among the enormous number of such applications, one of the most interesting is the *Distance Learning*. The potential of the Web for providing rich materials and experiences, the possibility and capability to learn more knowledge implied by digital technologies are factors of increasing importance in a world where the amount of information that needs to be learned grows very rapidly and becomes obsolete very quickly. As a matter of fact, the proliferation of Local Area Networks (LANs), and Wide Area Networks (WANs) for telecommunications, information and data applications has brought the enabling technological framework needed to bring network-based multimedia training to full availability of millions of people world-wide.

Interactive training delivered via a computer has been reported to be more effective than traditional classroom lectures, and, moreover, to reduce training time and costs [1], [2]. Exploiting computer delivered training it is possible to increase training effectiveness by increasing student participation, interest and retention of knowledge and reducing attrition level [3]. Fletcher [4] summarized a set of supporting evidences for the benefits of technology based learning systems coming from numerous analyses and specific studies. His conclusions can be summarized as follows.

- Technology can be used to teach: in the absence of any other instruction, technology based learning systems improve student achievement.
- Technology improves instructional effectiveness compared with the "conventional instruction" (lecture, text-based materials, hands-on experience).
- Technology reduces time to reach instructional objective: analyses covering a wide range of content areas (military training, adult education, and higher education) shows an average reduction of the 30% of time if compared with "conventional instruction".
- Technology can be used to teach "soft skills" (soft skills are knowledge and skills associated with social interactions).
- Students enjoy using technology: they are more likely to say they enjoy technology based instructions than conventional mechanisms.

Benefits of computer based training relies on the fact that they exploit a "learner-centered" training paradigm in place of the classical "tutor-centered". Such approach focus on needs, skills and interests of the learner. At the heart of the modern instructional design there is, in fact, the idea that people learn best when engrossed in the topic, motivated to seek out new knowledge and skills because they need them in order to solve the problem at the hand [5].

The purpose of this paper is to present ABITS, an innovative solution for intelligent training over the Internet able to address all these topics. Its features include automatic learners evaluation (through profiling) and intelligent course tailoring based upon user needs and inferred user profiles. ABITS includes and integrates several state-of-the-art technologies: metadata and conceptual graphs for knowledge manipulation, intelligent agents and fuzzy user profiling. ABITS is Web-based: it requires zero cost installation for end-users and can allow them to take training without time and place constraints. Moreover ABITS is content open: it allow easy integration of content from multiple courseware providers and authoring-tools in order to reuse existing didactic material.

The following paragraph is dealt with an overview of ABITS functions while paragraphs 3 and 4 will depict the ABITS internal architecture based on software agents. Finally paragraph 5 will show ABITS in action in a real case.

2 WHAT IS ABITS

ABITS stands for "Agent Based Intelligent Tutoring System". It is a Multi-Agent System (MAS) able to extend a traditional Course Management System (CMS) with a set of "intelligent" functions allowing student modeling and automatic curriculum generation. The purpose of such functions is the improvement of the learning effectiveness based upon the adaptation of the didactic material to student skills and preferences.

This chapter is thought as an introduction to these functions. In particular, paragraph 2.2 is dealt with student modeling while paragraph 2.3 describes the ABITS implemented algorithm for curriculum generation. Such functions are depicted in the UML Use Case Diagram of figure 1 where the *Evaluate Curriculum* case is dealt with curriculum generation while the *Evaluate Preferences* and the *Evaluate Cognitive State* cases are related to user modeling.

ABITS functions found their effectiveness on a set of rules for knowledge indexing based on Metadata and Conceptual Graphs. This point is treated in paragraph 2.1.

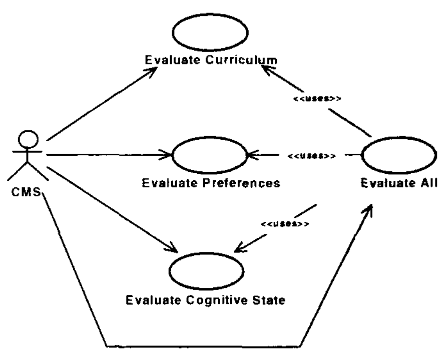

Figure 1: ABITS Use Case Diagram

2.1 Knowledge indexing

ABITS didactic material is organized in *Learning Objects* and is stored in a Course Material File System. A Learning Object is any entity which can be used, re-used or referenced during technology-supported learning. Learning Objects must be indexed in order to let the system know what each one of them is about and how it can be used during the learning process. Some kind of information about Learning Objects is so required. This is *Metadata*.

"Metadata is information about an object, be it physical or digital and its main goal is to locate in efficient and effective way resources over a system or a computer network" [7]. In the field of learning materials, several organizations such as IEEE, EDUCOM etc focused their attention on the creation of Metadata standards specifying the syntax and the semantics of the so-called *Learning Object Metadata* (LOM).

A LOM standard defines the minimal set of properties needed to allow Learning Objects to be managed, located, and evaluated. They accommodate, moreover, the ability for locally extending the basic properties. ABITS adopts the IEEE LTSC LOM standard [7] to index learning material. Many advantages come in fact from referring to a Learning Object Metadata standard:

- to take advantage of a complete syntax and semantic created by experts of the Learning Technology;
- to enable the automatic importation of extern learning objects that adopt the same Metadata standard;
- to enable the exportation/sale of learning objects to extern systems/clients that adopt the same Metadata standard;

Metadata not only have to provide information about a single Learning Object. They have to provide information about object relations and interdependency too. For this purpose the IEEE LOM standard has a Metadata element called *Idea* that supports *Domain Conceptualizations*. A Conceptualization is an abstract, simplified view of the world that we wish to represent. A *Conceptual Graph* is an explicit specification of a Conceptualization [8]. Conceptual Graphs are graph-like structures composed by *Concepts* and *Conceptual Relations* where every arc links some Conceptual Relation r to some concept c.

With the term *Concept* we intend an abstract notion that refers to a particular Conceptual Graph. Conceptual Graphs are used to link Concepts underlying the knowledge domain with several kinds of relations: (prerequisite, sub-concept, general relation, etc). As we will see, Conceptual Graphs are massively used by ABITS functions in conjunction with Metadata fields for Cognitive State modeling and automatic Curriculum Generation.

2.2 Student Modeling

ABITS student models are composed by a *Cognitive State* and a set of *Learning Preferences*.

The ***Cognitive State*** contains the knowledge degree, reached by a particular student, of every ABITS tested domain Concept [6]. We represent this information by using an array of *Fuzzy Numbers* (one for each concept). The decision to use Fuzzy Numbers [9] in ABITS Cognitive States arises from the necessity to manage uncertainty in the student evaluation process. In this way, in fact, we can admit different kind of evaluations with different degree of reliability.

As an example, when a student reads an expositive Learning Object (i.e. a lesson) with a given set of Concepts involved, ABITS forecasts a little increase in the knowledge of such Concepts (maintained in the Cognitive State) for this student but with a large degree of uncertainty (read doesn't mean understood). Conversely, when the same student answers correctly to a test related to the same set of Concepts, ABITS can increase again the knowledge degree of such Concepts but with a lower degree of uncertainty (user now is tested). To represent this kind of information we use more and more narrow fuzzy numbers.

Moreover, in order to model the attitude that have humans to forget what they learn, ABITS applies a *Forgetting Function* to Cognitive States. This algorithm, in order to signify that evaluations are more and more unreliable over the time, provides to widen the amplitudes of Conceptual knowledge degrees inside Cognitive States.

Within ***Learning Preferences*** we enclose information about the student perceptive capabilities i.e. to which kind of resources a specified student is shown to be more receptive [6]. To evaluate student preferences ABITS exploits Metadata elements contained in the *Educational* IEEE Metadata Category such as: *Format* (kind of media), *Difficulty*, *Pedagogical Approach*, *Interactivity Level* and *Semantic Density*.

To evaluate student Preferences ABITS exploits this idea: during the learning process there are *Milestones* (points in the student Curriculum) chosen by tutors where the Cognitive State is updated with respect to activities performed by students. After this point, a new evaluation is given for each Concept involved in student performed activities. ABITS can evaluate the pedagogical effectiveness of Learning Object typologies by exploiting the variation between concept evaluations and the Educational Metadata information about visited Learning Objects between couples of subsequent Milestones.

ABITS calculated information about Student Models can be exploited directly by tutors or re-used by ABITS in the Automatic Curriculum Generation procedure.

2.3 Automatic Curriculum Generation

Each student can be assigned to one or more different *Courses*. An ABITS Course is composed by a set of *Learning Goals* and by a *Curriculum*.

With ***Learning Goals*** (that are strongly different from Learning Objects) we intend a set of key Concepts necessary to be learnt to successful complete a specific Course. Such Concepts (as all other Concepts) are part of a Domain and are represented inside the Conceptual Graph of such Domain.

With ***Curriculum*** we intend, instead, an ordered list of Learning Objects that can be used to provide to a specific student all necessary knowledge to complete a specific Course. While Learning Goals indicate what (which Concepts) a student has to learn, Curriculum specify how these Concepts has

http://www.springer.com/978-0-7923-7508-1

Electronic Business and Education
Recent Advances in Internet Infrastructures
Chin, W.; Patricelli, F.; Milutinovic, V. (Eds.)
2002, XII, 400 p., Hardcover
ISBN: 978-0-7923-7508-1