An Intelligent Web Teacher System for Learning Personalization and Semantic Web Compatibility

Nicola Capuano[1,3], Matteo Gaeta[1,3], Alessandro Micarelli[1,4] and Enver Sangineto[2]

1 CRMPA Centro di Ricerca in Matematica Pura ed Applicata, c/o Università degli Studi di Salerno – DIIMA, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy.

2 CRMPA Centro di Ricerca in Matematica Pura ed Applicata, Sezione "Roma Tre", c/o Università degli Studi Roma Tre – DIA, Via della Vasca Navale, 79, 00146 Roma, Italy.

3 DIIMA Dipartimento di Ingegneria dell'Informazione e Matematica Applicata, Università degli Studi di Salerno, Via Ponte Don Melillo, 84084 Fisciano (SA), Italy.

4 DIA Dipartimento di Informatica e Automazione, Università degli Studi Roma Tre, Via della Vasca Navale, 79, 00146 Roma, Italy.

Abstract. We propose a Web tutoring system in which Artificial Intelligence techniques and Semantic Web approaches are integrated in order to provide an automatic tool able both to completely customize learning on the student's needs and to exchange learning material with other Web systems. IWT (Intelligent Web Teacher) is based on an ad hoc knowledge representation which describes the didactic domain by means of an Ontology. The student can select the concepts belonging to the Ontology she/he is interested in which. The system planning mechanism builds the most suitable Learning Path for that student.

Keywords: Intelligent web-based learning environments, Innovative use of AI languages in teaching and learning, Semantic Web.

1 Introduction

We present in this paper an innovative e-learning platform which integrates some techniques coming from different research areas such as Artificial Intelligence, Information Retrieval and Semantic Web. The aim of the platform is to provide a flexible e-learning instrument able to support learners during the whole training cycle, from the definition of the objectives to the assessment of the results, through the construction of custom self-adaptive courses.

Some of the proposed methods have been previously described in other scientific works such as [1, 2]. The platform is now available as a commercial product in its version 1.0 [3] with the name of IWT (Intelligent Web Teacher). Finally, we are currently working on it for the introduction of further innovative features (which will be also presented in the present paper) under the European Community founded project Diogene (5th Framework Programme, Information Society Technologies, contact number IST-2001-33358, see [4] for more details).

IWT aims to fill some lacks of the present e-learning systems, such as:

- The common e-learning systems usually do not allow the customisation of the courses on the student profile but simply propose standard courses.
- They usually limit themselves to deliver learning material to the student, without trying to interact with him through a *model of her/ his mind*.
- They usually exploit possible test results only for checking the student's acquired knowledge level but do not use such a knowledge also for changing the quantity and quality of the delivered learning material itself.
- They do not take into account the student's preferred *learning styles*.

In the following we will explain how IWT faces these aspects merging different emerging technologies and State-of-the-Art scientific developments.

The rest of the paper is organized as follows. Section 2 gives a quick overview of the main "intelligent" aspects of the IWT platform. In Section 3 we present our knowledge representation methodology and in Section 4 the student model. In Section 5 we show the inference mechanism able to construct a personalized course and we conclude in Section 6.

2 A Platform Overview

Figure 1 shows the IWT platform. The main idea is that a student approaching the system can choose among a set of *Domain Concepts* she/he wants to learn. Such Domain Concepts belong to an Ontology vocabulary which is written in SHOE [5] or in DAML + OIL (in order to be open to other e-learning platforms) and satisfying the knowledge representation shown in Section 3. However, the student only access to the Ontology vocabulary, in which each Domain Concept is associated with a brief explanation of its meaning. Once the student has chosen the Domain Concepts she/he is interested in, the system's planning mechanism builds a suitable course (a *Presentation*) which satisfies both the student present knowledge (or *Cognitive State*) and her/his preferred learning style (or *Learning Preferences*). This is done as follows.

The student's chosen Domain Concepts are contained in the set *Target Concepts*, input to the system. Both the student's *Cognitive State* and the *Learning Preferences* are stored in the *Student Model*, which is continuously updated by the test results (see Section 4). The planning mechanism matches the Target Concepts with the domain description (the *Ontology*) and the Cognitive State. The result of this match is the set of Domain Concepts (*Learning Path*) which is necessary to the student to learn all the Target Concepts.

In this way the system can cope for possible student's lack by automatically adding necessary pre-requisite Domain Concepts to the set she/he has chosen, or it can decompose such a set in a more detailed one or, finally, order it in the best didactic way.

Finally, for each Domain Concept of the Learning Path the system chooses a *Learning Object*, i.e. a Web-deliverable resource completely explaining the Domain Concept. A Learning Object (LO) is defined [2] as any entity which can be used, re-used or referenced during technology-supported learning. In our case, a Learning Object is a logical container that represents an atomic Web-deliverable resource such as a Lesson (an HTML page), a Simulation (a Java applet), a Test (an HTML page with an evaluating form) and each kind of Web-deliverable object. It is worth underlying that each Learning Object is treated by the system as an atomic piece of knowledge, without interfering in its internal structure.

Learning Objects are chosen taking into account the student Learning Preferences. First of all, the system asks to the student if she/he prefers high level learning material (which must be paid), or free, unguaranteed material found on the Web. In the first case the system matches the student's Learning Preferences with the Learning Objects' Metadata (LOMs) contained in a suitable system's local repository (see Figure 1). This LOM data base is updated by the Content Providers of the systems and contains Metadata describing Learning Objects' features owned by the Content Providers themselves. Once the Presentation is ready, each Learning Object is asked for to the corresponding Content Provider and returned to the student.

Vice versa, if the student asks for free material, then the system exploits Learning Objects previously found on the Web and off-line categorized. The categorization works as follows. Each *atomic* Domain Concept (see Section 3) is used as a category. Each textual Web document (Learning Object) found on the Web is classified by means of standard textual categorization techniques in order to associate it with an atomic Domain Concept (see Section 5). This link is finally used by the planning mechanism to construct the (free) Presentation returned to the student.

In the following sections each phase above mentioned is explained more in details. We will assume that the Automatic Course Generation (ACG), containing the planning mechanisms (see Figure 1) is called with an input given by a set (*TargetC*) of Target Concepts.

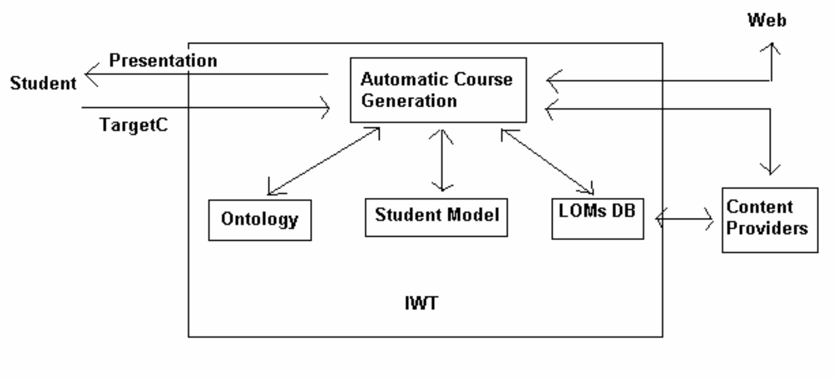

Figure 1. The main IWT Structure.

3 Didactic Knowledge Representation

We propose some Knowledge Representation rules able to describe fundamental properties of didactic Domain Concepts. These rules have to be:

1. Domain independent. Indeed, IWT is thought for a general-purpose application.
2. They have to describe relations among Domain Concepts (DCs) able to be exploited by the planning mechanism in order to include in the generating Learning Path all those Domain Concepts necessarily related with the Target Concept set.
3. They are not intended to provide an exhaustive knowledge representation framework. Indeed they are tailored to our planning purposes and needs, while a universal treatment of Ontology representation (valid also for not-didactic purposes) is out of the scope of this work.

We provide in the following the admissible relations among the Ontology's Domain Concepts.

- *HP* (Has Part): *HP (x, y1, y2, ..., yn)* means that the concept *x* is composed of the concepts *y1, y2, ..., yn* , that is to say: to learn *x* it is necessary to learn *y1* and *y2*, and, ..., and *yn*.
- *R* (Requires): *R (x, y)* means that to learn *x* it is necessary to have already learnt *y*. This relation poses a constraint on the Domain Concepts' order in a given Learning Path.
- *SO* (Suggested Order): *SO (x, y)* means that it is preferable to learn *x* and *y* in this order. Note that also this relation poses a constraint on the DCs' order but now it is not necessary to learn *y* if we are interested only in *x*.

Furthermore, we have the following relation linking Domain Concepts and Learning Objects' Metadata:

- *EB* (Explained By): *EB(d, l)* means that the Domain Concept *d* can be explained by means of the Learning Object indexed by the Metadata *l* (*l* is sufficient to explain *d*).

3.1 Some properties of the Ontologies

The *HP* relation is used to describe a decomposition of an abstract concept (e.g., Math Analysis) in a set of more specific sub-concepts (e.g., Limits, Derivatives, Integrals and Series, see Figure 2). It is worth to underline that the knowledge representation methodology chosen does not allow for disjunction representation. Namely, no alternative decompositions are allowed (e.g., Limits, Derivatives, Integrals and Differential Equations versus Limits, Derivatives, Integrals and Series). Indeed, each knowledge representation approach must be a trade-off between expressiveness and computational costs. We have chosen to prevent disjunction representation both to avoid backtracking mechanisms in the planning phase and because it is not clear how the system could choose among different decompositions.

The relations *R* and *SO* are *inherited* through *HP*. Formally we have that:

Property 1. *Decomposition Inheritance*

1. If *HP(d, ..., d1, ...)* and *R(d, d2)* then *R(d1, d2)*
2. If *HP(d,..., d1, ...)* and *SO(d2, d)* then *SO(d2, d1)*.

Definition 1. A *Presentation P* is an ordered list of LOMs (*l1, ..., ln*) with the following properties:

1. The union of LOMs of *P* is sufficient to explain to the student all the *Target Concepts* of the input list *TargetC*.
2. For each l_i, $l_j \in P$, if : $EB(d_1, l_i)$ and $EB(d_2, l_j)$ and $d_1 \prec d_2$, then $i<j$. Where the partial order relation $\prec$ between Domain Concepts is recursively defined as follows:
   a. if $R(x, y)$ then $y \prec x$
   b. if $SO(x, y)$ then $x \prec y$
   c. if $HP(z, ..., x, ...)$ and $HP(w, ..., y, ...)$ and $z \prec w$ then: $x \prec y \wedge x \prec w \wedge z \prec y$.
3. *P* meets the student's Learning Preferences of the current student state.

While Points 1 and 3 of the above definition are self-explaining, Point 2 needs some remarks. It is a consequence both of the transitivity property of the order relations *R* and *SO* and of Property 1. Intuitively, it means that the elements of a didactic domain are partially ordered. As a consequence of this, LOMs belonging to a same Presentation have to respect this partial order. If, for instance, a Presentation contains *li* and *lj*, which explain, respectively, the concepts "Derivatives" and "Limits", then *lj* has to precede *li*. The same situation holds when *li* and *lj* explain DCs not directly linked to each other by an order relation (Requires or Suggested Order) but components of DCs directly linked by an order relation.

In the following sections we will show the general planning algorithms needed to generate a partial ordered Presentation satisfying Points 1 and 3 above together with the linearization procedure to totally order a partial ordered Presentation according to Point 2. Before concluding this section, we provide a last definition which will be useful in further sections.

Definition 2. A concept *x* is named *atomic* if there is no concept *y* s.t. *HP(x, y)*.

It is important to note that only atomic DCs can be linked to LOMs by means of the relation *EB*. Indeed, a not atomic concept *x* is composed of sub-concepts (e.g., *y*, *z*) and the learning of *x* is equivalent to the learning of its components *y* and *z*. Only when no more decomposition is possible, we then "switch to the lower knowledge representation level", i.e. from Ontology to Metadata.

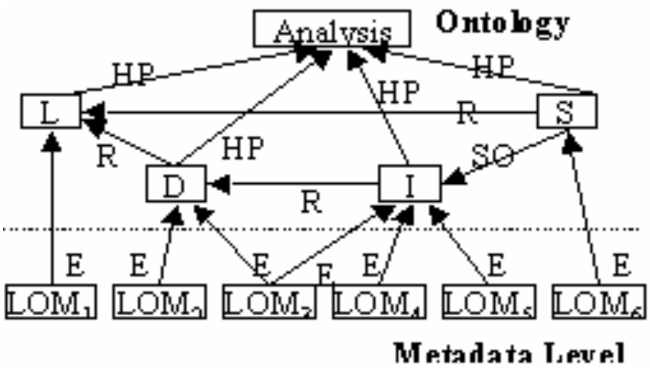

Figure 2. An example of knowledge representation belonging to the didactic domain "Math". L stands for Limits, D for Derivatives, I for Integrals and S for Series.

4 The Student Model

The IWT Student Model is composed of two modules: the Cognitive State and the Learning Preferences. We use the former to describe the knowledge degree achieved by each student about every DC. This evaluation regards both previously acquired student knowledge and skills learnt using our proposed e-learning platform. Moreover, the Learning Preferences module groups information about the student perceptive capabilities, i.e., the preferred types of resources and/or learning styles for the specific student.

Cognitive State and Learning Preferences describe the student present state by means of two different sets of atomic facts associated with a fuzzy truth degree. More precisely, we adopt a couple of fuzzy values given by:

Evaluation = <Degree, Reliability>. (1)

Evaluation is a pair of fuzzy values ranging in $[0, 1]^2$. They indicate, respectively, the degree of knowledge/preference the student has shown concerning the associated fact, and the reliability degree that the system has about this evaluation. For example, if a DC is associated to a value of *Evaluation = <0.8, 0.6>*, this indicates that IWT believes the student knows quite much about the related DC (0.8), and estimates the reliability of this statement about her/him knowledge with a certainty degree of 0.6.

The values of each *Evaluation* are continuously and automatically updated by the system in the following manner. During the learning process the planning module fixes some Milestones in which the student is evaluated exploiting suitable tests. Student answers allow to directly estimate and modify her/his Student Model.

4.1 The Cognitive State

The *Cognitive State* is composed of a set of student beliefs representing "the system knowledge about each student knowledge". If the Ontology (see Section 3) is composed by *n* DCs (*d1,…,dn*), then the Cognitive State of a given student is represented by the set:

Beliefs = {B1, B2,…, Bn},

each *Bi* being a belief so defined:

$Bi = <di, Evaluation_i >$,

where *di* is the i-th Ontology DC, while $Evaluation_i$ is a pair of fuzzy values as defined in (1). If $Evaluation_i = <Degree_i, Reliability_i>$, then *Bi* associates the fact: "the student knows the concept d_i " with the truth fuzzy value $Degree_i$ and the reliability fuzzy value $Reliability_i$. All the beliefs of the set *Beliefs* are assumed to be in logic and. Finally, we remark that all beliefs are represented using or extending the data structures of the Learner Information Packaging (LIP) standard [6].

4.2 The Learning Preferences

Learning Preferences are used in order to evaluate the student's preferences about learning resource and learning styles. Concerning preferred learning resources, we focalise our attention on some LO attributes contained into a few fields of the Educational Metadata category of the IMS learning standard. They refer to generic features of a LO, such as its language, context (the student educational level), age range, typical learning time, interactivity level, learning resource type, difficulty and so on.

Concerning preferred learning style, we classify each student using some of the four couples of learning categories proposed by Silverman and Felder [7] and we propose the Soloman and Felder psychological tests to the student to this purpose. The Silverman and Felder couples of learner categories which we adopt are:

- *Sensing versus Intuitive Learner*. It represents the abstraction level of the documents the user prefers. A *Sensing* student tends to like learning facts, he likes to use the same methods. The student will need more practical case studies. An *Intuitive* student often prefers discovering possibilities and relationships. He likes innovation and dislikes repetition and too much memorisation. The student is more comfortable with abstractions.
- *Visual versus Verbal Learner*. It indicates whether the student prefers textual or visual documents. A *Visual* student remembers best what he sees: pictures, diagrams, flow charts, films, and demonstrations. A *Verbal* student gets more out of words, written and spoken explanations. He has to write summaries or outlines of course material in his own words. Working in groups (through discussion groups, chat or teleconference) can be effective, he gains understanding of material by hearing co-students explanations and he also learns more when he does the explaining.

In order to maintain a descriptive uniformity with the Cognitive State, each feature of the Learning Preferences is described by an atomic fact associated with a pair of fuzzy values of type *Evaluation* as defined in (1). Thus, if we take into account *m* features (*f1, ..., fm*), the student Learning Preferences are represented by the following set:

LP = {P1, P2,…, Pm},

each *Pi* being a preference so defined:

$Pi = <eq_i, Evaluationi>$, (2)

where eq_i is an atomic fact with the following syntax:

$f_i = v_i$.

f_i is an attribute name while v_i is an admissible value for f_i. Let us look to the examples below:

<Learning_Resource_Type = Simulation, <0.7, 0.2>>, (3)

<Learning_Resource_Type = Sensing, <0.8, 0.7>>. (4)

Equation (3) means that the student likes simulations (with a truth fuzzy value of 0.7), but the system up to now has collected very few cues supporting this statement (indeed, the reliability value, 0.2, is very low). Conversely, (4) classifies the student as a Sensing student who prefers practical case studies.

5 Automatic Course Generation and Tailoring

Given a set of Target Concept *TargtC*, the Automatic Course Generation expands *TargetC* by following the relations *HP* and *R* contained in the Ontology and involving any DC of *TargetC*. Note that *SO* relation is not taken into account in this phase. A DC *d* is added to the list *LearningPath* iff it can be achieved by means of *HP* and *R* relations starting from *TargetC* and *d* is not known to the student (which is checked in the Cognitive State).

The *LearningPath* so obtained is then linearized in order to respect Property 1. To this aim, we build the *Augmented Graph*, a graph representation of the Ontology augmented with relations *R* and *SO* following Property 1. For example, if: *HP(d, ..., d1, ...)* and *R(d, d2)* then we explicitly add the relation *R(d1, d2)*, represented in the Augmented Graph with a node link. We are now able to linearize *LearningPath*: starting from the nodes in the Augmented Graph corresponding to *TargetC* we only have to follow the relational links in a pre-order visit of the Graph. The result is a total ordering of *LearningPath*, i.e. an order in which, for each two DCs *d1* and *d2*, it is always possible to state $d1 \prec d2$ or $d2 \prec d1$. This is the order in which the DCs will be presented to the student.

5.1 Presentation Generation

Once the Learning Path (*LearningPath*) has been obtained and ordered, we need to choose suitable LOs for each DC of it. We have two main cases. In the first case we propose to the student high-quality learning material taken by our content providers. This material is paid by the student. Content providers' LOs are off-line indexed with Metadata describing their features using some fields of the Educational Metadata category of the IMS learning standard and other IWT-defined features. Hence, we can easily match these values concerning LOs' descriptions with the corresponding values contained in the student's Learning Preference. In particular, when in (2) the fuzzy values of $Degree_i$ and $Reliability_i$ of $Evaluation_i$ are enough high, we can match the feature name (f_i) and its value (v_i) with LOs' Metadata to choose the most suitable for the present student.

In the second case, we give to the student the possibility to cope the Learning Path with free (ungarateed) LOs previously found on the Web and off-line categorized. LOs' categorization is obtained using the well-known technique called *Term Frequency X Inverse Document Frequency* (*TFIDF*), which computes the frequency of each tem (keyword) contained in a given document (a LO in our case) taking into account the frequency of the same term in all the document training set. We use as training set the set of LOs provided with a Metadata and a link to a DC. Finally, in the training phase, we take into account also textual information which can be retrieved by analysing the LOM attached to the LO.

6 Conclusions

In this paper we have presented the system IWT which realizes the intelligent functionalities of an automatic Course Management System. The system contains a representation of the student's knowledge and her/his learning preferences. Moreover, it is provided with a representation of the didactic domain (an Ontology). Both the student state and the didactic knowledge are written using modern e-learning and Semantic Web standards but satisfy our proposed knowledge representation methodology. This allows the system to efficiently perform inferences on the student input query and compute the best presentation for the didactic concepts she/he chooses. Finally, the system is able to both provide high-quality indexed learning material and learning objects found on the Web.

Acknowledgements

We want to tank Giorgio Di Nunzio and Gianfranco Pernice for their contribution to this work given by studing a suitable mechanism of document categorization for not indexed Learning Objects. This research was supported by the Information Society Technologies project Diogene (IST-2001-33358). Further details can be found on the project Web site: http://www.diogene.org [4].

References

[1] Capuano N., Gaeta M., Micarelli A. and Sangineto E. An integrated Architecture for Automatic Course Generation. *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, Kazan, Russia, 2002.

[2] Capuano N., De Santo M., Marsella M., Molinara M. and Salerno S. Personalised Intelligent Training on the Web: A Multi Agent Approach. Electronic Business and Education, Recent Advances in Internet Infrastructures, Kluwer: Multimedia Systems And Applications Series, vol. 20, chap. 5, 2001.

[3] IWT: Intelligent Web Teacher. White Paper. CRMPA. 2002.

[4] Diogene: a Training Web Broker for ICT Professionals (IST-2001-33358): http://www.diogene.org.

[5] Heflin and J. Hendler. Searching the Web with SHOE. In *AAAI-2000 Workshop on AI for Web Search*. 2000.

[6] http://www.imsproject.com/profiles/lipinfo01.html).

[7] Felder R. M., Learning and Teaching Styles in Engineering Education, *Engr. Education* 78 (7), 674-681, 1988.