Univ Access Inf Soc
DOI 10.1007/s10209-007-0101-0

LONG PAPER

Adaptive course generation through learning styles representation

Enver Sangineto · Nicola Capuano ·
Matteo Gaeta · Alessandro Micarelli

Abstract This paper presents an approach to automatic course generation and student modeling. The method has been developed during the European funded projects Diogene and Intraserv, focused on the construction of an adaptive e-learning platform. The aim of the platform is the automatic generation and personalization of courses, taking into account pedagogical knowledge on the didactic domain as well as statistic information on both the student's knowledge degree and learning preferences. Pedagogical information is described by means of an innovative methodology suitable for effective and efficient course generation and personalization. Moreover, statistic information can be collected and exploited by the system in order to better describe the student's preferences and learning performances. Learning material is chosen by the system matching the student's learning preferences with the learning material type, following a pedagogical approach suggested by Felder and Silverman. The paper discusses how automatic learning material personalization makes it possible to facilitate distance learning access to both able-bodied and disabled people. Results from the Diogene and Intraserv evaluation are reported and discussed.

Keywords E-learning · Automatic course generation and personalization · Learning Styles · Human–Computer interaction

E. Sangineto (✉)
DI, Dipartimento di Informatica, Università degli Studi
"La Sapienza", Via Salaria, 113, 00198 Rome, Italy
e-mail: sangineto@di.uniroma1.it

N. Capuano · M. Gaeta
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

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

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

1 Introduction

The possibility to automatically generate and personalize an e-learning course based on the student's learning necessities and methodological preferences is an interesting open research area with large application perspectives. Currently, the structure and the didactic material of on-line courses are usually statically defined by some teacher (or *learning designer*) who cannot take into account the student's peculiarities in preparing a course for an anonymous, generic learner. On the other hand, modern pedagogical theories underline the necessity to personalize, whenever it is possible, the didactic offer in order to enhance the students' learning characteristics. For instance, Felder and Silverman [15] have shown that each person is characterized by different modalities in the way she prefers to *receive* and *elaborate* information. Receiving information concerns the sensory channels (e.g., the auditory or the visual channel) used in the training activity, while information elaboration can be characterized by different abstraction levels and by a more or less sequential activity. Experimental results carried out by the authors in "traditional" frontal-lessons (in university, engineering courses) showed how matching the students' preferred learning styles with corresponding teaching styles makes it possible to considerably improve the learners' performance. This paper presents an e-learning platform which automatically generates and personalizes on-line courses by selecting and assembling learning material. The platform uses the learner's on-line feedback in order to continuously update her profile. The learner profile contains information concerning both the knowledge degree and learning preferences of the student, and is used by the system for the future course personalization (partially) implementing the Felder and Silverman approach. The system is thought to be used by every kind of learners, exploiting the learner's current knowledge and her learning characteristics in order to facilitate the learning experience. This is true for both able-bodied people with more or less developed reception and elaboration capabilities and for disabled learners having difficulties, e.g., in using some sensory channel. For instance, dyslexic people can take advantage from receiving learning material in which visual information is enhanced with respect to textual information (Sect. 3). Generally speaking, the possibility to personalize a course according to the learner's capabilities and preferences has been pointed out as one of the desirable main feature of an accessible e-learning software application by the IMS Global Learning Consortium, which recently presented a document [10, 21] on this matter. The six principles proposed by the document are.

1. Allow for customization based on user's preferences.
2. Provide equivalent access to auditory and visual content based on user's preferences.
3. Provide compatibility with assistive technologies and include complete keyboard access.
4. Provide context and orientation information.
5. Follow IMS specifications and other relevant specifications, standards and/or guidelines.
6. Consider the use of XML.

The system proposed in this article directly addresses Points 1 and 2 as its main goal. As it will be discussed in Sect. 4.2, the Learning Path built by the system and provided to the user is an abstract description of the course and its prerequisites, which gives a (partial) context and orientation information to the user (Point 4). Finally, standard compliance (Points 5 and 6) is achieved by representing the system's Learning Objects by means of the IMS Metadata Standard [17] and the Ontology with a XML format (see Sect. 4).

More in detail, this article presents a novel approach to automatic course generation and personalization developed in the European funded project *Diogene* (IST-2001-33358), [12], ended in October 2004. The system is now available as a commercial product (called IWT, see [20]) The Web-based e-learning platform built in the Diogene project accepts as an input a student query which specifies the topics the student is interested in. Course generation/personalization takes into account both the student's already acquired knowledge and her learning preferences.

The platform's learning material is composed of a set of Web deliverable objects (textual or hyper-textual documents, interactive exercises, simulations and others), called *Learning Objects* (LOs). The LOs are mostly provided by professional teaching centers and stored in their own servers (the *content providers' servers*). The system's LOs belong to a given didactic domain (e.g., "Computer Science", "Math", "History", etc.). The system represents the didactic domain by means of an *Ontology*, composed of a set of *Domain Concepts* (DCs) linked by means of specific pedagogical relations (see Sect. 4.1). Each DC corresponds to a topic of the domain and can be associated with one or more LOs. The relations among DCs allow the system to generate an abstract description of the course called *Learning Path*. A Learning Path is composed of all the DCs necessary in order to learn the topics contained in the student's query. The actual course, called *Presentation*, is then built starting from the Learning Path and selecting those LOs whose type is the closest possible to the student's learning styles, following the approach suggested by Felder and Silverman [15] (see Sect. 2). In fact, Felder and Silverman propose to categorize students in different learning categories (*learning styles*) and to use similar categories for the classification of the learning material (*teaching styles*). By matching learning styles with teaching styles, it is possible to improve the student's learning performance by providing the didactic material most suitable for individual learning preferences and capability.

In this process, the student's representation, i.e., the system's *Student Model*, as well as the learning styles-teaching styles matching, play a fundamental role. Teaching styles are off-line manually associated with each LO using the information represented in the LO's *Metadata*. The student's learning styles can be initialized either directly by the user, or indirectly using the results of the on purpose Felder and Soloman on-line psychological test [15]. The first solution is less accurate than the second, but makes the initialization session lighter for a given learner. The system provides also default initialization modalities for people who do not care to set up their learning preferences. However, whatever initialization is chosen, the initial values are automatically updated by the system using statistics extracted from the student's feedback. The statistical analysis is based on the results of special interactive LOs of type *test*, which are included in the Presentations and delivered to the student.

The combination of static pedagogical information (the Ontology) and a dynamic description of the user model based on a statistical analysis, makes it possible to select and assemble LOs without any human intervention.

The paper is organized as follows. The following section provides an overview of the main e-learning adaptive systems. The Felder and Silverman's pedagogical theory is briefly introduced in Sect. 3. Section 4 presents an overview of the entire system, showing its knowledge representation structures and the course generation and personalization process. Section 5 presents the Student Modeling aspects of the system, and discusses how the student's representation can be automatically updated using statistics on the provided feedback. Section 6 presents some concrete examples of courses generated using a large Ontology (having more than 1,500 elements) and shows different user interface components and other facilities of the system. Finally, in Sect. 7 reports and discusses the evaluation results of the platform. Section 8 concludes the paper.

2 Background

The main characteristic of the system proposed in this paper is its adaptation capabilities and its active participation in the learning process, as opposed to common e-learning platforms usually playing only the role of passive LOs' containers. For instance, the well known Ariadne platform [14] is based on a digital library of LOs which are indexed by using educational metadata standards (as in the work presented here). However, the Ariadne platform only aims at building a library of reusable learning components for the sharing of such components among different (human) teachers. The library is therefore a passive repository, and there is no automatic building of courses nor any adaptation according to the user profile.

This situation is very common in the new generation of Web-based learning platforms, especially in the commercial systems (e.g., [6, 26]). Indeed, they are born to support modern distance learning (e.g., in long-distance university courses), but they usually only work as passive content containers, in which teachers and students can exchange documents and information. In [32], the authors propose a system able to collect the students' feedback during the entire duration of the course in order to give the teacher the possibility to adapt the course itself to the classroom needs. Nevertheless, no automatic adaptation is expected and the feedback is analyzed by hand.

More sophisticated systems provide automatic tools for material management, classification and retrieval, user modelling, course tailoring, automatic or semi-automatic exercise corrections, etc. For example, at the University of Genoa (Italy), a Learning Management System (LMS) has been developed which merges typical LMS functionalities with Artificial Intelligence Agent-based methods [3]. The LMS tools range from Learning Objects authoring to classroom and group management, learners' competencies management, learner assessment and tracking for accreditation and certification, learning path personalization, tutoring/mentoring service, Web and streaming-based communication and computer supported collaborative learning. All these functionalities have been fully implemented and the platform is now going to be integrated into the Genoa University information system. On the other hand, the same authors have developed a prototype system with more advanced automatic tools based on an Intelligent Agent approach. Learners can delegate their learning processes to a population of agents. Indeed, each learner owns (and sometimes shares) a set of agents whose main tasks are profiling, searching, indexing, retrieval, control and supplying of LOs to the learner.

In [2] the authors propose a system for on-line testing which is able to:

- maintain a test database independent of what the tests will be used for;
- present the tests to the students starting from which items to select based either upon the teacher's decision or upon the system's rules, and ending with a presentation suitable for a particular situation;
- correcting and collecting results to be used by the student or the teacher for statistical purposes regarding course evaluation or item validation.

In [30], a complete system for knowledge management in an e-learning scenario is presented, thus exhibiting a greater automation degree with respect to previous solutions. The system is composed of the following entities:

- a course database;
- semantic metadata attached to each course;
- a Knowledge Navigator (KN);
- an Automatic Link Generator (ALG);
- an Automatic Metadata Generator (AMG).

The used learning material (i.e., the LOs of the course database) is composed of typical courses or lectures archived in any LMS. Moreover, the authors use a Metadata Catalogue to store the associations between each LO and its (set of) topic(s). Topics are organized in an Ontology expressing membership and subclass relations, as well as other relations such as pre-requisite relation, conceptual similarity relation, etc. The authors have chosen to build their Ontology on the basis of two assumptions: that an ontology needs to collect a large consensus in its community's specific area in order to be valid; and that the system focuses on Computer Science topics. For these reasons, they have selected and extended the Association of Computing Machinery (ACM) "Computing Curricula 2001 for

Computer Science". Furthermore, an important platform instrument is the Knowledge Navigator. The KN allows browsing of ontological relations. For instance, the learner can find the topic which is related to a given LO and then find either similar topics or pre-requisite topics whose study can be important for the LO itself. Moreover, while the KN is a tool enabling "explicit navigation of the Ontology", the ALG is a tool for "implicit navigation". The AGL has the responsibility of automatically performing a limited exploration of the Ontology, in the region near to the LO which the student is presently using. As a result, it produces a list of LOs that are related to what the student is presently studying. This list is made available to the student by means of a suitable button dynamically attached to the LO's Web pages. Finally, the authors suggest a (not yet implemented) tool for automatic or semi-automatic LO Metadata generation (AMG). The AMG sketch proposal is composed of standard techniques for document classification which can automatically provide a (tentative) classification of a new LO in order to facilitate human work in LOs' Metadata building. Even if this proposal shows an interesting system exploiting semantic relations among LOs, there is no customisation of the results according the student knowledge and learning preferences, because all the knowledge of the system is based on the didactic domain description without a student modeling process. Moreover, although the student can use the ALG and the KN tools for either implicit or explicit navigation among the topics of a given didactic domain, the system does not generate a complete, well defined course (including prerequisite arguments) but these instruments need to be controlled by the student who has the responsibility to possibly select the interesting related concepts.

In [35], Trigano and Giacomini propose a system called CEPIAH for helping teachers to implement pedagogical web sites and produce on-line courses. Using CEPIAH, the teacher can automatically generate educational Web site structures, adding the pedagogical contents into these structures, then visualize, manage and participate to the courses. The students can visualize and participate to the courses using the navigator integrated in the system. Both the Web sites and the courses are generated starting from the answers given by the user (the teacher) to two interactive questionnaires: a pedagogical questionnaire and a GUI related questionnaire. Within this conceptual framework, the authors take into account two major aspects, namely: the Human–Computer Interface (HCI) of the Web sites (the colours, the shapes of menus and buttons, etc.) and the modeling, by means of IMS Learning Design, of teaching scenarios which are based on various pedagogical approaches. For instance, the course structure automatically generated by the system online is based on teaching scenarios which integrate, according to the case, the features of different pedagogical theories such as behaviourism, constructivism, socio-constructivism, etc. The system's inference mechanism is based on a rule-based engine. The rules specify the way in which pedagogical models are assembled. The modules are IMS Learning Design components, which can be created either at generation time or conceived beforehand and recorded in XML files respecting the standard.

Benayache and Abele [4] developed the eMEMORA system for automatic course generation. In the eMEMORA system, resources, information and other knowledge are organized in a "course memory". This memory can be accessed by teachers when they want to re-use resources, as in a thematic resource base, as well as by learners who can directly use the memory for their learning necessities. eMEMORA is based on two Ontologies, the first (Application Ontology) describes the specific didactic domain, while the second (Domain Ontology) represents the teaching resources (persons, documents, etc.). The latter uses some concepts of the Learning Object Metadata standards. The system can also describe pedagogical relations (e.g., the "pre-requisite of" relation) using Topic Maps. Each learning resource is directly attached to one or more topics by an "Occurrence link".

However, in both the CEPIAH and in the eMEMORA systems automatic facilities are focused on the course representation and manipulation, while they do not take into account any information regarding the specific student, which on the other hand could help the course construction process by choosing personalized learning material and pedagogical strategies.

Weber et al. in [37] present an adaptive course generation system called NetCoach. Similarly to the system presented in this paper, the NetCoach knowledge base represents the concepts of the didactic domain the system is specialized on and, as it happens with Diogene, NetCoach is able to exploit prerequisite relations among the concepts in order to create a sequential ordering of the concepts needed by the student for her learning goals. Moreover, the system provides another type of relation: the "inference" relations. A concept A is linked to the concept B by means of an inference relation when it holds that every time a learner knows A, the learner also knows B. In the Student Model of a given learner, those concepts for which the student's knowledge has not been directly tested by the system, but it is only inferred via inference relations, are specially marked. However, it is not clear if it is always reliable to assert the student's knowledge of a concept which is generically related to another concept successfully studied by the student. Moreover, differently from Diogene (see Sect. 4.1), the knowledge representation proposed in NetCoach contains neither hierarchical nor pedagogical ordering relations, which have been shown to be necessary components of a didactic description of a (non-trivial) domain [11, 22]. Indeed, the aim of NetCoach is to provide the student with possible warnings concerning the learning path choosen inside a prefabricated curriculum (e.g., a digital book made of chapters and subchapters linked with the above mentioned relations), which is a checking activity, while the aim of Diogene is to build a course by assembling heterogeneous learning material (LOs). The latter is a construction activity which needs a deeper description of the didactic relationships among the domain concepts. Finally, even if the Student Model proposed in [37] is very flexible and permits the learner herself to directly modify it ("adaptable facilities" [37]), it does not describe the student's learning preferences and there is no possibility to customize the courses ("adaptive facilities" [37]) based on a general pedagogical strategy which accounts for the learner peculiarities, as in the case of the Diogene system.

A Student Model based on the same principles of [37] is used in [31] for ECSAIWeb, an environment for the automatic creation of Web-delivered courses. The ECSAIWeb engine is a production-rule based system in which LOs are described by means of *learning units*. A learning unit is composed of a "contents" part containing the references to a given LO, a "pre-conditions" part which defines the LO's prerequisites, and a "post-actions" part specifying how the system must change the Student Model status after the student has passed the test corresponding to the unit's LO.

Koper et al. [5, 25] emulate the behaviour of a swarm of insects which are able to self-organize a path by means of the aggregation of multiple pheromone tracks. In fact, the authors propose to represent in a transition matrix the most frequent transitions made by previous learners from every LO *A* to every LO *B* of the system's repository. This matrix is used in order to recommend to other learners possible learning paths composed of a sequence of LO transitions, each transition being chosen using the statistical information contained in the transition matrix. Of course, this recommendation system is based on the possibility to monitor a sufficiently large number of correct autonomous learner behaviours.

WebCT [36] is another example of on line course generation system. The WebCT courses are composed of HTML pages linked in order to suggest to the student a learning path.

The TopClass system [34] builds courses by assembling Units of Learning Material (ULMs), such as pages, tests or hierarchically structured ULMs. TopClass assesses the students' knowledge by tracking their results in the interactive tests.

The main novelties of the system proposed in this paper with respect to the abovementioned learning management platforms concern the possibility to automatically generate a course taking into account pedagogical knowledge about the didactic domain, as well as the student's knowledge degree about the specific topics of the domain and individual learning preferences. The latter feature allows the system to customize courses by selecting LO types according to the student's learning style [15], which is automatically updated by the system using a statistic analysis of the student's feedback. The next section briefly discusses the Felder and Silverman approach for "traditional" frontal lessons, while in the rest of the paper presents how it has been (partially) implemented in the on-line, automatic Diogene system.

3 The Felder and Silverman's pedagogical categories

In [15], the authors propose five pairs of student's categories (which will later become four pairs). These categories derive both from previous psychological studies (such as the Jung's well-known theory of psychological types) and from some empirical observations of the authors. The most important observation is that each person usually shows individual preference for one or more modalities in the way information is received and elaborated. For example, one can prefer to receive information by means of either the visual sensory channel or the auditory sensory channel. In the former case, the individual learning process is better suited for visual material (e.g., pictures), in the latter for sounds or texts (which are composed of words, then strictly related to the auditory information processing). It is important noticing that the studies carried out by the authors showed that this is true not only for disabled people, who can have some problem concerning one or more sensory channels, but for every individual, who always show a more or less noticeable predisposition for a given channel.

For this reason, Felder and Silverman propose to categorize learners in different learning categories. Each category is characterized by two opposite attributes (e.g., Visual versus Verbal) which represent the extremes of the range of possibilities for that category (a student can be more or less visual or verbal). Moreover, for each learning category, there is a corresponding teaching category, which indicates the type of teaching most appropriate for the corresponding preferred way to receive-process information during learning. The Felder and Silverman's main four categories are the following:

- *Sensing versus Intuitive Learner*. It represents the abstraction level of the learning material the student prefers. A Sensing student tends to like learning facts and using the same methods repeatedly. The student will need more practical case studies. An Intuitive student often prefers discovering possibilities and relationships. She likes innovation and dislikes repetition and too much memorization. The student is more comfortable with abstractions.

- *Visual versus Verbal Learner*. It indicates whether the student prefers auditory (textual) or visual documents. A Visual student remembers best what she sees: pictures, diagrams, flow charts, movies, demonstrations, etc. A Verbal student gets more out of words, written and spoken explanations, and often write summaries or outlines of course material in own words. Working in groups (through discussion groups, chat or teleconference) can be effective. The student gains understanding of material by hearing co-students explanations, and also learns more when doing the explanation.
- *Active versus Reflective*. It indicates how the student prefers to process information: actively (through engagement in activities or discussions) or reflectively (through introspection).
- *Sequential versus Global*. It indicates how the student progresses toward understanding, either in continual steps (sequentially) or in large jumps, holistically (globally). Sequential students prefer sequential explanations while global students usually prefer an initial overview of the involved topics which possibly describes to them the most important steps and relations they are going to study.

The first two categories regard the way in which people prefer to receive information during the learning process, while the other two classes regard the way in which people prefer to elaborate it. In its initial proposal, Felder and Silverman also had included a fifth category (Inductive vs. Deductive learning/teaching) which has been subsequently discarded.

It is worth noting that the four pairs of categories are not mutually excluding. On the contrary, each learner can be classified using a combination of values, one for each of the four categories. For instance, a given learner can be very intuitive, slightly more visual than verbal, strongly reflexive and indifferent with respect to the sequential-global choice.

The Felder and Silverman's categories have been adopted in the work reported here for two reasons. First of all, the underlying approach is based on a sufficiently large experimentation which has validated the proposed classes on an engineering student population. Second, although other approaches are maybe based on a stronger cognitive model formalization, the Felder and Silverman's theory provides some useful pragmatic instruments to customize teaching depending on the student's profile.

Of course, the proposed automatic course generation platform cannot take into account all the teaching suggestions proposed by Felder and Soloman. Indeed, some of these suggestions have been explicitly thought for human made frontal lessons. Nevertheless, if one is able to categorize each student according to learning categories and associate (off-line) the most suitable teaching categories with each LO of the system's database (see Sect. 4), then an efficient and effective way is obtained to select the most appropriate learning material for each given student.

Finally, Felder and Soloman later proposed a psychological test [16] for learners' categorization which is also (optionally) provided to the Diogene's students in their first approach to the system (see Sect. 5).

4 Automatically building a course

The Diogene platform is able to act as an intermediary between the learners and different "content providers", which are specialized training organizations providing learning material. The latter is suitably indexed by means of common Semantic Web standards and following a knowledge representation paradigm developed during the project [7, 8] (see Sect. 4.1). The knowledge representation framework allows the system to efficiently assemble and personalize the learning material in order to answer to the student's training requests.

The knowledge representation framework the system is based on is composed of three different sub-structures, respectively used for the representation of the student (the *Student Model*), the Learning Objects (indexed by means of *Metadata* coded using the well-known e-learning standard "IMS Metadata Standard" [17]) and the *Ontology*, which describes the pedagogical relationships among the Domain Concepts. The Ontology is structured using the relations presented in Sect. 4.1 and stored in a XML file. When the learner submits a training query (i.e., a set of DCs of interest) to the system, Diogene generates a course by choosing and assembling the available Los, and taking into account the didactic information encoded in the Ontology, as well as the student's profile. Figure 1 schematically

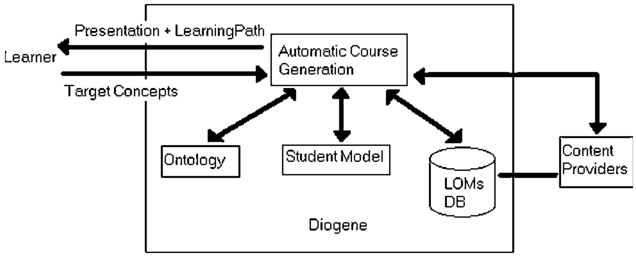


Fig. 1 The Diogene platform's architecture

shows the main components of the Diogene platform's architecture.

The pedagogical information of the Ontology includes prerequisite, decomposition and ordering relations among DCs, which altogether allow the system to generate a set of DCs (called *Learning Path*) starting from the DCs contained in the student's query (*Target Concepts*) and including and ordering all the necessary other DCs. The Student Model is used in order to include in the Learning Path only those DCs not already known by the student. Finally, the system selects a set of LOs explaining all the concepts of the Learning Path by choosing those LOs which best fit the student's learning styles as described in the Student Model.

The following subsections provide some more details on the course generation strategy. Section 5 we focuses on the Student Model representation and on the adaptive capabilities of the system, which makes it possible to use the proposed learner representation for implementing the Felder and Silverman's pedagogical approach in a completely automatic e-learning platform.

4.1 Representation of the Didactic domain

The system's knowledge about a specific didactic domain (e.g., math analysis, computer science, ancient history, etc.) is represented by means of an *Ontology* describing the main concepts of the domain and their pedagogical relationships. The Ontology is represented using a semantic network, in which each node corresponds to a *Domain Concept* (*DC*), and different nodes can be linked by oriented edges representing the Ontology relations. Such a representation is common in e-learning systems. Nevertheless, the novelty of this proposal is based on the conjunction of two points:

- The Ontology is an abstract description of the domain, independent of the specific Learning Objects of the system's database.
- The Ontology's relations are based on a minimal set of relations able to describe the most important pedagogical relationships necessary for (automatic) course generation.

Concerning the first point, while common Ontologies are usually directly represented by (some of) the attributes of the Learning Objects' Metadata, the Diogene's Ontology is described without any reference to any specific learning material. It is an abstract description of the topics the didactic domain is composed of, and of their relations. These relations do not depend on the specific material used to explain the topics. For instance, in the math analysis domain, limits are a prerequisite argument for derivatives, without referring to any mathematical text. Vice versa, pedagogical relations represented in the Metadata of the specific Learning Objects usually depend on the didactic content and methods of the Learning Objects. For instance, a given Learning Object containing a chapter of a book could be a prerequisite for a subsequent chapter of the same book (see, e.g., [37]). In this case, the prerequisite relation is not based on the topics but on *how* such topics are explained. What is needed is a description of the domain which is independent of the specific Los, since the process of LOs' selection needs to be dependent only on the student's learning styles once the Learning Path has been built (see Sect. 5).

A pedagogical knowledge representation which satisfies the two abovementioned properties is based on the following DCs' relations.

- *HP* (*Has Part*): $HP(x, y_1, y_2, ..., y_n)$ means that the concept x is composed of the concepts $y_1, y_2, ..., y_n$, that is to say: to learn x is *equivalent* to learn the set of Concepts $\{y_1, y_2, ..., y_n\}$. The elements $y_1, y_2, ...$ and y_n are all necessary, none excepted.
- *R* (*Requires*): $R(x, y)$ means that to learn x it is necessary to have already learnt y. This relation establishes also a constraint on the Domain Concepts' order in a given Learning Path (y must precede x).
- *SO* (*Suggested Order*): $SO(x, y)$ means that it is preferable to learn x and y in this order. Note also that this relation establishes a constraint on the DCs' order, but in this case it is not necessary to learn y if the learner is interested only in x.

Furthermore, the following relation links 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).

The *EB* relation is a one-to-many mapping, and the same for its inverse. $EB(d,l_1)$, $EB(d,l_2)$ means that l_1 and l_2 are two alternative LOs explaining d, while $EB(d_1,l)$, $EB(d_2,l)$ means that l explains both d_1 and d_2.

The *HP* relation is used to describe the decomposition of an abstract concept (e.g., Math Analysis) into a set of more specific sub-concepts (e.g., Limits, Derivatives, Integrals and Series, see Fig. 2). The relations *R* and *SO* are *inherited* through the relation *HP*. Formally, it holds that:

Property 1 Ordering Inheritance

1. If $HP(d, ..., d_1, ...)$ and $R(d, d_2)$ then $R(d_1, d_2)$.
2. If $HP(d, ..., d_1, ...)$ and $SO(d_2, d)$ then $SO(d_2, d_1)$,

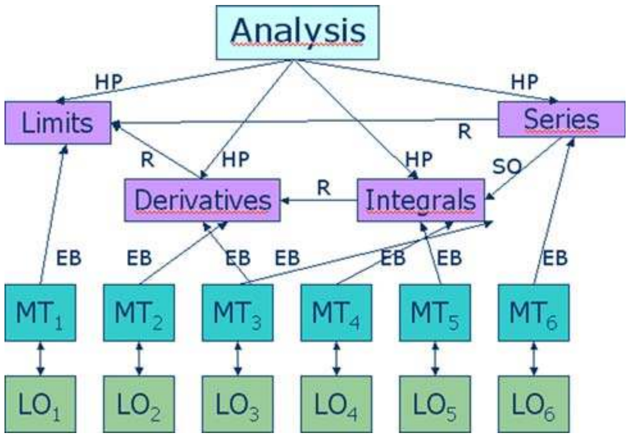


Fig. 2 A simple example of knowledge representation for the didactic domain "Math". LO_1, LO_2, etc., are the system's Learning Objects, respectively represented by the Metadata MT_1, MT_2, ... Note that the relation *SO*(*Series, Integrals*) is weaker than the relation *R*(*Series, Limits*). In fact, the former represents a pedagogical suggestion of the Ontology designer while the latter is a mandatory pre-requisite relation. Finally, *Limits, Derivatives, Integrals* and *Series* in this example are all atomic concepts while *Analysis* is not

where, coherently with the definition of $HP(x, y_1, y_2, ..., y_n)$ as a *decomposition* of x in $\{y_1, y_2, ..., y_n\}$, it is always assumed that:

- if $HP(d, y_1, ..., y_n)$ and $R(d, d_2)$ then $y_i \neq d_2$ $(1 \leq i \leq n)$ and
- if $HP(d, y_1, ..., y_n)$ and $SO(d_2, d)$ then $y_i \neq d_2$ $(1 \leq i \leq n)$.

Definition 1 A *Presentation* P is an ordered list of LOs $(l_1, ..., l_n)$ with the following properties:

1. The union of the LOs of P is sufficient to explain to the student all the Target Concepts her query is composed of.
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 is composed of LOs whose teaching style matches as much as possible the learning style of the student represented in her Student Model.

While Points 1 and 3 of the above definition are self-explanatory, Point 2 needs some remarks. It is a consequence of both the transitivity property of the order relations R and SO and the above defined Property 1. It defines a partial order on the DCs. As a consequence, LOs belonging to the same Presentation have to respect this partial order. If, for instance, a Presentation contains the LOs l_i and l_j, which explain, respectively, the Concepts "Derivatives" and "Limits", then l_j has to precede l_i. The same situation holds when l_i and l_j explain DCs not directly linked to each other by an order relation (Requires or Suggested Order), but which are components (by means of the Has Part relation) of DCs recursively linked by an order relation (see Step 2.c above).

Definition 2 A concept x is named *atomic* if there is no concept y such that $HP(x, ..., y, ...)$.

It was chosen to link LOs (by means of the relation *EB*) only with atomic DCs. Indeed, a non-atomic concept x is composed of sub-concepts (e.g., y, z) and to learn x is equivalent to learn its components y and z (see the definition of *HP*). Only when no further decomposition is possible by means of *HP*, then learning material is attached to the resulting DC set. This is not a limiting constraint, because a possible LO l explaining the entire concept x can be automatically attached to *both* y and z, satisfying the constraint. In Sect. 5 it will be seen that LOs possibly dealing with DCs not contained in the Learning Path can be easily excluded from the final Presentation, thus avoiding to provide the student with material concerning not requested topics. On the other hand, the Presentation construction process avoids selecting two different LOs for the same DC (see Sect. 5).

Finally, there are no subset relations. Indeed sets are not represented at all. There is no class definition and all the elements of the representation (the DCs) are (implicit) instances of the same class (the didactic domain). The simple reason is that representing sets is not necessary. As a consequence, the proposed knowledge representation schema is not a "classic" semantic network, since traditional semantic networks usually provide the definition of classes and the subset relation among classes.

The necessity to have a minimal set of well defined relations among concepts is shown by the work of Dicheva and Dichev [11], who have conducted a study to find out what are the main difficulties that authors of educational Topic Maps face. They compared the behaviour of a number of instructors invited to use a Topic Map authoring tool (TM4L) for the description of nine different domains, including Modern Algebra, Discrete Mathematics, and Software Systems. The results, presented in [11], show that one of the main difficulty is the subjectivity in the choice of possible relations among topics, which leads to problems for the reusability and interoperability of the created Topic Maps. The solution to this problem proposed in [11] is to provide authoring tools for learning domain descriptions in which the set of available relationships among concepts is pre-established and minimal. In the article five relations are proposed: three hierarchical relations ("superclass-subclass", "whole-part" and "class-instance") and two "horizontal" relations ("related to" and "similar to"), suggesting that horizontal relations should have a "weak semantics", i.e., their meaning does not need to be exactly defined. In Diogene only one hierarchical relation (*HP*) and two horizontal relations (*R* and *SO*) are used. In particular, concerning the horizontal relations, it is worth noticing that the authoring facilities provided by Diogene for building a domain Ontology are thought for its use in subsequent *automatic* course generation. Vice versa, common authoring tools such as TM4L only aim at supporting human teachers in the domain Ontology navigation in order to *manually* select a set of arguments and LOs for a course. For this reason, horizontal relations employed in automatic course planning need to be associated with a *strong operational semantics*, because their use cannot rely on a human interpretation.

Similar limitations hold in common Learning Design formal approaches, which mainly aim at proposing a method for human-made course construction. For instance, the "Educational Modeling Language (EML)" [24, 33] formally defines a software engineering-like approach to assemble LOs in a *unit of study*. Each unit of study expresses the semantic relationships among the contained LOs. Its main components are LOs, learning objectives, prerequisites, and learners' and facilitators' roles in the learning activity, which are described by means of Unified Modeling Language (UML) diagrams. Paquette [27] proposes an authoring tool (MOT+) and some learning design principles (the MISA method [28]) to support the creation of learning designs following pedagogical needs. Both EML and MISA take into account pedagogical requirements in defining a framework expressing the relationships among Los, but the richness of the proposed description methods and the lack of a precise operational semantics make such instruments less suitable for automatic course generation purposes.

In [23], Kitamura and Mizoguchi develop an ontological framework composed of different layered ontologies and a rich set of concepts and relations for helping learning designers in sharing knowledge. Mizoguchi et al. in [22] propose an Ontology to support Japanese IT teachers in defining the learning objectives of their technological courses. In this work the authors argue that the two main necessary hierarchical decompositions of concepts are, respectively, the "is-a" and the "part-of" relation. The former is used to describe the topics of a given domain and how they are specialized in superclasses and subclasses. On the other hand, the second relation is used to describe the necessary detailed subcomponents of a given concept *C*, which a learner must learn to know *C*. The "is-a" relation links *C* with possible subordinate concepts, while the "part-of" relation provides an operational description of what are the components of *C* necessary to be understood to learn *C*. Once again, the operational-oriented knowledge representation approach of Diogene requires only the "part-of" relation, which corresponds to the *HP* relation above mentioned.

The following subsection shows how the proposed knowledge representation is used by the system for the Presentation building.

4.2 Presentation construction

The first step in the automatic construction of a Presentation is to determine the set of all the necessary DCs for a given student's query. As mentioned above, such a set is called Learning Path. Starting from the set of Target Concepts (*TargetC*), the Automatic Course Generation algorithm follows all the *HP* and *R* relations of the Ontology and iteratively includes new DCs in the Learning Path. Note that the *SO* relation is not taken into account in this phase. A DC *d* is added to the list *LearningPath* (initially set to *TargetC*) if and only if:

1. it can be reached by means of a sequence of *HP* and/or *R* relations starting from the DCs included in *TargetC* (i.e., *LearningPath* is the transitive closure of *TargetC* with respect to *HP* and *R*) and
2. *d* is not already known by the student (this fact can be checked using the Cognitive State, see Sect. 5).

The *LearningPath* so obtained is then pruned excluding the non-atomic DCs. As a final step, it is ordered with respect to Property 1 (Sect. 4.1). To this aim, an *Augmented Graph is built*, i.e., a graph representation of the Ontology which is explicitly augmented with relations *R* and *SO* following Property 1. For instance, if: $HP(d, ..., d_1, ...)$ and $R(d, d_2)$, then an edge linking the nodes corresponding to d_1 and d_2 is explicitly added to the Augmented Graph and labelled with the relation $R(d_1, d_2)$ (see Figs. 3, 4). Note that this operation is performed off-line, once for a given Ontology, and is $O(n^2)$, where n is the number of DCs of the Ontology.

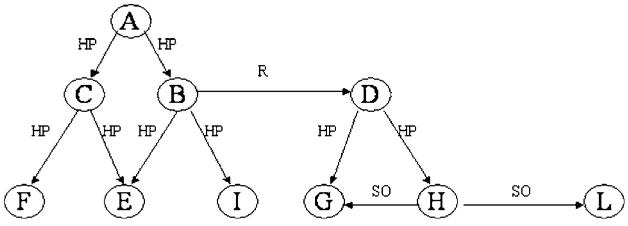


Fig. 3 An example of graph representing an Ontology

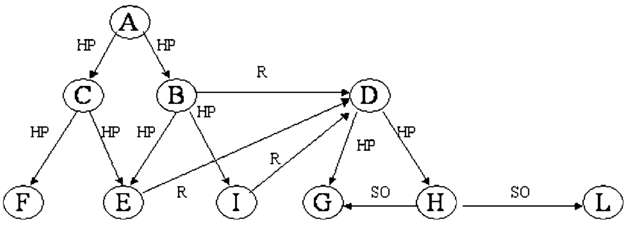


Fig. 4 The Augmented Graph of the Ontology shown in Fig. 3

The next step is to order the (on-line) obtained *LearningPath*: starting from the nodes in the Augmented Graph corresponding to *TargetC*, the edges of the graph are followed in a depth-first visit, which is a standard technique for topological ordering of graphs [9]. The result is a total ordering of *LearningPath*, i.e., an order in which, for every two DCs d_1 and d_2 ($d_1 \neq d_2$), either $d_1 \prec d_2$ or $d_2 \prec d_1$. This is the order in which the DCs will be presented to the student. Note that in this phase the relation *SO* is also used.

Figure 3 shows an example of Ontology. Suppose a student inputs the system with the following Target Set:

$TargetC = \{C, B\}.$

Then, the corresponding Learning Path is:

$LearningPath = \{F, E, I, G, H\}.$

Figure 4 represents the Augmented Graph corresponding to the Ontology of Fig. 3. Using such a graph, a total order can be given to the Learning Path, resulting in the following ordered list of concepts:

$LearningPath = \{F, G, H, E, I\}.$

Note that, from a given Partial Ordered Set (such as those defined by an Augmented Graph) it is possible to derive more than one total ordered set. For instance, in this case, also {*G,H,E,I,F*} would have been a solution consistent with the ordering constraints defined in Sect. 4.1. Since the ordering process is operationally implemented by means of a depth-first visit of the Augmented Graph, the solution selected by the system is uniquely defined. Different ordering relations can be defined by the Ontology designer using the *SO* (or the *R*) relation.

Furthermore, it is worth noticing that the construction of the Augmented Graph is an important preliminary step. A depth-first visit of the original graph of the Ontology (without the additional edges for Property 1) is not able to produce an ordering of the *LearningPath* which satisfies the Property 1 and the semantics of the relations. For instance, the (semantically) inconsistent loop shown in Fig. 5 (left) can not be detected by a depth-first visit of the original graph following the edge orientations. The reason is that the *HP* relation is an *equivalence* relation and not an order relation, and it needs to be dealt with differently from the *R* and *SO* relations.

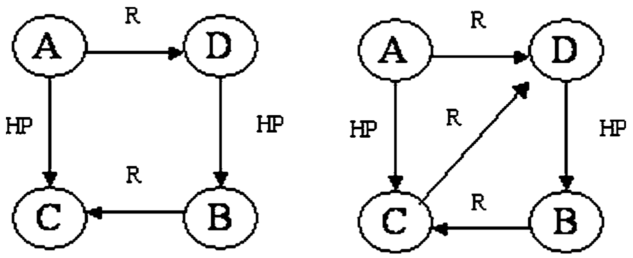


Fig. 5 *Left* an example of Ontology whose semantic inconsistence cannot be found looking for loops in its graph representation. *Right* the Augmented Graph of the same Ontology in which a loop does exist

The Ontology consistency check is performed off-line and gives important feedback to the Ontology Designer. For instance, the Ontology built during the Diogene project (see Sect. 6) is composed of more than 1,500 DCs, and it has been built by three different organizations, belonging to three different European countries (Bulgaria, France and Greece), each dealing with different parts of the domain representation, which have finally been merged. Such a large Ontology could not have been constructed without automatic consistency check tools, which in fact have frequently helped the Ontology Designers in correcting their errors both in the "local" construction phases and in the merging phase.

Once the Learning Path has been obtained and ordered, suitable LOs need to be choosen for each of its DCs. The system proposes to the student learning material taken from the content providers' repositories. Each content provider's LO is indexed off-line by means of Metadata describing its teaching styles (see Sect. 4). Hence, the system can match the LOs' teaching styles with the student's learning styles in order to select the best LOs for a given Learning Path. The next section discusses how the learning styles are represented in the Student Model, how the teaching styles are associated with every Learning Object, and how these features are matched by the system in searching for the best Presentation.

5 The student model

The Diogene's Student Model is composed of two modules: the *Cognitive State* and the *Learning Preferences*. The former describes the knowledge degree achieved by each student about every DC of the Ontology. This evaluation regards both previously acquired student knowledge and skills learnt using the Diogene's platform. On the other hand, the Learning Preferences module represents information about the student's learning styles (Sect. 3) and other student's data which have to be taken into account during the LO selection process (see below).

The Cognitive State is represented by the set $CS = \{B_1, B_2, ..., B_n\}$, each B_i being a triple of the type:

$$B_i = \langle d_i, e_i, N_i \rangle, \tag{1}$$

where d_i is the ith Ontology's Domain Concept, e_i is a fuzzy value ($e_i \in [0, 1]$) representing the current knowledge degree reached by the student about the topic $d_{i,}$ and N_i is the number of test evaluation performed so far by the platform on the concept d_i (see Sect. 5.1). For instance, the triple: ⟨"Private Variables", 0.6, 2⟩ means that the student has shown a sufficient degree of knowledge or comprehension of the concept "Private Variables", and this has been tested twice by the system.

When a given student first interacts with the platform, for every DC d of the Ontology, the system includes in the student's Cognitive State the triple $\langle d, 0, 0\rangle$, thus making the assumption that d is completely unknown by the learner. This evaluation is maintained until a test on the concept d is evaluated: the result of the test will update the system's estimation on the student knowledge about d (see Sect. 5.1).

In order to decide if the concept d is known to the student and has to be included or not in the Learning Path (Sect. 4.2), fixed threshold K is used (currently, $K = 0.5$). If d is a DC, such that $\langle d, e, N\rangle \in$ CS and $e \geq K$, then d will not be included in the Learning Path during its construction. It is worth noticing that, while fuzzy values are commonly used to represent knowledge degrees, in this case a decision needs to be taken on whether to include d in the Learning Path or not, which leads to the necessity of a "crisp" decision threshold. In fact, since each DC of the Learning Path will be subsequently "transformed" into Los, it is necessary to decide whether a given LO for a given topic must be delivered to the student or not.

The Learning Preferences (LP) are represented by the set $LP = \{P_1, P_2, P_3, P_4\}$, where:

$$\begin{aligned} P_1 &= \langle \text{``}\textit{Educational Context}\text{''}, v_1 \rangle, \\ P_2 &= \langle \text{``}\textit{Age}\text{''}, v_2 \rangle, \\ P_3 &= \langle \text{``}\textit{Languages}\text{''}, v_3 \rangle, \\ P_4 &= \langle \text{``}\textit{Learning Styles}\text{''}, \textit{Categories} \rangle. \end{aligned} \tag{2}$$

The Property P_1 indicates the educational context of the learner, as proposed by the IMS Metadata Standard (IMS, 2001). Indeed, even if the standard is thought to be used for the LO representation, it is also used by Diogene as a reference for the representation of some student's attributes that are matched by the system with the LOs' Metadata in the course generation and personalization process. The values of v_1 range in the set of admissible values selected by the standard (i.e., strings such as: "University", "Secondary School", "Continuous Formation", and others). Similarly, the Properties P_2 and P_3 indicate, respectively, the age and the main languages of the learner. The values of v_2 are integers, while v_3 is a set of at most three strings chosen among the IMS vocabulary (e.g.: "English", "Italian", "French", and so on [17]). Finally, Property P_4 describes the learning styles preferred by the student following the Felder and Silverman's categorization (Sect. 3), and is represented by means of the quadruple:

$$\textit{Categories} = \langle C_1, C_2, C_3, C_4 \rangle, \tag{3}$$

each C_i ($1 \leq i \leq 4$) being in turn a pair of the type: $C_i = \langle T_{i,} e_i \rangle$, where T_i represents one of the four possible types of learning style categories ($T_i \in \{$"*Sensing-Intuitive*", "*Visual-Verbal*", "*Active-Reflective*", "*Sequential-Global*"$\}$) and e_i is a fuzzy value ($e_i \in [0, 1]$) representing the current system's estimation of the student's preference with respect to the category T_i. For instance, a student whose Learning Styles are given by:

$$\begin{aligned} &\textit{Categories} = \langle \langle \text{``}\textit{Sensing-Intuitive}\text{''}, 0.7 \rangle, \\ &\langle \text{``}\textit{Visual-Verbal}\text{''}, 0.3 \rangle, \langle \text{``}\textit{Active-Relative}\text{''}, 0.6 \rangle, \\ &\langle \text{``}\textit{Sequential Global}\text{''}, 0.9 \rangle \rangle \end{aligned} \tag{4}$$

is more Intuitive than Sensing (0.7 against 0.3), more Visual than Verbal (the opposite with respect to the previous one: 0.3 against 0.7), a bit more Reflective than Active and very much oriented to a Global processing of information rather than a Sequential one (0.9 against 0.1).

The first time a student approaches the Diogene's platform, the system sets the values of the Properties $P_1 - P_3$ of the Learning Preferences directly asking the necessary information to the student. Concerning the Learning Styles (property P_4), the platform offers to the learner the optional possibility to use the Soloman and Felder's psychological test [16] for on-line classification. The results of this test is used to instantiate the values of the Learning Styles. The use of the Soloman-Felder test is not mandatory for the learner in order to avoid a heavy impact with the system. In the case in which the student prefers not to use this preliminary psychological test, she can directly set the values of the category types choosing an estimated value for each category (using a slider-based interface). Finally, for those people who do not want or are not able to estimate their own learning styles, the system sets the initial values of all the category types to 0.5:

$$\begin{aligned} &\textit{Categories} = \langle \langle \text{``}\textit{Sensing-Intuitive}\text{''}, 0.5 \rangle, \\ &\langle \text{``}\textit{Visual-Verbal}\text{''}, 0.5 \rangle, \langle \text{``}\textit{Active-Relative}\text{''}, 0.5 \rangle, \\ &\langle \text{``}\textit{Sequential Global}\text{''}, 0.5 \rangle \rangle, \end{aligned} \tag{5}$$

which means that the student is (initially evaluated as) indifferent with respect to any learning style preference. The following subsection shows how the values of both the Cognitive State and the Learning Styles are automatically updated by the system taking into account the results of the tests included in each Presentation.

This section ends with a discussion of how the system performs the matching among the LOs and the student's Learning Preferences. In fact, each LO is off-line (manually) indexed by means of a Metadata using the IMS Metadata Standard [17] when it is initially stored in the content provider's repository (see Sect. 4). A Metadata is a set of attributes describing information concerning the LO. This information is organized in several categories and concerning both content-dependent and content-independent (e.g., copyright issues, title, version and so on) data. Diogene uses the "Classification" and the "Educational" (content-dependent) categories of the standard in order to represent and use information concerning the LO. For example, the attribute "TaxonPath" contained in the category "Classification" is used to link the LO with one or more DCs of the Ontology (thus implementing the inverse of relation *EB*). At run-time, once the Learning Path has been generated (Sect. 4.2), for each atomic DC *d* of the Learning Path the system selects a LO among all those LOs linked to *d* by means of the TaxonPath attribute. Since a given LO *l* is allowed to be (off-line) linked to more than one DC (e.g., $EB(d_1, l), ..., EB(d_n, l)$), then *l* is excluded from this selection if at least one of its parents $d_1, ..., d_n$ does not belong to the Learning Path. This is because of need to avoid providing the learner with material either not requested or concerning topics whose prerequisites have not been checked by the system against the learner's Student Model.

The choice of the best LO for *d* is performed by the system by matching the attributes of the "Educational" category of the LOs' Metadata with the student's Learning Preferences. The attributes of the "Educational" category of the IMS Metadata Standard that are used are: "Learning Resource Type", "Context", "Typical Age Range" and "Language". The last three attributes are matched with the corresponding properties of the student's Learning Preferences: the system will exclude those LOs whose "Context" attribute is different from the property "Educational Context" contained in the Learning Preferences of the student (Property P_1). In a similar manner, LOs whose "Typical Age Range" do not include the value of the "Age" property (Property P_2) of the student or LOs whose "Language" attribute is not included in the set of values of the "Languages" property (Property P_3) of the student are excluded from the set of possible choices for *d*.

Finally, if $A(d)$ is the set of the admissible LOs for *d*, i.e., those LOs which are linked to *d* by the TaxonPath attribute and which are consistent with the properties $P_1 - P_3$ of the student's Learning Preferences, the system selects among $A(d)$ a LO taking into account the value of the attribute "Learning Resource Type" of its Metadata. This is done as follows. The values of the "Learning Resource Type" attribute of the IMS Metadata Standard are strings ranging in the set: {*"Exercise"*, *"Simulation"*, *"Questionnaire"*, *"Diagram"*, *"Figure"*, *"Graph"*, *"Index"*, *"Slide"*, *"Table"*, *"Narrative Text"*, *"Exam"*, *"Experiment"*, *"ProblemStatement"*, *"SelfAssesment"*}. With the help of pedagogic experts, all these types of resources have been classified using the Felder and Silverman's teaching styles, by associating to each resource type (*R*) a quadruple (*Teaching Styles*(*R*)) of the same type of (3). For instance, *"Exercise"* and *"Simulation"* are respectively classified as:

$$\begin{aligned}&\mathit{Teaching\ Styles}(\mathit{Exercise}) = \langle\langle\text{``}\mathit{Sensing\text{-}Intuitive}\text{''}, 0\rangle,\\&\langle\text{``}\mathit{Visual\text{-}verbal}\text{''}, 0.5\rangle, \langle\text{``}\mathit{Active\text{-}Relative}\text{''}, 0\rangle,\\&\langle\text{``}\mathit{Sequential\text{-}Global}\text{''}, 0.5\rangle\rangle,\end{aligned} \tag{6}$$

$$\begin{aligned}&\mathit{Teaching\ Styles}(\mathit{Simulation}) = \langle\langle\text{``}\mathit{Sensing\text{-}Intuitive}\text{''}, 0.3\rangle,\\&\langle\text{``}\mathit{Visual\text{-}verbal}\text{''}, 0\rangle, \langle\text{``}\mathit{Active\text{-}Relative}\text{''}, 0\rangle,\\&\langle\text{``}\mathit{Sequential\text{-}Global}\text{''}, 0.5\rangle\rangle.\end{aligned} \tag{7}$$

At run-time, if *l* is a LO belonging to $A(d)$ (for a given DC *d*), and $R = \mathit{Resource}(l)$ is the resource type of *l* as described in the "Learning Resource Type" field of its Metadata, the system computes the matching score between *l* and the student's learning styles as follows. Suppose the student's learning styles are represented by:

$$\mathit{Categories} = \langle\langle T_1, e_1\rangle, \langle T_2, e_2\rangle, \langle T_3, e_3\rangle, \langle T_4, e_4\rangle\rangle \tag{8}$$

and the teaching styles associated with the resource type are represented by:

$$\mathit{Teaching\ Styles}(R) = \langle\langle T_1, f_1\rangle, \langle T_2, f_2\rangle, \langle T_3, f_3\rangle, \langle T_4, f_4\rangle\rangle, \tag{9}$$

then the distance between *l* and the student's profile is computed by means of the following:

$$\mathit{Dist}(R) = (e_1 - f_1)2 + (e_2 - f_2)2 + (e_3 - f_3)2 + (e_4 - f_4)2. \tag{10}$$

The Presentation is built by choosing a set of LOs which minimizes the sum of $\mathit{Dist}(R_j)$ for all the resource R_j of the set.

Finally, every time a LO *l* is selected to be included in the Presentation, all its parents (e.g., $EB(d_1, l), ..., EB(d_n, l)$), are excluded from the current Learning Path and the process is iterated until the Learning Path is empty.

5.1 Student model updating by means of the student's feedback

In each Presentation delivered to the student, the system includes some special LOs of type *test* which consist of interactive multiple-choice tests evaluated by Diogene in order to assess the knowledge degree reached by the learner concerning the DCs of the Learning Path. All the tests are attached at the end of the Presentation, which is then split in two consecutive sequences (delivered to the student respecting this order): a learning phase, composed of normal LOs, and a test phase, composed of tests. Suppose that:

$$Learning\ Path = \{d_1, \ldots, d_n\} \quad \text{and} \tag{11}$$

$$Presentation = \left\{l_1, \ldots, l_m, t_1, \ldots, t'_m\right\}, \tag{12}$$

where t_1, ..., $t_{m'}$ are the LOs of type test representing the test phase. Note that neither the number of learning LOs (m) nor the number of test LOs (m') is requested to be equal to the number of DCs (n). In fact, a given LO (learning or test) can be used to explain or evaluate more than one DC, depending on the TaxonPath relation with the elements of the Ontology. On the contrary, a given DC of the Learning Path is associated with one and only one LO of the Presentation using the relation *EB* (see Sect. 4.1). Finally, the course generation and personalization procedure guarantees that, altogether, l_1, ..., l_m cover the arguments d_1, ..., d_n and t_1, ..., $t_{m'}$ test the same topics d_1, ..., d_n. Since tests are a special kind of LOs, they are dealt with by the system as LOs. The Presentation construction phase is simply split in two phases, the first for learning LOs (leading to the production of the sequence l_1, ..., l_m) and the second for the test phase (whose output is the sequence t_1, ..., $t_{m'}$). Note that test LOs are associated with Metadata as well, and the corresponding "Learning Resource Type" attribute is used in combination with the function *Teaching Styles*() in order to categorize each test resource. It is then possible to select test types which are the most suitable for a given learner profile using the same methodology presented for "normal" LOs (see above).

Suppose now that the result of the whole test phase gives the following set of fuzzy values: $\{v_1, ..., v_n\}$, v_i $(1 \leq i \leq n)$ being the degree of knowledge shown by the student for the concept d_i in the just evaluated test phase. Hence, if $\langle d_i, e_i, N_i\rangle \in CS$ (where CS is the student's Cognitive State before all the tests' evaluation), then CS is updated by substituting $\langle d_i, e_i, N_i\rangle$ with $\langle d_i, e_i', N_i'\rangle$, where:

$$N_i' = N_{i+1}, \tag{13}$$

$$e_i' = (e_i N_i + v_i)/N_i'. \tag{14}$$

Equation 14 updates the system's evaluation about the student's knowledge on d_i by computing the average value of all the test phases' results (included the last v_i) performed so far.

Concerning the Learning Preferences of the Student Model, there is no modification of the initial values of the properties $P_1 - P_3$, while the student's Learning Styles (property P_4) are updated as follows. Each type of learning style T_k $(1 \leq k \leq 4)$ is associated with an histogram $H[T_k]$ which represents information related to the (discretized) statistic distribution of the preferences so far shown by the student concerning T_k. For example, H[*"Sensing-Intuitive"*] is represented by an histogram of 10 bins (a discretization of the range [0, 1] in 10 steps) which describes the statistic data collected by the system on the preferences shown by the student for Sensing and Intuitive resources. $H[T_k]$ is represented by the following function:

$$H[T_k] : \{1, 2, \ldots, 10\} \rightarrow \Re \tag{15}$$

The highest bin of H[*"Sensing-Intuitive"*] represents the preference shown by the student for the Sensing-Intuitive choice of the learning resources. For instance, in Fig. 6 the bin with the maximum value of H[*"Sensing-Intuitive"*] is the third, which corresponds to the pair ⟨"*Sensing-Intuitive*", 0.3⟩ in the *Categories* representation in the Student Model. We have that, for each $\langle T_k, g_k\rangle \in Categories$ $(1 \leq k \leq 4)$:

$$g_k = (\arg\max\ h \in \{1, 2, \ldots, 10\} H[T_k](h))/10. \tag{16}$$

The histograms is initialized by setting:

$$H[T_k](h') = M, \tag{17}$$

$$H[T_k](h') = 0, \quad for\ \text{all}\ h \in \{1, 2, \ldots, 10\}, h \neq h', \tag{18}$$

where M is a (small) pre-fixed value and h' is the initial preference of the student, either obtained by means of the Solomon-Felder test, directly set by the student herself, or a default value (in the last case h' = 5: see above).

The histograms are used to take trace of the student's behavior related to the different learning resources used in

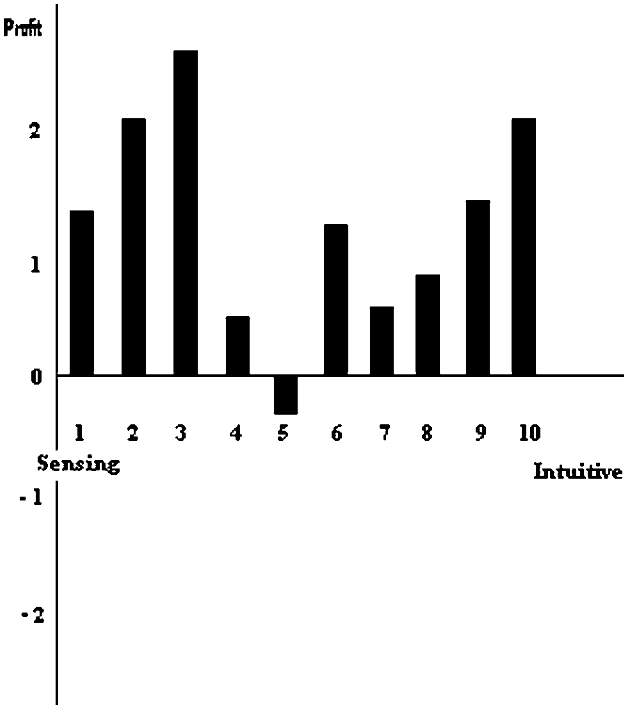


Fig. 6 An example of Sensing-Intuitive Histogram. The fifth bin has a negative value

the Presentation. For every LO l_j ($1 \le j \le m$) of the Presentation (12) and every learning category, the histogram bins corresponding to the resource type of l_j are updated, and then the (possible new) maximum bins of the new histograms are taken as the representation of the student's learning styles. More in detail, suppose that:

$$Categories = \langle\langle T_1, g_1\rangle, \langle T_2, g_2\rangle, \langle T_3, g_3\rangle, \langle T_4, g_4\rangle\rangle \tag{19}$$

is the classification of the student's learning styles before the last test phase, and, for each $d_i \in$ Learning Path, $w_i = e_i' - e_i$ is the difference between the evaluation of the DC d_i before and after the test phase as mentioned above. Moreover, for each LO l_j contained in the Presentation, suppose that l_j has been used to explain to the student the set $D_j = \{d_{j1}, \ldots, d_{jz}\}$ (this information is stored in the TaxonPath attribute of l_j). Hence, the overall benefit of l_j with respect to D_j can be computed as:

$$W_j = \sum_{s=1}^{z} w_{j_s}. \tag{20}$$

Finally, suppose that $R_j = Resource(l_j)$ and:

$$Teaching\ Styles(R_j) = \langle\langle T_1, f_1\rangle, \langle T_2, f_2\rangle, \langle T_3, f_3\rangle, \langle T_4, f_4\rangle\rangle. \tag{21}$$

For each l_j and for each category T_k ($1 \le k \le 4$) the hth bin of $H[T_k]$ is incremented, where $h = Int(f_k \times 10)$, being $Int(x)$ the function which returns the integer best approximating the real value x. The increment is obtained by substituting $H[T_k]$ with the new $H'[T_k]$ defined as:

$$H'[T_k](h) = H[T_k](h) + W_j. \tag{22}$$

$$H'[T_k](l) = H[T_k](l), \quad \text{for all } l \in \{1, 2, \ldots, 10\}, l \neq h. \tag{23}$$

In other words, for each T_k, f_k is used to select the bin to update and W_j as the updating weight. Note that W_j can possibly be a negative value (even if this is a seldom event). For this reason $H[T_k]$ is defined in $\Re$ (see (15) and the histogram example in Fig. 6) and not in $\Re^+$.

When the updating operation has been repeated for all the LOs of the Presentation, obtaining the final histograms H'', the system substitutes the old *Categories* with the new:

$$Categories' = \langle\langle T_1, g_1'\rangle, \langle T_2, g_2'\rangle, \langle T_3, g_3'\rangle, \langle T_4, g_4'\rangle\rangle, \quad \text{where} \tag{24}$$

$$g_k' = (\arg\max\ h \in \{1, 2, \ldots, 10\} H''[T_k](h))/10 (1 \le k \le 4). \tag{25}$$

6 A case study

This section shows some functionalities of the Diogene platform in a specific domain, namely Information and Communication Technology (ICT), chosen during the Diogene project to test the course generation and personalization potential of the system [12, 13].

Concerning the Ontology creation, choices similar to those suggested in [30] were adopted, selecting taxonomies approved by authoritative entities in the ICT area, with the double objective of covering the target domain (ICT topics) in an exhaustive way and of collecting a large consensus in the specific didactic community. For this reason, the Ontology has been built on the base of the ACM "Computing Classification System" (CCS) [1]. This classification is mainly used by ACM for classification purposes, publications, etc. ACM was chosen for different reasons: it has a widespread ICT taxonomy, it is easily expressed using the Diogene relations, and it is directly linked with the ACM & IEEE Computing curricula guidelines, describing how courses in the field of computing have to be arranged in suitable training programs at various levels. Starting from this taxonomy:

1. topics (DCs) were added in the lowest levels of the taxonomic hierarchy (which is represented by our *Has Part* relations) and
2. all the resulting DCs were linked with the necessary *Requires* and *Suggested Order* relations.

The current Ontology, used in the conducted tests, contains 1,652 concepts. This Ontology is linked with a total amount of 716 LOs, plus a few hundreds of interactive tests. Figure 7 shows a zoom of a particular piece of the Ontology regarding the argument "dynamic Web pages creation" which has entirely been added to the original ACM classification.

For Ontology editing, the Protégé tool [29], which is a domain independent ontology editor, was adopted and customized with the specific knowledge representation constraints of Diogene (see Sect. 4). Moreover, some plugins were added to Protégé, which is an open-source software, such as the building of the Ontology's graph, the Augmented Graph construction and the consistency check tool (see Sect. 4.2). Indeed, the consistency check facilities provided by Protégé itself are not suffient for the knowledge representation semantics of Diogene, because they are not able to take into account complex semantics loops such as those shown in Fig. 5. Figure 8 shows a screen snapshot of the Protégé template customized for the needs of Diogene.

Concerning LO Metadata editing, a suitable Metadata editor called *Knowledge Management System* (KMS) was built. Figures 9, 10 and 11 present some snapshots of the interface of the KMS module. Figure 9 shows the creation of a LO Metadata, while Figure 10 shows an example of link creation between some LO Metadata and the corresponding DCs (i.e., the *EB* relation).

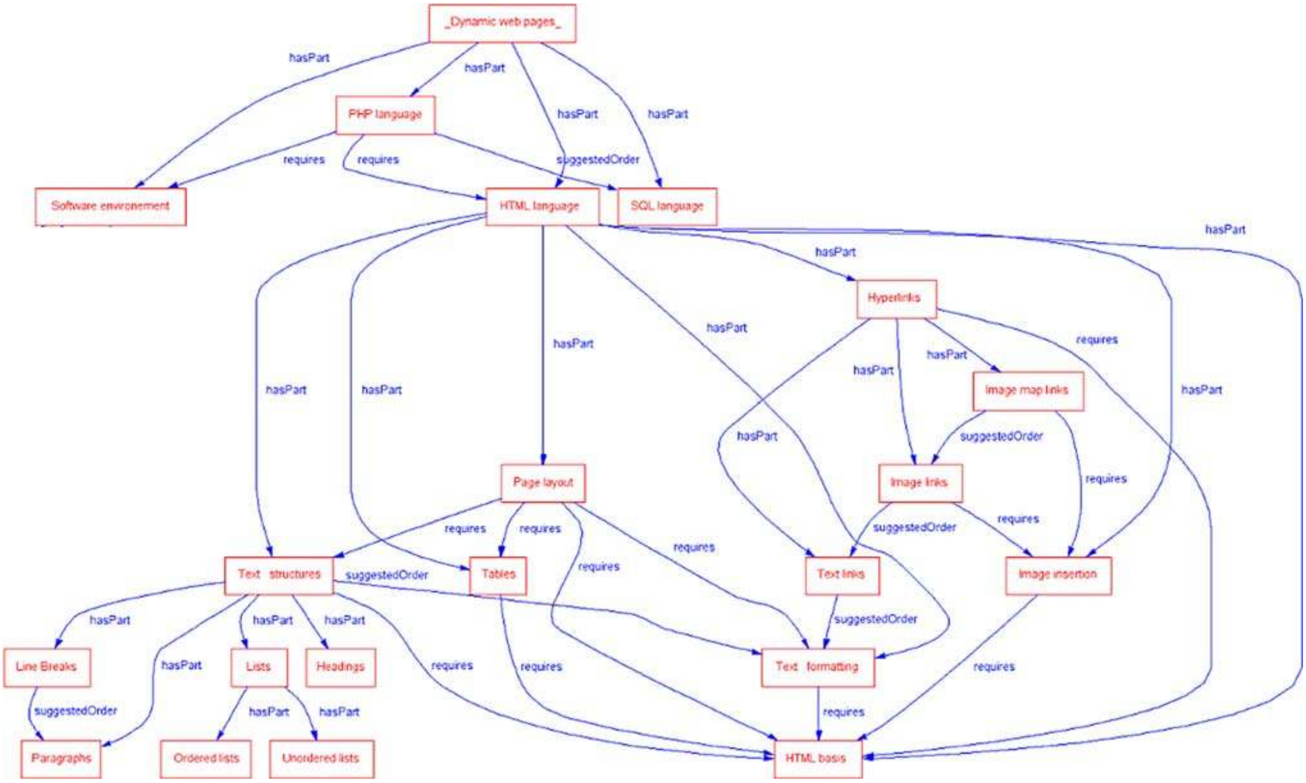
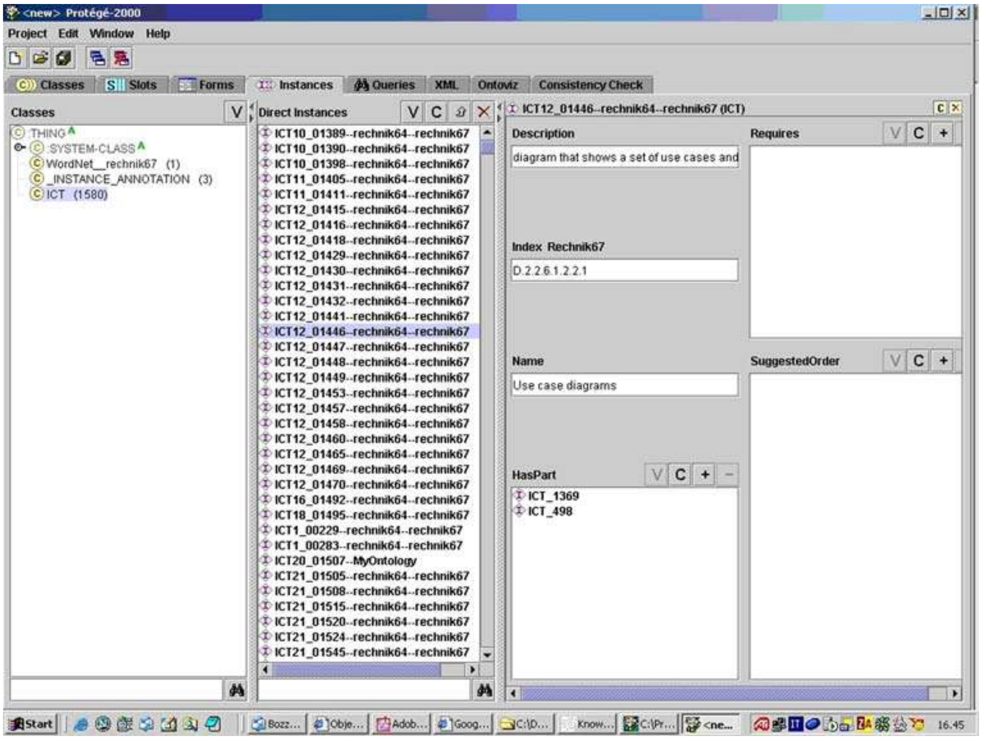


Fig. 7 A zoom of the ICT Ontology


<new> Protégé-2000
Project Edit Window Help
Classes Slots Forms Instances Queries XML Ontoviz Consistency Check
Start
16.45

Classes

- THING
  - :SYSTEM-CLASS
  - WordNet__rechnik67 (1)
  - _INSTANCE_ANNOTATION (3)
  - ICT (1580)

Direct Instances

ICT10_01389--rechnik64--rechnik67
ICT10_01390--rechnik64--rechnik67
ICT10_01398--rechnik64--rechnik67
ICT11_01405--rechnik64--rechnik67
ICT11_01411--rechnik64--rechnik67
ICT12_01415--rechnik64--rechnik67
ICT12_01416--rechnik64--rechnik67
ICT12_01418--rechnik64--rechnik67
ICT12_01429--rechnik64--rechnik67
ICT12_01430--rechnik64--rechnik67
ICT12_01431--rechnik64--rechnik67
ICT12_01432--rechnik64--rechnik67
ICT12_01441--rechnik64--rechnik67
ICT12_01446--rechnik64--rechnik67
ICT12_01447--rechnik64--rechnik67
ICT12_01448--rechnik64--rechnik67
ICT12_01449--rechnik64--rechnik67
ICT12_01453--rechnik64--rechnik67
ICT12_01457--rechnik64--rechnik67
ICT12_01458--rechnik64--rechnik67
ICT12_01460--rechnik64--rechnik67
ICT12_01465--rechnik64--rechnik67
ICT12_01469--rechnik64--rechnik67
ICT12_01470--rechnik64--rechnik67
ICT16_01492--rechnik64--rechnik67
ICT18_01495--rechnik64--rechnik67
ICT1_00229--rechnik64--rechnik67
ICT1_00283--rechnik64--rechnik67
ICT20_01507--MyOntology
ICT21_01505--rechnik64--rechnik67
ICT21_01508--rechnik64--rechnik67
ICT21_01515--rechnik64--rechnik67
ICT21_01520--rechnik64--rechnik67
ICT21_01524--rechnik64--rechnik67
ICT21_01545--rechnik64--rechnik67

ICT12_01446--rechnik64--rechnik67 (ICT)

Description: diagram that shows a set of use cases and

Index Rechnik67: D.2.2.6.1.2.2.1

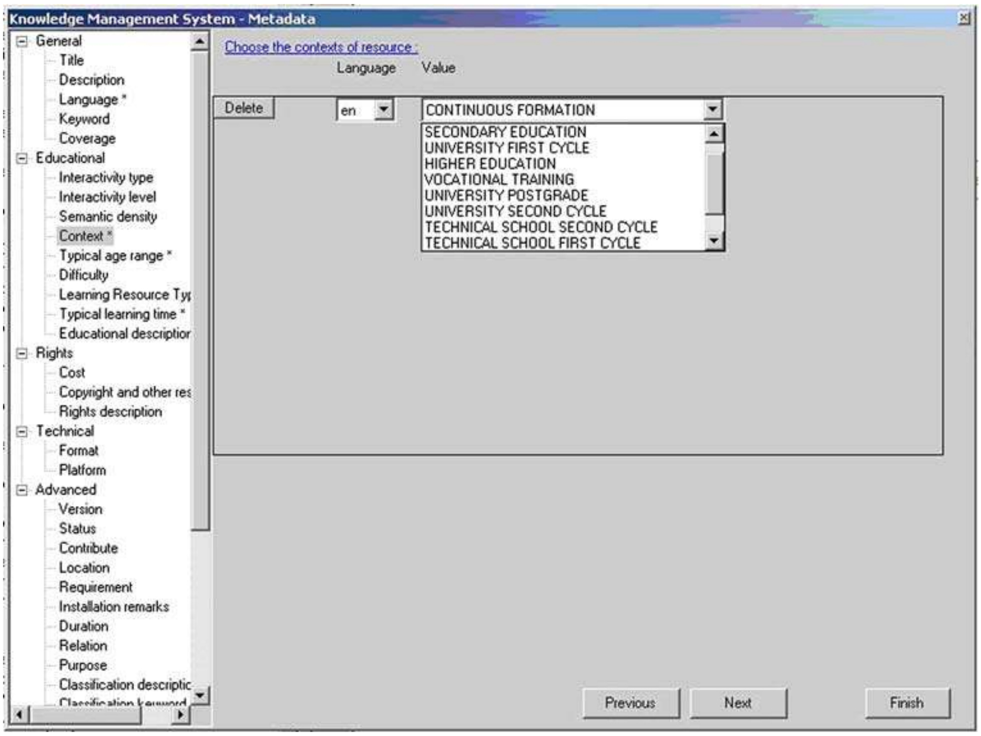
Name: Use case diagrams

HasPart: ICT_1369, ICT_498

Requires

SuggestedOrder

Fig. 8 The Protégé template customized for the Diogene's knowledge representation necessities



Language: en
Value: CONTINUOUS FORMATION

Fig. 9 The editing of a LO Metadata by means of the KMS

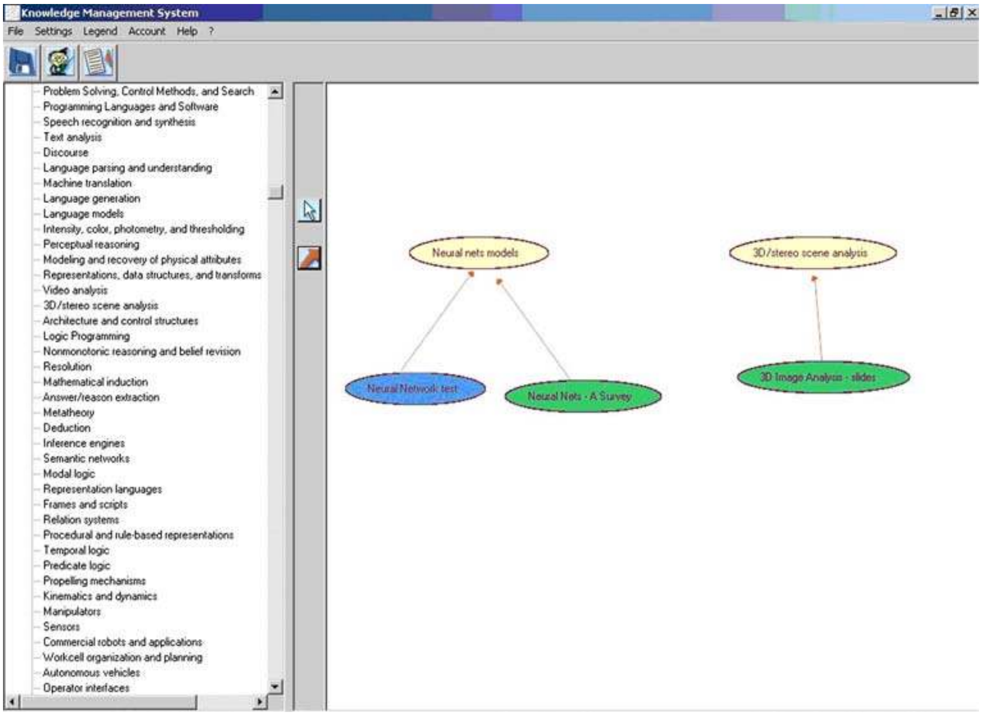
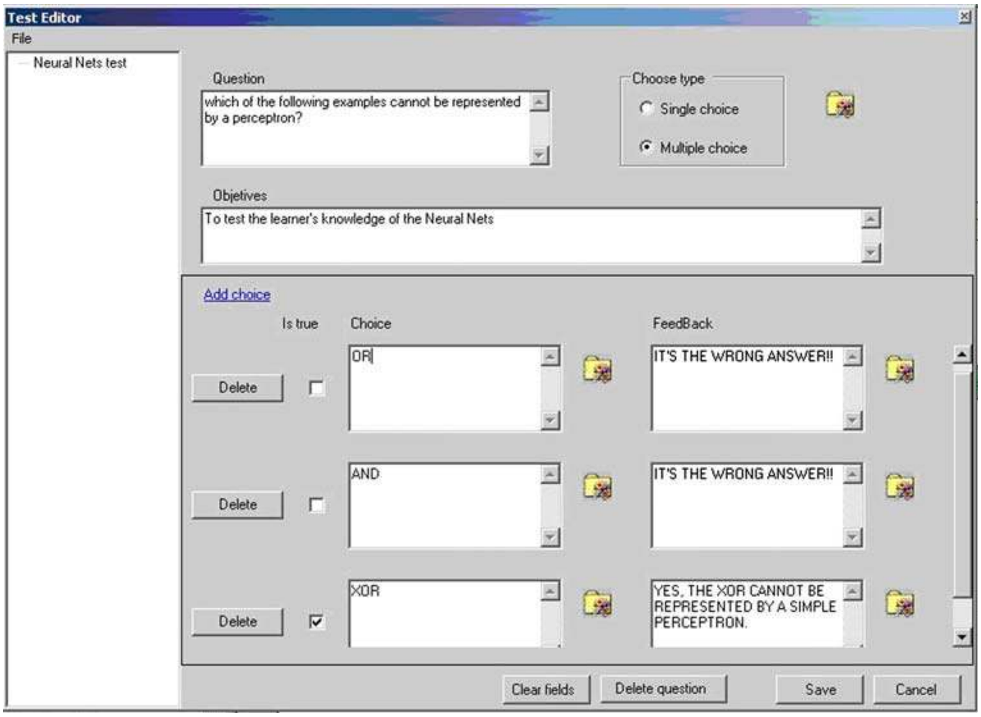


Fig. 10 The creation of links between the LO Metadata and the DCs by means of the KMS

The KMS is also used for the creation of the interactive tests used in the user assessment (see Sect. 5). Currently, the test types which can be built using KMS are single or multiple-choice tests composed of the following elements.

- A question (a text with possible associated images, Flash files and/or HTML documents).
- A set of possible answers which are shown on-line to the student together with the question.

Fig. 11 An example of test creation by means of the KMS



Neural Nets test

Question: which of the following examples cannot be represented by a perceptron?

Choose type: ○ Single choice ● Multiple choice

Objetives: To test the learner's knowledge of the Neural Nets

Is true	Choice	FeedBack
□	OR	IT'S THE WRONG ANSWER!!
□	AND	IT'S THE WRONG ANSWER!!
☑	XOR	YES, THE XOR CANNOT BE REPRESENTED BY A SIMPLE PERCEPTRON.

- One or more possible right answers (the teacher can choose among single or multiple choice tests).
- A possible feedback associated to each correct or wrong answer. The feedback is composed of a text document and its possible associated images.

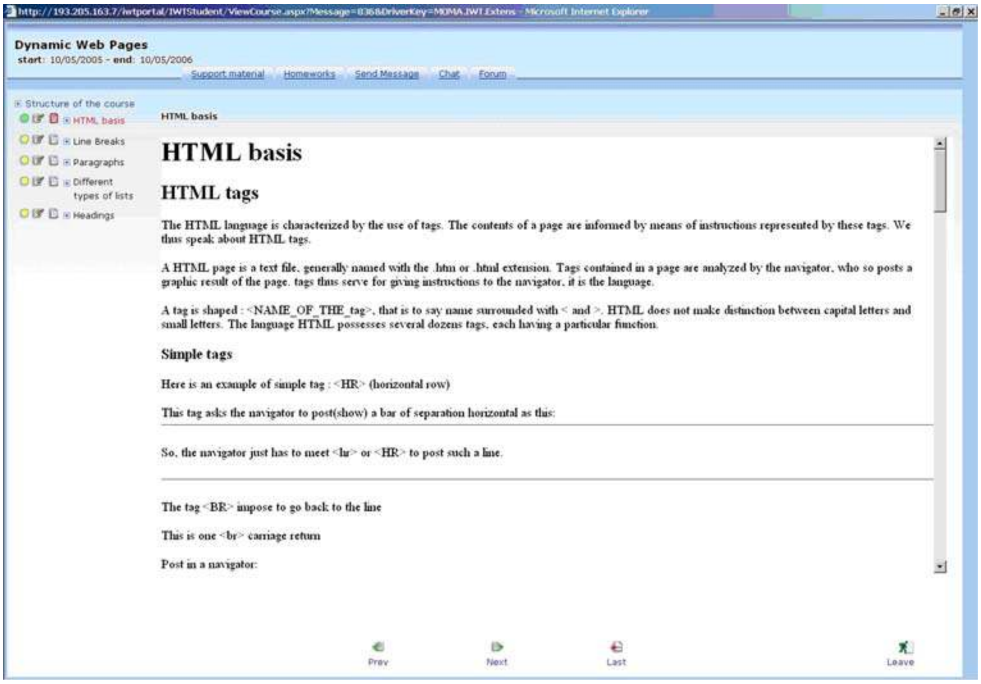
Figure 11 shows an example of test creation.

In conclusions, the system's knowledge off-line set-up operations are performed by, respectively, the Ontology Designer(s), using a modified version of Protégé, and by the content providers' teacher(s), who can use the KMS both as an authoring tool for test creation and as a Metadata editing tool for the LO indexing.

Concerning the on-line operations performed by the learner, the first time the student accesses to the system, she is invited by Diogene to initialize her Learning Preferences. As discussed in Sect. 5, this initialization is not mandatory and the system will assume default values for "lazy" learners. The Student Model is not a static representation of the student's knowledge and preferences but a dynamic one, and the initial values are continuously updated using the on-line tests' results.

Finally, when the Student Model has been initialised, the student can input the system with a query composed of a set of Target Concepts she is interested in. The Target Concepts are selected from the list of the system's topics (i.e., the names of all the available DCs) being the Ontology structure not shown to the learners. The system's output is computed building the Learning Path and the corresponding Presentation, both provided to the student. Figure 12 shows the Presentation corresponding to the query *Text Structures* using the Ontology of Fig. 7. The Learning Path produced by the system was: *HTML Basis, Line Breaks, Paragraphs, Ordered List, Unordered List, Headings*. In this execution example a Cognitive State was used with every concept associated with the value ⟨*0, 0*⟩ and a Learning Preferences module with default values for the preferred learning categories (5). Note that, despite the fact that *Text Formatting* is atomic (see Definition 2, Sect. 4.1), it is not included in the final Learning Path because the link with *Text Structure* is only a *SO* association. The entire Presentation structure is shown on the left hand side of Fig. 12: *HTML Basis, Line Breaks, Paragraphs* and *Headings* are the names of the LOs associated to the homonymous DCs, while *Different Types of Lists* is a LO chosen for both *Ordered List* and *Unordered List*.

The last three figures show some snapshots from Intelligent Web Teacher (IWT [20]), a commercial platform derived from Diogene and currently available at the url address http://elearning.diima.unisa.it/IWTPortal/. Figures 13 and 14 show a course for English beginners. Figure 13 shows the Presentation generated by the system for the learner "Mr. Smith", with a verbal learning style preference. As usual, the course structure is shown on the left hand side of the learner's window. In this case, the Presentation is composed of four LOs: "To Be Verb", "To Be Verb2" (a test), "Vowels" and "How to present myself". The currently shown LO is "Vowels" (in the centre of the window), composed of an (Italian) text explaining some basic differences and analogies between



Dynamic Web Pages
start: 10/05/2005 - end: 10/05/2006

HTML basis

HTML basis

HTML tags

The HTML language is characterized by the use of tags. The contents of a page are informed by means of instructions represented by these tags. We thus speak about HTML tags.

A HTML page is a text file, generally named with the .htm or .html extension. Tags contained in a page are analyzed by the navigator, who so posts a graphic result of the page. tags thus serve for giving instructions to the navigator, it is the language.

A tag is shaped : <NAME_OF_THE_tag>, that is to say name surrounded with < and >. HTML does not make distinction between capital letters and small letters. The language HTML possesses several dozens tags, each having a particular function.

Simple tags

Here is an example of simple tag : <HR> (horizontal row)

This tag asks the navigator to post(show) a bar of separation horizontal as this:

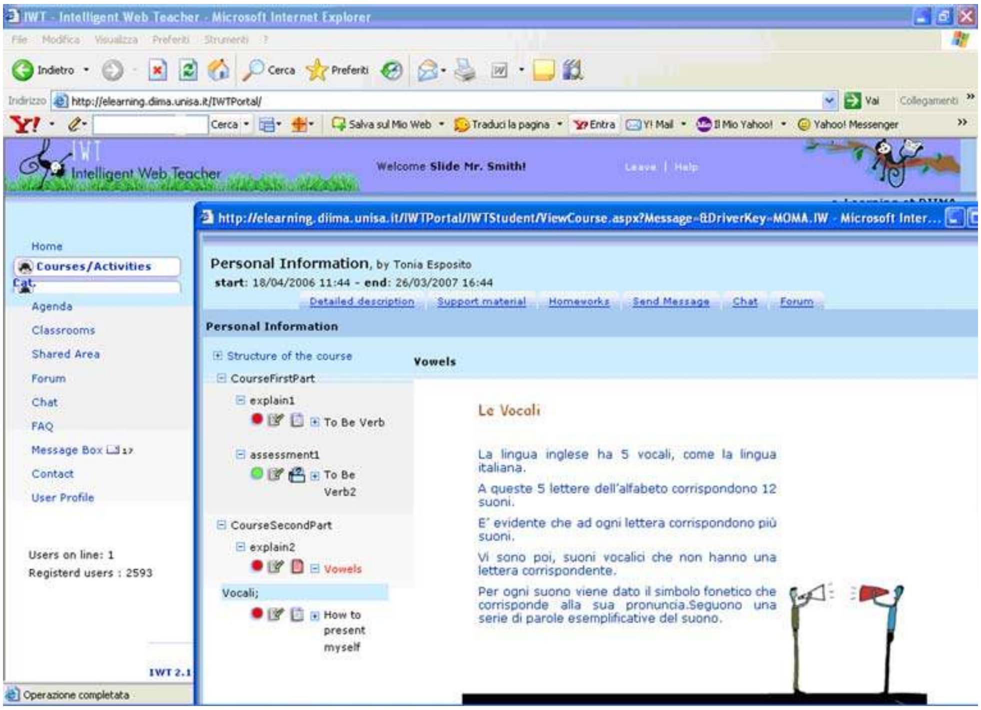
So, the navigator just has to meet <hr> or <HR> to post such a line.

The tag <BR> impose to go back to the line

This is one <br> carriage return

Post in a navigator:

Fig. 12 An example of Presentation built by the system. The full list of the names of the LOs is shown on the left hand side of the screen while the window on the centre is used to show each single LO once the student clicks on its name



Welcome Slide Mr. Smith!

Users on line: 1
Registerd users : 2593

Personal Information, by Tonia Esposito
start: 18/04/2006 11:44 - end: 26/03/2007 16:44

Personal Information

Vowels

Le Vocali

La lingua inglese ha 5 vocali, come la lingua italiana.

A queste 5 lettere dell'alfabeto corrispondono 12 suoni.

E' evidente che ad ogni lettera corrispondono più suoni.

Vi sono poi, suoni vocalici che non hanno una lettera corrispondente.

Per ogni suono viene dato il simbolo fonetico che corrisponde alla sua pronuncia.Seguono una serie di parole esemplificative del suono.

Fig. 13 A Presentation for a "verbal" learner

the Italian and the English vowels. Conversely, Fig. 14 shows the same course, obtained using the same Ontology and the same Target Concepts but with the Student Model of a different learner ("Mr. Brown", in this case), whit a visual learning style preference. In this case the Presentation is composed of the following LOs: "To Be Verb", "To Be Verb4" (a test), "Vowels-Sim" and "How to present myself". While the first and the last LOs are the same of the previous example, but the test LO and the LO dedicated to vowels differ. In particular, the latter is shown in the central user's window: it is an animation in which vowels move on the screen and a speaker explains some simple notions about English vowels. People with difficulties in reading long texts usually prefer animation-based LOs and

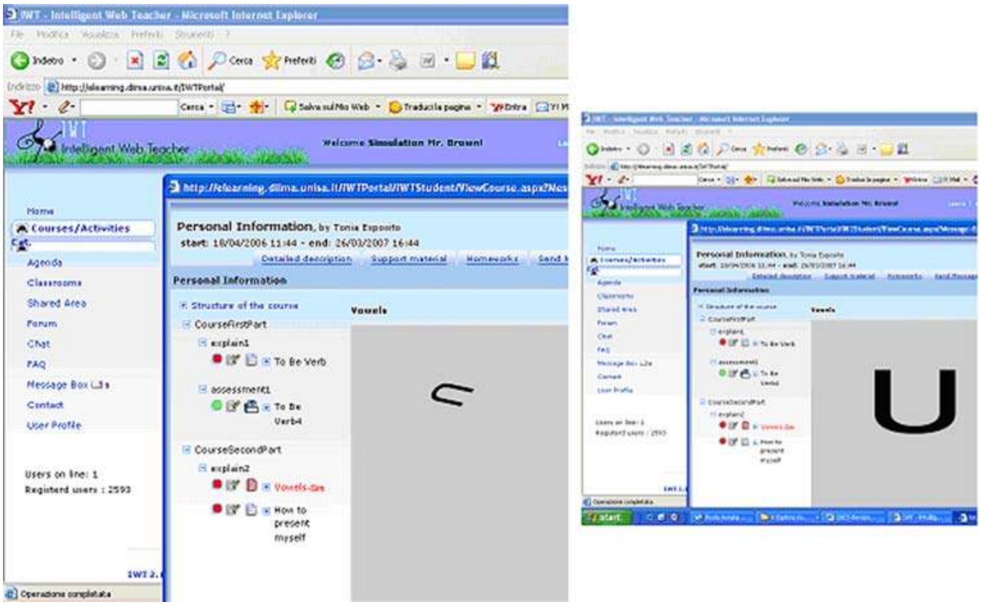


Fig. 14 Two different frames of an animation composing a Presentation for a “visual” learner

the system is able to provide them with suitable material by monitoring their feedback with respect to the past LO types.

Finally, Fig. 15 shows some additional facilities provided by the system, such as the annotation tool, which can be activated by clicking on the pencil-like icon on the left side of each LO (see Figs. 13, 14). When the user clicks on the pencil-like icon associated with a given LO, the system displays the small window appearing at the centre of the screens in the snapshots of Fig. 15, in which the learner can write some textual annotation. The learner can insert in the annotation also mathematical formulas (the “+Formula” Button, Fig. 15, left), phonetics symbols (the “+Fonetica” Button) or either logical or electronic circuits (the “+Circuits” Button, Fig. 15, right). Annotations are stored in the student’s profile (a more general structure containing the Student Model as described in this paper and other information specific to a given learner) and associated with the specific LO they refer to. Such annotations can be used by the learner (i.e., by the author of the annotations) in possible future accesses to the same LO, in the same fashion a manual annotation on a text book is used by a student in a possible future reading of the same arguments of the book.

7 Evaluation

A preliminary version of the system has been tested in the European funded project *Intraserv* (Fifth Framework Programme, Information Society Technologies, contact

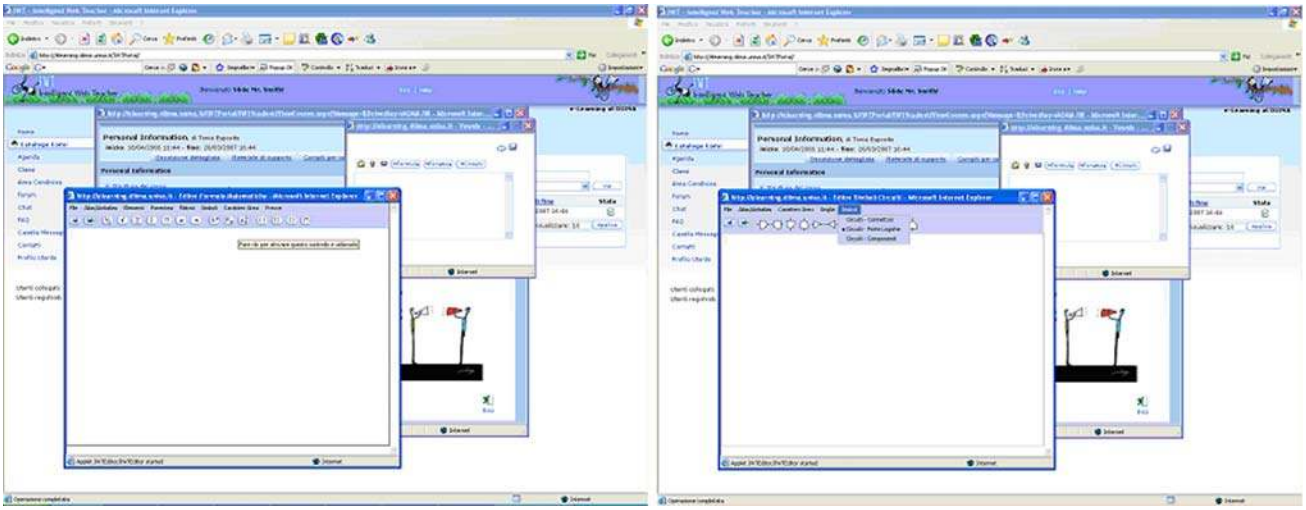


Fig. 15 The annotation facilities provided by the IWT system

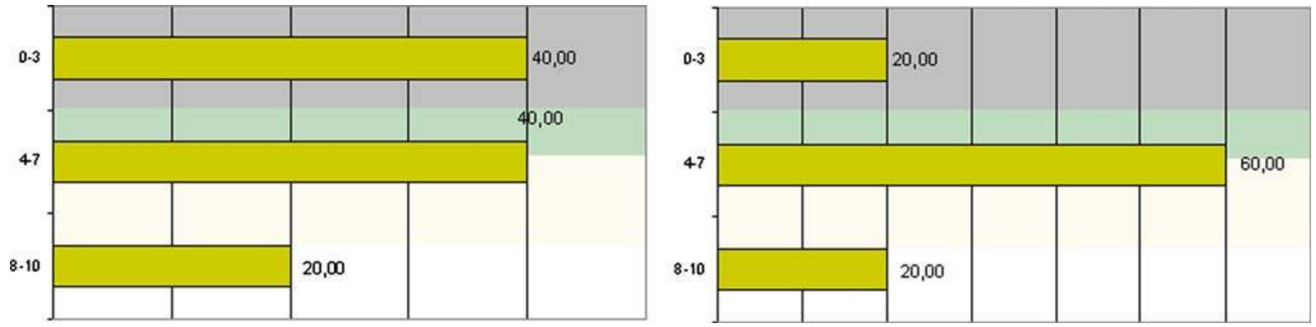


Fig. 16 “Simple” course. The table on the *left* shows the percentages of learners who have scored, respectively, low-level (0–3), medium-level (4–7) and high-level (8–10) competencies in the test performed before the training phase. The table on the *right* shows the competencies achieved by the same learners after an e-learning phase (*without automatic course customization*) with the proposed platform (Intraserv final experimentation [19])

number IST-2000-29377, see [18, 19] for more details), while the complete version has been tested in the *Diogene* project (Fifth Framework Programme, Information Society Technologies, contact number IST-2001-33358, see [12, 13] for more details). This section shows some experimental results from both the projects.

7.1 Intraserv evaluation

A first prototype version of the system has been tested in the Intraserv project. This version focused on the Learning Path and the Presentation generation (Sects. 4, 5) but differs from the final version because the Learning Preferences only include the “static” properties $P_1 - P_3$ (see Sect. 5), not taking into account the treatment of the learning/teaching styles. Nevertheless, course adaptation is still possible using the learner’s Cognitive State as well as the selected Target Concepts and the system’s planning mechanism.

For evaluation purposes, a group of voluntary platform users was set up composed of 28 learners, all workers belonging to Spanish and Italian Small and Medium Enterprises (SMEs). Figures 16, 17 and 18 illustrate the test results. All the voluntary learners were tested before and after a training phase realized by means of the developed platform on a “business decision” domain. In all the tests the students’ skills in the chosen domain were quantified using three ability ranges: low-level (0–3 scores), medium-level (4–7 scores) and high-level (8–10 scores). The group of learners was then split in two separate subgroups. Figure 16 shows the experiments performed on a first group composed of 20 students who have accessed to all the platform’s e-learning facilities (collaborative tools, LOs’ on-line delivering, etc.), except the automatic course generation and personalization described in this paper. Conversely, Fig. 17 shows the results of the second group composed of eight learners who have used the platform combined with its automatic course generation and personalization system. As can be seen, the progress made by the second group of students is much sharper with respect to the first group that have not had access to the learning personalization facilities. Finally, Fig. 18 shows the average test results on the entire group of 28 learners (with and without personalization).

Besides these tests, interviews to different platform’s users were also conducted in order to evaluate the overall system’s friendliness and the user satisfaction level.

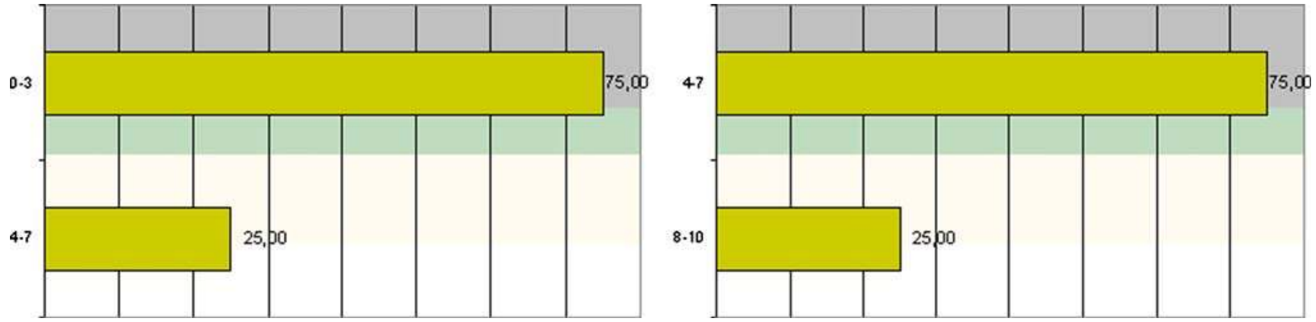


Fig. 17 “Intelligent” course. The table on the *left* shows the learners’ competencies in the test performed before the training phase. The table on the *right* shows the competencies achieved by the same learners after an e-learning phase performed *by means of the automatic course customization system* (Intraserv final experimentation [19])

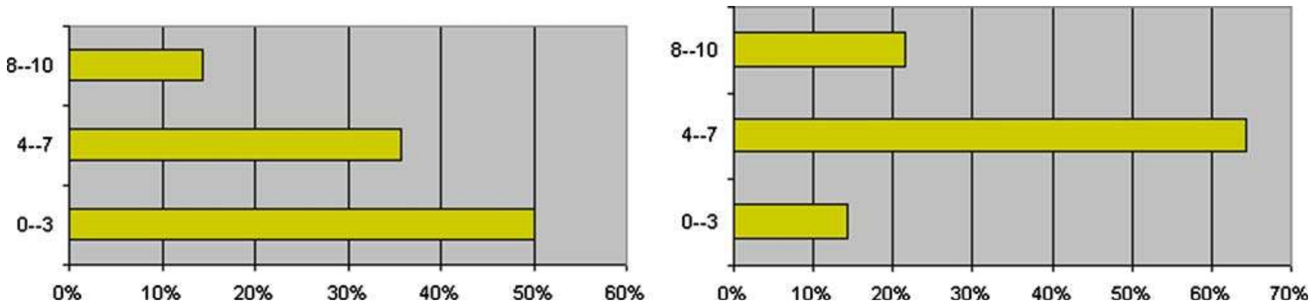


Fig. 18 The table on the left summarizes the test results on the whole group of 28 learners before the training phase. The table on the right shows the average competencies achieved by the learners after an e-learning phase with the Intraserv platform. In this case both the simple and the intelligent course results have been summed up (Intraserv final experimentation [19])

Beyond the 28 learners already mentioned, 7 worker-supervisors (one work-supervisor for each of the seven companies in question), a tutor (playing the role of teacher in the platform) and a training expert (responsible of the Ontology creation) were also interviewed. Table 1 summarizes some of the questions included in the used questionnaires with the corresponding results. From Table 1 it arises that most of the interviewed people (65.39%) felt satisfied with respect to the tool's navigation and interaction facilities, even if some of them (23.07%) had some technical problems in using the platform. Following this experiment, the interface utilities of the Intraserv prototype system have been improved during the Diogene project.

7.2 Diogene evaluation

The final version of the system, including course generation and personalization, as well as learning/teaching styles treatment, has been finally tested in the context of a second European project (Diogene). The Diogene's evaluation has been based on a set of 137 learners, belonging to 6 of the 8 different companies and University departments, partners of the project. The evaluation method was based on well-known evaluation instruments: performance recordings, on-line questionnaires, interviews of the involved learners and analysis of the system's reports and log information. The evaluation has been conducted between April and October 2004 and built up on several use scenarios that realised a real life learning environment.

Many evaluators pointed out that one of the major benefits provided by Diogene is the transparency of the learning process (e.g., see Point 4 of the IMS Global Learning Consortium principles mentioned in Sect. 1). In fact, the Learning Path inspection allows a student to understand what topics are related to her set of interests, and what are her learning necessities due to lack of knowledge. Moreover, learners stated in the interviews that their user profiles provided good support in selecting LOs close to their learning preferences and that this fact helped the learning process providing a coherent strategy in the learning material choice. More in details, about 70% of the learners felt that their user profile provided good support to define their learning behaviours (Fig. 19). They showed highly satisfied with the type of LOs suggested by the system with respect to their learning preferences (72%) and the adequacy of the courses that were suggested by the system to their lack of knowledge (78%). More details are available in the project's Final Report (see [13]).

Finally, Fig. 20 illustrates the overall learning progresses obtained by the learners (the question "work

Table 1 The questionnaires' results. The values in the Columns 2–6 show the percentage of users' satisfaction regarding each question in Column 1 (Intraserv final experimentation [19])

	Yes, very much	Yes, sufficiently	Not enough	Not at all	No answer
Is the tool versatile and suitable for your needs?	15.38	42.31	30.77	11.54	0
Was the audiovisual environment clear?	42.31	30.77	3.85	11.54	11.54
Did you have any problem in the installation phase?	7.69	11.54	7.69	73.08	0
Did you feel satisfied with the tool's navigation and interaction facilities?	42.31	23.08	23.08	11.54	0
Have you had any technical problem which made the learning difficult?	15.38	7.69	15.38	61.54	0

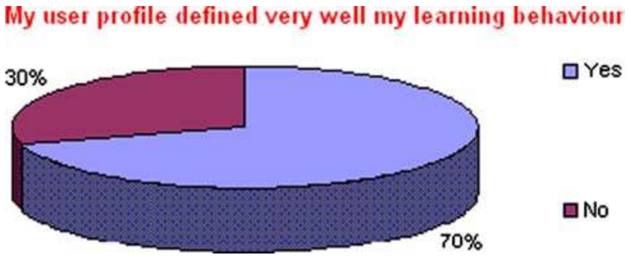


Fig. 19 Summative evaluation of the Student Model (Cognitive State and Learning Preferences) [13]

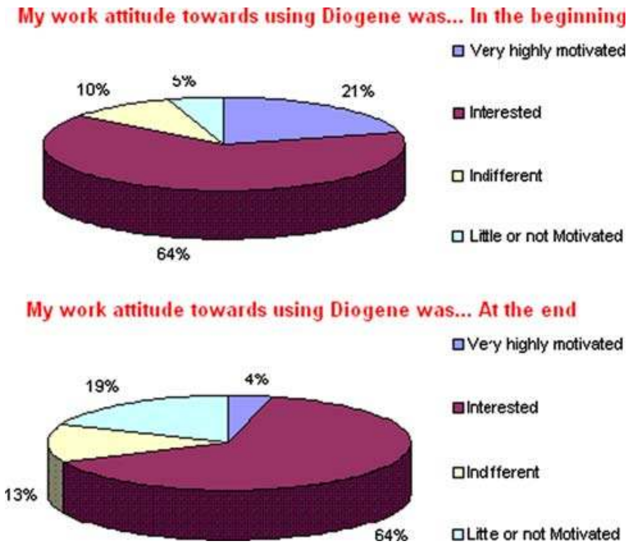


Fig. 20 Learning progresses achieved by the learners [13]

attitude” refers to the fact that all the courses used for the evaluation concern topics of interest of the learners’ real life work).

8 Conclusions

This paper has presented the Diogene platform, which performs the intelligent functionalities of an automatic Course Management System. These functionalities include the automatic generation and personalization of (complex) courses starting from the system’s knowledge on (1) the student’s current skills and preferences and (2) the didactic domain. This is possible because the system contains a representation of the student’s knowledge and individual learning preferences. Moreover, the platform is provided with both an abstract representation of the didactic domain (by means of an Ontology) and the description of each single Learning Object (by means of an associated Metadata).

The Ontology is represented using XML files, and the Learning Object Metadata are described by means of the IMS Metadata Standard, which enables knowledge sharing among different platforms. Moreover, the system’s knowledge representation satisfies a new methodology for didactic domains described in this paper. This allows the system to efficiently perform inferences on the student query and to compute the best presentation of the didactic concepts she is interested in.

Particular attention has been paid to the personalized selection of the LO resource types provided to the user in accordance with individual learning styles and following the Felder’s and Silverman’s pedagogical approach. This is possible by means of an explicit representation of the student learning styles, which is continuously updated by the system exploiting the feedback of the interactive tests.

It is believed that LO personalization can facilitate the student’s learning experience through the automatic adaptation of the system resources to the student’s preferences and necessities. Current research is focused on studying semi-automatic systems for Learning Object Metadata creation, in order to simplify the off-line teachers’ work in filling-in the fields of the IMS Metadata for every LO of the system.

Acknowledgments This research was supported by the Information Society Technologies projects Diogene (IST-2001-33358) and Intraserv (IST-2000-29377). Further details can be found, respectively, on the project Web sites: http://www.crmpa.it/diogene [12] and http://www.crmpa.it/intraserv/ [18].

References

1. ACM Web Site. http://www.acm.org/class/
2. Adorni, G., Bianchi, D., Calabrese, E.: A versatile system for on-line testing, Proceedings of the Workshop on Artificial Intelligence and e-learning, Eighth National Congress of Italian Association for Artificial Intelligence, Pisa, Italy 23–26 (2003)
3. Adorni, G., Sugliano, A.M., Vercelli, G.: AI embedded into LMS agents: the UniGe E_Learning portal experience and future scenario. In: Proceedings of the Workshop on Artificial Intelligence and e-learning, Eighth National Congress of Italian Association for Artificial Intelligence, 23–26, Pisa, Italy (2003)
4. Benayache, A., Abel, M.-H.: Using knowledge management method for e-learning. In: Chiazzese, G., Allegra, M., Chifari A., Ottaviano S. (eds.) Methods and Technologies for Learning. WIT Press, ISBN 1-84564-155-8 (2005)
5. Bert van den Berg, René van Es, Tattersall, C., Janssen, J., Manderveld, J., Brouns, F., Kurvers, H., Koper, R.: Swarm-based sequencing recommendations in e-learning. In: Proceedings of the fifth International Conference on Intelligent Systems Design and Applications, 2005. ISDA ’05, pp. 488–493 (2005)
6. Blackboard Web Site (1997): http://www.Blackboard.com/
7. Capuano, N., Gaeta, M., Micarelli, A., Sangineto, E.: An integrated architecture for automatic course generation. In: Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 02), pp. 322–326. Kazan, Russia (2002)
8. Capuano N., Gaeta, M., Micarelli, A., Sangineto, E.: An intelligent Web teacher system for learning personalization and

semantic web compatibility. In: Proceedings of the Eleventh International PEG Conference, St. Petersburg, Russia, 28 June–1 July (2003)

9. Cormen, T.H., Stein, C., Rivest, R.L., Leiserson, C.E.: Introduction to Algorithms, 2nd edn. McGraw, New York (2001)
10. De Marsico, M., Kimani, S., Mirabella, V., Norman, K.L., Catarci, T.: A proposal toward the development of accessible e-learning content by human involvement. Univers. Access Inform. Soc. **5**, 150–169 (2006)
11. Dicheva, D., Dichev, C.: Confronting some ontology-building problems in educational topic map authoring. In: Workshop on Applications of Semantic Web Technologies for e-Learning (SW-EL@AH'06), Ireland (2006)
12. Diogene, A.: Training Web broker for ICT professionals (IST-2001-33358), official Web site: http://www.crmpa.it/diogene (2001)
13. Diogene, the project's Final Report, available at: http://www.crmpa.it/diogene/archive.html (2004)
14. Duval, E., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., Hendrikx, K., Wentland Forte, M., Ebel, N., Macowicz, M., Warkentyne, K., Haenni, F.: The Ariadne knowledge pool system. Commun. ACM **44**(5), 73–78 (2001)
15. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. Engr. Educ. **78**(7), 674–681 (1988)
16. Felder, R.M., Soloman, B.A.: The Learning Styles Questionnaire. Available at: http://www.engr.ncsu.edu/learningstyles/ilsweb.html (1993)
17. Learning Resource Meta-Data Best Practice and Implementation Guide, version 1.2.1. Final Specification http://www.imsproject.org/specifications.html (2001)
18. IntraServ Web Site. InTraServ: Intelligent Training Service for Management Training in SMEs, FP5 IST Project (IST-2000-29377). http://www.crmpa.it/intraserv/
19. Intraserv Impact Evaluation Results and Analysis. Available on: http://www.crmpa.it/intraserv/archive.htm
20. IWT: Intelligent Web Teacher. White Paper. CRMPA (2002)
21. IMS Global Learning Consortium Inc.: IMS guidelines for developing accessible learning applications, version 1.0, white paper, retrieved December 1, 2004 from: http://www.imsproject.org/accessibility/ (2002)
22. Kasai, T., Yamaguchi, H., Kazuo Nagano, Riichiro Mizoguchi.: Building an ontology of IT education goals. Int. J. Cont. Eng. Educ. Lifelong Learn. **16**(1/2), 1–17 (2006)
23. Kitamura, Y., Mizoguchi, R.: An ontological schema for sharing conceptual engineering knowledge. In: Proceedings of the International Workshop on Semantic Web Foundations and Application Technologies, pp. 25–28, Nara, Japan (2003)
24. Koper, R.: Modeling units of study from a pedagogical perspective. The pedagogic meta-model behind Educational Modelling Language (E.M.L.), Open University of the Netherlands, Educational Technology Expertise Centre, Valkenburgerweg (2001)
25. Koper, E.J.R.: Increasing learner retention in a simulated learning network using indirect social interaction. J. Artif. Soc. Soc. Simulat. (2005)
26. Lotus LearningSpace Web Site: http://www.pugh.co.uk/Products/lotus/learningspace.htm (2004)
27. Gilbert Paquette.: Educational modeling languages, from an instructional engineering perspective. In: McGreal, R. (ed) Online education using learning objects, pp. 331–346. Routledge/Falmer, London (2004)
28. Paquette, G., De la Teja, I., Léonard, M., Lundgren-Cayrol, L., Marino, O.: Using an instructional engineering method and a modeling tool to design IMS-LD units of learning. In: Koper, R., Tattersall, C. (eds.) Learning Design: A Handbook on Modelling and Delivering Networked Education and Training, pp. 161–183. Springer, Heidelberg (2005)
29. Protégé Web Site. http://protege.stanford.edu
30. Ronchetti, M., Saini, P.: The knowledge management problem in an e-learning system: a possible solution, Proceedings of the Workshop on Artificial Intelligence and e-learning, Eighth National Congress of Italian Association for Artificial Intelligence, 23–26, Pisa, Italy (2003)
31. Sanrach, C., Grandbastien, M.: ECSAIWeb: A web-based authoring system to create adaptive learning systems. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.) Adaptive Hypermidia 2000, pp. 214–226. LNCS 1892 (2000)
32. Seal, K.C., Przasnyski, Z.H.: Using the World Wide Web for teaching improvement. Comput. Educ. **36**, 33–44 (2001)
33. Tattersall, C., Vogten, H., Brouns, F., Koper, R., van Rosmalen, P., Sloep, P., van Bruggen, J.: How to create flexible runtime delivery of distance learning courses. Educ. Technol. Soc. **8**(3), 226–236 (2005)
34. TopClass Web Site. http://www.wbtsystems.com
35. Trigano, Ph., Giacomini, E.: Toward a Web based environment for Evaluation and Design of Pedagogical Hypermedia. J. Educ. Technol. Soc. IEEE Learn. Technol. Task Force **7**(3), (2004)
36. WebCT Web Site. http://www.webct.com
37. Weber, G., Kuhl, H.C, Weibelzahl, S.: Developing adaptive Internet based courses with the authoring system NetCoach. In: Proceedings of the Third Workshop on Adaptive Hypermedia (UM2001) (2001)