An Ontology-based Planning System for e-Course Generation

E. Kontopoulos, D. Vrakas, F. Kokkoras, N. Bassiliades, I. Vlahavas

Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, 54124, Greece
{skontopo, dvrakas, kokkoras, nbassili, vlahavas}@csd.auth.gr

1. Abstract

Researchers in the area of educational software have always shown a great deal of interest for the automatic synthesis of learning curricula. During the recent years, with the extensive use of metadata and the emergence of the Semantic Web, this vision is gradually turning into a reality. A number of systems for curricula synthesis have been proposed. These systems are based on strong relations defined in the metadata of learning objects, which allow them to be combined with other learning objects, in order to form a complete educational program. This article presents PASER, a system for automatically synthesizing curricula using AI Planning and Semantic Web technologies. The use of classical planning techniques allows the system to dynamically construct learning paths even from disjoint learning objects, meeting the learner's profile, preferences, needs and abilities.

2. Introduction

Search in the World Wide Web (WWW) is currently based on search engines that simply feature keyword-based search functionality. Thus, there exist no widely adopted methods for searching *by content* in the WWW. This turns the pursuit for educational material into a formidable problem for the instructors and learners, especially when the material addresses particular learning and pedagogical goals.

Several education-related standards have been proposed to address this problem, providing automation and personalization in searching and accessing educational resources, as well as interoperability among them. These standards concern recommended practices and guidelines for software components, tools, technologies and design methodologies that facilitate the development, deployment, maintenance and interoperation of computer implementations of educational components and systems.

As more educational e-content is becoming available on-line, the need for systems capable of automatically constructing personalized curricula by combining appropriate autonomous educational units (or *learning objects*, as they are called) is becoming more intense. This article reports on such a system, called *PASER* (*P*lanner for the *A*utomatic *S*ynthesis of *E*ducational *R*esources), which extends the authors' previous work on a methodology presented in [1]. The system consists of (a) a metadata repository, storing learning object descriptions, learner profiles and ontological knowledge for the educational domain under consideration, (b) a deductive object-oriented knowledge base system for querying and reasoning about RDF/XML metadata, called R-DEVICE and (c) a planning system called HAP_{EDU} that automatically constructs course plans. Emphasis is given on the competencies ontology, which is based on a widely accepted vocabulary (see section 4) for expressing the structure and content of a variety of concept scheme paradigms. Details regarding the functionality of the overall application and implementation issues are further discussed. PASER follows all the evolving educational metadata standards that describe learning resources (LOM), content packaging (CP), educational objectives (RDCEO) and learner related information (LIP).

The rest of the article is organized as follows: the following section describes the overall system architecture, section 4 presents the competencies ontology utilized by the system, while the next section is concerned with the knowledge representation and the reasoning process. Section 6 presents the planning system that comprises the backbone of PASER, followed by Section 7 that features a case study scenario. Finally, Section 8 thoroughly reports on similar systems and the article ends with the section that includes the conclusions and poses directions for future work.

3. System Architecture

PASER is a synergy of five processing modules (Fig. 1), namely a planner, an Ontology & Metadata Server, the R-DEVICE module and two data converters. The system assumes the availability of three more metadata repositories that feed its modules with certain educational metadata. More specifically, there exists a LOM repository that stores metadata about the available learning objects, a repository of LIP compliant metadata describing the learners that have access to the system and an RDCEO metadata repository. The latter provides competency definitions that are referenced by the other two. In addition, it is used by the Ontology & Metadata Server providing this way a system-wide consistent competency vocabulary. We also assume that all metadata are checked by an expert user before they are entered into the system. This may introduce additional workload, but ensures that a common terminology and semantics are used in the enterprise or organization, in which the system is installed.

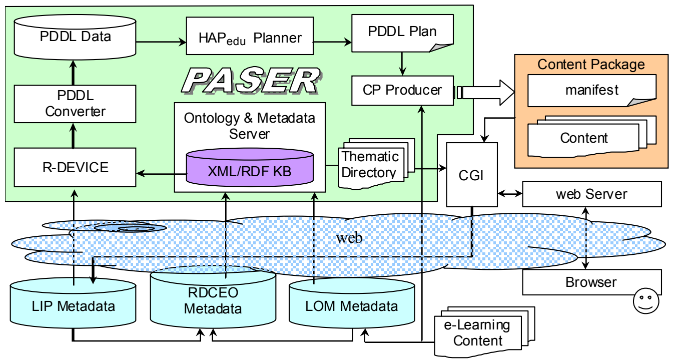


Fig. 1. PASER – System Architecture

The system supports two main types of users, namely content providers and learners. The content providers interact with the system, in order to add new educational material into the system and provide the appropriate metadata (LOM records). In order for PASER to automatically synthesize curricula, the LOM records must contain information about the prerequisites and the objectives of the resources expressed in terms registered within the PASER ontology. This is achieved manually with the content provider being responsible for selecting the appropriate terms from the ontology. However, we are currently working on a text mining module for proposing the terms that best match textual descriptions of both the resource's prerequisites and objectives.

On the other hand, the learner interacts with the PASER system in order to specify his/her educational objectives. The learner is presented (by means of a web page) with a dictionary of themes for which the system may be able to provide educational material.

As soon as the user selects a subject, the R-DEVICE module of PASER filters out the available learning objects based on (a) the user's preferences and knowledge status, as they described in his LIP record and (b) the PASER's understanding of the theme, as it is described in the Ontology & Metadata Server module. R-DEVICE [2] is a deductive object-oriented knowledge base system for querying and reasoning about RDF/XML metadata. It transforms RDF and/or XML documents into objects and uses a deductive rule language for querying and reasoning about them. The properties of RDF resources are treated both as first-class objects and as attributes of resource objects. This way, resource properties are gathered together in one object, resulting in superior query performance than the performance of a triple-based query model.

The output of R-DEVICE is a set of LOM objects that are directly or indirectly related with the theme selected by the user. Based on these records and keeping only a limited subset of the LOM record elements, the PDDL converter module produces a description of the user's request as a planning problem, encoded in the PDDL language.

HAP_{EDU} is a state – space planning system, based on the HAP planner [3], which is modified in order to implicitly support abstraction hierarchies that are needed in course planning problems.

The PDDL expressed plan produced by the HAP_{EDU} planner is forwarded to the CP producer module, which, in turn, creates a content packaging description (compliant to the CP metadata specification) of the learning objects involved in the plan. The produced CP record is finally forwarded to the user. Notice that at the current stage we do not take into account the performance of the user regarding the supplied educational material. In the future, assessment results should be taken into account, in order to determine the learner's performance and update his LIP record accordingly. At the moment, we provide the user with a simple verification form, related to the material provided, in which he simply verifies that he studied (and learned) the material. This verification properly updates his LIP record.

The series of the aforementioned steps is also described in a following section (section 7), where a scenario (use case) is featured, regarding a learner wishing to reach certain education goals using PASER.

4. The Ontology

The PASER competencies ontology deployed for the PASER system was developed using RDF Schema[1], a W3C Recommendation framework that provides mechanisms for describing application-specific classes and properties.

The ontology consists of a number of AI-related competencies (310 in total) that formulate an *isPartOf* hierarchy. More specifically, the root element of the ontology is the *AI* node that consists of a number of sub-competencies (e.g. Machine Learning,

[1] http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

Planning, Knowledge Representation etc.), which, in turn, consist of a further layer of sub-competencies and so on. In the current version of the ontology, the depth of the competencies hierarchy tree is 5.

The competencies ontology is based on the *SKOS Core* model[2], developed by the *Semantic Web Best Practices and Deployment Working Group*[3]. It defines an RDF vocabulary for describing the structure and content of a variety of concept scheme paradigms as well as other types of controlled vocabulary.

Each competency in the PASER ontology is represented as an instance of the **`paser:Competency`** class, which is a subclass of the **`skos:Collection`** class, while each **`isPartOf`** property is a sub-property of the **`skos:member`** property. In other words, competencies are considered unsorted collections of sub-competencies, with each sub-competency being a member of the "parent" competency. This does not suggest, however, that the total of all members of a competency completely defines the specific competency, since new members can be added at a later time, due to initial omissions or further additions. Finally, every competency is assigned a primary (*preferred*) and, optionally, a secondary (*alternative*) label, using the SKOS properties **`skos:prefLabel`** and **`skos:altLabel`**. Fig. 2 shows a code fragment of the ontology that illustrates the definition of *Hill-Climbing Search*, as a sub-competency of *Heuristic Search*.

```
<paser:Competency rdf:about="http://www.paser.org/rdfs#ai_a_04_02">
     <skos:prefLabel>Hill-Climbing Search</skos:prefLabel>
     <skos:altLabel>HC</skos:altLabel>
     <paser:isPartOf>
          <paser:Competency rdf:about="http://www.paser.org/rdfs#ai_a_04">
               <skos:prefLabel>Heuristic Search</skos:prefLabel>
          </paser:Competency>
     </paser:isPartOf>
</paser:Competency>
```

Fig. 2. PASER Ontology fragment, illustrating the definition of a sub-competency

5. Knowledge Representation and Reasoning

The PASER system makes extensible use of the various educational metadata specifications developed in the recent years or being under development at the present time. Specifically, learning objects are described based on the IEEE LOM specification, as it is defined in the IMS Learning Resource Meta-Data specification. The characteristics of a learner that are needed for recording and managing learning-related goals, accomplishments, etc. are described based on the IMS Learner Information Package. The XML binding of both specifications is used.

During the object filtering phase performed by the R-DEVICE module of PASER, a phase that will feed the planner with the appropriate objects, extensible usage of the *classification* elements of LOM records is performed. These elements allow the classification of the host LOM record based on competencies such as educational objectives and prerequisites. This can be formally established using the RDCEO specification. The latter is an emerging specification of an information model for describing, referencing and exchanging definitions of competencies, in the context of e-Learning. The same competency definitions are also used to describe the goals and accomplishments of the learner, in a controlled way. As a result, it is possible to establish links among learning objects and between learning objects and

[2] http://www.w3.org/2004/02/skos/core/
[3] http://www.w3.org/2001/sw/BestPractices/

characteristics of the learner. This information together with other constraints imposed over the learning objects due to the learner's preferences, are exploited by R-DEVICE, in order for the learning object repository to be filtered out and keep only the "promising" objects. Informally encoded examples of the competency related information located in LOM and LIP metadata, are presented in Fig. 3 (a) and (b), respectively.

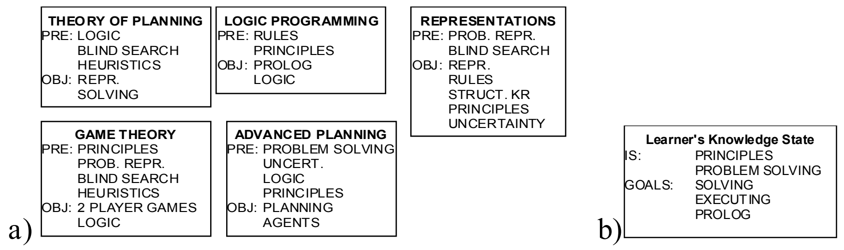


Fig. 3. Prerequisites and educational objectives of some, informally presented, learning objects (left) and initial knowledge state (IS) and learning objectives (GOALS) for a learner (right)

Finally, the same terms defined in the RDCEO metadata, are also organized as depicted in Fig. 4. This organization allows the decomposition of learning objectives into sub-objectives. As a result, the system will be able to relate learning objects with learner objectives in various levels of granularity. Notice that the hierarchy of Fig. 4 is a *part-of* hierarchy that is represented in a proprietary ontology of PASER, i.e. it is not represented directly in RDCEO because the latter does not allow the representation of hierarchical relationships.

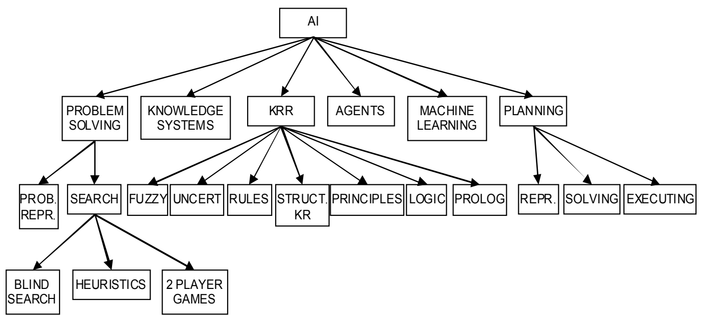


Fig. 4. Sample ontology for the Artificial Intelligence area

The following query filters out the LOM records: *"Find all learning objects that have as educational objective the learner's request or learning objects that have as educational objective the prerequisites of already selected learning objects. At the same time, ensure that all the constraints introduced by the learner's profile are met"*.

These queries are formulated in R-DEVICE using deductive rules. An example of such rules follows. We assume that the learner's request is stored in R-DEVICE as objects of the form: **`(learner-request (competency <string>))`**. For example, if the user requested educational material for learning Prolog, the stored object will be: **`(learner-request (competency "Prolog"))`**.

```
(deductiverule r1
  (learner-request (competency ?comp))
  ?lom <- (lom ((value purpose classification) "Educational Objective")
             ((entry taxon taxonpath classification) ?comp))
 =>
  (result (lomid ?lom))
)
(deductiverule r2
  (result (lomid ?lom))
  ?lom <- (lom ((value purpose classification) "Prerequisite")
             ((entry taxon taxonpath classification) ?comp))
 =>
  (learner-request (competency ?comp))
)
```

Fig. 5. Example of querying the metadata using R-DEVICE deductive rules

The R-DEVICE rules presented in Fig. 5 perform the following:

- Rule r_1 keeps the IDs of LOMs that achieve learner requests.
- Rule r_2 recursively searches for prerequisite LOMs, from the already selected ones, and augments the learner requests.

The filtered set of metadata produced by R-DEVICE is transformed into PDDL and is fed to the planning module in order to find a course plan. The details concerning the planning system are presented in the following section. After the planner has constructed the course plan, the CP producer creates a "package" of e-learning content (encoded in XML) and forwards it to the learner.

6. The Planning System

The core of the PASER system is a planning engine, capable of providing curricula that achieve the educational goals of the learner. The problem of synthesizing curricula from a database of educational resources, given the learners objectives and his current knowledge state, can be considered as a planning problem and such a view enables the development of fully autonomous systems that generate course plans for each student separately, meeting his needs and abilities.

A planning problem is usually modelled according to *STRIPS* (Stanford Research Institute Planning System) notation. A planning problem in STRIPS is a tuple $<I,A,G>$ where I is the Initial state, A a set of available actions and G a set of goals.

States in STRIPS are represented as sets of atomic facts. All the aspects of the initial state of the world, which are of interest to the problem, must be explicitly defined in I. State I contains both static and dynamic information. For example, I may declare that object John is a truck driver and there is a road connecting cities A and B (static information) and also specify that John is initially located in city A (dynamic information). State G on the other hand, is not necessarily complete. G may not specify the final state of all problem objects even because these are implied by the context or because they are of no interest to the specific problem. For example, in the logistics domain the final location of means of transportation is usually omitted, since the only objective is to have the packages transported. Therefore, there are usually many states that contain the goals, so in general, G represents a set of states rather than a simple state.

Set A contains all the actions that can be used to modify states. Each action A_i has three lists of facts containing:

- the preconditions of A_i (noted as $prec(A_i)$)
- the facts that are added to the state (noted as $add(A_i)$) and
- the facts that are deleted from the state (noted as $del(A_i)$).

The following formulae hold for the states in the STRIPS notation:

- An action A_i is applicable to a state S if prec(A_i) $\subseteq S$.
- If A_i is applied to S, the successor state S' is calculated as:
  $S' = S \setminus \text{del}(A_i) \cup \text{add}(A_i)$
- The solution to such a problem is a sequence of actions, which if applied to I leads to a state S' such as $S' \supseteq G$.

Usually, in the description of domains, action schemas (also called operators) are used instead of actions. Action schemas contain variables that can be instantiated using the available objects and this makes the encoding of the domain easier.

6.1. Problem Representation

There may be a few alternatives in formalizing the problem of automatic synthesis of educational resources, as a planning problem. A straightforward solution adopted by PASER is the following:

a) The facts of the problem are the competencies defined in the ontology of the thematic area of interest.
b) A state in the problem is a set of competencies, describing the current knowledge state of the learner.
c) The initial state of the problem is the set of competencies currently mastered by the learner as described in the Learner Information Package.
d) The goals of the problem are defined as a set of competencies that the learner wishes to acquire, as defined in the Learner Information Package.
e) There are three operators in the definition of the problem:
   - Consume an educational resource *con(L)*, where L refers to the specific educational resource as described by the IEEE LOM standard. The preconditions of *con(L)* are the competencies described in the *Classification-prerequisite* field. Similarly, the add effects of *con(L)* are the competencies described in the *Classification-educational objective* field. The delete list of *con(L)* is empty.
   - Analyze a goal *anl(G)*, which consults the ontology in order to find a set of sub-goals Z that can replace G. This operator is similar to the *methods* in Hierarchical Task Network Planning and it is used in order to allow the definition of competencies in various abstraction levels. The precondition list of *anl(G)* contains only G. The add list contains the sub-goals in which G can be analyzed (Z) and the delete list contains G.
   - Synthesize a set of goals *sth(S)*, which consults the ontology in order to find a single goal that can replace a set of sub-goals S. This operator is opposite to *anl(G)* and is also used in order to allow the definition of competencies in various abstraction levels. The precondition list of *anl(S)* contains S. The add list contains the goal G which subsumes S and the delete list contains S.

Consider for instance, the example in Fig. 4. The specific problem is modelled as described in Fig. 6.

```
IS (Initial state) = [principles, problem solving]
G (Goals) = [solving, executing, prolog]
con(Theory of Planning): prec=[logic, blind search,
      heuristics], add=[repr, solving], del=∅
...
anl(ai): prec=[ai], add=[problem solving, knowledge systems,
      krr, agents, machine learning, planning], del=[ai]
...
sth(ai): prec=[problem solving, knowledge systems, krr,
      agents, machine learning, planning], add=[ai],
      del=[problem solving, knowledge systems, krr, agents,
      machine learning, planning]
...
```

Fig. 6. Educational request modeled as a planning problem

6.2. Translation to PDDL

PDDL (*P*lanning *D*omain *D*efinition *L*anguage) is the standard language for encoding planning problems. The basic feature of PDDL is the separation of the domain data from the planning data. The domain describes a family of similar problems and contains all the information that is general and does not depend on the actual instance (problem). Technically, the domain consists of the definitions of the object classes, the relations (predicates) between the classes and the operators (actions with un-instantiated variables) of the domain. The problem on the other hand, contains the information for the specific objects that take part in the problem, the initial state and the goals of the problem.

One difficulty in translating a course planning problem to PDDL is the fact that, although according to our representation there are only three operators, each action differs in the number of preconditions and effects, since this depends on the LOM that the action is considered to consume, for example. Therefore, the process of creating a general operator for the consume family of actions is not straightforward.

One way to overcome this is to model the specific actions of the problem directly and feed the planner with this information, without modelling the domain in PDDL. However, this process is planner-dependent and the PASER system will lose its modularity, as it won't be able to use a different planning module. Moreover, most planners, including HAP$_{EDU}$, have a pre-planning phase in which the domain is analyzed in order to extract information for the guiding mechanisms and this phase must be reorganized if not omitted in order to cope with direct action specifications.

The way to overcome the difficulty that was finally adopted by PASER is to use conditional effects, universal preconditions and explicit declaration of the relations in the definition of the domain. More specifically, the domain contains two classes, named **`Competency`** and **`LOM`** and the relations

- **`holds(?Competency)`**: which states that a specific competency is true in a state
- **`requires(?LOM,?Competency)`**: which states that the **`Competency`** is required in order to consume the LOM
- **`adds(?LOM,?Competency)`**: which states that the **`Competency`** is learned by the learner after the consumption of the **`LOM`**.
- **`is-part-of(?Competency1,?Competency2)`**: which states that **`Compe-tency2`** is a part of **`Competency1`**. This hierarchy information is extracted from the ontology and is used to define competencies in various levels of abstraction.

We suggestively provide the definition of the operator *consume* in PDDL:

```
  (:action con:parameters(?LOM1)
   :precondition(and (LOM ?LOM1)
(forall (requires ?LOM1 ?Competency1) (holds ?Competency1)))
   :effect(and ((forall (?Competency2)
(when (adds ?LOM1 ?Competency2) (holds ?Competency2)))))
```

The definition above suggests that the operator **con** (consume) for a specific LOM can be consumed if all the competencies (universal precondition) that are required by the LOM hold in the current state. The operator uses conditional effects in order to state that all the competencies that are added by the LOM will hold in the successor state.

6.3. The HAP$_{EDU}$ Planner

The planning system that was embedded in PASER is called HAP$_{EDU}$, as already stated, and is able to handle universal preconditions and conditional effects in a simple way, also adopted by the vast majority of the planning systems. Specifically, it fully instantiates the operators in a preliminary phase and uses the ground actions throughout the rest of the planning process. HAP$_{EDU}$ is a state – space planning system, based on the HAP planner [3] which is modified in order to implicitly support abstraction hierarchies that are needed in course planning problems.

The support for levels of abstraction is realized through actions that analyze competencies in their parts (operator **anl**) and synthesize higher-level competences from their parts (operator **sth**). Moreover the planning system must be aware of the existence of different abstraction levels in the encountered facts and deploy the appropriate logical tests, in order to examine whether for example the competencies required by a LOM are present in the current state. Following the example in Fig. 4, note that the LOM "REPRESENTATIONS" can be consumed although the competencies "PROB. REPR." and "BLIND SEARCH" are not included in the initial state, as they are parts of the "PROBLEM SOLVING" competency according to the ontology.

The HAP$_{EDU}$ system works in two phases. In the first phase the systems analyzes the problem structure in order to estimate the distances between all the problem's actions and the goals. The distance between a state *S* and an action *A* is merely the number of actions that need to be applied to *S* in order to reach another state *S'*, in which the preconditions of *A* hold. The fact that the heuristic function of HAP$_{EDU}$ is based on distances of actions rather than facts enables it to keep better track of the various interactions between the facts, and therefore produce better estimates. In the second phase, the proposed heuristic is used by a regression planner employing a weighted A* search strategy and various other speedup mechanisms.

7. Case Study

This section briefly presents an scenario of a learner logging into the PASER system, in order to request instructional material for achieving his educational goals. The learning objects PASER gives access to are located either in the system's storage space or in remote sites owned by content providers. In both cases, proper LOM metadata records should be created and stored in the system, by using a form provided by PASER (Fig. 7).

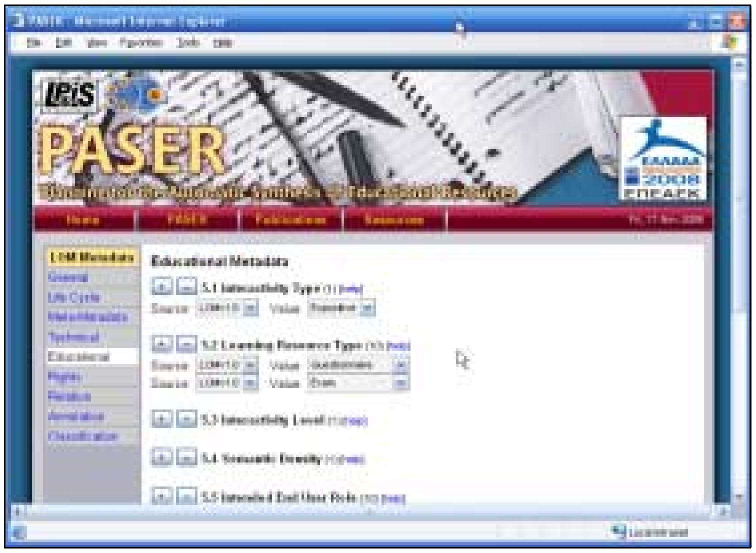
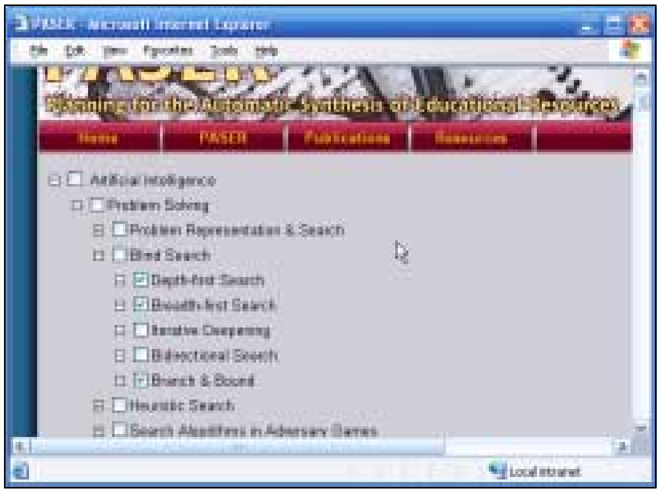


Fig. 7. Creating LOM metadata in PASER.

To access the services of PASER, a learner must be registered into the system. Besides defining the log-in credentials, the learner should also create a profile by providing some personal details (like age, native language, etc.) as well as technical details (like Internet connection type, operating system, browser, etc.). Additionally, the user is able to specify his current state of knowledge selecting the appropriate terms of the competencies ontology, presented to him in a tree hierarchy, as depicted in Fig. 8. Note that, although this is not compulsory, in the current implementation the user is responsible for stating his current knowledge state (if he doesn't, the system assumes he is a novice learner). However, we are studying the possibility of updating the Learners Information Package (LIP) dynamically through a series of evaluation tests.

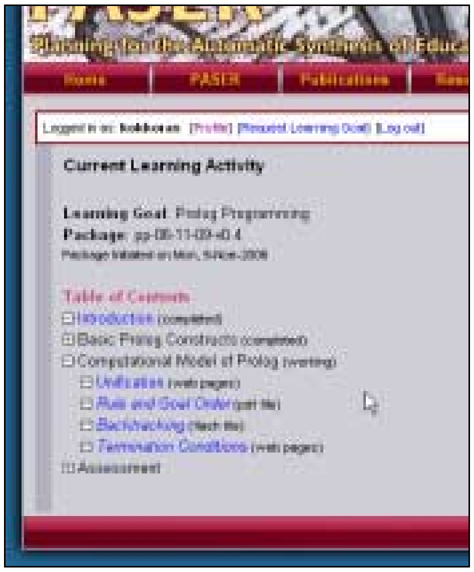

PASER - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Planning for the Automatic Synthesis of Educational Resources
Home
PASER
Publications
Resources
Local intranet

- ☐ Artificial Intelligence
  - ☐ Problem Solving
    - ☐ Problem Representation & Search
    - ☐ Blind Search
      - ✓ Depth-first Search
      - ✓ Breadth-first Search
      - ☐ Iterative Deepening
      - ☐ Bidirectional Search
      - ✓ Branch & Bound
    - ☐ Heuristic Search
    - ☐ Search Algorithms in Adversary Games

Fig. 8. Forming the learner's knowledge state

The next step for the learner is to select a number of competencies that correspond to his educational objectives (goals). This can be done from a hierarchy of RDCEO terms similar to the one presented to him for forming his current knowledge level (Fig. 8).

At this point the PASER system has all the necessary information for transforming this educational task in a planning problem, consisting of the initial state (current knowledge level), the goals (educational objectives) and the actions (available learning objects). At the request of the learner PASER calls the $\mathrm{HAP}_{\mathrm{EDU}}$ system in order to solve the planning problem and retrieve a sequence of learning objects, which are presented to the user as an educational content package (see Fig. 9), the "consumption" of which will eventually lead the learner in the desired knowledge state.



Logged in as: kokkoras [Profile] [Request Learning Goal] [Log out]

Current Learning Activity

Learning Goal: Prolog Programming
Package: pp-08-11-09-40.4
Package initiated on Mon, 8-Nov-2006

Table of Contents
- Introduction (completed)
- Basic Prolog Constructs (completed)
- Computational Model of Prolog (working)
  - Unification (web pages)
  - Rule and Goal Order (pdf file)
  - Backtracking (flash file)
  - Termination Conditions (web pages)
- Assessment

Fig. 9. A Content Package formed from the plan retrieved by $\mathrm{HAP}_{\mathrm{EDU}}$

As depicted in Fig. 9, the content package is presented to the user as a table of contents consisted of direct links to learning objects (web pages, pdf files, etc.). The order of the material has been decided by $\mathrm{HAP}_{\mathrm{EDU}}$ while access to the material is provided gradually. For example, in Fig. 9, the "*Rule and Goal Order*" learning object will become available after completion of the "*Unification*" unit. The user can interrupt the learning activity at will and continue at a latter time.

8. Related Work

Automatic course generation has been an active research field for almost two decades. One of the first attempts in creating an automatic system, using planning techniques for the synthesis of educational resources is the work by Peachy and McCalla [4]. The learning material is structured in concepts and prerequisite knowledge is defined, which states the causal relationships between different concepts. Then, planning techniques are used in order to find plans that achieve the learning goals and to monitor the outcomes of the plan.

Karampiperis and Sampson have conducted a lot of research in the field of Instructional planning for Adaptive and Dynamic Courseware Generation. In a recent approach [5], they use ontologies and learning object metadata in order to calculate the best path through the learning material. Their approach is based on a strong connection between the domain ontology (similar to our competencies ontology) and the available Learning objects. Their method implies that content creators/providers should connect new learning objects with existing ones, based on the 'Relation' category of the IEEE LOM specification. Their proposal does not restrict addition of disjoined learning objects; however the planning algorithm will not be able to take them into account while constructing the learning path.

There are a number of systems that serve as course generators that automatically assemble learning objects retrieved from one or several repositories. These systems usually adopt the Hierarchical Task Network (HTN) planning framework. In [6] Ulrich uses the JShop2 HTN planner in order to represent the pedagogical objectives as tasks and the ways of achieving the objects as methods in order to obtain a course structure. The proposed method is quite efficient in terms of planning time and it allows the domain engineer to encode methods for different pedagogical goals. The main disadvantage of the approach is the extensive need for domain information to be encoded in the definition of the planning problem. Furthermore, the LOMs are considered to share a common level of abstraction, contrary to the competencies hierarchy maintained by the PASER ontology. However, the planning engine itself is a very effective one and it is in our future plans to examine the possibility of embodying a similar one in PASER.

Baldoni et al [7] propose the selection and composition of learning resources in the Semantic Web, using the SCORM framework. The learning resources are represented in the knowledge level in terms of prerequisites and objectives, in order to enable the use of automated reasoning techniques. Their proposal shares similar principles with PASER, i.e. construction of learning paths is based on competencies rather than predefined relations among the learning objects. However, the lack of an ontology connecting the competencies used in the LOM specifications, does not allow their approach to overcome unsupported competencies.

TANGRAM [8] is an integrated learning environment for the domain of Intelligent Information Systems. It is implemented as a Web application built on top of a repository of educational content and intended to be useful to both content authors and students interested in the domain of IIS. The key feature of the system is its capability to decompose Learning Objects into smaller units, which can be later reassembled into new Learning Objects, personalized to the user's domain knowledge, preferences, and learning styles. Furthermore, similarly to PASER, TANGRAM also provides guidance and directions towards the most appropriate learning path that the student has to follow each time.

Another similar system is *OntAWare* [9], a knowledge management, courseware creation and delivery ontology-based system. Actually, OntAWare comprises a set of software tools for learning content authoring, management and delivery. The primary characteristic of the authoring environment is the semi-automatic generation of learning objects, applying graph transformations on appropriate domain ontologies. Adaptation is also among the system's key attributes, allowing users to assume the role of the author/instructor and customize the teaching and learning strategies in the generation of learning objects as well as produce strictly sequenced curricula or allow variations in student navigation.

DCG [10] (Dynamic Courseware Generator) represents one of the first attempts towards WWW-based dynamic learning content generation. The system generates individual courses, according to the learner's goals and previous knowledge (initialized with a pre-test) and dynamically adapts the course, according to the learner's success in acquiring knowledge. DCG utilizes "*concept structures*" (e.g. topic maps) as a road-map, in order to generate learning paths for the courses. On the other hand, the system displays a number of drawbacks, since it comprises a relatively old implementation: the generation of a learning path is based on simplistic planning techniques and heuristics, while the variety of learning objects handled is quite limited (i.e. HTML resources).

9. Conclusions and Future Work

This article presented PASER, a system aiming at augmenting the educational process in the e-Learning environment. PASER is able to store, manage and compose electronic educational material (learning objects), in order to provide personalized curricula to the learner. We presented the overall architecture of the system, focusing mainly in the core modules, namely the planning sub-system responsible for synthesizing the curricula, the ontology and metadata repository and the knowledge base module that queries and reasons on learning metadata.

However, there are still many open design and implementation issues. The project is still in its early stages and there is still a lot of work left to be done. Additionally, there are design aspects that need further investigation in order to improve the system in terms of functionality and efficiency. For instance, one of our imminent plans involves an extension to the system, namely TCS (Text Classification System) that will be able to automatically match descriptions expressed in free text, concerning the resource's prerequisites and objectives, to the RDCEO terms used in the ontology. An extensive user evaluation is also a non-trivial issue we are planning to address soon.

10. Acknowledgments

This work was partially supported by the PYTHAGORAS II program which is jointly funded by the Greek Ministry of Education (EPEAEK) and the European Union.

11. References

1. Vrakas, D., Kokkoras, F., Bassiliades, N., Vlahavas, I,. (2006). Towards Automatic Synthesis of Educational Resources through Automated Planning, *Proceedings of the 4th Hellenic Conference on Artificial Intellligence (SETN '06)*, pp. 421-431.

2. Bassiliades N., Vlahavas I., "R-DEVICE: An Object-Oriented Knowledge Base System for RDF Metadata", *International Journal on Semantic Web and Information Systems*, Vol. 2, No. 2, pp. 24-90, 2006.

3. Vrakas, D., Tsoumakas, G., Bassiliades, N., Vlahavas, I. (2005). HAPrc: An Automatically Configurable Planning System, *AI Communications*, Vol. 18 (1), pp. 1-20.

4. Peachy, D. R., Mc-Calla, G. I. (1986). Using planning techniques in intelligent tutoring systems. *International Journal of Man-Machine Studies*, Vol. 24, pp. 77–98.

5. Karampiperis, P., Sampson, D. (2004). Adaptive instructional planning using ontologies. Proceedings of the *4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*, pp. 126–130.

6. Ullrich, C. (2005). Course generation based on HTN planning. Proceedings of *13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems*, pp. 74-79.

7. Baldoni, M., Baroglio, C., Patti, V., and Torasso. L. (2004). Reasoning about learning object metadata for adapting SCORM courseware. Proceedings of the *International Workshop on Engineering the Adaptive Web, EAW'04: Methods and Technologies for personalization and Adaptation in the Semantic Web*. Eindhoven, The Netherlands, pp. 4-13.

8. Jovanovic, J., Dragan, G. & Devedzic, V. (2005). TANGRAM: An Ontology-based Learning Environment for Intelligent Information Systems. In G. Richards (Ed.), Proceedings of *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2005* (pp. 2966-2971). Chesapeake, VA: AACE.

9. Holohan, E., Melia, M., McMullen, D., and Pahl, C., "Adaptive E-Learning Content Generation based on Semantic Web Technology", In *Proc. Int'l Workshop on Applications of Semantic Web Technologies for E-Learning*, Amsterdam, The Netherlands, 2005.

10. Vassileva J.: "Dynamic Course Generation on the WWW", *Proceedings 8th World Conference on AI in Education (AI-ED97)*, Knowledge and Media in Learning Systems, Kobe, Japan, 1997.