PASER: A Curricula Synthesis System based on Automated Problem Solving

DIMITRIS VRAKAS[1], GRIGORIOS TSOUMAKAS[1], FOTIS KOKKORAS[1], NICK BASSILIADES[1], IOANNIS VLAHAVAS[1], DIMOSTHENIS ANAGNOSTOPOULOS[2]

[1]Department of Informatics
Aristotle University of Thessaloniki
GREECE
{dvrakas, kokkoras, greg, nbassili, vlahavas}@csd.auth.gr http:\\lpis.csd.auth.gr
[2]Department of Geography
Harokopio University, Athens
GREECE
dimosthe@hua.gr

Abstract: - This paper presents PASER, a system for automatically synthesizing curricula using AI Planning and Machine Learning techniques on an ontology of educational resources metadata. The ontology is a part–of hierarchy of learning themes which correspond to RDCEO competencies. The system uses an automated planner, which given the initial state of the problem (learner's profile, preferences, needs and abilities), the available actions (study an educational resource, take an exam, join an e-learning course, etc.) and the goals (obtain a certificate, learn a subject, acquire a skill, etc.) constructs a complete educational curriculum that achieves the goals. PASER is accompanied by a Machine Learning module that classifies textually described users' learning requests to competencies registered within the ontology. Furthermore, the ML module interactively assists content providers in constructing educational resources metadata (LOM records) that comply with the ontology concerning both learning objectives and prerequisites.

Key-Words: - Curricula Synthesis, Automated Planning, Text Mining

1 Introduction

The lack of widely adopted methods for searching the Web by content makes difficult for an instructor or learner to find educational material on the Web that addresses particular learning and pedagogical goals. Aiming at providing automation and personalization in searching and accessing educational material, as well as and interoperability among them, several education related standards have been developed. These standards concern recommended practices and guides for software components, tools, technologies and design methods 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.

In this paper we report on an ongoing project for the development of such a system. The proposed system, called PASER (Planner for the Automatic Synthesis of Educational Resources) 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 c) a planning system called HAP_{EDU} that automatically constructs course plans and d) a Text Classification System, called TCS, which is able to automatically extract RDCEO competencies from textual descriptions.

PASER is compliant with 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 paper is organized as follows: Section 2 previous related work on the area of automated course synthesis. Section 3 presents the overall architecture of the proposed system, whereas Sections 4, 5 and 6 present in more detail its major subsystems. Finally, section 7 concludes the paper and poses future directions.

2 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 (1986), in which the learning material is structured in concepts and prerequisite knowledge is defined, which states the causal relationships between different concepts. Then they use planning techniques in order to find plans that achieve the learning goals and to monitor the outcomes of the plan.

Karampiperis and Sampson have carried a lot of research in the field of Instructional planning for Adaptive and Dynamic Courseware Generation. In a recent approach (Karampiperis and Sampson

2004) they use ontologies and learning object metadata in order to calculate the best path through the learning material.

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 HTN planning framework (Currie and Tate 1991, Erol, Hendler and Nau 1994). In (Ulrich 2005) 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. Similarly, Baldoni et al (2004) propose a system for selecting and composing learning resources in the Semantic Web, using the SCORM framework for the representation of learning objects. The learning resources are represented in the knowledge level, in terms of prerequisites and knowledge supplied, in order to enable the use of automated reasoning techniques.

In (Bassiliades et al 2003), X-DEVICE, an intelligent XML repository system for educational metadata is presented. X-DEVICE can be used as the intelligent back-end of a WWW portal on which "learning objects" are supplied by educational service providers and accessed by learners according to their individual profiles and educational needs. X-DEVICE transforms the widely adopted XML binding for educational metadata into a flexible, object-oriented representation and uses intelligent second-order logic querying facilities to provide advanced, personalized functionality.

An older approach for a tool that generates individual courses according to the learner's goals and previous knowledge and dynamically adapts the course according to the learner's success in acquiring knowledge is DGC (Vassileva 1997). DGC uses "concept structures" as a road-map to generate the plan of the course.

3 System Architecture

PASER is a synergy of six processing modules (Fig. 1), namely a planner, an Ontology & Metadata Server, the R-DEVICE module, the TCS classification system 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 later provides competency definitions that are referenced by the other two. In addition, it is used by the Ontology & Metadata Server providing in this way a system-wide consistent competency vocabulary. We also assume that all metadata are checked by an expert user before they are entered in to 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.

The system supports two main types of users, namely content providers and learners.

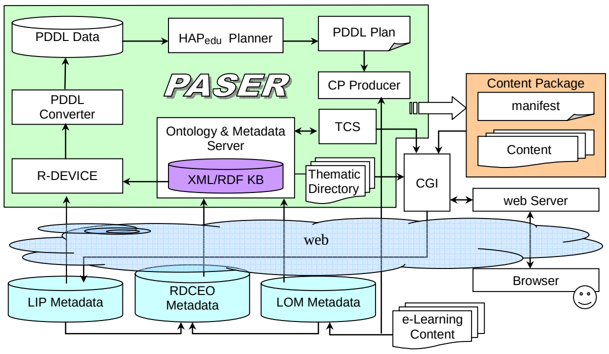

Fig. 1 PASER – System Architecture

The content providers interact with the system in order to add new educational material in the system and provide the appropriate metadata (LOM record). In order for the PASER to automatically synthesize curricula the LOM record must contain information about the prerequisites and the objectives of the resource expressed in terms registered within the PASER's ontology. This is either achieved manually with the content provider being responsible for selecting the appropriate terms from the ontology or semi-automatically with the TCS module 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 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. Furthermore, the learner is able to give a textual description of his educational goals and let the TCS classification system propose a set of themes that correspond to them.

As soon as the user selects a theme, 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 (Bassiliades and Vlahavas 2006) 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. In 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 (in R-DEVICE terminology) describing learning 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 (Vrakas et al 2005) 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. Note that, at the current stage we do not take into account the performance of the user regarding the supplied educational material. In a later stage, 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 updates his LIP record, properly.

4 Data Models, 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 (2006). 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 data preparation phase performed by the R-DEVICE module of PASER, a phase that will feed the planner with the appropriate data, extensible usage of the *classification* elements of LOM records is done. 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 between learning objects and between learning objects and characteristics of the learner. This information together with other constraints imposed over the learning objects due to the learner's preferences, are exploited by the R-DEVICE module, in order to filter out the learning object repository and keep only the "promising" objects. Informally encoded examples of the competency related information located in LOM and LIP metadata, are presented in Fig. 2 (a) and (b), respectively.

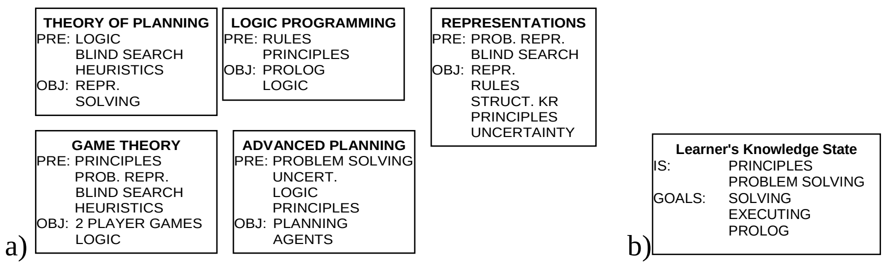

Fig. 2 Prerequisites and educational objectives of some, informally presented, learning objects (left) and initial knowledge state (IS) and learning objectives (GOALS) for a learner (right)

Finally, competencies are organised in a part-of hierarchy, as depicted in Fig. 3. This hierarchy allows the decomposition of learning objectives into sub-objectives, since the completion of all sub-objectives is a prerequisite to accomplish the super-objective. As a result, the system is able to relate learning objects with learner objectives in various levels of granularity. The part-of hierarchy is represented in RDF and not directly in RDCEO because the latter does not allow the representation of hierarchical relationships. However, each competency of the ontology (RDF resource) is related to its RDCEO counterpart.

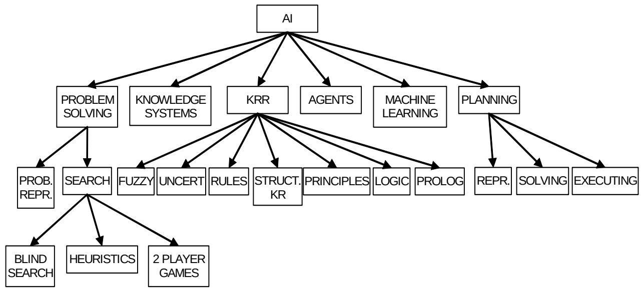

Fig. 3 Sample ontology for the Artificial Intelligence area

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")).

The R-DEVICE rules presented in Fig. 4 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.

```
(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. 4 Example of querying the metadata using R-DEVICE deductive rules

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.

5 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 capabilities.

5.1 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 that is 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 (Currie and Tate 1991, Erol, Hendler and Nau 1994) 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. 3. The specific problem is modeled as described in Fig. 5.

5.2 The HAP_{EDU} Planner

The planning system that was embedded in PASER is called HAP_{EDU} and is based on the HAP planner (Vrakas et al 2005) which is modified in order to implicitly support abstraction hierarchies that are needed in course planning problems.

```
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. 5 Educational request modeled as a planning problem

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 see whether for example the competencies required by a LOM are present in the current state. Following the example in Fig. 3, 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.

6 The Text Classification System

The PASER system requires the specification of information related to educational resources according to the terms in its ontology. However, directly using a complex hierarchy with part-of relations is cumbersome. To deal with this problem PASER allows its users to insert information in free-text, which is a user-friendly and easy-to-use format even for people not familiar with computer technology. Subsequently, the Text Classification System (TCS) of PASER matches the textual description of an educational resource to one or more nodes of the educational terms ontology of PASER. These nodes are presented to the users in order to assist them in the task of selecting the proper terms that correspond to their educational resources.

The TCS is engaged in two cases within the PASER system. The first case is when content providers (professors, teachers, etc) are constructing metadata of educational resources (LOM records) and the second case is when users are entering their learning requests. In the first case the content providers have to specify the goals and the requirements of the educational resource, while in the second case the users must specify the educational goals that they want to achieve.

Within TCS, textual descriptions of educational resources are represented using the "bag-of-words" approach. According to this approach each description is associated with a vector of words. The elements of the vector can have either Boolean values (0 or 1) in order to denote presence or absence of the particular word in the description, or weights (usually numerical values between 0 and 1) to denote the importance of the word for the description (e.g. word frequency). We use the former approach for simplicity.

Each description that is stored in PASER is associated with one or more terms of the ontology. This renders the learning problem a multi-label classification problem, as TCS has to predict all related terms of a new description. In order to deal with such a problem we follow the one-versus-rest approach, which learns as many classifiers as the terms, where the task of each classifier is to classify a text as related with this term or not.

We use the Naïve Bayes classifier in TCS, because it is suitable for textual data and is incremental. It is important for TCS to be incremental as initially there might be no knowledge available in the system. As users start to enter educational resources and specify the corresponding terms of the ontology, the system will gradually learn to predict the proper terms for new descriptions.

In addition we use a feature selection methodology in order to remove redundant, noisy and irrelevant words that might influence the performance of the classifier. Previous studies have shown that feature selection improves the classification accuracy of Naïve Bayes classifiers. Specifically, we employ the chi-squared measure to evaluate the predictive ability of each word and keep the 2000 words with the highest chi-square measure. The values of the chi-square measure are incrementally updated with every new textual description that is stored in PASER.

7 Conclusions

This paper presented PASER, a system aiming at facilitating the educational process in the e-Learning environment. PASER is able to store, manage and synthesize electronic educational material (learning objects) to provide personalized curricula to the learner. We presented the overall architecture of the system, focusing mainly in the core modules, namely the ontology and metadata repository, the knowledge base system that queries and reasons on these metadata, the planning sub-system responsible for synthesizing the curricula and the Text Classification System that assist users in selecting the appropriate competencies.

However, there are still many open design and implementation issues. As stated in the paper, the project is still in its early stages and although initial implementations of some sub-systems have been realized, there is a lot of work to be done. Additionally, there are design aspects that need further investigation in order to improve the system in terms of functionality and efficiency.

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.

References:

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

[2] Bassiliades, N., Kokkoras, F., Vlahavas, I., & Sampson, D. (2003). An Intelligent Educational Metadata Repository. Intelligent Systems, Techniques and Applications, 4, 297-337.

[3] Bassiliades N., & Vlahavas I. (2006). R-DEVICE: An Object-Oriented Knowledge Base System for RDF Metadata. International Journal on Semantic Web and Information Systems, 2(2), 24-90.

[4] Currie, K., & Tate, A. (1991). O-Plan: The Open Planning Architecture. Artificial Intelligence, 52(1), 46-86.

[5] Erol, K., Hendler, J., & Nau, D. (1994). UMCP: A Sound and Complete Procedure for Hierarchical Task Network Planning. 2nd International Conference on Planning Systems, Chicago, Illinois, 88-96.

[6] Fox, M., & Long, D. (2003). PDDL2.1: An extension to PDDL for expressing temporal planning domains. Journal of Artificial Intelligence Research, 20, 61-124.

[7] IMS Global Learning Consortium (2006), Specifications, http://www.imsglobal.org/specifications.html.

[8] Karampiperis, P., & Sampson, D. (2004). Adaptive instructional planning using ontologies. 4th IEEE International Conference on Advanced Learning Technologies, 126–130.

[9] Peachy, D. R., & Mc-Calla, G. I. (1986). Using planning techniques in intelligent tutoring systems. International Journal of Man-Machine Studies, 24, 77–98.

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

[11] Vassileva, J. (1997). Dynamic Course Generation on the WWW. 8th World Conf. on AI in Education (AI-ED97), Knowledge and Media in Learning Systems, Kobe, Japan, 498-505.

[12] Vrakas, D. Tsoumakas, G., Bassiliades, & N., Vlahavas, I. (2005). HAPrc: An Automatically Configurable Planning System, AI Communications, 18 (1), 1-20.