

Document downloaded from:

<http://hdl.handle.net/10251/79349>

This paper must be cited as:

Garrido Tejero, A.; Morales, L.; Serina, I. (2016). On the use of case-based planning for e-learning personalization. *Expert Systems with Applications*. 60:1-15.
doi:10.1016/j.eswa.2016.04.030.



The final publication is available at

<http://dx.doi.org/10.1016/j.eswa.2016.04.030>

Copyright Elsevier

Additional Information

This is the author's version of a work that was accepted for publication in *Expert Systems with Applications*. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in *Expert Systems with Applications*, 60, 1-15, 2016. DOI:10.1016/j.eswa.2016.04.030

On the Use of Case-Based Planning for E-learning Personalization

Antonio Garrido^a, Lluvia Morales^b, Ivan Serina^c

^a*Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia (Spain)*

^b*Universidad Tecnológica de la Mixteca, Oaxaca 69000 (Mexico)*

^c*University of Brescia, via Branze, 38, Brescia (Italy)*

Abstract

In this paper we propose *myPTutor*, a general and effective approach which uses AI planning techniques to create fully tailored learning routes, as sequences of Learning Objects (LOs) that fit the pedagogical and students' requirements.

myPTutor has a potential applicability to support e-learning personalization by producing, and automatically solving, a planning model from (and to) e-learning standards in a vast number of real scenarios, from small to medium/large e-learning communities. Our experiments demonstrate that we can solve scenarios with large courses and a high number of students. Therefore, it is perfectly valid for schools, high schools and universities, especially if they already use *Moodle*, on top of which we have implemented *myPTutor*. It is also of practical significance for repairing unexpected discrepancies (while the students are executing their learning routes) by using a Case-Based Planning adaptation process that reduces the differences between the original and the new route, thus enhancing the learning process.

Keywords: e-learning, learning route personalization, planning, plan adaptation, case base planning

Email addresses: agarridot@dsic.upv.es (Antonio Garrido),
lluviamorales@mixteco.utm.mx (Lluvia Morales), ivan.serina@unibs.it (Ivan Serina)

1. Introduction

E-learning is the process of providing on-line courses on the Internet for students so that they can study and learn from any place and computing device (personal computer, mobile phone, tablet, etc.) by using electronic media, information, Internet technologies and platforms, such as Learning Management Systems (LMSs). Although e-learning has become an increasingly popular training option, it cannot rely just on the upload of contents to the Internet or the developments of new standards. On the contrary, it needs to offer a feasible, personalized way that facilitates and enhances the students' learning process by combining such contents appropriately.

As proposed in related literature (Caputi & Garrido, 2015; Comi et al., 2015a; Essalmi et al., 2015; Garrido & Onaindia, 2013; Garruzzo et al., 2007b; Kurilovas et al., 2015; Rosaci & Sarne, 2010), a revolutionary key challenge of the current century is advanced personalized learning to promote adaptivity and fully tailoring of the e-courses. The idea is to use intelligent systems to construct and recommend a personalized learning route of contents that fit the individual requirements of each student, and even the device each student is using at that moment (Garruzzo et al., 2007a; Rosaci & Sarne, 2010).

As a motivating example, let us assume two students with different background (initial knowledge) and learning outcomes, interested in the same course. Obviously, under a fully personalized perspective, the LOs and their sequence cannot be the same for both students. A different subset of LOs can be combined in different ways according to the learning style, current knowledge and learning goals of each student. For instance, one student will need a shorter route than the other, or a particular type of contents, different to the other's. Also, although in some scenarios the LOs could be the same, the learning route to be planned will have to be different according to the specific needs. Therefore, we require some kind of planning to select the best sequence of LOs, and in the best order.

More precisely, rather than having a predefined flow of contents the student

has to follow in a course, which may be too teacher-oriented and somewhat frustrating for the student, we want to have an individualized sequence of contents that is generated and accommodated to what the student needs, thus being 100% student-oriented. Achieving such a high level of individualization is not
35 a straightforward enterprise as it requires: i) the combination of pedagogical theories (Brusilovsky & Vassileva, 2003), and ii) to take into account the causal relationships among the tasks to be done in the course (Caputi & Garrido, 2015; Garrido et al., 2012). For instance, a given task has some prerequisites to be held before it can be initiated (e.g. when some previous knowledge has to be ac-
40 quired), which means some orderings among tasks may be arbitrary but others are compulsory and, therefore, enforced in any individual route.

The construction of personalized routes requires an intelligent decision-making procedure to recommend the most adequate content for each student in every step of his/her learning process. Unfortunately, e-learning content selection is
45 difficult. It depends on many variables, involving learning contents, their (semantic) ontology, their degree of difficulty, the time required, how long the course lasts, the available time each student has, the student's preferences and learning styles, the resources that are available, the devices to be used, and also the level of cooperation and peer-to-peer (P2P) group formation among tutors
50 and students (Messina et al., 2013). As we will discuss in the related work section, many techniques can be applied here and, particularly, AI planning is very valuable not only to recommend contents that fit the students' needs, but also to find the right order in which such contents need to be sequenced (Brusilovsky & Vassileva, 2003; Caputi & Garrido, 2015; Castillo et al., 2010; Garrido & On-
55 aindia, 2010, 2013; Ullrich & Melis, 2009). After all, planning can be seen as an intelligent reasoning process to select the right contents and to place them as an ordered route of executable tasks to reach certain goals subject to several constraints.

In this paper we present an approach, named *myPTutor*, which takes as
60 an input an e-learning model described in a standard e-learning language and produces a PDDL (*Planning Domain Definition Language*) model as an output.

In particular, it applies standard AI planning and CBP (*Case Base Planning*) techniques to the generation and sequencing of e-learning routes, which are fully tailored to the students' profiles and necessities. Our main contributions address
65 the following topics:

- Knowledge representation, in which we analyze and extract metadata information from learning contents encoded in e-learning standards, and produce an automated compilation of standard PDDL domain+problem files. This PDDL representation allows us to use any PDDL-compliant
70 planner, thus making our model planner (i.e. solver) independent.
- Learning route personalization, not only in terms of contents but also in terms of their sequencing. In planning terminology this means a plan, generated by a case-based planner or any other planner.
- Content and rules datasource, as a CBP repository for planning domain+file
75 compilations that contains students' learning information to be reused in the future. This has some resemblance to a collaborative recommendation technique, which reuses some recommendations that appear the most similar to similar students. In *myPTutor*, the stored compilations are successively analyzed by our case-based planner (Serina, 2010), which re-
80 trieves the best element that fits the current requirements and only adapts it if necessary (Fox et al., 2006).
- Learning designs development. Particularly, we provide a simple translation of the resulting sequence learning contents (plans) into another standard representation, namely learning design (IMS, 2008), that provides a
85 usable manifest for standard on-line learning platforms, thus closing the e-learning cycle.
- Extension of *Moodle*, a well-known and widely used LMS. We have implemented a full vision that encompasses all the previous aspects on top of *Moodle*, as a flexible way to make curriculum authoring easier. All in all,
90 our contribution shows a practical significance to help tutors and teachers

choose the most suitable learning route and semi-automatically adapt it in accordance with the students' goals and individual features.

The remaining part of the paper is structured as follows. Section 2 explores some related work and how planning technology can be useful in e-learning. Section 3 describes the problem and introduces the role of planning for learning routes personalization. In Section 4 we present our general approach in detail, describing its structure, main elements, the e-learning-to-planning compilation and the CBP techniques we use. In Section 5 we explain our current implementation and how it is integrated on top of *Moodle*. A thorough evaluation with a large collection of experimental results is provided in Section 6. In Section 7 we discuss the lessons learnt, and the strong and weak features of our planning approach within an e-learning setting. Finally, in Section 8 we present the conclusions and the future work.

2. Related work and how planning can help

2.1. Related work

There are many aspects within e-learning in literature, which are beyond the scope of this paper. But in general, e-learning and course personalization has been traditionally addressed from a double perspective: student's modeling and adaptive+dynamic courseware composition.

On the one hand, student's modeling can be defined as the process of gathering relevant information to infer the current cognitive state of the student and to represent it to be accessible and useful in e-learning (Chrysafiadi & Virvou, 2013). There are many approaches to construct a student's model. For example, the overlay model that represents the student's knowledge level in Gaudioso et al. (2012); statistical, data mining and machine learning techniques to understand and improve the performance of the student's learning process (Campagni et al., 2015; Natek & Zwilling, 2014; Pena-Ayala, 2014); cognitive theories to explain human behavior (Alepis & Virvou, 2011); fuzzy logic modeling techniques and Bayesian networks to deal with the uncertainty of students' diagnosing

120 (Jeremic et al., 2012); and ontologies to reuse students' models (Comi et al.,
2015b; Clemente et al., 2011). These approaches can be used on its own or be
combined, thus building a hybrid model to personalize contents according to the
students' needs and available resources (Kyriacou, 2008). Although our work
could effectively adopt these modeling techniques, we do not explicitly focus
125 on students' modeling. We have limited our analysis to the students' informa-
tion necessary for automatically creating an AI planning representation. We
follow the approach described in Baldiris et al. (2007), based on Felder's clas-
sification (Felder & Silverman, 1988) and SCORM (2004), which allows us to
enrich the standard IMS-LIP representation (IMS, 2008) and improve students'
130 personalization.

On the other hand, the idea with course composition is to recommend and
personalize contents to students to ensure they complete all the activities that
an instructor deems important. Moreover, an interesting issue is to assist stu-
dents while navigating throughout the contents, and to monitor their progress
135 and interaction in order to dynamically adapt the contents to their specific
requirements.

From the point of view of personalizing e-learning contents, many techniques
have been applied, such as neuronal networks, adjacency matrices, constraint
programming models, soft computing methods, integer programming, machine
140 learning, multi-agent approaches, swarm intelligence models and recommenda-
tion techniques (Anaya et al., 2013; Brusilovsky & Vassileva, 2003; Comi et al.,
2015a; de Oliveira et al., 2013; Essalmi et al., 2015; Garrido et al., 2008; Idris
et al., 2009; Kurilovas et al., 2014, 2015; Martinez et al., 2004; Rosaci & Sarne,
2010). They all have in common the interest in simulating human decision-
145 making and recommending course contents in the form of different learning
tasks/objects (e.g. text documents, pictures, audio, video and, in general, any
type of multimedia contents). Previous works assign contents to predefined
classes or students' profiles while designing the adaptation rules. In de Oliveira
et al. (2013), a system that aids instructors in semi-automatic assessment is
150 proposed. The recommendation is treated as a multi-label classification task,

using the machine learning multi-label k-nearest neighbor algorithm, in which each student's profile is associated with one or more classes of programming activities, with individualized activities. Similar works also use machine learning techniques to define a recommendation decision table which warns of some problematic circumstances, and a pedagogical support system (decision tree) that provides a visual explanation of the recommendation suggested (Anaya et al., 2013). Some works exploit recommendation based on each student's profile and his/her previous contents (content-based recommendation), whereas others provide contents based on similar students (collaborative filtering recommendation), and even hybrid approaches that mix techniques. These recommenders analyze students' behavior and characteristics to personalize contents that could be useful to other students (Champiri et al., 2015; Park et al., 2012). But to do this, recommenders require to track and analyze previous students' interactions. In that line, some approaches emphasize such interaction by using effective knowledge sharing among the students (Garruzzo et al., 2007b; Rosaci & Sarne, 2010), thus supporting group formation methods and protocols that promote P2P e-learning, and exploit the opinions of other students as a kind of social recommendation (Messina et al., 2013).

From the point of view of monitoring the interaction of the students within the e-learning system, most of the existing works propose their own intelligent tutoring systems with adapted presentation *w.r.t.* the graphical aspect, but do not use existing LMSs.

The idea of combining both previous points of view naturally leads to a multi-agent learning approach (Comi et al., 2015a; Garruzzo et al., 2007a,b; Messina et al., 2013; Rosaci & Sarne, 2010). In particular, Garruzzo et al. (2007a,b) propose two learning systems, *ISABEL* and *MASHA-EL*, which associate to each student a device agent to autonomously monitor the student's behavior when accessing e-learning Web sites. The idea of handling different students' devices and offer different personalization *w.r.t.* them is very interesting, but it makes it necessary to explicitly model the differences in the graphical representation; e.g. considering the resolution of a picture, or the size of a video. Moreover,

each site is associated with a teacher agent, and when a student visits an e-learning site, the teacher agent collaborates with some tutor agents associated with the student to provide useful recommendations, although the student is
185 the last responsible to evaluate whether the content is relevant or not (Rosaci & Sarne, 2010).

Despite the advances of tutoring systems for e-learning, most systems still show some limitations: i) they do not fully use e-learning standards; or ii) they recommend personalized contents, but not a particular learning route as a
190 sequence of contents to attain the student's learning outcomes; or iii) they are not implemented/integrated as part of an existing LMS; or iv) they are usually limited to a specific solver. In contrast to related systems, our approach tries to overcome these limitations. It deals with standard e-learning metadata, which is automatically extracted and compiled as a PDDL model. We provide the
195 total or partial ordered sequence of contents. It is important to note that our approach differs from the recommender systems' idea in the sense that we do not only recommend contents to be used and re-analyzed step by step, but a fully tailored learning route to give students the opportunity to be aware of their progress in the learning process. We also allow teachers to define compulsory
200 or optional goals, an extra feature in many LMSs. We use *Moodle*, a well-known and one of the most used LMSs (Caputi & Garrido, 2015). Although we cannot directly offer personalization of the display or the device used, we rely on *Moodle* for every task related to the visualization. Currently, *Moodle* is evolving to allow users to make full use of their mobile devices. Finally, we do
205 not adopt a particular solver since we can use any PDDL-compliant planner as a solver.

2.2. Motivation for using planning

Metaphorically speaking, generating a personalized learning route resembles planning closely, which is enough motivation to face e-learning via planning. In
210 e-learning the main elements are: i) the students' background, ii) the learning goals to be attained, iii) the profile-adapted LOs with their prerequisites and

learning outcomes, iv) the ordering relations, and v) the tailored learning route. We can simply map these elements into planning terms, respectively, as: i) initial state, ii) top level goals, iii) actions with preconditions and effects, iv) causal link relations, and v) the solution plan.

The main objective of planning is to provide a plan as a sequence of actions, without the necessity of having previous information about other students, which is a limitation in the earlier stages of collaborative recommendation. Planning moves one step further and selects contents that fit the students' needs and also provides the order in which such contents need to be sequenced. In other words, planning is ideal for contents sequencing. Planning also deals with multi-objective metrics, very appealing for e-learning: students (and teachers) prefer a good learning route, in terms of time, competence, resources or cost, and not simply yet another route. As presented in Essalmi et al. (2015), metrics are a good support for analyzing and deciding which strategy and plan should be applied to improve personalization.

Our planning approach applies Case-Based Planning (CBP) techniques (Borrajó et al., 2015; Serina, 2010; Spalazzi, 2001). Intuitively, a learning route that is good for a given student will be also valid for a similar student, which can reduce the planning burden, analogously to collaborative recommendation techniques. Focusing on planning algorithms, CBP takes advantage of former problem-solving experiences by storing in a plan library previously generated solutions that can be reused to solve similar problems in the future (Serina, 2010). This is essential in e-learning. First, teachers want to include didactic issues that cannot be simply represented in terms of preconditions, effects and causal links. And when they can, they are reluctant to use complex and tedious constructs that are unfamiliar to them. Second, if the execution of a learning route fails it needs to be fixed. Although a new route starting from scratch can be replanned, it does not seem to be reasonable; a route adaptation that minimizes the differences between the original and the recalculated route seems more reasonable. Informally, students and teachers prefer a kind of *inertia* in the learning routes, which enhances the continuity in the learning process. In

other words, CBP techniques are ideal in this scenario for the definition, mem-
orization (of alternatives which are motivated by reasons beyond causal links
245 and orderings), retrieval and adaptation of learning routes to new requirements.

3. Description of the problem

The Web is full of interoperable digital resources, known as LOs, imple-
mented in XML standards such as SCORM, IMS or LOM (IMS, 2008; LOM,
2002; SCORM, 2004). But LOs in themselves are insufficient to accommodate
250 the different studying styles and preferences of the students in large courses.
The right selection and combination of LOs within these courses is the basis
for e-learning. For instance, according to pedagogical issues, a diagram is rec-
ommendable for visual students but not for verbal ones, and just the opposite
happens for a lecture. Consequently, e-learning must provide a student-centered
255 solution trying to offer a learning experience where the whole learning process
of users is fully tailored to their objectives, learning styles, needs and, more
generally, to their profiles.

After generating a tailored learning route, it needs to be executed in a LMS.
This means to assist students in navigating the route, monitoring it in order
260 to check their progress compared with the expected and proposing an actions'
sequence when inconsistencies are detected, i.e. a new temporal constraint,
appear. If this happens, a flexible process to make the remaining learning route
executable (by adding or removing LOs) becomes essential. And it is expected
that this process will not ignore the original students' interests, but try to use
265 the original learning route again as much as possible.

3.1. *AI planning for learning personalization*

The idea of personalizing learning routes to individual students' needs and
profiles comes from the observation that the traditional mode of instruction can-
not fully satisfy the different students' needs, especially in case of heterogeneous
270 classes, in presence of very different backgrounds and motivations.

As previously said, different techniques have been used to ensure flexibility within course personalization. In particular, we adopt AI planning techniques to effectively adapt the learning route of each student to his/her constraints. The effective integration of these techniques in LMSs is a complex task that
275 requires first, to provide to teachers and the students an effective access to the planned contents/LOs; second, to monitor the students' learning progresses in order to satisfy new necessities, or react to deviations from the planned learning route and, third, to allow the teachers to supervise the whole process.

This paper addresses the personalization of learning routes from a planning
280 perspective, and more particularly by using a case-based planning approach (Serina, 2010). But before describing this approach, we first introduce some essential background on planning.

3.2. AI planning and CBP techniques

AI planning is notoriously a very hard search problem. Its objective is to
285 define a plan, i.e. a partially ordered sequence of actions, that allows one or more agents to satisfy their goals starting for a specific initial configuration. Each action is specified by a set of preconditions, a set of effects, a duration and (probably) a cost. Preconditions are facts that must be true before the action execution, whereas the effects determine how the world changes as a
290 consequence of the action execution. The action cost is defined in terms of a specific metric and is usually related to the resource consumption and action duration.

Unfortunately, solving a planning problem is PSPACE-hard (Backstrom & Nebel, 1995) and a number of techniques and heuristics have been proposed
295 in literature to effectively use AI planning in applicative contexts. Case-Based planning is one of such techniques, with the aim of reusing previous experiences in order to solve similar problems in the future and efficiently manage complex problems.

In particular, using the formalization of Liberatore (2005), we define a *plan-*
300 *ning case* as a tuple $\langle \Pi_0, \pi_0 \rangle$, where Π_0 corresponds to a planning problem and

π_0 to its solution. A *case-base* or *plan library* corresponds to a set of relevant cases encountered by the system during its *life*: $\{\langle \Pi_i, \pi_i \rangle | 1 \leq i \leq m\}$.

In CBP, when a new (*target*) problem Π must be solved, the following phases are carried out:

- 305 1. *Plan Retrieval* to retrieve elements from the library similar to Π . However, one may also want to consider the case-base elements that are similar to Π after their objects have been systematically renamed; this corresponds to identifying, for each case-base element $\langle \Pi', \pi' \rangle$ similar to Π , a matching (or mapping) among the objects of Π' and the objects of Π that minimize
310 the adaptation effort in order to reuse π' . This is extremely important in our context, where teachers could decide to reuse a course, or a part of a course, that they or their colleagues have been already effectively used. Obviously, the students will not be the same, but, if their profile, goals and the resources available are similar to the ones in the case base, our
315 system will be able to propose a new high quality plan (learning route) to the teacher with a limited number of changes *w.r.t.* the one stored in the library.
2. *Plan Adaptation*, which allows us to reuse the solution plan of the case-base element most similar to Π , without repeating the whole search process but
320 concentrating only on the discrepancies in the input plan.
3. *Plan Revision*, which allows us to monitor and execute the solution plan π for Π and repair it if failures occur during its execution. A failure in planning is any type of situation that makes the execution of the plan differ from what expected (i.e. planned).
- 325 4. *Plan Storage*, which allows us to store $\langle \Pi, \pi \rangle$ as a new element in the library or simply discard it.

Moreover, as it will be better described in Section 4.3, we have adopted *plan merging techniques* (Garrido et al., 2012; Yang et al., 1992) so as to reuse parts of plans previously stored in the case base improving the efficiency of the
330 approach.

4. Our approach: *myPTutor*

We propose a complete approach for e-learning personalization by using AI case-based planning, and implement it as a prototype named *myPTutor*¹. *myPTutor* provides a mixed-initiative architecture that allows teachers and students
335 to work together during the learning cycle. Figure 1 shows the overall architecture used in *myPTutor* that encapsulates such a circle. The structure is relatively simple and consists of several modules that are executed sequentially, thus elaborating the workflow, and involving several technological issues numbered in Figure 1:

- 340 1. It uses standard LO repositories and modeling (authoring) tools.
2. It makes use of compilation techniques and algorithms for students' information acquisition to automatically create the PDDL planning domains.
3. It applies planning and CBP solving methods for generating the plans, i.e. the personalized learning routes.
- 345 4. It shows these learning routes on a LMS and monitors the students' progress when executing/navigating the course to detect discrepancies.

Next, we describe this four issues in more detail.

4.1. An (authoring) tool to define the course

Currently there are many LO repositories to create courses. But the LOs
350 in themselves, as isolated elements, are insufficient to generate meaningful instruction (Polsani, 2003), so it is important to define efficiently how LOs are related and for what pedagogical purposes. Several editors can be used for modeling and defining the course, from simple editors that provide pre-defined textual templates (e.g. RELOAD, <http://www.reload.ac.uk>, or eXeLearning,
355 <http://exelearning.org>, etc.) to graphical tools that allow drag&drop of visual components (Garrido et al., 2009), thus facilitating the course definition. In

¹More info in <http://servergrps.dsic.upv.es/myptutor>

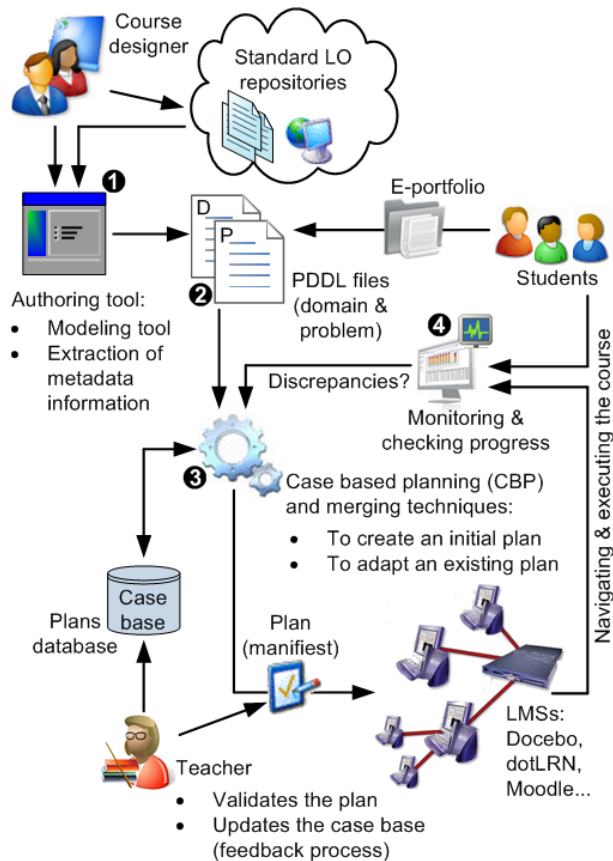


Figure 1: Architecture of *myPTutor*.

both cases, the teachers (playing the role of course designers) define the course by creating the LOs or reusing them from available repositories.

In particular, *myPTutor* uses an authoring graphical planning-oriented tool, further described in Garrido et al. (2009), which provides an easy definition of LOs, their relationships and profile adaptation. The aim of using a planning-oriented tool is twofold. First, to reuse and/or define classical LOs within a course. Second, to be able to extend these LOs with more flexible features that are very valuable in planning technology, such as hard/soft requirements in LOs, resource necessities (and their cost) in LOs, and metric to be optimized. Obviously, finding out these planning features is not always a straightforward

task and course designers sometimes fail in providing this information. If some information is missed we are still able to take advantage of planning technology. But clearly, the more features are defined in the LOs, the more flexibility has the planner to find better plans; and the initial effort pays off to enhance the
370 the ulterior learning process.

4.2. From e-learning to planning. A translator module for PDDL compilation

4.2.1. LO metadata information and its use in modeling courses

An instructional course is formed by a set of LOs which are interrelated
375 by means of causal dependencies. That is, the relationship between two LOs represents the precedence requirement between the two of them; in essence, one must be executed before the other in a cause-effect way. The relationships and the properties of the LOs rely on the metadata information defined in each LO.

Metadata specification for LOs is usually specified in an XML standard format, such as LOM (LOM, 2002). This specification has many useful entries for pedagogical theories, but there are only three that are key to support planning personalization, as seen in Figure 2. First, we need the technical platform requirements (number 4 in Figure 2), seen as the particular resources for the LOs. For instance, if multimedia or specific equipment is required for that LO. Second,
385 the educational information about the student's learning style, difficulty of the LO and its typical learning time (number 5 in Figure 2). This gives us an idea of the necessary time to deal with this LO for a student with the profile recommended for this LO. Third, the relations as the content dependencies which comprise hierarchical structures and orderings among LOs (number 7 in
390 Figure 2). The hierarchical structures use the *IsPartOf* relationship to represent a complex aggregation of LOs. On the other hand, there are also three types of ordering relationships to represent causal dependencies, which are based on the Dublin Core Metadata Initiative (<http://dublincore.org>): i) *Requires*, ii) *IsBasedOn*, and iii) *References*. They represent hard preconditions, both conjunctive and disjunctive requirements (*Requires* and *IsBasedOn*, respectively)
395 and soft preconditions as a recommended requirement (*References*).

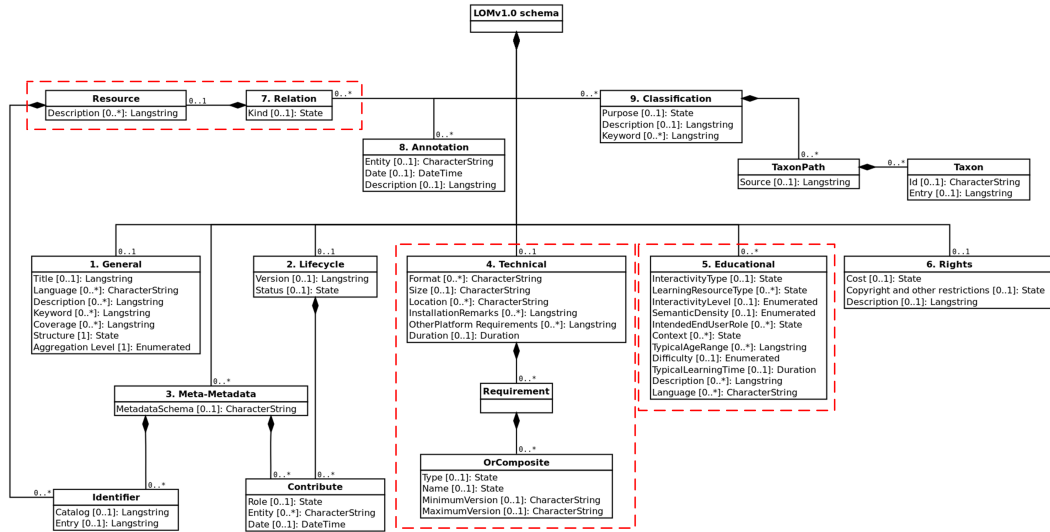


Figure 2: The three main elements required to support planning personalization (from http://en.wikipedia.org/wiki/Learning_object_metadata). Image used under the terms of the GNU Free Documentation License.

4.2.2. Compilation of the PDDL model

The implicit formalism behind PDDL is the separation of the domain data, which describes the general scenario, from the problem data, which describes the particular situation within such a scenario. Thus, a PDDL model requires two text files. First, the domain file that contains all the available actions for the planner to use. Second, a complete description of the initial state and the goals to be satisfied from that state.

myPTutor uses a knowledge engineering method, based on Garrido et al. (2009, 2013), to automatically extract the metadata information and compile the set of LOs as a PDDL domain. On the other hand, the students' characteristics and interests are compiled as the planning problem.

The PDDL domain. The compilation of the PDDL domain is a polynomial process that iterates all over the LOs and generates one action template (operator in PDDL terminology) per LO. A thorough description of this process is given in Garrido et al. (2009), but in short each operator consists of:

- A unique name taken from the LO name/identifier.
- One parameter, the student, to support the personalization and to make the definition of preconditions/effects more flexible.
- 415 • The duration, as the LO learning time.
- The preconditions, to filter the students' profile and to support the dependency relations. Other elements, such as the students' role, technical or educational requirements can be also modeled as preconditions.
- 420 • The effects, to represent the LO outcome. Other elements, such as optional expressions on rewards *w.r.t.* the student's profile, resource costs or additional metric values can be easily included.

An example of a PDDL action for a given LO, named `L0-example`, is as follows:

```

425 (:durative-action L0-example
:parameters (?s - student)
:duration (= ?duration (L0-example-learning-time ?s))
:condition (and
(at start (not (L0-example ?s done)))
(at start (L0-ex1 ?s done))
430 (or
(at start (L0-ex2 ?s done))
(at start (L0-ex3 ?s done))))
:effect (at end (L0-example ?s done)))

```

As can be seen, the structure of the action template for `L0-example`, where 435 the student `?s` is defined as the only parameter, is simple. The duration of `L0-example` is extracted from the learning time given by its metadata information. In this example, the preconditions include the fact of not having this LO done (to avoid repetitions), having `L0-ex1` previously done (*Requires* dependency) and `L0-ex2` or `L0-ex3` done (*IsBasedOn* dependency). The effect

440 achieved at the end of this LO is simply the fact of having **L0-example** done,
with the competence it involves.

The PDDL problem. The compilation of the PDDL problem models the initial state, the goals and, when necessary, the metric to optimize. It is created by extracting the relevant student's features from his/her e-portfolio. The initial
445 state represents the students' learning style, initial background and previous knowledge, and some other information (e.g. special equipment or resource availability). The goal represents the learning outcomes, which generally involve to pass the entire course or a part of it, usually in terms of a set of LOs.

An example of a PDDL problem, for a given student **std1**, is as follows:

```
450 (:init
    ;Profile and learning style
    (learning-style student1 theoretical)
    (performance std1 high)
    ;Academic trajectory on languages and availability (initial background)
455 (language-level spanish std1 high)
    (language-level english std1 medium)
    (availability std1 much))
(:goal (and
    (L0-ex1 std1 done)
460 (L0-ex3 std1 done)))
(:metric minimize (total-time)) ;shortest time plan
```

The structure of the problem is also simple. It includes the initial information for **std1** according to his/her particular profile, i.e. learning style and performance, the languages (s)he knows (the LO usually includes this information as
465 part of its educational metadata, as depicted in number 5 of Figure 2) and time availability. In this example, the learning outcomes require both **L0-ex1** and **L0-ex3** done. Finally, we define **total-time**, i.e. the plan makespan, as the criterion to be optimized by the planner.

4.3. Solving the planning problem via CBP

470 Analogously to Case-Based Reasoning (CBR) (Leake, 1996), our approach adopts two assumptions on the nature of the world: the *world is regular* and the problems the agents encounter *tend to recur*. Thus we assume that to similar problems correspond similar solutions, and that new problems are likely to be similar to problems already encountered.

475 Our Case-Based approach is built on top of the OAKPLAN system (Serina, 2010; Bonisoli et al., 2015), which computes an appropriate mapping between the students and the objects of the reuse candidate and the corresponding ones of the current instance. This mapping is computed in polynomial time using kernel functions in order to define an accurate mapping function.

480 The different e-learning routes that our system adopts can be generated in different ways; they can be imported by courses manually generated by teachers or can be generated automatically by our case-based planner (or by any other PDDL planner (Chen et al., 2006; Gerevini et al., 2003; Gerevini & Serina, 2010; Richter & Westphal, 2014; Vidal, 2014)), although in this case they have to be
485 validated by a teacher in order to be effectively used. Moreover, since we suppose to store in our library only high quality e-learning routes, we have adopted *plan merging techniques* (Yang et al., 1992) in order to have the possibility to also reuse subparts of stored e-learning routes. In particular, we decompose the solution plans into subparts and then we store these subparts in the plan library
490 (if they are not already present) (Garrido et al., 2012).

In short, when a new case base element $\langle \Pi', \pi' \rangle$ is inserted into the CB, we try to identify the subplans of every solution plan π' that allow us to satisfy every single goal, or a set of interrelated goals, and that can be inserted into the plan library to increase the *competence* of the library in itself (Gerevini
495 et al., 2013a,b; Smyth & McKenna, 2001; Tonidandel & Rillo, 2002). The sets of interacting goals can be easily computed considering the causal link relations and the actions in the subplans of π' that are in common with the different goals.

When a teacher requests an e-learning route for a new course (step 1 of

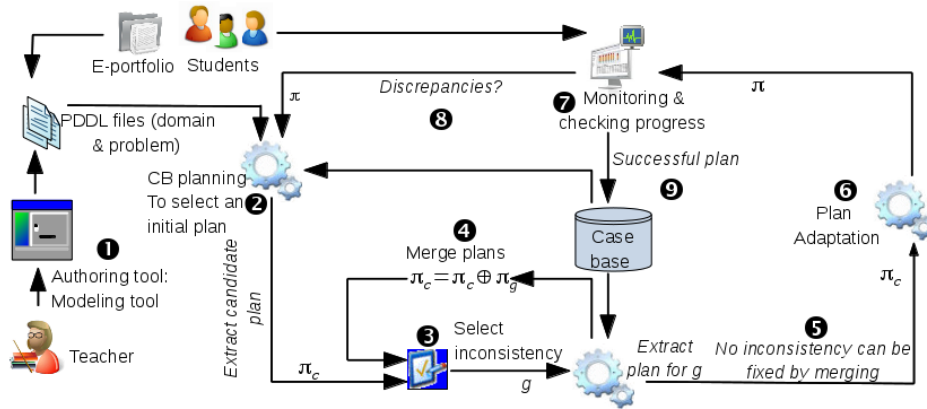


Figure 3: Case Base planning steps of *myPTutor* in more detail.

500 Figure 3), the system searches in the plan library a plan that already solves the current e-learning planning problem. Otherwise, the system tries to progressively identify (sub)plans in the plan library that can be used in order to reach the current goals. Intuitively, this phase consists in reusing parts of the retrieved plans to complete a new one. This way of proceeding allows teachers

505 to easily analyze and validate the suggested e-learning plan. In fact, they can avoid reconsidering the full proposed plan and simply examine the new parts of the plan that, for example, have been introduced for satisfying new goals of the students or changes in the resource availability.

More precisely, we first apply the standard retrieval phase of OAKPLAN

510 (Serina, 2010) in order to find the best plan π that satisfies the current goals with a heuristic adaptation cost significantly lower than the corresponding generation cost (step 2 of Figure 3). Next, we progressively analyze the unsatisfied goals

of the current problem Π (step 3 of Figure 3), trying to identify in the library a subplan π_g that can be merged with the current partial plan π (step 4 of Figure 3) in order to satisfy the selected goal g (and other unsatisfied goals if possible) and reduce, at the same time, the global heuristic adaptation cost². The process stops when all the goals and preconditions are satisfied or when there is not a suitable plan that can be extracted from the library that satisfies the remaining unsupported goals (step 5 of Figure 3). Obviously, if the resulting plan π still contains discrepancies, it does not represent a solution plan; however, it can be used as a starting point for the *Plan Adaptation* phase in order to find a solution plan for the current planning problem (step 6 of Figure 3). Thus, the goal of the plan adaptation phase is to reuse and modify an existing input plan in order to solve a new problem avoiding to repeat the whole search process but concentrating only on discrepancies in the input plan.

The goal of the *Plan Revision* phase is to solve failures that can happen during the e-learning route execution (step 7 of Figure 3). In the e-learning context, a failure is anything that prevents one student from achieving his/her goals. There are many examples for this, such as a particular LO the student cannot achieve (eg. a test that is failed); when the student takes too long to finish a LO and, consequently, fails to start/complete the next LO; or when a temporal constraint appears and the new agenda is incompatible with the planned LOs. The system analyzes the plan execution trace in order to identify failures that might prevent the correct execution of the remaining part of the plan. In such a case, the system can react repairing the failures or aborting the plan and restarting the whole CB process (step 8 of Figure 3). When the plan execution ends, the corresponding plan can be saved in the library or abandoned (step 9 of Figure 3).

Regarding the *Plan Storage* phase, given a plan (or a subplan) π^* and the corresponding goal set, we identify the goals that are required for its execution defining in this way a new possible case base element c^* ; then we search in the

²In our tests we have considered the earliest and the latest part of π where g can be satisfied

case base if a case-base element c_j exists whose goals and initial state perfectly match with the current goals and initial state, respectively. In this case, we have to decide whether to insert c^* in the case base and remove c_j , or simply skip the
545 insertion of c^* . In our tests we have used an update policy that maintains the plan with the lowest number of actions, but other policies could be used as well considering, for example, the plan qualities, their makespan, or the robustness to exogenous events.

It is important to point out that the definition of an appropriate e-learning
550 route for a set of students depends on different criteria, such as pedagogical issues, teacher’s preferences, students’ learning styles, etc. In general, supposing that our plan library contains good quality cases that have been accurately generated and validated, we are not only interested in e-learning routes that best fit the students’ requirements but also in trying to generate routes that
555 are structurally similar to one or more elements of the case base. This concept corresponds to the notion of *plan stability* in the AI planning context (Fox et al., 2006; Nguyen et al., 2012), which measures the differences a process induces between an original plan and a new plan. As previously exposed, it is assumed that problems tend to recur; this means that with high probability
560 the new problem could be structurally similar, although not identical, to one already solved or to a subpart of it. Moreover, in our context it is fundamental to preserve the stability of the e-learning route of each student maintaining the LOs planned and their relative order. In particular, it could be extremely frustrating for a student obtaining a completely different route given a slightly
565 change in the current state or in the goals, and this should be obviously avoided especially when the original LOs are still valid.

Hence, we have adopted a simple although effective notion of plan *distance* (Fox et al., 2006), which is defined in terms of the number of different actions (LOs in our context) between two different plans, i.e. the number of LOs that
570 appear in the first e-learning route but are not included in the subsequent, plus the LOs that appear in the second plan but are not found in the first one. Supposing that the current plan is no longer valid as a consequence of a change,

the notion of stability (Fox et al., 2006; Nguyen et al., 2012) is defined as the distance between the actual solution plan and the original one.

575 4.4. From planning to e-learning and its execution within a LMS

Once the planner has generated a plan, i.e. a learning route per student, the sequence of LOs is transformed into a learning manifest, which can be shown and executed in any LMS that integrates a SCORM player. Consequently, the LMS identifies, for each student, the instructional design that must be followed by
580 the student under IMS-CP or SCORM specifications. The LMS is fundamental not only to provide the students a sequence of LOs but also to visualize the learning plan to the teachers, who can manually modify the LOs proposed by the planner and, finally, to validate the instructional design the students have to perform.

585 The main objective of the LMS is to provide the navigation mechanisms within the sequence of LOs of the course to each student. In addition to this, it is also valid to monitor the progress of the student throughout the course, as well as to detect serious discrepancies between the current situation of its learning sequence and the situation that was originally planned to be executed so far.
590 These discrepancies appear due to many factors, such as a change in the profile information, the hardware or software resource availability, or the expected execution of the LOs. These situations are automatically recovered from the LMS database (which includes the student's e-portfolio), as it is immediately updated after an evaluation activity is finished and/or graded.

595 5. Integration with Moodle. Putting all together

myPTutor is general enough to be compatible with any LMS and to support any PDDL planner. As a proof of concept and to study its viability, we have implemented it on top of *Moodle* (<http://moodle.org>). Although the technical details for the integration with *Moodle* are beyond the scope of this paper,
600 we have implemented several extensions to allow a mixed-initiative mechanism

between users (students and teachers), the database, GUIs and the planning services. These extensions allow next, both to students and teachers, to close the cycle presented in Figure 1, which means that the plan progress is monitored and, in case of discrepancies, the plan is adapted to the new scenario.

605 *5.1. Extending Moodle to enhance personalized e-learning*

Moodle is a complex platform with many capabilities and utilities but, unfortunately, it was not originally designed to support the kind of e-learning personalization that *myPTutor* provides, which limits its direct applicability. Therefore, we have had to tune some of its modules to solve these limitations.

610 First, we have defined new tables in the relational database to support planning preconditions, the students' background (i.e. the initial state), the goals that are compulsory for each course and those that can be optionally chosen by each student. Second, new forms have been added to the teacher's GUI to define the course goals (compulsory and optional) and the initial profile required
615 for the students (see Figure 4-1). Similar forms have been designed for students to input their profile information and desired optional goals. Once this information has been input into the system, it automatically invokes the planning Web service to calculate a personalized learning route per student, which can be generated from scratch or by reusing previous cases from the CBP library
620 (Figure 4-2).

Finally, other forms are designed to execute and monitor the state of the learning route. *myPTutor* also outputs the learning route as a Gantt chart in Microsoft Project format (see Figure 5). From the teacher's point of view, (s)he can observe this chart of expected LOs and manually compare it with the *already*
625 *done* sequence of LOs. From the student's point of view, this chart shows the schedule of the next LOs to be executed. Again, if during the student execution of the learning route the system detects discrepancies between the expected and real state, a message is displayed to the student, as depicted in Figure 4-2. This message indicates the remaining LOs that cannot be executed from the current
630 state, and the process stops until the system performs an adaptation of the

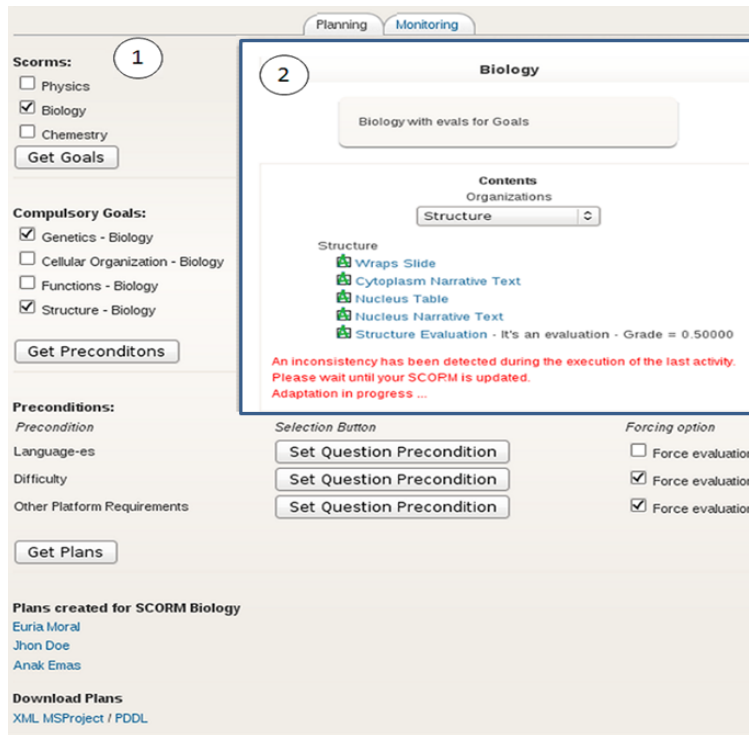


Figure 4: 1): module that allows teachers to select the course, its goals, background pre-conditions and adapt SCORMs to different students' profiles through planning services. 2): monitoring during the student SCORM execution. Note that the GUI is still very simple because we are more interested in validating our approach than in having a ready-to-deploy tool.

route. This process continues until the learning route is completed and all goals are achieved.

5.2. Closing the e-learning cycle

As Figure 1 depicts, steps 1–3 perform the generation of a personalized learning route. But once this route (i.e. plan) has been generated it needs to be uploaded to the LMS (*Moodle* in our case) to be used by both students and teachers (step 4) as part of a learning manifest. The LMS identifies, for each student, the instructional design that must be followed by the students under the IMS-CP or SCORM specifications. This allows students to navigate and

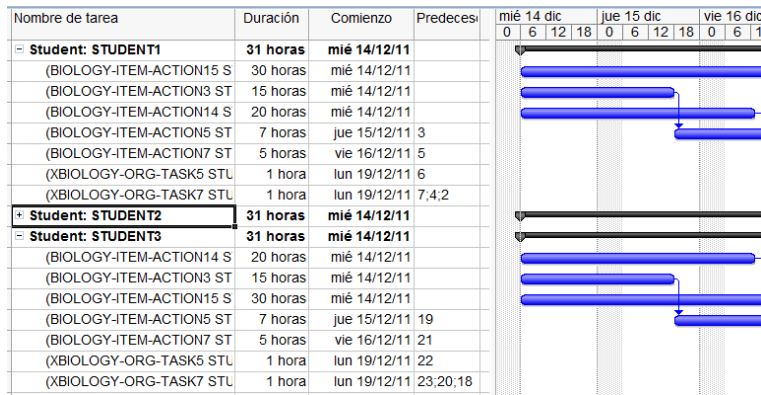


Figure 5: Gantt chart that shows the learning routes of three students.

640 execute the course.

Moodle does not only offer the students the sequence of LOs to use, but it also provides a simple and efficient way to monitor the progress of each student in his/her learning route. In other words, *Moodle* is very useful for navigation matters of LOs, giving information on the structure and calendar of LOs. But 645 it also keeps trace of the students' progress, i.e. the LOs that a student takes. Also, when some tests or questionnaires are applied to evaluate the students' comprehension of the objectives, it automatically stores their performance in the database. This is an extremely good opportunity to monitor the progress of the students, to check the plan execution and detect discrepancies.

650 Discrepancies appear during execution due to many factors (Morales et al., 2011), such as changes on the background/profile information, or the execution of the LOs in themselves. For instance, on the one hand, the students' profile can be easily modified in *Moodle*: incrementing/decrementing their performance levels according to previous evaluation tasks, improvement in their 655 foreign language levels, acquisition of new resources (e.g. multimedia equipment or software), etc. On the other hand, differences between the expected state (obtained by a flawless execution of the generated plan) and the real state (resulting from the current execution of the plan) may arise, such as when the student does not pass a critical evaluation task, or when a LO with dependen-

660 cies takes more time than initially scheduled. These situations, and others that
can be defined in the future can be modified by the students, by the teachers or
directly updated by *Moodle*. In any case, this information can be automatically
recovered from the LMS database.

When discrepancies are detected, the system executes the remaining part of
665 the learning plan to identify if it contains flaws. If a flaw is found, it is highlighted
to the teacher to be repaired manually or to invoke a planner to find a new plan
to fix the flaw. If it is no a flaw found, it is provided a new schedule of the LOs
that have not been achieved or provided yet.

The plan adaptation stage consists in reusing and modifying previously gen-
670 erated plans to solve a new problem and overcome the limitations of planning
from scratch. But *myPTutor* uses the same step 3 of Figure 1 for this, in which
the entire part of the plan is provided to the CBP, thus increasing the possibili-
ties to fix the flaws with minimal modifications. The core idea here is to preserve
the learning sequence and its components initially proposed to each single stu-
675 dent as much as possible. In fact, it makes no sense to propose a sequence of LOs
completely different to the previous ones and this should be avoided, because
in many cases most of the LOs that have been already executed can be reused.
After finding the plan in the library and repairing it with the plan adaptation
techniques discussed in Section 4.3, the solution plan can be inserted into the
680 library or discarded. When the plan execution finishes successfully, the plan is
stored in the case base, closing the learning cycle.

6. Evaluation and experimental results

In this section we report the experimental results obtained by conducting
both a qualitative and a quantitative evaluation of the techniques proposed to
685 evaluate our approach. For this, we have experimented with several courses
defined in *Moodle*, but in this paper we focus just on four courses:

- one medium-size *Course1* with 35 LOs on Discrete Maths used in Castillo
et al. (2010);

- one real, large-size *Course2* with 90 LOs on Natural Sciences based on <http://www.profesorenlinea.cl>;
- two real, large-size courses: *Course3* with 88 LOs on Metaheuristics, and *Course4* with 62 LOs on Structured Programming. Both courses are part of the syllabus for the Bachelor's degree in Computer Engineering of the Universidad Tecnologica de la Mixteca (Mexico), <http://www.utm.mx/~computacion/tabla2008.html>.

6.1. Qualitative evaluation

For this evaluation we have used the Structured Programming course (*Course4*). All the LOs were defined by a professor with experience in this topic and following the course' guidelines defined in the syllabus. We also considered three *classes* of students with different profiles; in particular, different learning styles, different temporal requirements, as well as different knowledge on the topic and foreign languages.

The evaluation was calculated according to several opinion questionnaires, which were given to a group of participants (i.e. ten professors that are giving or have given this course, and ten students that are currently enrolled in the course), to assess different aspects such as:

- *Adequacy* of the learning routes to the student's profiles.
- *Size* and *duration* of the route *w.r.t.* the students' profiles and their previous experience in the course topics.
- The *contents* and *structure* of the learning routes, in order to maintain the congruence with other LOs of the course.

The evaluation was divided into two tests: Test 1 showing the participants the original learning route for each student's profile, and Test 2 making significant changes in the profiles and showing professors and students the new adapted learning routes.

The learning routes for Test 1 were provided by each planner (OAKPLAN, LPG and SGPLAN6³) and we took the best “quality” plan, in terms of *reward*. Reward is a measure that increases when a LO that best fits with the student’s learning style is selected to be included in his/her route, based on the weights
720 proposed in Baldiris et al. (2007). In particular, the planner that provided the best quality plans in this test was OAKPLAN. For Test 2, we used a new optimization metric, which measures equally the reward and stability, by using OAKPLAN and OAKPLAN-merge for replanning (OAKPLAN-merge (Garrido et al., 2012) is our case-based approach with merging techniques, which has been
725 implemented on top of OAKPLAN (Serina, 2010)). In this test, the planners that provided the best results were OAKPLAN-merge and OAKPLAN, with the best learning routes in two classes and one class of students, respectively.

After generating the learning routes, and for the qualitative evaluation, we showed the learning routes to our participants by using a Gantt diagram that
730 included information about each LO, such as duration, activity type, difficulty, language and hardware/software requirements. The results of the questionnaires for both tests are given in Tables 1 and 2, respectively. In Table 1 we can find that the length in terms of LOs, duration and adaptation were considered good for both professors and students in most of the 50% of the cases. On the
735 other hand, adaptation was better considered by students than by professors, as observed in Table 2. Also students thought the main problem of the original learning route was the length, whereas professors thought it was more related to the difficulty and limitations in resources and students’ performance.

All in all, professors think that before using a learning route (both the origi-
740 nal and, primarily, the adapted one when necessary) it has to be, at least quickly, checked to detect weaknesses that jeopardize students’ success with the course. Also, most of the disadvantages found are related to the lack of variety in the LOs for very different learning styles. Thus, creating more and better LOs is essential. Finally, professors agree that giving just one duration to each LO is

³For a further description of these planners see <http://ipc.icaps-conference.org>

Questions	Professors		Students	
	Yes	No	Yes	No
Are the number of LO's appropriate for the course?	7	3	6	4
Is the duration of the sequence appropriate?	5	5	7	3
Is the adaptation of the contents to the student's learning style appropriate?	8	2	8	2

Table 1: Qualitative evaluation for the initial learning route for *Course4* (Test 1).

Questions	Professors			Students		
	Yes	No		Yes	No	
<i>Is the new sequence similar to the first one?</i>	5	5		7	3	
<i>Is it well adapted to the changes in the student's profile?</i>	8	2		9	1	
<i>What was the main reason for failing executing the original course? Eg. difficulty of the course, length, time constraints, learning outcomes that are difficult to achieve, others.</i>	Difficulty of the course.		Length	Not enough resources or students' capabilities		
<i>Do you consider the new sequence better, equal or worse than the original one?</i>	<i>Better</i>	<i>Equal</i>	<i>Worse</i>	<i>Better</i>	<i>Equal</i>	<i>Worse</i>
	3	7	0	3	7	0

Table 2: Qualitative evaluation after making changes in the profiles, thus having to adapt the original learning route used in Table 1 (Test 2).

745 a very imprecise task, and defining a min-max interval for the learning time for each LO is more recommendable, while it also helps find better quality and more flexible learning routes. However, this means that a new way to represent this interval in state-of-the-art planners needs to be considered.

6.2. Quantitative evaluation

750 This evaluation was performed in order to analyze the performance and scalability of the techniques proposed at the increase of the number of students involved and the number of discrepancies *w.r.t.* the original learning route. The compilation of a standard PDDL model facilitates the use of any modern planner and, therefore, we can run many computational experiments. In particular, we tested the effectiveness of our case-based approach with merging techniques (OAKPLAN-merge), *vs.* plan generation techniques when discrepancies appear

within execution. We focus on: i) the scalability by means of the CPU time needed to obtain (retrieve&adapt or replan) the sequences, ii) the best solutions as regards plan quality with a better reward, and iii) the best stability obtainable
760 within a fixed deadline.

Regarding *Course1*, we have simulated four configurations (using 20, 40, 60 and 80 groups of students, respectively), and also simulated 10 variants or executions per group, then we have defined 44 planning problems in total. For *Course2*, we have simulated nine initial profile' configurations (with 10, 20...90
765 fictitious students, respectively), and defined 10 variants per configuration (plus one additional variant for the 90-th problem), then we have defined 100 planning problems (the 9 initial configurations plus the 91 variants). When we say variants it means artificial changes of the student profile and/or simulations of the LOs execution that can be positive or negative according to the performance
770 of the student and that may happen in an incremental way, i.e. by making some equipment no longer available, including new restrictions, etc.

In *Course3* and *Course4*, we have tested a random variants creation considering up to 100 students. In fact, we have simulated ten initial student profiles (given 10, 20...100 fictitious students, respectively), and generated 9 variants
775 for each configuration, using in this way 100 learning routes. The variants are generated by randomly modifying the initial facts of the original problem. Moreover, the number of modifications is increased systematically considering the number of students: we allowed 4 modifications per student, applying thus 40 modifications to the initial state of planning problems with 10 students and
780 400 modifications of planning problems with 100 students.

In addition to OAKPLAN and our case base planner OAKPLAN-merge, we have made use of two state-of-the-art planning systems, LPG and SGPLAN6. Tests were performed on an Intel(R) Xeon(TM) CPU E5-2620 2.00GHz with 10GB of RAM, and censored after 30 minutes. Since LPG and OAKPLAN use a
785 local search approach, the results correspond to the median value over five runs. In our experiments, the solution plans (i.e. the learning routes) inserted in the case base were calculated using the best quality plans obtained by SGPLAN6,

LPG or OAKPLAN itself on the planning problems of the initial-configuration used to generate the relative variants.

790 Figure 6 illustrates the results of the different planners in order to solve our benchmark domains. On the left we compare OAKPLAN (with and without merging techniques) *vs.* LPG and SGPLAN6 using a “complete” case-base that contains all the base problems and the corresponding solutions (the case-base for the merging variants contains also the selected subplans of the base problems).
795 Here we can observe the general good behavior of the case-based techniques, which are comparable in terms of CPU-time to SGPLAN6. The results show that the case-based approach is in the worst case not slower than replanning, and sometimes faster. In addition, plan retrieval techniques do not prove so useful in case of significant changes and when establishing the route requires a
800 higher amount of computational resources than regenerating a new one. But, as Table 4 shows, the advantages in making this effort pay off as regards plan stability.

In Figure 6-right, we analyze the behavior of OAKPLAN and OAKPLAN-merge to study the impact of using a case base considering: i) a case base
805 created using only the smallest base problem (with 10 students), ii) a case base (“Incremental” lines) where the base problems are progressively inserted *after* the corresponding variants have been evaluated (it initially contains only the smallest base problem). In the first case, we primarily want to evaluate the ability of the merging techniques to reuse the solutions available in the case
810 base at the increase of the “differences” (in terms of number of students) among the current situation and the elements stored in the case base. In particular, we want to examine the scalability in terms of number of students, which is extremely important in our context since a teacher could decide to evaluate the effectiveness of an e-learning course considering a limited number of students
815 before using it for the whole class. Considering, for instance, the tests with the *small* case base and the *Course2*, we can observe the general good behavior of OAKPLAN-merge, while OAKPLAN *without* merging techniques is able to solve only 55 variants.

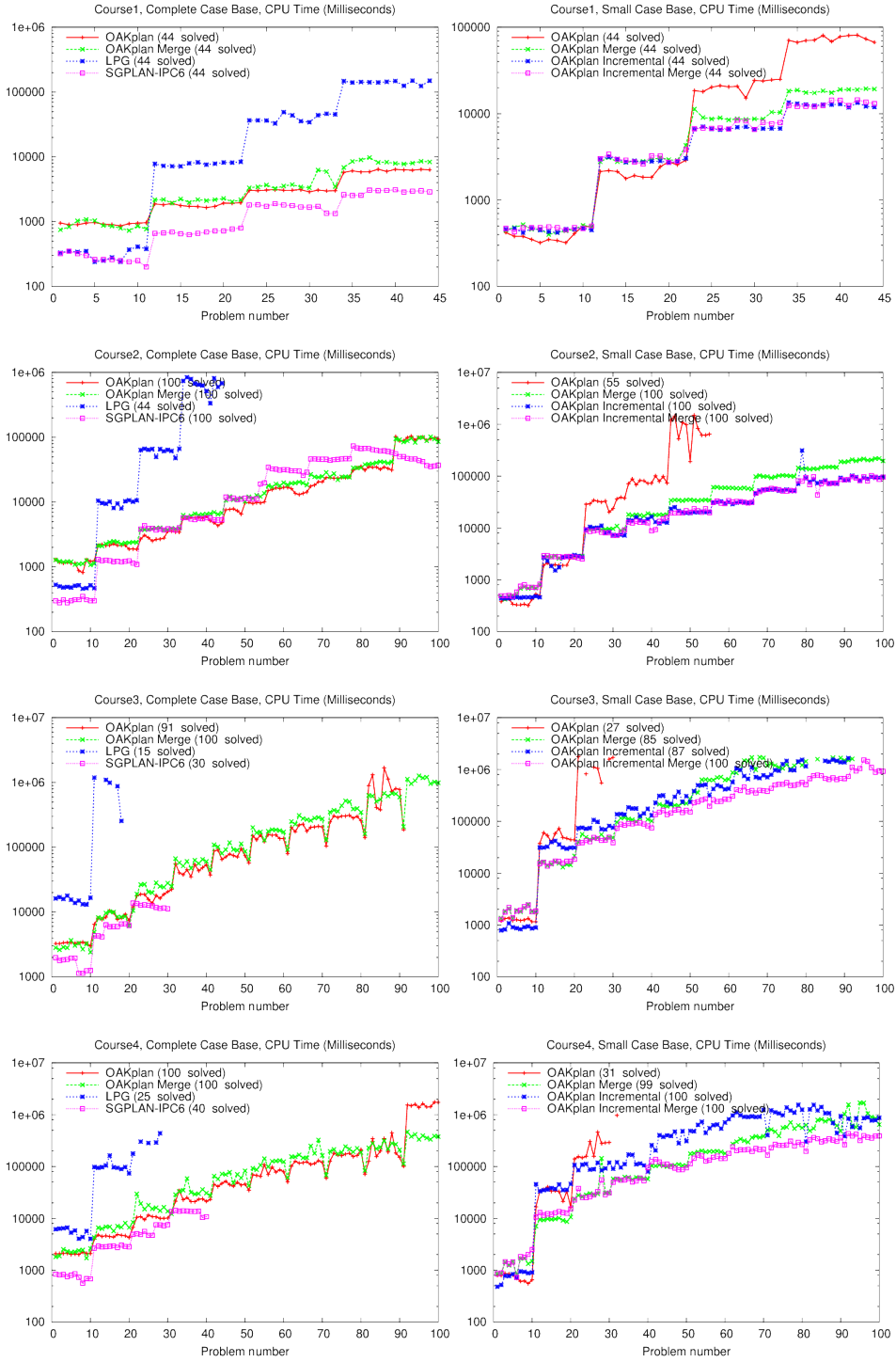


Figure 6: CPU time in milliseconds (on a logarithmic scale) required by the different planners to solve our benchmark domains. In the plots on the left we compare the case-based approaches (OAKPLAN and OAKPLAN-merge) *vs.* replanning (LPG and SGPLAN6). On the right, we compare the case-based approaches considering different input case bases.

Regarding the tests with the *incremental* case base, we want to analyze the
820 behavior of OAKPLAN considering a case base that contains elements which are
structurally not too much different *w.r.t.* the current situation. For example,
considering the solution of the variants with 50 students, the case base does not
contain the base problem with 50 students but it contains the base problems
with 10, 20, 30 and 40 students. Here we want to examine the situation where
825 a teacher has already used a course in different classes and wants to reuse the
stored experiences in a new (slightly bigger) class. As expected, the behavior
of OAKPLAN-merge does not change significantly. On the contrary, the CPU-
time of OAKPLAN *without* merging techniques decreases significantly since it
can replan starting from a case base element with a slightly lower number of
830 students *w.r.t.* the current situation. Moreover, it is now able to solve all the
variants considered.

Table 3 summarizes the experimental results for the different planners con-
sidered in terms of percentage of solved problems, average total CPU-time in
seconds and plan quality. Note that we want to maximize the total reward
835 of the students and so the higher the quality, the better the plan is. Here
we can observe the general good behavior of the case-based techniques, which
is particularly evident considering the IPC scores reported between brackets.
Briefly, each planner receives a score between 0 and 1 for each problem solved.
The score is the ratio between the time/quality/distance⁴ value of the solu-
840 tion found and the time/quality/distance value of the best solution found by
any other planner in the table. The score is summed across all problems for
a given planner; the higher the score, the better the planner is. In particular,
OAKPLAN is the fastest, followed by OAKPLAN-merge and SGPLAN6. Note
that the retrieval and adaptation process sometimes has an impact in the qual-
845 ity: the route needs to be modified to be suitable for a new configuration instead
of being constructed from scratch. Our tests demonstrate that the quality for
the case-based approach may obtain better results than replanning, especially

⁴Distance values are considered in Table 4

Planner/Courses	% Sol.	Time (score)	Quality (score)
OAKplan			
Course1	100.0 %	2.97 (31.31)	34284.773 (43.00)
Course2	100.0 %	21.49 (90.14)	72157.200 (99.00)
Course3	91.0 %	174.42 (84.02)	130806.538 (90.00)
Course4	100.0 %	205.47 (87.14)	114094.930 (99.00)
Total	97.3 %	159.59 (294.75)	93131.189 (334.00)
OAKplan-merge			
Course1	100.0 %	3.81 (29.69)	34284.773 (43.00)
Course2	100.0 %	22.92 (84.66)	72189.000 (99.00)
Course3	100.0 %	254.83 (88.55)	142099.650 (99.00)
Course4	100.0 %	118.62 (82.25)	114051.130 (99.00)
Total	100.0 %	115.71 (287.38)	99832.872 (343.00)
LPG-td			
Course1	100.0 %	46.80 (23.15)	34253.409 (43.00)
Course2	39.0 %	158.66 (21.47)	31163.974 (38.00)
Course3	12.0 %	368.02 (4.95)	29485.167 (11.00)
Course4	21.0 %	161.40 (8.75)	33148.476 (20.00)
Total	33.7 %	1217.05 (60.12)	10965.863 (115.00)
SGPlan			
Course1	100.0 %	1.38 (43.00)	34205.682 (43.00)
Course2	100.0 %	23.43 (88.50)	69883.500 (99.00)
Course3	30.0 %	6.55 (28.89)	47385.333 (29.00)
Course4	40.0 %	5.69 (38.95)	51018.875 (39.00)
Total	62.2 %	688.45 (202.34)	34754.608 (213.00)

Table 3: Percentage of problems solved, average CPU-time (seconds) and average Quality of OAKPLAN, OAKPLAN-merge, SGPLAN6 and LPG. The corresponding IPC scores are reported between round brackets.

in the most difficult instances. Furthermore, it is important to point out that the plan quality obtained by LPG and SGPLAN6 are penalized by the low percentage solution rate. In fact, easy problems are characterized by low values of plan quality. Similarly, the solution times reported in SGPLAN6 for *Course3* and *Course4* are related to the solved problems, while in the definition of the “Total” (average) CPU-time we have associated the time limit (1800 seconds) to the unsolved problems.

Table 4 summarizes the experimental results for the different OAKPLAN configurations considered in terms of percentage of solved problems, average total CPU-time in seconds, matching time in seconds, plan quality, plan distance and

Planner/Courses	Sol. %	Time (Score)	Match. Time	Quality (Score)	Dist. (Score)	Stab.
OAKplan						
Course1	100.0 %	2.97 (39.6)	2.066	34285 (43.13)	9.9(32.8)	0.995
Course2	100.0 %	21.49 (93.0)	8.181	72157 (99.38)	54.6(82.4)	0.989
Course3	91.0 %	174.42 (85.4)	23.706	130807 (92.23)	233.9(34.8)	0.958
Course4	100.0 %	205.47 (91.9)	16.964	114095 (100.59)	251.1(36.7)	0.949
Total	97.3 %	159.59 (311.8)	14.217	93131 (338.33)	152.0 (189.7)	0.969
OAKplan-merge						
Course1	100.0 %	3.81 (36.9)	2.629	34285 (43.13)	2.6(43.0)	0.998
Course2	100.0 %	22.92 (87.5)	9.736	72189 (99.35)	12.5(87.3)	0.997
Course3	100.0 %	254.83 (89.4)	76.556	142100 (101.37)	222.1(46.0)	0.963
Course4	100.0 %	118.62 (85.7)	43.565	114051 (100.61)	208.8(37.5)	0.956
Total	100.0 %	115.71 (301.4)	38.085	99833 (347.47)	129.2 (216.9)	0.975
OAKplan-small						
Course1	100.0 %	24.25 (30.8)	0.569	34387 (43.01)	454.9(16.0)	0.641
Course2	55.0 %	198.33 (33.5)	0.595	41534 (56.86)	649.9(30.4)	0.582
Course3	27.0 %	483.77 (15.9)	0.821	51090 (26.17)	497.2(10.8)	0.730
Course4	31.0 %	170.68 (18.5)	0.566	45258 (30.71)	506.0(10.0)	0.681
Total	45.6 %	1050.43 (101.3)	0.621	19127 (159.75)	246.2 (70.3)	0.644
OAKplan-small-merge						
Course1	100.0 %	7.80 (32.5)	2.092	34387 (43.01)	32.7(18.5)	0.982
Course2	100.0 %	64.83 (66.8)	5.168	69268 (103.84)	5.6(81.6)	0.998
Course3	85.0 %	594.14 (56.6)	38.081	127151 (85.30)	47.7(81.6)	0.986
Course4	99.0 %	316.30 (68.4)	34.349	108080 (104.38)	50.4(94.6)	0.983
Total	95.3 %	323.07 (226.8)	22.092	87057 (339.52)	32.1 (279.3)	0.988
OKplan-incr						
Course1	100.0 %	3.46 (40.6)	0.844	34332 (43.06)	230.8(16.8)	0.826
Course2	100.0 %	50.72 (82.8)	5.253	70814 (101.89)	356.6(52.9)	0.864
Course3	68.0 %	346.55 (44.9)	7.268	104002 (67.70)	601.3(18.6)	0.822
Course4	100.0 %	290.93 (67.0)	8.645	115920 (99.79)	577.0(22.6)	0.842
Total	90.6 %	323.03 (238.4)	6.158	79233 (315.43)	419.7 (113.9)	0.842
OAKplan-incr-merge						
Course1	100.0 %	5.96 (33.8)	2.225	34330 (43.06)	16.5(20.5)	0.989
Course2	100.0 %	33.47 (75.7)	9.921	71189 (101.46)	22.7(66.8)	0.993
Course3	100.0 %	320.24 (78.5)	94.853	142966 (100.58)	215.0(52.5)	0.964
Course4	100.0 %	150.94 (76.1)	57.077	115254 (100.48)	186.7(42.1)	0.961
Total	100.0 %	147.46 (266.7)	47.334	100150 (348.59)	125.5 (185.0)	0.975

Table 4: Percentage of problems solved, average CPU-time (seconds), average matching Time, average Quality, average Distance and average Stability of OAKPLAN, OAKPLAN-merge, OAKPLAN-small, OAKPLAN-small-merge, OAKPLAN-incr, OAKPLAN-incr-merge. The corresponding IPC scores are reported between round brackets.

plan stability. In particular, OAKPLAN is the fastest, followed by OAKPLAN-merge and OAKPLAN-incr-merge. Note that the matching time of the “merge”
860 versions is significantly higher than the others since they have also to compute the matching with the subplans; this determines higher CPU times for the retrieval phase but it allows us to significantly improve the percentage of solutions obtained, the Distance Score and the plan stability. Standard OAKPLAN-incr-merge produces the best quality plans, strictly followed by OAKPLAN-merge,
865 but OAKPLAN-small-merge and OAKPLAN also perform extremely well.

Regarding the plan distance and plan stability, we can observe the good behavior of OAKPLAN-small-merge with values extremely different *w.r.t.* the other planners. This is related to the merging of small subplans in order to obtain the complete solution plan. Anyway, we can observe the general good
870 behavior in terms of plan distance and plan stability of the “merge” versions *w.r.t.* the others. Their distance values are obtained considering the amount of different actions as regards the matching plan provided by the retrieval phase, which is not necessarily obtained directly by a single solution plan stored in the case base (as in OAKPLAN), but also using the different subplans (highlighted to
875 the teachers) obtained by the analysis of the case-based elements that best fit the current goals and initial state. This is highly appreciated in an e-learning setting because both students and teachers prefer a kind of *inertia* in the (already known) learning routes.

Regarding the plan distances produced by OAKPLAN *without* merging techniques
880 *w.r.t.* the solution of the base problems we can observe values which are similar to the ones obtained for replanning. This is not surprising since the elements stored in the small and in the incremental case bases are very different *w.r.t.* the new planning problems. On the contrary, the performance of OAKPLAN-merge is extremely good, both in the small and incremental case
885 bases. However, it is important to point out that in this case the comparison in terms of plan distances are performed considering directly the plan provided by the retrieval phase. It is up to the teacher to decide if (s)he wants to validate elements that only deal with previously executed courses or also subparts of

them.

890 Globally, we can observe that the use of plan merging techniques is potentially very effective in our context, allowing us to obtain efficiently new e-learning routes which are very similar to the combination of previously executed ones (or subparts of them). This is extremely important since it allows to highlight the teachers the changes *w.r.t.* already executed learning routes, facilitating the
895 validation process. Furthermore, the stored plans can also contain some notes regarding, for example, the pedagogical motivations associated to the selection or combination of specific LOs, annotated by the teacher during the creation of the original learning route, or during previous executions of the learning route. In this case, these notes can be easily re-examined by the teachers facilitating
900 the learning route validation process.

7. Discussion and lessons learnt

The key lesson we have learnt is that planning technology has demonstrated very helpful for e-learning personalization, but it must be introduced transparently to the user. Obviously, teachers and students do not want to deal with
905 PDDL constructs. On the contrary, they want to continue using their courses, LOs and LMSs with no changes. And here an automated knowledge engineering compilation based on LOs' metadata is very useful. But this metadata definition is sometimes tricky. In our qualitative evaluation, professors pointed out that this is one of the main drawbacks when defining courses by combining LOs, as
910 they are not always fully specified and varied enough. Consequently, this is still a challenging issue: to find LO repositories designed to be interoperable, thus reducing the effort necessary to establish relations with other LOs and create big courses.

Personalization of contents in e-learning has been addressed by other ap-
915 proaches, but they have some limitations *w.r.t.* finding the best sequence of contents. On the contrary, the generation of a learning route is one of the main strengths of *myPTutor*. But this is not the only one. The fact of using

AI planning techniques allows us to find an adequate sequence of contents in an automated way. We automatically compile a PDDL planning model from the LOs' metadata. This standard model makes our PDDL planning approach solver-independent, which is a good property in any approach. In particular, any PDDL-compliant planner can be used to solve the model and generate a fully-tailored learning route per student. An additional advantage of our PDDL planning approach *vs.* (collaborative) recommender approaches is that it does not need to deal with previous information about other students to create the route. In other words, we just deal with the information of the current student.

Another strong point that we consider very useful here is the application of case-based planning for adaptation when problems appear during execution. Our CBP approach not only recommends personalized learning contents that remain static during the course, but it also considers previous recommendations to similar students profiles that can be used when some of the profile settings change during the learning route execution, or when a learning objective has not been properly reached, making it necessary to include additional LOs to the route. Loosely speaking, we use previous cases for repairing, while also trying to improve the quality of the original route if possible. This resembles collaborative recommenders, in terms of reusing recommendations of previous similar students. Although this resemblance is very interesting and shows very effective, it requires, first, to acquire information about past students and, second, to store and process CBP repositories that may become large and expensive. And these may involve some limitations.

Our approach still has some limitations. First, the PDDL planning approach cannot make use of what has been learnt from previous students because we want to be able to use any planner. "*What is good for past students may also be good for a new similar student*" is an idea that we can only use in the CBP approach, when we need to adapt the plan to new requirements and keep a stable solution. This means our PDDL planning model does not take advantage of social network scenarios (Comi et al., 2015a). On the contrary, the CBP approach is effective when a significant number of learning sessions have been

performed. In this sense, these two approaches are somewhat complementary:
950 the PDDL planning approach is particularly useful at the early stages of an
e-learning course, whereas the CBP approach is more indicated after a number
of learning sessions.

Second, the integration in *Moodle* is not straightforward and requires a sig-
nificant amount of extra coding and changes to the database of *Moodle*. This
955 makes it difficult to use our approach as a plug&play module. This also prevents
our approach from offering easy personalization of the display, as we depend on
Moodle's visualization capabilities. However, this can also be seen as an advan-
tage because we will take advantage of the new developments of *Moodle* at low
cost.

960 Finally, our approach does not consider the use of an ontology by different
teachers. In fact, the use of an ontology could significantly increase the reusabil-
ity of different LOs, as proposed in Comi et al. (2015b). This would allow us to
derive a global representation from personal ontologies of different agents, tak-
ing advantage of the multi-agent paradigm. This way, each agent might enrich
965 its own ontology by using semantic negotiation.

As a final lesson, personalization of e-learning routes is important for both
educational and enterprise organizations, as it supports a lifelong learning pro-
cess. But based on our experience, students seem more enthusiastic about
personalization than teachers, who still believe in e-learning as an instructor-
970 centered process and want to check, even in a fast way, the learning routes that
are automatically generated. Consequently, adopting this and other types of
personalization still raise further challenges and its horizon is not fully clear.
The possibility of directly encoding in the e-learning metadata standards all the
information related to group formation+collaboration and resource constraints
975 is still a hot topic, and it will increase the applicability of planning and, conse-
quently, of our approach.

8. Conclusions and future work

We have described a flexible approach for intelligent personalization of e-learning routes, which combines students' and profiles' modeling, knowledge
980 representation, planning and case-based planning techniques to recommend the best learning route (of contents) to each student. As a proof of concept, our approach has been integrated on top of *Moodle*, thus using widely approved educational standards, such as SCORM and IMS. In *Moodle* we can monitor the students' progress and interaction in the course and use an adaptation method
985 to repair unexpected discrepancies, which is very interesting in large courses with a high number of students.

In order to check the applicability of our approach, we have run a significant number of experiments. They show a good performance according to the time required to calculate and re-adapt the learning route, as well as to the
990 satisfaction degree after this process.

Our future work addresses four main issues. First, to extend the definition of the LO metadata with more flexible fields, such as range-based durations, students' profiles and languages, etc. to help define courses more easily, thus allowing for more flexible learning routes. Second, to enhance the concept of
995 plan stability to manage more complex resource and temporal constraints, such as those involving group participation tasks. The idea is to introduce in the new route the "structural properties" of the original learning route modeled as *preferences* that should be satisfied. Third, to extend our Web services to provide a standard API to allow LMS-based agents to monitor changes in the
1000 students' profiles, thus improving the users' interaction to represent more issues about the LOs, especially for teachers. This will facilitate the evaluation of our approach in a large number of scenarios. Fourth, to include an ontology system to increase the reusability of different contents.

9. Acknowledgements

1005 This work has been partially funded by the Consolider AT project CSD2007-0022 INGENIO 2010 of the Spanish Ministry of Science and Innovation, the MICINN project TIN2011-27652-C03-01, the MINECO and FEDER project TIN2014-55637-C2-2-R, the Mexican National Council of Science and Technology, the Valencian Prometeo project II/2013/019 and the BW5053 research
1010 project of the Free University of Bozen-Bolzano.

Alepis, E., & Virvou, M. (2011). Automatic generation of emotions in tutoring agents for affective e-learning in medical education. *Expert Systems with Applications*, 38, 9840–9847.

Anaya, A., Luque, M., & García-Saiz, T. (2013). Recommender system in collaborative learning environment using an influence diagram. *Expert Systems with Applications*, 40, 7193–7202.
1015

Backstrom, C., & Nebel, B. (1995). Complexity results for SAS+ planning. *Computational Intelligence*, 11(4), 625–655.

Baldiris, S., O.C., S., Barrera, C., Boticario, J., Velez, J., & Fabregat, R. (2007). Linking educational specifications and standards for dynamic modelling in ADAPTAPlan. In *Proceedings of the CONTEXT 2007 Workshop on Representation Models and Techniques for Improving E-Learning (ReTieL 2007)* (pp. 56–63).
1020

Bonisoli, A., Gerevini, A. E., Saetti, A., & Serina, I. (2015). Effective plan retrieval in case-based planning for metric-temporal problems. *Journal of Experimental & Theoretical Artificial Intelligence*, 27, 603–647.
1025

Borrajo, D., Roubířková, A., & Serina, I. (2015). Progress in case-based planning. *ACM Computing Surveys*, 47, 1–39.

Brusilovsky, P., & Vassileva, J. (2003). Course sequencing techniques for large-scale web-based education. *International Journal of Continuing Engineering Education and Lifelong Learning*, 13, 75–94.
1030

- Campagni, R., Merlini, D., Sprugnoli, R., & Verri, M. (2015). Data mining models for student careers. *Expert Systems with Applications*, *42*, 5508–5521.
- Caputi, V., & Garrido, A. (2015). Student-oriented planning of e-learning contents for moodle. *Journal of Network and Computer Applications*, *53*, 115–127.
- Castillo, L., Morales, L., Gonzalez-Ferrer, A., Fdez-Olivares, J., Borrajo, D., & Onaindia, E. (2010). Automatic generation of temporal planning domains for e-learning. *Journal of Scheduling*, *13*(4), 347–362.
- Champiri, Z., Shahamiri, S., & Salim, S. (2015). A systematic review of scholar context-aware recommender systems. *Expert Systems with Applications*, *42*, 1743–1758.
- Chen, Y., Wah, B., & Hsu, C. (2006). Temporal planning using subgoal partitioning and resolution in SGPlan. *Journal of Artificial Intelligence Research*, *26*, 323–369.
- Chrysafiadi, K., & Virvou, M. (2013). Student modeling approaches: a literature review for the last decade. *Expert Systems with Applications*, *40*, 4715–4729.
- Clemente, J., Ramirez, J., & de Antonio, A. (2011). A proposal for student modeling based on ontologies and diagnosis rules. *Expert Systems with Applications*, *38*, 8066–8078.
- Comi, A., Fotia, L., Messina, F., Pappalardo, G., Rosaci, D., & Sarne, G. (2015a). Forming homogeneous classes for e-learning in a social network scenario. In *Proc. Int. Conference on Distributed Computing (IDC 2015)* (pp. 131–141).
- Comi, A., Fotia, L., Messina, F., Pappalardo, G., Rosaci, D., & Sarne, G. (2015b). Using semantic negotiation for ontology enrichment in e-learning multi agent systems. In *Proc. Int. Workshop on Intelligent Computing at Large Scale (ICLS-2015)* (pp. 474–479).

- de Oliveira, M., Ciarelli, P., & Oliveira, E. (2013). Recommendation of programming activities by multi-label classification for a formative assessment of students. *Expert Systems with Applications*, *40*, 6641–6651.
- Essalmi, F., Ben Ayed, L., Jemni, M., Graf, S., & Kinshuk (2015). Generalized metrics for the analysis of e-learning personalization strategies. *Computers in Human Behavior*, *48*, 310–322.
- Felder, R., & Silverman, L. (1988). Learning and teaching styles in engineering education. *Engr. Education*, *78(7)*, 674–681.
- Fox, M., Gerevini, A., Long, D., & Serina, I. (2006). Plan stability: Replanning versus plan repair. In *Proc. 16th Int. Conference on Automated Planning and Scheduling (ICAPS-2006)* (pp. 212–221). AAAI Press.
- Garrido, A., Fernandez, S., Morales, L., Onaindia, E., Borrajo, D., & Castillo, L. (2013). On the automatic compilation of e-learning models to planning. *The Knowledge Engineering Review*, *28(2)*, 121–136.
- Garrido, A., Morales, L., & Serina, I. (2012). Using ai planning to enhance e-learning processes. In *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling (ICAPS 2012)* (pp. 47–55).
- Garrido, A., & Onaindia, E. (2010). On the application of planning and scheduling techniques to e-learning. In *Proc. 23rd Int. Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE 2010) - LNAI 6096* (pp. 244–253). Springer.
- Garrido, A., & Onaindia, E. (2013). Assembling learning objects for personalized learning. an ai planning perspective. *IEEE Intelligent Systems*, *28(2)*, 64–73.
- Garrido, A., Onaindia, E., Morales, L., Castillo, L., Fernandez, S., & Borrajo, D. (2009). Modeling e-learning activities in automated planning. In *Proceedings of the 3rd International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS-ICAPS 2009)* (pp. 18–27).

- Garrido, A., Onaindia, E., & Sapena, O. (2008). Planning and scheduling in an e-learning environment. a constraint-programming-based approach. *Engineering Applications of Artificial Intelligence*, *21(5)*, 733–743.
- Garruzzo, S., Rosaci, D., & Sarne, G. (2007a). ISABEL: a multi-agent e-learning system that supports multiple devices. In *Proc. Int. Conference on Agent Technologies (IAT 2007)* (pp. 485–488). 1090
- Garruzzo, S., Rosaci, D., & Sarne, G. (2007b). MASHA-EL: a multi-agent system for supporting adaptive e-learning. In *Proc. Int. Conference on Tools with Artificial Intelligence (ICTAI 2007)* (pp. 103–110).
- Gaudio, E., Montero, M., & Hernandez-del Olmo, F. (2012). Supporting teachers in adaptive educational systems through predictive models: a proof of concept. *Expert Systems with Applications*, *39*, 621–625. 1095
- Gerevini, A., Roubíčková, A., Saetti, A., & Serina, I. (2013a). Offline and online plan library maintenance in case-based planning. In *Proceedings of the 13th Conference of the Italian Association for Artificial Intelligence*. Springer. 1100
- Gerevini, A., Roubíčková, A., Saetti, A., & Serina, I. (2013b). On the plan-library maintenance problem in a case-based planner. In *Proceedings of the 21st International Conference on Case-Based Reasoning*. Springer.
- Gerevini, A., Saetti, A., & Serina, I. (2003). Planning through stochastic local search and temporal action graphs in LPG. *Journal of Artificial Intelligence Research*, *20*, 239–290. 1105
- Gerevini, A., & Serina, I. (2010). Efficient plan adaptation through replanning windows and heuristic goals. *Journal of Algorithms in Cognition, Informatics and Logic*, *102*, 287–323.
- Idris, N., Yusof, N., & Saad, P. (2009). Adaptive course sequencing for personalization of learning path using neural network. *International Journal of Advances in Soft Computing and its Applications*, *1(1)*, 49–61. 1110

- IMS (2008). IMS global learning consortium. Available at <http://www.imsglobal.org> (accessed April 2016).
- 1115 Jeremic, Z., Jovanovic, J., & Gasevic, D. (2012). Student modeling and assessment in intelligent tutoring of software patterns. *Expert Systems with Applications*, *39*, 210–222.
- Kurilovas, E., Zilinskiene, I., & Dagiene, V. (2014). Recommending suitable learning scenarios according to learners' preferences: an improved swarm
1120 based approach. *Computers in Human Behavior*, *30*, 550–557.
- Kurilovas, E., Zilinskiene, I., & Dagiene, V. (2015). Recommending suitable learning paths according to learners' preferences: Experimental research results. *Computers in Human Behavior*, *51*, 945–951.
- Kyriacou, D. (2008). A scrutable user modelling infrastructure for enabling life-
1125 long user modelling. (pp. 421–425). Springer volume 5149 of *Lecture Notes in Computer Science*.
- Leake, D. (1996). *Case-Based Reasoning*. Cambridge, Massachusetts: The MIT Press.
- Liberatore, P. (2005). On the complexity of case-based planning. *Journal of*
1130 *Experimental & Theoretical Artificial Intelligence*, *17(3)*, 283–295.
- LOM (2002). Draft standard for learning object metadata. IEEE. rev. 16 february 2005. Available at <http://iee-SA.centraldesktop.com/ltsc/> (accessed April 2016).
- Martinez, F., Magoulas, G., Chen, S., & Macredie, R. (2004). Recent soft computing approaches to user modeling in adaptive hypermedia. In *Proceedings of Adaptive Hypermedia (AH'04)*, LNCS 3137 (pp. 104–114).
1135
- Messina, F., Pappalardo, G., Rosaci, D., & Corrado, G., S. Sarne (2013). A distributed agent-based approach for supporting group formation in P2P e-learning. In *Proc. Int. Conference on Advances in Artificial Intelligence* (pp.
1140 312–323).

- Morales, L., Garrido, A., & Serina, I. (2011). Planning and execution in a personalised e-learning setting. In *Proc. Conferencia Española para la Inteligencia Artificial (CAEPIA). LNAI 7023* (pp. 232–242). Springer-Verlag.
- Natek, S., & Zwillig, M. (2014). Student data mining solution-knowledge
1145 management system related to higher education institutions. *Expert Systems with Applications*, *41*, 6400–6407.
- Nguyen, T., Do, M., Gerevini, A., Serina, I., Srivastava, B., & Kambhampati, S. (2012). Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence*, *190*, 1–31.
- 1150 Park, D., Kim, H., Choi, I., & Kim, J. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications*, *39*, 10059–10072.
- Pena-Ayala, A. (2014). Educational data mining: a survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, *41*, 1432–
1155 1462.
- Polsani, P. (2003). Use and abuse of reusable learning objects. *Journal of Digital Information*, *3(4)*. Available at <http://journals.tdl.org/jodi/article/view/89/88> (accessed April 2016).
- Richter, S., & Westphal, M. (2014). The LAMA planner: Guiding cost-based
1160 anytime planning with landmarks. *CoRR*, *abs/1401.3839*. URL: <http://arxiv.org/abs/1401.3839>.
- Rosaci, D., & Sarne, G. (2010). Efficient personalization of e-learning activities using a multi-device decentralized recommender system. *Computational Intelligence*, *26*, 121–141.
- 1165 SCORM (2004). Sharable Content Object Reference Model. Available at <http://scorm.com> (accessed April 2016).

- Serina, I. (2010). Kernel functions for case-based planning. *Artificial Intelligence*, *174*, 1369–1406.
- Smyth, B., & McKenna, E. (2001). Competence models and the maintenance
1170 problem. *Computational Intelligence*, *17*, 235–249.
- Spalazzi, L. (2001). A survey on case-based planning. *Artificial Intelligence Review*, *16*, 3–36.
- Tonidandel, F., & Rillo, M. (2002). The FAR-OFF system: a heuristic search
case-based planning. In *International Conference on Artificial Intelligence
1175 Planning and Scheduling (AIPS)* (pp. 302–311).
- Ullrich, C., & Melis, E. (2009). Pedagogically founded courseware generation
based on HTN-planning. *Expert Systems with Applications*, *36*, 9319–9332.
- Vidal, V. (2014). Yahsp3 and yahsp3-mt in the 8th international planning
competition. In *Proceedings of the 8th International Planning Competition
1180 (IPC-2014)*. Portsmouth, USA.
- Yang, Q., Nau, D., & Hendler, J. (1992). Merging separately generated plans
with restricted interactions. *Computational Intelligence*, *8(4)*, 648–676.