

# LRNPlanner: Planning Personalized and Contextualized E-learning Routes

Eva Onaindia and Antonio Garrido and Oscar Sapena  
Departamento Sistemas Informaticos y Computacion  
Universidad Politecnica de Valencia  
Camino de Vera s/n 46022 Valencia

## Abstract

*The aim of educational systems is to design a sequence of learning objects on a set of topics tailored to the learner's goals and individual properties. However, some of the main difficulties actual educational systems have to face is the generation of learning routes for multiple learners, the lack of an explicit management of time and resources or the synchronization of group activities. We claim that AI planning provides the necessary technology to address all these missing issues in actual e-learning environments, and this paper elaborates in this direction.*

## 1 Introduction

The application of Artificial Intelligence (AI) planning techniques has reported significant advances in the process of designing e-learning routes due to the similarity of both processes [5, 8, 10]. Designing an e-learning route implies to generate a sequence of learning objects tailored to the learner's goals and individual properties. The goal is to select the most suitable learning objects and order them properly so as to form the route a learner must follow during the development of the e-learning course. The purpose of AI planning is to achieve a course of actions that attain a set of goals starting from an initial situation [4]. Even though the purpose of both processes is essentially the same, e-learning is more concerned with the activation and connection of learning objects to create the e-learning route, whereas AI planning is directly concerned with the activities to be performed by learners. This conceptually different view makes it important to define the relationship between the domain concept represented by a learning object and the necessary activities a learner must perform to attain that concept [7].

The aim of an e-learning environment is to offer intelligent personalization capabilities based on the different characteristics of each individual learner. One of the main diffi-

culties actual systems have to face is the simultaneous generation of a course for multiple learners, trying to accommodate the different learning and studying styles, strategies and preferences of diverse learners [9]. Other missing aspects in actual systems are the lack of an explicit management of time, consumption of resources or synchronization of group activities [3]. Current approaches are aimed at obtaining a profile-adapted course for a single learner without considering the particular characteristics of the context where the learner will carry out the proposed activities. In other words, current e-learning systems focus on the design of learning routes and ignore the practicability of executing the routes on a particular learning scenario.

In this paper we claim that the application of AI planning methods can be successfully used to overcome the difficulties of personalization and contextualization of current learning platforms. We present a thorough analysis on the practical aspects which are necessary to consider in order to generate a personalized and contextualized learning route. In particular, we present LRNPlanner, an integrated tool for the definition and resolution of learning routes that uses heuristic AI planning techniques.

The paper is organized as follows. Next section introduces a simple scenario case to motivate our work. Section 3 details the components of an e-learning planning problem and presents a graphical authoring tool for the definition of learning routes. Section 4 outlines the particular planning characteristics of our working domain and it then shows the resolution process. Section 5 shows two application examples and, finally, section 6 concludes and outlines the future work.

## 2 Motivation

In this section we present a simple scenario case to motivate our work. Two learners, John and Rebecca wish to follow a course on Java programming which is delivered as a special programming training in an academic center. The

two learners have different profiles and backgrounds. John is a learner at the first year of B.S. in Computing Science and Rebecca is a self-taught graphic designer with low-level programming skills. Both learners have jobs that demand their attention during their working hours and they need some flexibility to follow the course. Therefore, e-learning seems the most logical way for them to pursue the course.

The academic center has structured the Java course in several modules. Each module is designed to acquire some basic knowledge which is necessary to carry on with the next module (see Figure 1). There are different ways (routes) that a learner can follow to attain the concepts of each module, which will depend on the learner's profile and previous knowledge. Each module consists of a set of tasks where some of them are traditional classroom delivery or lab sessions (in-person attendance), and others are e-learning tasks that learners can perform at their most convenience.

AI planning techniques are an ideal mechanism to tackle this type of problems as it is necessary to take into account the initial background of the problem actors, the relationships and coordination of tasks, the synchronization and scheduling of group activities, and, finally, the logistics on the academic center in terms of resource availability. However, designing e-learning routes under this complex setting can be seen as a very particular planning problem with specific features and the application of a general-purpose AI planning algorithm may not be the most appropriate solution. In the rest of the paper we will show the adaptation of traditional AI planning techniques to efficiently solve the design of personalized and contextualized e-learning routes.

### 3 The e-learning planning problem

AI planning is concerned with the construction of a sequence of formally-described world-states resulting from the application of actions defined in a planning domain. Each action may be executed only in some particular set of world states (when its preconditions hold), and has some particular set of results on its world state (its effects). A planning problem consists of a planning domain together with an initial state of the world and a desired goal state, or set of goal states, of the world. A planner solves a planning problem by producing a sequence of actions, each of which is legal in its starting world state, which takes the initial state to a goal state.

The e-learning planning problem can be identically described with the same terms but with some exceptions. An e-learning domain describes the tasks of a particular course; tasks are used to achieve concepts (learning objects) and they specify the realization of actions over the concepts. For instance, the Java programming course of section 2 may contain a concept like *Vars&types* and tasks like *Read-*

*ing, Exercise* or auto-assessment *Quiz*. Tasks will form the learning route of each learner, i.e. the sequence of actions the learner must follow to achieve the course's goals. Concepts can act as prerequisites and/or learning outcomes of a task, analogously to the preconditions and effects of an action. For example, the effect of task *Reading\_Vars&types* can be *Basics\_Vars&types*, which is a prerequisite for the task *Exercise\_Vars&types*, and whose effect, likewise, is a prerequisite for doing the task *Quiz\_Vars&types* (see Figures 1, 2). On the other hand, the initial state is formed by the learners' data, (e.g. personal profile, previous knowledge and resource availability) and the information of the particular context where the course will take place (e.g. resource availability, capacity constraints, etc.) Finally, the goal state is to attain the course's goals for all the involved learners.

As can be seen, defining an e-learning problem is not a simple task. Therefore, a graphic representation results to be a user-friendly way for both defining and visualizing e-learning problems. This way, the next section is devoted to explain the graphical tool we have developed to define this type of problems.

#### 3.1 The graphical authoring tool

We can identify three main components in an e-learning problem: concepts, tasks and learner-profiles. This way, the tool offers three views according to the component is being designed:

- The **conceptual** view allows the instructor to define the course contents in terms of concepts and their relations, i.e. *what* knowledge is going to be given.
- The **task** view is the design layer after the conceptual view. At this stage the instructor applies a decomposition of concepts into tasks, i.e. *how* the knowledge is going to be given.
- The **adaptation** view is the last design layer. At this stage the instructor represents the information about the learners, their profiles and background as well as the context characterization, i.e. *who* is the knowledge going to be given to and under which resources.

The overall idea is to create two abstract layers and one final layer, where all components join. The tool guides the user to follow the procedure an instructor would go through when creating a course. The first thoughts will mainly contain general concepts and objectives the instructor wants to teach. Therefore, the conceptual view permits the creation of the general outline of a course using the most general components. When the instructor has finished the outline of the course (s)he would probably think about different ways

and tasks to achieve the concepts. The task view is defined as a detailed view of the first layer, where the instructor specifies the necessary tasks to attain a concept and how they are related. Having created the outline of concepts and the tasks to achieve these, the instructor would maybe want to do more specific adjustments on the tasks or concepts. This leads to the adaptation view, which now contains all components: concepts, tasks, profiles and how they are related.

**The conceptual view.** The conceptual view builds a network of concepts and relations between them based on the Learning Object Metadata (LOM) [6]. LOM is basically an XML dialect to describe a learning object, no matter of what kind or in which environment it exists. The actual components of LOM are mostly optional, giving the user free choice about how detailed (s)he wants to describe the object. In the conceptual view of LRNPlanner we only use the component **Relation** as this is the only one relevant for planning. This component contains all relations to other LOM objects. LOM offers several types of relations like *is-required-by*, *is-basis-for* or *is-referenced-by* and all their inversions too. The possibility to describe a single object and easily link it with other objects makes it very powerful and extremely useful for the design of the conceptual view in LRNPlanner.

The conceptual view contains only concepts. Not all the concepts an instructor creates for a course do necessarily have to be done by the learner to complete the course. It might be that only some of them need to be done for the completion of the course. Hence, the relations are divided in AND- and OR-Relations, which refer to the relations *is-required-by* and *is-basis-for* in LOM, respectively (this is a personal interpretation of the computational meaning of the LOM relations). All concepts with an AND-Relation have to be completed and at least one of the concepts with an OR-Relation as well. This option already enables a quite realistic course design concerning the concepts. Additionally, instructors might recommend some concepts before doing another and so the Reference-Relation is created (relation *is-referenced-by* in LOM). This relation does not denote a strong requirement but a recommendation for finishing one concept before starting another. This rounds out the conceptual view and builds a good basis for the task view.

Figure 1 shows the conceptual view of a simple introductory module in the Java programming course. Each module contains a *Start* and *End* node which mark the start and end, respectively, of the module. This view contains concepts such as *structure of a program*, *variables & types*, *conditional flows*, *if/else statements*, etc. and the relations linking these concepts. As can be seen in the figure, the learner can finish the route either by learning the concepts *if/else* and *switch* or by having previous knowledge on these concepts in similar programming languages (*prev. if/else*).

**The task view.** This layer inherits the information from the conceptual view and associates tasks to each concept in the first layer. The task view consists entirely of tasks, and it is thought to be a detailed view of the first layer where each concept is *translated* into the necessary tasks to attain it. This view, being of a similar nature as the conceptual view, receives the same relations because tasks can have the same requirements or recommendations as concepts. For every concept in the conceptual view it has to be a task view of the concept in the second layer. The simplest task view is to associate a single task to each concept, but the instructor may create a detailed task network as desired. Additionally, it is possible to define a duration for each task and even model the case when a task can be repeated several times as a reinforcement task, also specifying the max number of repetitions allowed per task. Figure 2 shows the task view of the concept *variables & types*. As can be seen in the figure, there exist several routes to reach the *End* node.

**The adaptation view.** The third layer inherits the tasks from the task view and includes user-profiles as the main component of the adaptation view. This view is a little different from the other two and so the relations differ from the other layers' relations. First thing to consider is that the adaptation view encompasses concepts and tasks and thus the links concept-task and task-concept now become the main relationships in this view. The information inherited from the preceding layer is automatically transformed into a flow structure where every task has a subsequent concept and every concept is preceded by a task. The adaptation view receives the relations among concepts and tasks from the first two layers and assigns a virtual concept to each task, with the basic function of having the 'task\_done', and it is used to link with other tasks. This creates a planning-like structure (in terms of precondition-action-effect) and gives the instructor the opportunity of furthermore refining the course.

Once the basic structure of the adaptation view has been automatically created, the instructor proceeds with the e-learning course profile-adaptation. Adaptation here means adding details to concepts and tasks to get a fully customized course. These details include specifying which tasks are more suitable for which types of learners, the achievement level of concepts or the type of material and resources a task requires to attain its goals:

1. **Learners' profiles.** Learners' profiles are defined at this stage to represent the different learning styles as classified by Felder [2], the different types of learning as expressed in the Bloom's taxonomy [1] or any other learning feature that is considered relevant for the personalization of a learning route. Profiles are then used to adapt the course tasks to the different learning characteristics. For example, it will be preferable to guide inductive learners through tasks that exemplify con-

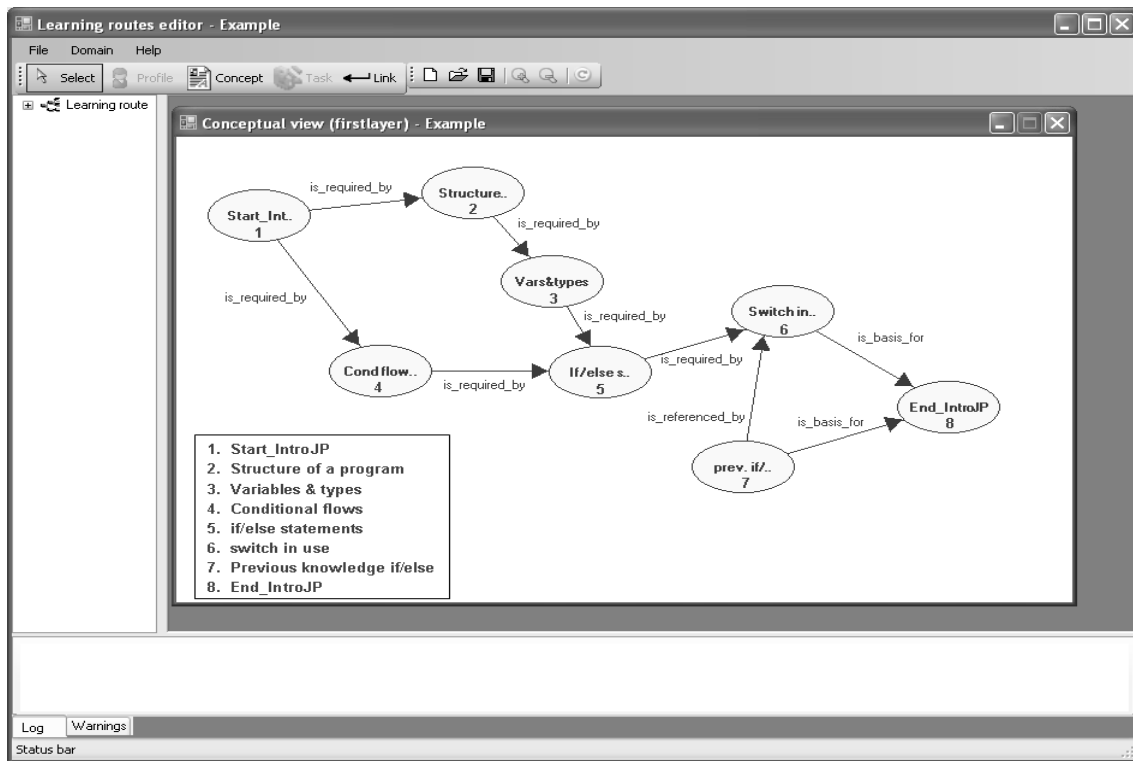


Figure 1. LRNPlanner: conceptual view of the introductory module in the Java Programming course

cepts or recommend some tasks only to learners with high-level language skills. For instance, the profile named *processing* can take on either the value *visual* or *verbal* and the profile *reasoning* can take on either the value *inductive* or *deductive*, according to Felder's classification. A more personalized profile might be *language-skills* or *programming-background* with values high, medium or low. The same relations defined for concepts and tasks (AND-, OR- and Reference-) are also used to create the profile(s)-adapted tasks with the same meaning.

2. **Achievement level of concepts.** The tool also allows the instructor to personalize the degree of achievement and/or necessity of concepts. Thus, concept-task and task-concept relations are associated a percentage value which is interpreted as the minimum degree a concept has to have in order to succeed. Percentages work as a kind of threshold and they are used to express that tasks attain and need concepts at a certain competence level. Learner's profiles can also be used to personalize the percentage values of concepts. Besides defining profile-dependent tasks, we can also set up the requirements and outcomes of a task according to the profile. Thus, a specific task  $t_1$  may attain

50% of concept  $c_1$  for learners whose profile value is *processing=visual* but 80% if the learner has a profile *processing=verbal*.

3. **Material and resources.** The tool also allows the user to specify the material required by a task like for example a *set course book*, a *Power Point* presentation for a task that requires animated slides or a *Flash presentation* for an interactive programme. The material assigned to a task is also associated with a resource type, i.e the necessary equipment to be able to use the task material. The resource associated to the material *set course book* may be just a *book* but the resource necessary to execute a *flash presentation* or a *video film* will be a *multimedia equipment* or *DVD player*.

Figure 3 shows the adaptation of the concept *variables & types*, including learner profiles and percentages of the required and attained concepts according to each profile. For instance, the task *animated slides* requires a 60% of the concept *done reading* for any type of learner (with no particular profile) and generates a 20% of *done animated slides* for everybody, plus 50%, 80% or 25% for visual, verbal or inductive learners, respectively. Under the AI planning perspective, all these elements comprise the domain of the problem. Learners' profiles describe the learners' typology,

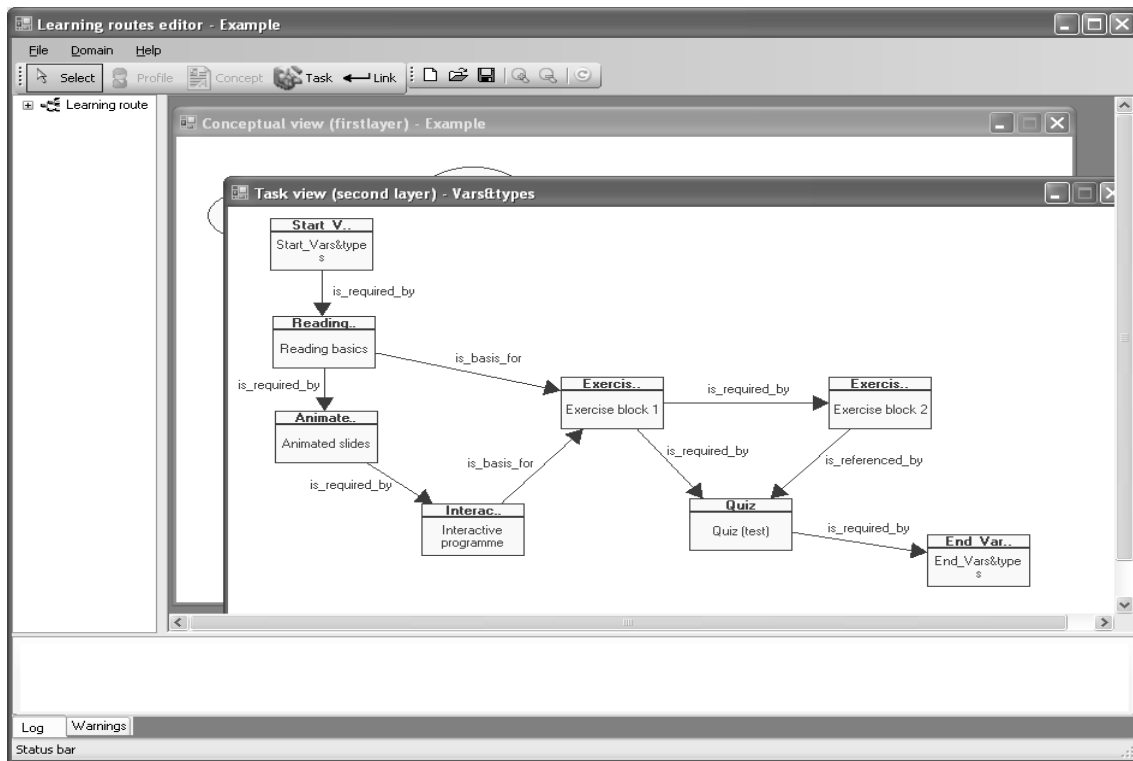


Figure 2. Task view for the concept *variables & types* in the Java programming course

which represents information shared by any e-learning domain. Concepts, tasks, percentages, materials and resources are the elements used to describe the particular problem domain — a module of the Java programming course in our case. Therefore, the conceptual view, the task view and the adaptations have only to be done once per course, which increases the reusability degree.

After generating the e-learning course, we have to specify the particular problem to be solved by introducing the data about the problem actors, i.e. the learners, and the problem context, i.e. the academic center. The former consists of entering the personal details of the learners such as their learning style, their previous knowledge level on the course topics, the goals to be attained and at which achievement degree (percentage), and their available resources. The latter refers to the particular characteristics of the environment where the course will take place, mainly regarding to resource availability such as the number of labs, and their capacity and temporal constraints. Note that in the problem definition it is important to model the personal resources the learners dispose of to pursue the course (e.g. a personal computer or a particular book). This is treated in LRNPlanner as a learner-dependent and unconstrained resource. However, if the learner does not dispose of personal resources, the academic center must schedule the usage of a

shared resource, which obviously introduces a major planning+scheduling effort.

#### 4 Planning contextualized learning routes

The basic principle underpinning the generation of e-learning routes can be formulated in the same terms as traditional planning: generate a sequence of ordered tasks such that the execution of this sequence will lead a learner to attain the goals of a course. Planning e-learning routes is also about temporal planning as tasks have different durations and the academic center must provide a scheduled course to the learner. This way, learners are usually more interested in attaining the goals in the shortest possible time (optimization of the overall plan duration or plan makespan) rather than minimizing the number of tasks. Additionally, actions can have associated a cost in terms of effort or resource consumption, and resources have a limited capacity, so the planner must be capable to work with other cost measures as well as satisfy the resource capacity constraints.

A particular characteristic of e-learning is that whilst the route of a single learner is a totally ordered sequence of tasks, the organization of a course for several learners involves a parallel execution of the tasks of different learners, the synchronization and scheduling of group activities and

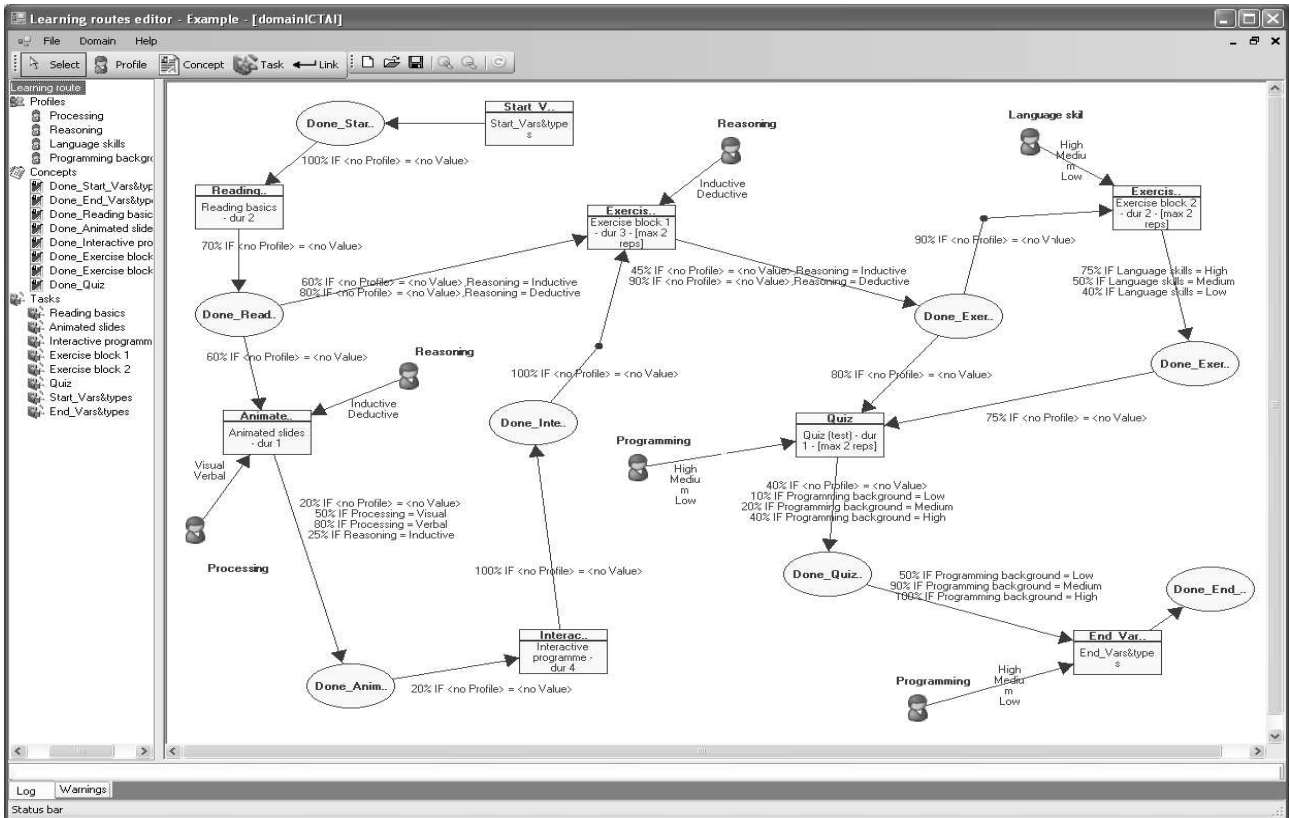


Figure 3. Adaptation view of the concept variables & types in the Java programming course

an explicit resource handling. Thus, the design of learning routes for several students is a planning+scheduling problem which introduces some specific features.

#### 4.1. Features of the e-learning domain

E-learning domains differ from traditional planning domains in several aspects:

- **Numeric variables.** Unlike most planning algorithms that use a propositional representation for facts, e-learning domains require a representation for handling numerical information (percentages) as concepts are numeric-valued items, which makes the planning task more complex.
- **Monotonic evolution.** E-learning tasks have only increasing effects because once a learner has attained a concept through the execution of one or several tasks, no further tasks can lessen the competence level the learner has acquired. Consequently, the learning process is a monotonically non-decreasing process.
- **Repetition of tasks.** In an e-learning setting, tasks can be executed more than once similarly as in any other

planning problem. However, repeating a task is a reinforcement activity for the learner to attain a determined competence level of a concept and, consequently, the number of possible repetitions is usually limited. This aspect makes a clear difference with respect to classical planning problems where there is no restriction on the number of times an action can be executed.

- **Synchronization.** Action synchronization is not usually addressed in traditional planning. In e-learning there are tasks that must be executed at the same time by a group of learners, e.g. doing a group activity, or time synchronized tasks, e.g. attending a lab session at 2–3.30 pm. Synchronization is also an important issue when generating a learning route for multiple learners.

The above aspects evidence that the emphasis must be put on the design of planning capabilities for coordinating the routes of several learners in time, dealing with consumption of resources, synchronization of group activities and even multi-criteria optimization. From all these considerations, we can conclude that developing an ad-hoc e-learning planner seems more beneficial than using a standard planning system.

## 4.2. Problem-solving in LRNPlanner

LRNPlanner uses a heuristic backward search process that extracts a valid plan by traversing the search space starting from the goal state. The starting point is a formula  $G$  that describes the set of goals for each learner, e.g.  $G = \{g_1, g_2\}$  where  $g_1 = \langle l_1, c_1, 80 \rangle$  and  $g_2 = \langle l_2, c_1, 100 \rangle$ , which denote that learner1 must attain concept  $c_1$  at level  $\geq 80$  and learner2 at level  $\geq 100$ .

The planner starts the process to search backward from  $G$  using the profile-adapted tasks defined in the domain. To do so, it must regress the formula  $G$  through the tasks operators to produce subgoals  $g_1'$  and  $g_2'$ . The regression of a formula  $g_i$  through a task operator  $t_j$  is the weakest formula  $g_i'$  such that if  $g_i'$  is satisfied by a state description before applying an instance of  $t_j$ , and  $g_i'$  satisfies the conditions of that instance of  $t_j$ , then  $g_i$  will be satisfied by the state description after applying that instance of  $t_j$ .

Let  $G = \{g_1, g_2\}$  be the goal state,  $I$  the initial situation which contains the following information  $I = \{\langle l_1, visual \rangle, \langle l_2, verbal \rangle, \dots\}$ , and  $A = \{t_1, t_2, \dots\}$ . Figure 4 shows the backward search process for this little example. The problem is to achieve  $G$  from a state  $I$  in which learner1 is visual and learner2 is verbal (as part of the information in the  $I$ ).  $t_1$  is a task for visual and verbal learners that increases the knowledge level of  $c_1$  in 30; and  $t_2$  is a task for verbal learners that increases the level of  $c_1$  in 70. Let us regress through  $apply(t_1, g_1)$  to produce  $c_1$  at level 30 for learner1. The next subgoal state (node 1 in Figure 4) contains the pending goals for both learners plus any preconditions of  $t_1$  not already in the goal description. A plan is reached when the regression process finds a subgoal state that is satisfied by the initial state description (node  $i$  in Figure 4). The plan is built by following upwards and instantiating the sequence of tasks that links the subgoal satisfied with the initial goal state.

It must be noticed that the obtained sequence of tasks (plan) contains the necessary actions for both learners to attain their goals. While the tasks of a single learner must be sequentially executed, the tasks of different learners can be executed in parallel. Thus, the sequence of tasks must be appropriately scheduled to obtain a consistent temporal plan. This is done by a Constraint Satisfaction Problem (CSP) module [3], which allocates tasks in time and checks the synchronization and resource availability constraints. Nevertheless, the sequence of tasks may not be schedulable; that is, it solves the logical structure of each single learner route but it is not an executable plan for the two learners under the specified time constraints. For this reason, the CSP module is not only invoked when a logical plan is reached but also during the regression process in order to prune the search space when a non-executable goal state is found. In Figure 4, the CSP is invoked in node 2 whose path-plan already

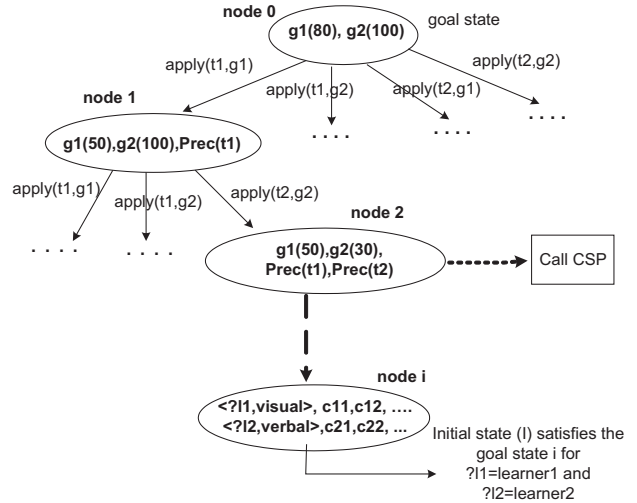


Figure 4. Backward search in LRNPlanner

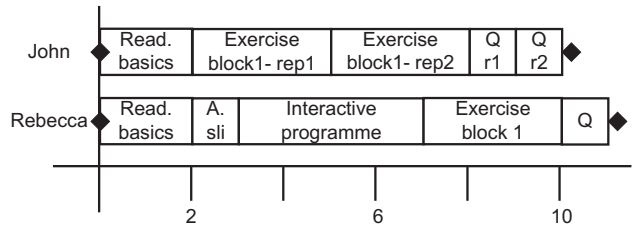


Figure 5. Routes for John and Rebecca

contains one task,  $t_1$  and  $t_2$ , for each learner, respectively.

## 5. Application examples

Let us revisit the simple scenario of the two learners, John and Rebecca, presented in section 2 to acquire a certain level on *variables & types* according to the e-learning course depicted in Figure 3. Once the e-learning domain has been defined, the instructor needs to model the data of the learners interested in pursuing the course. We will assume that John's profile values are: inductive, verbal, lang.skill=medium and prog.back=medium, whereas Rebecca's values are: deductive, visual, lang.skill=low and prog.back=low. The competence level that learners want to attain for concept *variables & types* is  $\geq 90$  for John and  $\geq 50$  for Rebecca. LRNPlanner finds the optimal makespan personalized learning route for each learner shown in Figure 5, where symbols ♦ represent the Start- and End- task nodes. Note that LRNPlanner can decide whether a learner needs to do the same reinforcement task more than once, as the auto-assessment task *Quiz* ( $Qr1$  and  $Qr2$ ) for John.

Planning routes for just a pair of learners does not usually

impose tight constraints on resources. This way, we have defined a new example with two types of learners; the former (Type 1 with 20 learners) with the profile values: inductive, visual, lang.skill=high and prog.back=medium, and the latter (Type 2 with 30 learners) with the values: deductive, verbal, lang.skill=medium and prog.back=high. The competence level required for concept *variables & types* is  $\geq 80$  and  $\geq 60$  for Type 1 and Type 2, respectively. The resulting plan is formed by 320 tasks (6x20 for Type 1 and 7x30 for Type 2) with the learning routes shown in Figure 6-a. Now, if we consider *Quiz* as a task that requires in-person attendance to a lab with max capacity for 20 learners which is available from time 10, the plan needs to be subsequently scheduled by the CSP to meet these constraints. The temporal plan, as a personalized schedule of tasks for each learner, appears in Figure 6-b. Note that the CSP module has split learners of Type 2 into two groups to satisfy the limited capacity of the lab, which makes the plan makespan a bit longer.

Additionally, LRNPlanner can detect if there is not a feasible learning route because of an inconsistency with one or more profiles or an overconstrained resource. For example, if the *Quiz* task can be done at most once it will be impossible to attain a value  $> 80$  for the *Done\_Quiz* concept, no matter the learner's profile (see Figure 3). Similarly, if the lab for *Quiz* is only available from 10 to 12 the CSP module will fail to find a feasible schedule. This is a very interesting property as it allows the instructor to find out whether a learning course is not well defined before learners start executing the first task. This way, the instructor can modify and/or include new tasks to check the validity of the course for different profiles without jeopardizing the educational process of the learners.

## 6. Conclusions and future work

In this paper we have presented an integrated tool for the definition and resolution of e-learning routes<sup>1</sup>. Our next immediate objective is to integrate LRNPlanner into an educational platform to access educational web sites such as those offering learning objects, learning scenarios and interactive exercises. The objective is to enrich the conceptual view and facilitate the user the discover, search and recovery of learning objects in both the e-learning content repositories and the global Semantic Web.

We also plan to test our tool with a group of different students, i.e. obtain the students learning routes, trace the learners evolution and check their level of achievement with regular tests. Additionally, we will introduce plan repairing

<sup>1</sup>This work has been partially funded by Consolider Ingenio 2010 CSD2007-00022 project and by the Spanish Government TIN2005-08945-C06-06 project.

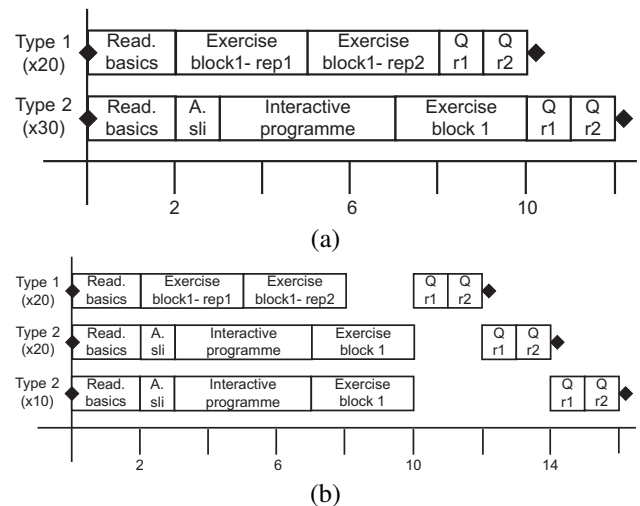


Figure 6. Routes for the two types of learners

techniques and dynamic adaptation to adjust the learning routes in view of the learners evolution.

## References

- [1] B. Bloom. *Taxonomy of Educational Objectives, Handbook I: the Cognitive Domain*. New York: David McKay Co Inc., 1956.
- [2] R. Felder and L. Silverman. Learning and teaching styles in engineering education. *Engr. Education*, 78(7):674–681, 1988.
- [3] A. Garrido, E. Onaindia, and O. Sapena. Planning and scheduling in an e-learning environment. A constraint-programming-based approach. *Engineering Applications of Artificial Intelligence*, 21(5):733–743, 2008.
- [4] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning. Theory and Practice*. Morgan Kaufmann, 2004.
- [5] P. Karampiperis and D. Sampson. Adaptive instructional planning using ontologies. In *4th IEEE Int. Conference on Advanced Learning Technologies*, pages 126–130, 2004.
- [6] LOM. Draft standard for learning object metadata. IEEE. 15 july 2002. 6 oct. 2007, 2002.
- [7] A. Staller. Merging domain knowledge and task analysis in an ontology. In *Current developments in Technology-Assisted Education*, pages 1585–1589, 2006.
- [8] C. Ullrich. Course generation based on HTN planning. In *Proc. 13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems*, pages 74–79, 2005.
- [9] C. Ullrich. *Course generation as a Hierarchical Task Network Planning Problem*. Ph.D. Dissertation, University of Saarlandes, 2007.
- [10] D. Vrakas, G. Tsoumakas, F. Kokkoras, N. Bassiliades, I. Vlahavas, and D. Anagnostopoulos. PASER: a curricula synthesis system based on automated problem solving. *Int. Journal on Teaching and Case Studies*, 1(1/2):159–170, 2007.