An AI Planning-based Approach for Automated Design of Learning Routes

Eva Onaindia, Oscar Sapena, Antonio Garrido
Department of Information Systems and Computation, Polytechnic University of Valencia, Valencia, Spain
onaindia@dsic.upv.es
osapena@dsic.upv.es
agarridot@dsic.upv.es

Abstract: This paper presents an approach for the automated design of learning routes, based on the application of AI techniques. Our educational planning system is composed of two main modules: an instructor-oriented *graphical authoring tool* to model the key elements of a course and an *automated planner* to compute a learning route for a specific student profile or a customised learning route for a group of students. More specifically, the proposed architecture works in three stages: (1) the instructor defines the elements of a course such as learning concepts, tasks, teaching materials, required resources, etc. Our authoring tool helps guide the instructor during the course design; (2) an AI planning process is applied over the course structure defined in the above stage to obtain a generic learning route for a given student profile. This learning route is composed of a set of partially-ordered activities which contemplates teaching goals at different levels of competence. The purpose of this generic learning route is to validate the course design described by the instructor in the authoring tool and check the existence of a feasible implementation for the course; (3) a second AI planning process permits to obtain a customised learning route for a specific group of students. This learning route is a course of actions allocated in time which accounts for the available resources and temporal constraints of the particular teaching context where the course will be taught.

Our educational planning system offers two novelties with respect to other tools for course generation: a user-friendly graphical tool aimed at providing instructors support during the specification of a course/subject and an automated planner to compute customised plans totally adapted to the particular learning context, considering time constraints, resource usage and all issues necessary to make a learning route realizable (executable plan). Additionally, our approach allows us to dynamically adapt the learning route during its execution according to the course evolution.

Keywords: Course design; Learning content management systems; e-learning strategies

1. Introduction

The application of AI planning techniques has reported important advances in the automatic course generation within e-learning. The ultimate goal of a course generator is to automatically assemble learning objects from one or several repositories to a greater unit. One of the first attempts in this direction was the work by Peachy and McCalla (Peachy 1986), in which the learning material is structured in learning concepts and prerequisite knowledge is defined, which states the causal relationship between different concepts. This instructional planner was one of the first approaches to combine instructional knowledge and artificial intelligence planning techniques to generate sequence of learning materials. In the same direction, Vassileva designs a system that dynamically generates instructional courses based on an explicit representation of the structure of the concepts/topics in the domain and a library of teaching materials (Vassileva 1997). Other approaches suggest the use of ontologies and learning objects metadata in order to calculate the best path through the learning material (Karampiperis 2004); others introduce hierarchical planners to represent pedagogical objectives and tasks in order to obtain a course structure (Ullrich 2005). Most recent works, such as the one presented in (Vrakas 2007), incorporate machine learning techniques to assist content providers in constructing learning objects that comply with the ontology concerning both learning objectives and prerequisites.

All these approaches as well as most current adaptive web-based educational systems rely on a subject domain ontology, specifying the domain concepts and the relations among them, which represent the instructional knowledge. However, from an instructional design perspective, domain concepts are not specific enough to guide learning, instruction and assessment. In contrast, *task analysis for instructional design* is based on learning outcomes, including their taxonomic classification, information processing analysis, and prerequisite analysis (Smith 2005). Unlike purely concept-based models, task analysis describes how domain concepts and task analysis results can be combined together, thus increasing didactic precision (Staller 2006).

Task analysis for instructional design keeps a strong resemblance with AI planning models. Information processing analysis specifies the mental steps (actions in planning) that are performed to complete a task. Prerequisite analysis identifies for each step what the learner needs to be able to do in order to achieve the step (planning preconditions). So classifying the learning outcomes (plan effects) corresponding to the learning goal and the prerequisites is an essential part of task analysis (plan generation).

The purpose of AI planning, as well as task analysis, is to represent the relation between learning outcomes and subject domain knowledge (domain concepts). In this direction, we present a novel approach for instructional design which is intended to help the instructor in the design, creation and monitoring of an instructional course or learning route. Our educational planning system consists of two main modules: an instructor-oriented *graphical authoring tool* to model the key elements of the course such as prerequisites, tasks and learning outcomes and an *automated planner* which can be used in two different ways:

- to generate a course for a specific student profile; this analysis does not take into account neither context constraints nor specific resources.
- to generate a course for a specific student or group of students; the outcome of this process is a customised plan for one or more students under a set of temporal and resource constraints.

The rest of the paper is organized as follows: Section 2 introduces the basic elements of the problem we want to tackle. Section 3 shows the architecture of our educational planning system and provides details of the graphical authoring tool and the planner. Section 4 shows an application example and section 5 concludes.

2. Problem definition

In this section we describe the basic elements that constitute our problem of automatic course generation. We divide these elements in three categories: the learner model, the task analysis and those items that are specifically considered for the customised plans.

2.1 Learner model

The learner model or learner profile records personal data about the student which will be used by the planner when calculating the course that better fits the student's strengths, interest, modalities, personal resources, etc. The learner model is characterized by the following attributes:

- *Learning style*: there are many learning styles analyses and instruments for determining learning style preferences, such as the Felder model (Felder 1988). The students learning styles determine the teaching style so the instructor designs the most appropriate tasks that best fit for the teaching-to-a-learning style of the students.
- *Previous knowledge*: it is important to identify the existing knowledge (prerequisites or required previous knowledge) for each student. The more actively students relate the new material to their previous knowledge, the more effective their learning will be.
- *Personal resources*: tasks may require the use of specific resources such as video, audio or other type of resource. In order to successfully perform a task, the learner will need access to a computer with a DVD drive, or some hard disk space to create a working directory or internet and e-mail access. If the learner does not have the necessary personal resources at disposal, the planner will search for alternative activities to reach the learning goal.

2.2 Task analysis: concepts and tasks

Concepts represent prerequisites as well as learning outcomes of a task. Generally speaking, we can say concepts are learning sub-goals that must be progressively attained in order to reach the final learning goal. In our model we can specify that a student attains a concept at a certain competence level (indicated as a percentage). This value will mainly depend on two input parameters: i) the student profile and ii) the knowledge level of the task prerequisites. Thus, if a student is able to attain a concept at a competence level of 65% this will highly influence the success of the remaining learning route. Our model also allows ordering constraints between concepts or time constraints to set a time deadline for attaining a concept.

Tasks are the core element of the instructional course in our educational planning system. Tasks represent units of learning that learners must follow to achieve a goal. The execution of a task refers to a piece of education, lesson, exam, exercise, etc. The instructor designs the tasks flow according to the students learning style, their previous knowledge and the competence level of the intermediate attained learning outcomes. This way, by linking prerequisites, tasks and learning outcomes the instructor designs a general course for a given subject. This general learning route will implicitly comprise many specific courses, one for each student profile and resulting learning outcome. Tasks are characterized by several attributes, some of which are essential for creating a general course and others for creating a customised plan. In the first group we have:

- *Student constraints*. Tasks can be restricted to a particular student profile. The model also allows applying tasks to students whose profile does not particularly fit the task at the cost of a penalty.
- *Knowledge constraints*. The task may require a certain competence level before realization. This mandatory prerequisite may come from the students' previous knowledge or from the acquisition through the execution of other learning activities.
- *Ordering constraints*. The model permits to specify a task must always be done before or after another task.

The planner manages the following task attributes when computing a customised plan:

- *Material.* The instructor describes the material to be used in a specific task. Material is classified into mandatory or optional (complementary). The mandatory material implies the learner must count on the personal resources to ensure he can pursue the task. Optional material is not a hard constraint but it may determine the competence level of the attained outcomes.
- *Exclusive resources*. The realization of a task may require the use of a resource in exclusivity (for instance, a lab). This way, only students who are doing that task can access the exclusive resource.
- *Cooperation/collaboration.* The teacher will be able to set a minimum/maximum number of students per task. This issue is particularly relevant for cooperative tasks. It is also possible to set an "ideal" number of students for the task ([min. students, max. students]).

2.3 Learning material, resources and time

These items are only considered when computing customised courses for a group of students in a particular teaching context.

Learning material is an important item for an instructional course as it enables learners to develop skills by progressing through a chosen theme. The necessity of specific learning material for a task may determine the necessity of specific personal resources; for instance, a video film will require a multimedia computer, a DVD player or a room with a video projector that the course organizing staff would make available to students.

Resources denote the available global resources to undertake a course, such as labs, printers or book lending service. The planner will consider, among others, the following attributes: the resource capacity (number of persons who can use the resource simultaneously), the quantity (number of available resource units) and the availability (time interval at which the resource is available).

Dealing realistically with *time* in e-learning may involve managing dates, days of the week, hours, etc. This factor brings a great complexity to the model so for the sake of simplicity we will just handle duration tasks as time intervals (this task lasts one hour and a half) or that a particular learning outcome must be attained before 40 hours from the course starting date.

3. Architecture

In this section, we present the overall architecture as a general scheme that shows the functionality of our approach (see Figure 1). First, the instructor interacts with an authoring tool to design all the elements that make up the course design. After this, a planner is invoked for a particular learner profile and the output will be a plan (learning route) that fulfils the learning goal for such a profile, which can be represented in the graphical tool or in a standard format. On the other hand, learners' role becomes more active when a particular customised plan for each learner is generated, calculated on the basis of the student profile and the particular constraints of the e-learning context.

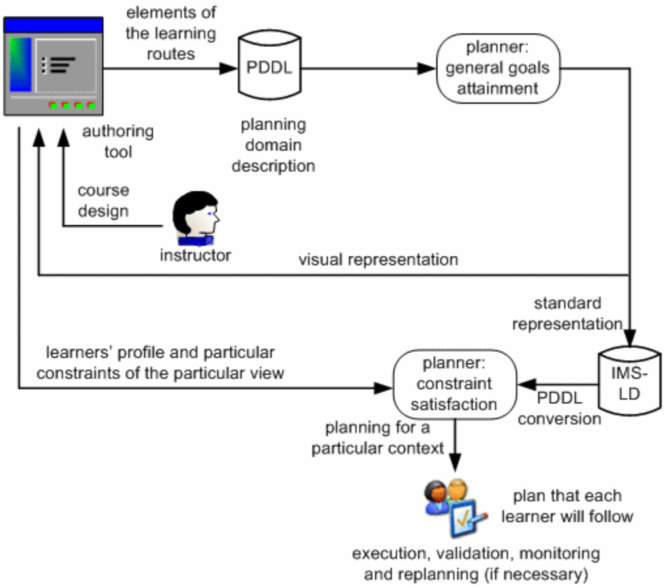


Figure 1: Basic architecture of our approach

3.1 Structure and workflow

The architecture is structured around four essential elements, 1) the instructor, 2) the authoring tool, 3) the planner, and 4) the learners, which interact in three high-level steps. In the first step, the instructor models all the key elements of the course design in a simple visual way by using the authoring tool, which allows the teacher to have a visual representation of the learning goals, tasks and their relations in the form of prerequisites and effects. This representation is later encoded as the planning domain description, which represents the domain knowledge. We use *PDDL* (McDermott 1998), a standard planning domain definition language for planners, to represent this knowledge.

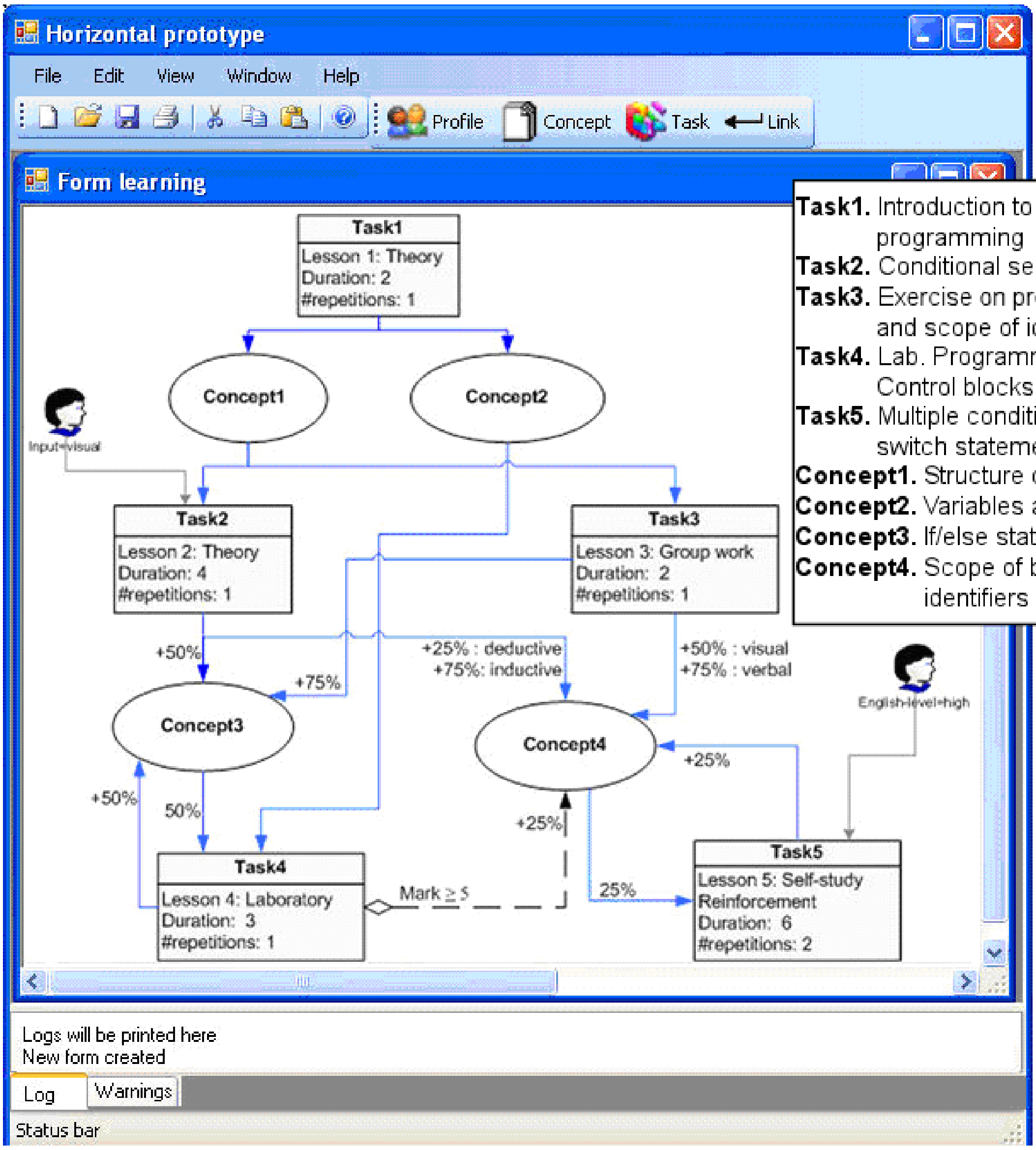
The second step focuses on the execution of the planner over this course design; the planner finds the learning route that satisfies the general goals without considering particular context constraints. This route can be visually represented in the tool or using a standard representation such as *IMS-LD* (IMSLD 2003). The third step focuses on the execution of the planner over a particular scenario; in this case, elements like the learners' profile and specific context constraints come into consideration to find a valid learning route, which must satisfy all constraints of each particular learner. Therefore, the task of the planner is similar to the task of the second step but now adapting the general learning route to take into consideration the number of students of each profile and the resources availability. Finally, when learners execute their learning routes, a monitoring stage to keep track of the tasks that have already been completed becomes necessary. This stage allows learners to map their position on the learning route and to determine what remains to be done to reach the goals of the route (Tattersall 2006). This might involve some re-planning when a learner does not fulfil the tasks as expected, e.g. when the learner does not pass a particular task.

3.2 Graphical authoring tool

The authoring tool, as shown in Figure 2, can be seen as a visual editor to assist the instructor on the definition of the elements introduced in Section 2. It offers a user-friendly easy-to-use interface; the instructor simply has to drag and drop the concepts, tasks and their relations, as the basic items necessary to model learning routes in a course design (Tattersall 2006):

- *Concepts*. In Figure 2, *Concept1* is a learning outcome (planning effect) of *Task1* and a prerequisite for *Tasks* 2 and 3. Different competence levels can be defined on the concepts, e.g. *Concept3* is achieved by *Tasks* 2 and 3 with increasing competence levels +50% and +75%, respectively. Particularly in Figure 2, *Task2* is only for learners with visual input profile, while *Concept4* is achieved at levels +25% and +75% for deductive and inductive organization profiles, respectively. On the other hand, *Task4* modifies the level of *Concept4* in +25% only in case the learner gets a mark greater or equal than 5 in such task.
- *Tasks.* Tasks have many attributes, such as resources or learning material, which are part of the model but have no graphical representation to keep the course design simple enough. Therefore, the tool only depicts graphically the flow of tasks that build the *learning routes*.

Clearly, different routes may be obtained by combining different profiles and conditions to be satisfied as long as the prerequisites are accomplished.



```
Logs will be printed here
New form created
```

Task1. Introduction to programming
Task2. Conditional sentences
Task3. Exercise on program blocks and scope of identifiers
Task4. Lab. Programming in Java. Control blocks: if/else
Task5. Multiple conditional: the switch statement
Concept1. Structure of a program
Concept2. Variables and types
Concept3. If/else statements
Concept4. Scope of blocks and identifiers

Figure 2: Snapshot of the GUI prototype of our tool. The elements shown are part of a simplified version of a programming course with 5 basic tasks and 4 concepts

The tool can also export the information about the course design to different standard formats, such as *PDDL* (planning language used by the planner to find an adequate plan, i.e. a learning route), *IMS-LD* (to represent the learning routes) and *XML* (to save both the visual information and the elements of the course design in a plain-text file).

3.3 Used techniques: planning and CSP

Planning is a search process aimed at finding a sequence of actions to accomplish a goal. A common approach to planning is representing the current state and determining the series of actions necessary to reach a goal state. In the problem we are addressing, the current state represents the learner characteristics (learning style and previous knowledge) and the goal consists of the acquisition of a certain competence level on some concepts.

The search process starts in the current state and the actions, which correspond to the learning tasks, are used to generate the list of possible successor states, until a state satisfying the goal-conditions is obtained. Each learning task can be easily translated to a *PDDL* action and each concept to a numeric variable. For example, the operator corresponding to *Task2* in Figure 2 is the following one:

```
(:action TASK2
   :parameters (?p - profile)
   :precondition (and (input-visual ?p) (>= (concept1 ?p) 100))
   :effect (and (increase (concept3 ?p) 50)
                (when (input-deductive ?p) (assign (concept4) 25))
                (when (input-inductive ?p) (assign (concept4) 75))
                (increase (duration) 4))
```

The action preconditions represent the student and knowledge restrictions for carrying out the task (a student with a visual learning style and a competence level of 100% on *Concept1*). As for the effects, *Task2* increases the competence level on *Concept3* +50% and +25% (+75%) on *Concept4* if the student is deductive (inductive). The task duration, 4 hours, is also modeled as an action effect.

For computing the general plans, all information about available resources and temporal constraints are not considered. Task durations are taken into account only for optimization purposes and they are represented through the numeric variable '*duration*'. With this representation, parallelism between tasks is allowed (provided that no causal links exist between them). Although, in practice, a student cannot carry out several tasks simultaneously, in the general view we permit this situation as we do not have enough criteria (resources, constraints, etc.) to establish a specific order. A general plan can be quickly obtained by using a general-purpose planner (Ghallab 2004) and it is useful for two reasons:

- First, the designer can validate the course model by checking that students with different profiles can achieve the learning goals through the defined tasks.
- Second, general plans are used as a starting point for obtaining the customised learning route for each student.

The final goal of our approach is to obtain a customised plan/learning route for each student, taking into account all resource and time constraints of the learning context. This task is very complex since it is necessary to schedule all activities of all students to obtain a consistent global solution. The hardest part in this task is the organization of collaborative tasks, where several students must work together to carry out a common task. This compels to establish some synchronization points in the learning routes of several students, thus losing the possibility of computing each learning route separately. In our approach, we compute a general plan for each student and then we try to coordinate them by using a *CSP* (*Constraint Satisfaction Problem*) solver (Liu 1998). A *CSP* allows the formulation of very complex constraints, so it is easy to model collaborative tasks and a wide range of resource and time constraints. If the *CSP* solver does not find a feasible solution, then it is necessary to identify the general plans that cause the conflict. Then, the planner computes alternative general plans for avoiding the conflict and the *CSP* solver is invoked again. When a solution is found, the learning route for each student is shown in the authoring tool (it is also possible to export them to the *IMS-LD* format). A learning route can contain paths that must be conditionally executed, depending on the results obtained by the student in an assessment point. The learning route is only shown until the first assessment point. Once the obtained result is available during the plan execution:

- If the result is the expected one, the following fragment of the learning route (until the next assessment point) is made available.
- Otherwise, a new plan is generated from the current state to solve this unexpected situation.

Section 4 shows a small application example that illustrates how customised plans are obtained for three students in the course example depicted in Figure 2.

4. Application example

This example is based on the course design shown in Figure 2. We suppose there are three students registered in the course with the following characteristics:

- *Student1*: deductive and visual learning style, with a high English level.
- *Student2*: deductive and verbal learning style, with a high English level.
- *Student3*: inductive and visual learning style, with a low English level.

The course goal is to achieve all concepts with a competence level of 100%, trying to minimize the required total time. We also define the following constraints for the course:

- *Task1* is an in-person class, so a lecture room is required. Since there are only three students, there are no problems with the available resources (number and capacity of the rooms), but all students must be present together during this task.
- *Task3* is a collaborative activity, so the three students have to work together at the same time.

The first step to obtain the customised plans is to generate a general plan for each student separately. These plans are shown in Figure 3. As it can be observed, general plans are a set of partially ordered tasks: arrows represent the order relationships between tasks. For *Student1*, for example, *Task2* and *Task3* must be executed necessarily after *Task1*, and *Task4* requires that *Task2* or *Task3* had been executed before (in this case, only one of the two tasks is required). General plans can also be used for validating the course design. In Figure 3, we can observe that for *Student2* there is not a feasible learning route if the mark obtained in the *Task4* assessment is below 5.

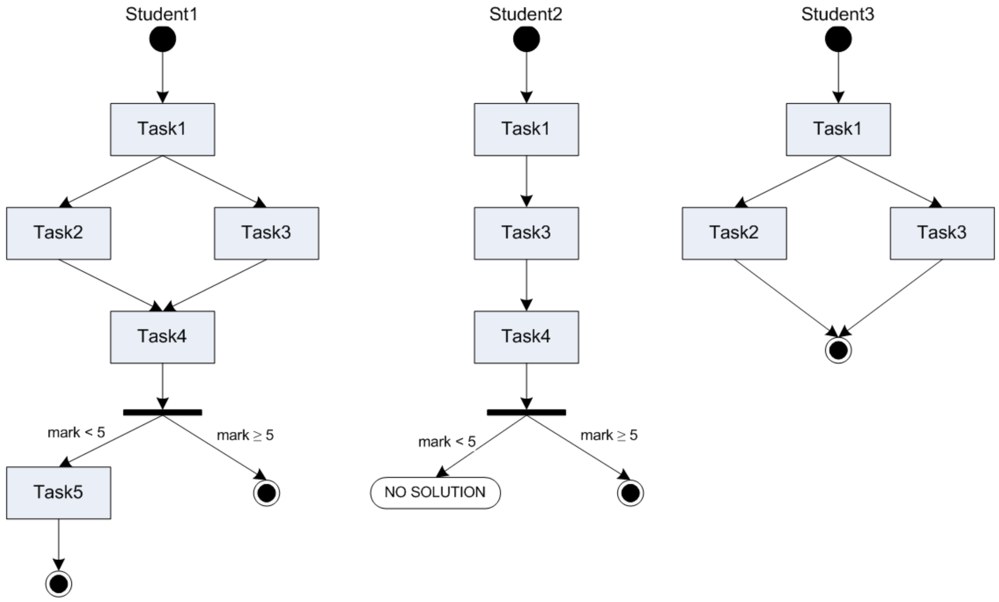


Figure 3: Individual general plans for each student

The next step is to use a *CSP* solver to schedule the general plans in order to meet the defined context constraints. The resulting customised plans are shown in Figure 4. On the left, we can see the full customised plan for *Student3*, and the customised plans for *Student1* and *Student2* until the time instant 7 (*t* = 7), which is the first assessment point. It can be observed that *Task1* and *Task3* are executed at the same time by all the students in order to satisfy the task's constraints.

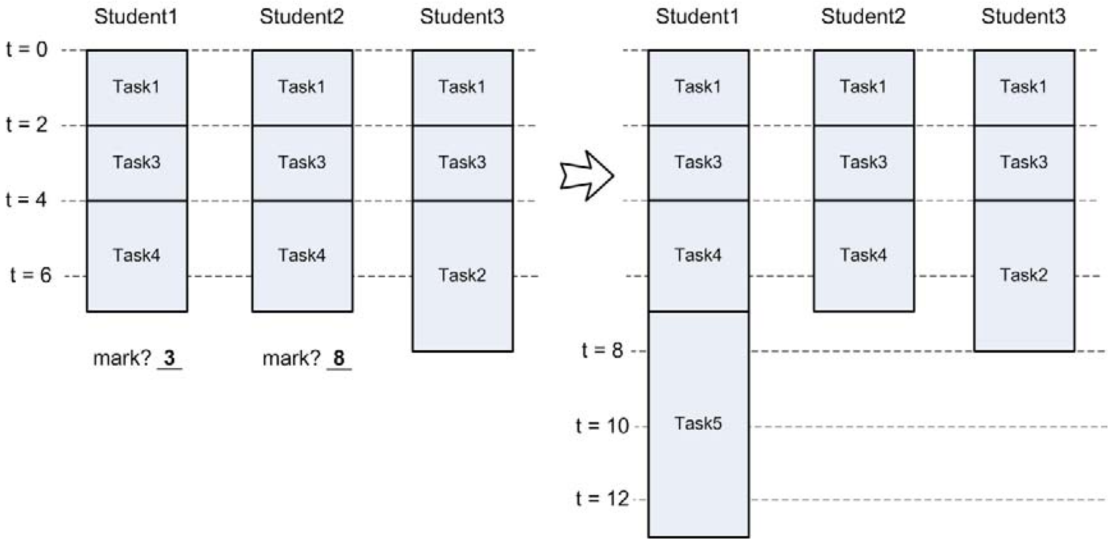


Figure 4: Customised plans for each student

Once the marks of students 1 and 2 for *Task4* are known (3 and 8 points out of 10, respectively), the rest of the customised plans are displayed (see the right side of Figure 4). As *Student1* has not passed *Task4* satisfactorily, he/she must carry out *Task5*, which is a reinforcement activity.

5. Conclusions

In this paper we have presented a novel approach for the automated design of learning routes using AI planning techniques. Our architecture is composed of two main modules, a graphical authoring tool and an automated planner. The authoring tool is intended to guide the teacher in the instructional design, providing the necessary elements to break the new learning goals into tasks and laying out the tasks in some logical order. The tool can be used by any instructor without any special skills on *IMS-LD* or any particular language syntax. Additionally, the tool makes possible the interoperability by simply providing one course design that can be used many times in different particular contexts, e.g. an instructor defines the course elements necessary to attain a competence level for a learning goal and the same course definition can be used in different terms and universities, which offers a very appealing way to share e-learning experiences among educational institutions.

The automated planner is applied over the data defined in the course design and obtains two different types of learning routes: a generic learning route for a specific student profile or a customised course for a group of people to be delivered in a concrete learning context. To our knowledge, this is the first educational system that considers the particular constraints of the learning-teaching context, thus generating a fully realizable course.

Acknowledgements

This work has been partially supported by the Spanish government project MCyT TIN2005-08945-C06-06 (FEDER), by the Valencian government project GV06/096 and by the Universidad Politecnica de Valencia under the PAID-04-07 programme.

References

Felder, R.M. and Silverman, L.K. (1988) "Learning and Teaching Styles in Engineering Education," Engr. Education, 78(7), 674-681.

Ghallab, M. and Nau, D. and Traverso, P. (2004) “Automated Planning. Theory and Practice”. Morgan Kaufmann.

IMSLD (2003). “IMS Learning Design Specification. Version 1.0 (February, 2003)”, *available at http://www.imsglobal.org/learningdesign (accessed June 2007)*.

Karampiperis, P. and Sampson, D. (2004) “Adaptive instructional planning using ontologies” 4th IEEE International Conference on Advanced Learning Technologies, pp.126-130.

Liu, Z. (1998) “Algorithms for Constraint Satisfaction Problems (CSPs)”, *Math thesis*, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.

McDermott, D. et al. (1998) “PDDL - The Planning Domain Definition Language”, *AIPS-98 Planning Competition Committee*.

Peachy, D.R. and Mc-Calla, G.I. (1986) “Using planning techniques in intelligent tutoring systems”. International Journal of Ma-Machine Studies, 24, pp. 77-98.

Smith P.L. and Ragan T.J. (2005) “Instructional design”. (3rd edition), Hoboken, NJ:Wiley.

Staller, A. (2006) “Merging domain knowledge and task analysis in an ontology”. Current Developments in Technology-Assisted Education, pp. 1585-1589.

Tattersall, C., Janssen, J., van den Berg, B. and Koper, R. (2006) “Modelling Routes towards Learning Goals”, *Campus-Wide Information Systems*, Vol 23, No. 5, pp 312–324.

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

Vassileva J. (1997) “Dynamic Course Generation”. Communication and Information Technologies, 5(2), pp. 87-102.

Vrakas D. et al. (2007) “PASER: A Curricula Synthesis System based on Automated Problem Solving”. International Journal on Teaching and Case Studies, Special Issue on "Information Systems: The New Research Agenda, the Emerging Curriculum and the New Teaching Paradigm", Inderscience, Vol. 1, Nos. 1/2, pp. 159-170, 2007.