Planning for Conditional Learning Routes

Lluvia Morales, Luis Castillo, and Juan Fernández-Olivares

Department of Computer Science and Artificial Intelligence, University of Granada
Daniel Saucedo Aranda s/n 18071 Granada, Spain
{lluviamorales,L.Castillo,faro}@decsai.ugr.es

Abstract. This paper builds on a previous work in which an HTN planner is used to obtain learning routes expressed in the standard language IMS-LD and its main contribution is the extension of a knowledge engineering process that allows us to obtain conditional learning routes able to adapt to run time events, such as intermediate course evaluations, what is known as the standard IMS-LD level B.

Keywords: Planning and Scheduling, Automatic Generation of IMS-LD level B.

1 Introduction

The e-learning institutions that offer large scale courses tend to provide tens to hundreds of courses that have to be delivered to thousands of students grouped into classes [1]. For this reason, the e-learning community has created a set of standards that allows, not only the reuse of educational resources, but also the design of personalized learning sequences to introduce the concepts of the course. So that, each of them could exploit at maximum the course.

This paper is based on a standard known as Learning Design or IMS-LD [2] which describes as many custom sequences of educational resources from a single online course as the number of students profiles on it. These sequences vary depending on students' features, but also according to a set of conditional variables and its possible values, e.g. evaluations that are made to determine the level of knowledge about a given concept. Hence, the generation of a IMS-LD by hand is expected to be too expensive in both time and human effort.

One of the most effective solutions to carry out this task is by using intelligent Planning and Scheduling (P&S) techniques[3]. There are several approaches in the literature that are discussed later, but none of them takes advantage of all the IMS-LD potential, particularly the possibility of producing conditional learning sequences, what is the main concern of this paper.

Our previous work [4] describes a knowledge engineering process applied to several data encoded through e-learning standards that allows to automatically generate a planning domain and problem. These planning domain and problem describe all personalization requirements of a course and they are used by the hierarchical planner [5] to generate a IMS-LD level A. However, this process does

A. Hernández Aguirre et al. (Eds.): MICAI 2009, LNAI 5845, pp. 384–396, 2009.
© Springer-Verlag Berlin Heidelberg 2009

not include the generation of domains and problems that allow us to generate a level B IMS-LD with conditional sequences.

Throughout this article we first give a brief introduction to the related e-learning terminology and to the planner we use to address the problem to generate a conditional learning design. Later we focus on defining a methodology that will allow us to solve that problem through modifications to the initial architecture and to improve the planning domain and problem generation algorithms. Results of experiments conducted with the planner [5] will be described in a special section. Methodology and experiments will be highlighted, discussed and compared with other current existing proposals, to finally arrive to the conclusions.

2 Preliminaries

This section shows some key concepts used in the e-learning community, and its relation with some concepts of P&S that are used in the next section. It also describes the main features of the planner which interprets the hierarchical domain and problem generated by the methodology presented in this article.

2.1 E-Learning Standards and Tools

Large scale on-line courses are usually provided through a *Learning Management System*(LMS) that includes not only the descriptive representation of the course through nested concepts by using educational resources for their full understanding and practice, but also the timing of the course, the application of evaluations, and students' features.

The *educational resources* of an online course are all those images, documents, Flash, Word, PDF, HTML, etc... with which a student interacts in some way to understand the concepts of the course, while the *concepts* are the representation of the chapters, lessons and units that can be organized in a hierarchical structure that defines educational resources at the lowest level.

The e-learning community has adopted several standards based on XML to represent features of educational resources and concepts of an online course, as well as the entities that interact with them such as questionnaires and students. The following points give a brief description of these standards that belong to the family of standards approved by IMS Global Learning Consortium[2]:

- Learning Resource Metadata (IMS-MD) is a standard that allows us to define, by default metadata values, the features of an educational resource. It is usually edited in a tool called RELOAD[6] and it is of vital importance to define the order and hierarchy between nested concepts like chapters, subchapters, lessons, or units and educational resources in a course. Certain features and requirements of these resources as optionality, duration, resource type, language, difficulty, media resources required, etc. are also described by the standard.
- Learner Information Package (IMS-LIP) integrates every students' profile data in a single XML document through nested labels as identifier, name, preferences, competencies, etc.

- Question and Test Interoperability (IMS-QTI) standard is a XML document that represents an evaluation and its possible outcomes, besides some rules to assess the knowledge level achieved by students.
- Learning Design(IMS-LD) standard is the main tool for designing the sequence of learning activities of a course. It helps to represent the interaction between different structures or sequences of learning activities, also called educational resources, and the different roles that bring together students from a course with similar profiles.

The Learning Design can be defined from three levels of abstraction A, B and C, where each level incorporates and extends the capabilities of the previous one. Our work aims to extend the generation of IMS-LD level A capabilities based on HTN planning techniques, already done in [4], to an IMS-LD level B which extends the ability of generate personalized learning sequences with a run-time dynamic that supports properties and conditions like intermediate evaluations.

It is also worth to mention that each course element has its equivalent within the hierarchical planning paradigm, and that it is possible to obtain relevant information to define domain and planning problems from e-learning standards.

Educational resources are represented as *durative actions* in a planning domain, while concepts and their hierarchical structure are equivalent to the *tasks*. These tasks and actions are described by the IMS-MD standard that has metadata which allows to define the duration and conditions for implementing an action, in addition to the preconditions for implementing the methods of a task and their relationships of order and hierarchy.

On the other hand, the IMS-LIP standard helps to define the initial state of a problem by representing the features of a student as predicates and/or functions that are intimately related to certain conditions and preconditions of the domain defined by IMS-MD. While the IMS-QTI standard defines an evaluation that is considered a goal within the scope of planning. These planning domain and problem structures used by the planner are described in next section.

2.2 HTN Planner

Hierarchical planners are used for most of researchers who have worked over the sequencing of educational resources. [7], [8], and [4] they agree that this type of planning allows to easily obtain a typical course concepts hierarchy without neglecting those courses that have a plain concepts structure.

The planner we use as reference for modeling the problem and domain of this work is described in [5], called SIADEX. This planner follows the hierarchical task network formalism established by SHOP2 in [9] which is based on a forward search procedure, that involves keeping the problem state during the search process. Primitive actions used by the planner are fully compatible with PDDL 2.2 level 3, and support durative actions.

Creators of this planner have defined an extension to PDDL that allows to represent hierarchical tasks and their methods within a related temporal framework that inherits their temporal restrictions to its child actions. This permits to determine deadlines for the plan execution, and other features of the planner [5]. The methodology to obtain a level B learning design by using this planner is explained in the following section.

3 Methodology

As mentioned above, in order to generate an IMS-LD level B, conditional values should be added to our previous approach, e.g. intermediate evaluations that may change during the execution of the course.

Unlike the previous proposal which generated an overall plan for the course and each student, now it will be necessary to generate as many plans as possible evaluations and outcomes that are defined for each student profile, where each profile groups the students of a course with the same features and requirements.

The mid-term evaluations of the course concepts and their possible outcomes are described by an ordered sequence of IMS-QTI documents that content the goals of a course. Moreover, each document defines particular satisfaction levels, with a range of scores related, and rules to change the requirements of a student according to his/her score in the evaluation and the level it comes in.

Furthermore, each evaluation has a main concept or task related which is fully defined by a instructor using the IMS-MD standard. This main task is composed of subtasks and actions, with relations and requirements also described by IMS-MD. These tasks have a hierarchical structure between them and must be carried out in sequence, in order to understand the main concept being taught.

Our approach takes the set of profiles and the sequence of IMS-QTI goals and follows an iterative process. For every profile, we iterate over the set of goals. Every new goal generates a planning problem, and all the resulting plans for each goal and every profile are increasingly added to the final IMS-LD.

However, to generate each of these IMS-LD "drafts" it will be necessary to design a robust hierarchical domain and planning problem able to support the planning of all the sequences of actions that could meet the goal in question. The amount of these sequences is determined, not only by the number of different profiles of students who take that course, but also by the conditions attached to the outcomes of the immediate evaluation prior to the evaluation being planned.

An IMS-LD level B includes conditions related to some properties of the course that can take different values at run time. We have used the process of applying intermediate evaluations as an example, because an evaluation can take different values at runtime according to the score obtained for each student, dynamic features of the student like the language or activity level.

In our example, possible score ranges of evaluations are used to define sequences of educational resources that are showed later under some conditions, which are defined during the planning process included in the architecture described in figure 1. This figure shows the process to generate an IMS-LD level B, summarized into the next three stages:

1. The first step is to get all the information provided by educational standards stored in a LMS called Moodle, through an XML-RPC web service.

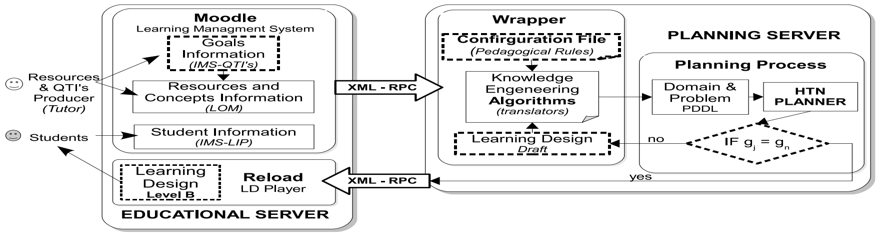

Fig. 1. IMS-LD level B generation process. The features which extend our previous work are highlighted by dotted boxes.

2. In the second stage a translation of the information obtained in first step is made through knowledge engineering algorithms and pedagogical rules described in a configuration file, in order to obtain a planning domain and problem, which satisfy the contents of the corresponding intermediate objective or goal. Sequences and conditions in IMS-LD "draft", if are not translating the first objective, are recovered and included in the planning domain.
3. Finally planning of sequences and conditions defined through the execution of a hierarchical planner is carried out to obtain a new learning design which includes these sequences and conditions, together with the previous ones. At the end of this stage, if the planned goal is to meet the objectives to make the last evaluation of the course, then the learning design is considered to be complete and can be shown through the RELOAD tool to the students of the course according to their profile. Otherwise, we understand that the learning design is incomplete and we will overwrite the last IMS-LD draft with this new one, and return to the second step.

In order to formalize, simplify and understand the previous process, a set of concepts have to be defined first.

Having $St = \{st_1, st_2, ..., st_l\}$ as a set of students in a course and $\mathcal{G} = \{g_1, g_2, ..., g_n\}$ as an ordered set of goals that must be met to satisfy the objectives of a course that are related to its evaluations.

Where p_i is a set that groups the students of a given course with the same features and requirements. Let us consider that for each profile p_i there are n goals $(g_1, g_2, ..., g_n)$ or intermediate evaluations to accomplish, according to the number of documents under the IMS-QTI standard that are linked to the course.

The outcome of each evaluation may be a continuous value ranging in the possible set of scores $[S_{min}, S_{max}]$. However, for practical reasons, instructors do not adapt to every possible outcome, but just for a discrete set of intervals.

Let us consider that instructors only distinguish h different levels of scores, then, they only take into account scores in the following intervals $[limit_1, limit_2)$, $[limit_2, limit_3), [limit_3, limit_4), ...[limit_{h-1}, limit_h]$ where $limit_1 = S_{min}$ and $limit_h = S_{max}$. Therefore, the set $\mathcal{L} = \{\mathsf{l}_{j,1}, \mathsf{l}_{j,2}, ..., \mathsf{l}_{j,h}\}$ describes h satisfaction levels related to each goal g_j.

On the other hand, for each level $l_{j,k}$ certain changes on the features of profile p_i have to be made in order to adapt it to the new profile performance and needs.

For example, if a satisfaction level of goal g_j is in the range of scores with limits [0,100] and there are 3 satisfaction levels defined within intervals [0,50), [50,80) and [80,100], then each level must require certain changes in some features of each profile p_i, such as the following:

If g_j`-level >=` *80 then*
  `performance-level` *of* p_i *turns High.*
If g_j`-level <` *80 and* `>=` *50 then*
  `performance-level` *of* p_i *turns Medium and actions with High* `difficulty` *must be required.*
If g_j`-level <` *50 then*
  `performance-level` *of* p_i *turns Low and actions with* `optional` *features must be required.*

On the other hand, a course $\mathcal{C} = \{(g_1, \mathcal{SC}_1), (g_2, \mathcal{SC}_2), ..., (g_n, \mathcal{SC}_n)\}$ is defined as a set of main tasks $\mathcal{SC}_j$ that are directly related to each goal in a course.

It is important to recall that each main task $\mathcal{SC}_j$ is compound of tasks and actions with their own relations and requirements. These actions are organized in a hierarchical structure and must be realized in sequence in order to understand the concept of the main task and accomplish its related goal g_j. Hence, the definition of a set of sequences $\mathcal{S}_j = \{s_{j,1}, s_{j,2}, ..., s_{j,h}\}$ is needed to understand that each main task $\mathcal{SC}_j$ has to be sequenced on h different ways for each profile p_i in order to accomplish each goal g_j.

That h number of sequences corresponds to the number of satisfaction levels on the previous goal g_{j-1}. It means that we have to plan sequences for each possible outcome that a student could have been obtained at the previous evaluation, because features of his/her profile may change in accordance, and next sequences must be adapted to any of these changes.

In the domain generation process some conditions have to be defined in order to select the best sequence $s_{j,k}$ that can accomplish the goal g_j for each profile p_i, according to its possible outcomes in g_{j-1}.

Figure 2 shows that the first adapted sequence for every profile is unique but, since intermediate evaluations are being considered, the sequence is split into h different branches, a branch for each of the h possible levels of the previous test. So, as one can imagine, the first draft of a learning design does not include

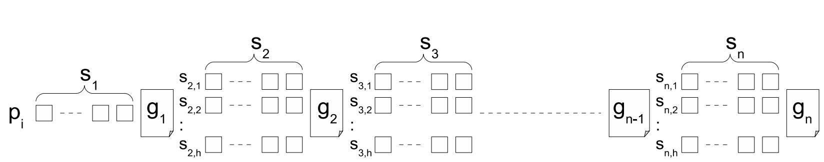

Fig. 2. Formalization of the planning process, where p_i is a profile, $\mathcal{S}_j$ is a set of adapted sequences of educational resources derived from the description of the set $\mathcal{SC}_j$, $s_{j,k}$ are the sequences of set $\mathcal{S}_j$ that are planned according to changes in the profile p_i based on the levels $l_{j,k}$ of the goal g_j that must be satisfied by any of these sequences

```
(:task Multiplication
  :parameters(?profile - profileId ?score - goalSatLevel)
 (:method Theoretical
  :precondition(learning-style ?profile ?score theoretical)
  :tasks(
    (task_addtion                        ?profile ?score)
    (action_multiplicationFormalization  ?profile ?score)
    (action_multiplicationSimulation     ?profile ?score)
    (action_multiplicationExperiment     ?profile ?score)))
 (:method Pragmatic
  :precondition(learning-style ?profile ?score pragmatic)
  :tasks(
    (task_addition                       ?profile ?score)
    (action_multiplicationExperiment     ?profile ?score)
    (action_multiplicationSimulation     ?profile ?score)
    (action_multiplicationFormalization  ?profile ?score))))
```

```
(:durative-action action_multiplicationExercise
  :parameters(?profile - profileId ?score - goalSatLevel)
  :duration(= ?duration 10)
  :condition( and
    (equipment ?profile ?score multimedia)
    (>= (language-level English ?profile ?score) 50))
  :effect(additionExercise ?profile ?score done)
)
```

Fig. 3. PDDL action and task examples

conditions for a same profile. Therefore, the way to design the hierarchical domain and planning problem, in order to generate the sequences for this design, is substantially different from what we will see in the following sections.

The algorithms to generate a hierarchical planning domain and problem are detailed in the following subsections, but it is worth noticing that only algorithms to generate the domain and problem that allow to plan sequences that meet from g_2 onwards are described because the domain and problem to meet the first goal is similar, if not equally, of that detailed described on [4].

3.1 Domain Generation

A planning domain, that describes features and relationships of the elements of a course, is generated once for every goal g_i of $\mathcal{G}$ and its generation is based on a knowledge engineering process on the features of tasks and durative actions that conforms the main task $\mathcal{SC}_i$ to be planned to meet the goal g_j.

The algorithm to implement the domain generation is the following:

1. All the features of elements that conform the main tasks $\mathcal{SC}_j$ are analyzed, dividing them into two subsets: tasks and durative actions.
2. Durative actions are generated according to its features and relations, obtaining its durations and execution conditions, in addition to a pair of auxiliar actions that helps us to retrieve the elements of IMS-LD draft.
3. Tasks and actions that are part of the same main task $\mathcal{SC}_j$ are ordered according to their hierarchy relations or pedagogical rules described in a configuration file.
4. Finally the domain tasks are defined and its conformant subtasks assigned to the appropriate methods according to the order obtained previously.

For example, elements of the main task $\mathcal{SC}_j$ which represent the concepts to accomplish, are placed in the tasks subset, and those that represents educational resources in the durative actions subset.

Durative actions definition described in step two is carried out as in figure 3 with the `action_multiplicationExercise` example encoded in PDDL.

Another two actions must be described to retrieve previous sequences and conditions of the IMS-LD draft. Those are:

- `(initAction)` that recalls from the learning design draft the profiles definition set $\mathcal{P}$, the components of past main tasks, and the sequences planned for those main tasks. Also, the property for the goal g_{j-1} is added and initialized.
- `(endAction)` that recovers the rest of the elements of IMS-LD draft. It is also used to define new conditions according to the values that can take the property for the goal g_{j-1} and the sequence $s_{j,k}$ that must be showed to each profile for the corresponding level $l_{j,k}$ part of that goal.

An example of a condition that can be inserted into the new IMS-LD through `endAction` is the following one:

```
if test-score-of-goal_{j-1} > 79
  show activity-structures of every profileId with score=90 and goal_{j-1}
else
  hide activity-structures of every profileId with score=70 or score=30, besides goal_j
```

Where, when the property value test-score-of-$goal_{j-1}$ is found in execution time at level $l_{j,3}$, corresponding to the [80, 100] interval, the corresponding sequence planned for goal g_j and score 90 will be showed, and sequences generated for the same goal but another levels will be hidden.

Finally, results of the steps three and four are shown in figure 3, where the main task `Multiplication` is encoded in PDDL with two decomposition methods for different order of durative actions, and where the task `addition`, which is related with the *Required* relation to `Multiplication`, is the reason of its first place in the order of both methods.

Through the algorithm explained by the above examples it is possible to define a hierarchical planning domain that completely describes tasks and actions in it, to facilitate the planning process for a main task $\mathcal{SC}_j$ of the course $\mathcal{C}$.

The main difference between this generation process and previous approaches is summarized in the next two points:

- The domain is generated as many times as goals in the course, not only once for the entire course as in the previous proposal.
- Adding a pair of auxiliar actions will be possible to retrieve the information of IMS-LD draft and add information resulting of the current planning process to a new learning design.

After generating the domain will be necessary to design a planning problem. The process for creating that problem is described in following section.

3.2 Problem Generation

A hierarchical planning problem, as already mentioned above, is generated as many times as the number of elements in the set of goals $\mathcal{G}$ derived from the IMS-QTI standard.

The generation of each planning problem, to goal g_2 onwards, is divided in two steps. The first one is the initial state definition and the second of the goals that must be meet in order to satisfy the current problem.

The initial state of a planning problem for goal g_j includes a description of all the profiles in the set $\mathcal{P}$, i.e. the specification of the features of each profile by means of predicates or functions in PDDL language.

But since the features of a profile are subject to change based on the results of an intermediate evaluation, the process of generation of these initial states is a bit more complex than that described in the preceding paragraph. This process first creates as many different copies of the profiles in $\mathcal{P}$ as levels of satisfaction to the goal g_{j-1}, which will generate mxh copies that allow us to further define the same number of "initial states" each different profile-level.

```
;
; profileOne with score 90
; (level between 80 and 100)
;
(= (performance profileOne 90) 90)
(learning-style profileOne 90 theoretical)
(= (availability profileOne 90) 1150)
(equipment profileOne 90 multimedia)
;Academic trajectory
(= (language-level en profileOne 90) 90)
(= (mark profileOne 90 requiredSequence) 80)
(= (mark profileOne 90 prevOptionalActions) 30)
(= (mark profileOne 90 prevDifficultActions) 80)
```

```
(:tasks-goal
  :tasks [
  (initAction)
  ( and ( <= ?end 1150 )
        (Multiplication profileOne 90) )
  ( and ( <= ?end 1750 )
        (Multiplication profileOne 70) )
  ( and ( <= ?end 2150 )
        (Multiplication profileOne 30) )
  :
  (endAction)
 ])
```

Fig. 4. PDDL initial state and goals

The definition of these copies is made because the profile of a student may change according to the results obtained in the different intermediate evaluations to be applied, and hence, personalized sequences of that profile might change too. Therefore, copies are essential to define not only the initial state, but also the goals of the problem.

Once profile copies are generated, the initial state of the problem has to be created through the assignment of a representative score value to each copy, according to its corresponding level, and the definition of its features using functions and predicates.

The PDDL profile in figure 4 describes features of the profile p_1 with score 90, meaning that its satisfaction level is in the [80,100] range which is in $l_{j,3}$ level. Therefore, its acquired features from IMS-LIP have been modified according to that level. This example shows 5 functions and 3 predicates that describe several features of the profile. Other 14 features definitions should be described in the initial state of the problem to obtain all the necessary profile copies.

The second stage consists in defining the goals of the planning problem. Thus, for each profile copy described in generation process of the initial state for planning problem, a conditional action like `(and (<= ?end availability) (S_jMainTask profile_i level_k))` must be stated as a goal.

The above predicate refers to tasks and actions that belongs to the main task `S_jMainTask` that must be planned for p_i profile according to its features for the copy in $l_{j,k}$ level. The main task duration should not exceed that allocated for `?end` variable according to the availability of this profile copy.

An example of goals definition is shown in figure 4. Where the main task $\mathcal{SC}_j$ has a related time that is accumulated in the variable `?end` according to the duration of the actions and tasks that conform it. This time should not exceed the time established by the goals conditions for that variable.

As we can see, defined goals, to accomplish the main task for each profile, and level are between `initAction` and `endAction`. Execution of both actions can retrieve information from IMS-LD draft and add conditions for the new sequences that must be generated between them.

Table 1. Execution times of each case $\mathcal{SC}_j$ during the two phases of our experiment

Resources Sets for goals one to four	Case 1	Case 2	Case 3	Case 4	Average Time (sec.)
Domain & Problem Generation	1.22	1.11	1.19	1.16	1.17
Planner Execution	0.99	0.84	0.97	0.92	0.93
D&P Generation + Planner Exec Time (sec.)	2.21	1.95	2.16	2.08	**2.1**

Through this process it is possible to generate a hierarchical planning problem for each set of tasks and durative actions related to a goal. This problem must be robust enough as to require the creation of all the possible sequences for every profile in $\mathcal{P}$ and the different levels that can obtain the previous goal g_{j-1}.

This ends the description of the process to generate an IMS-LD level B described at the beginning of this section and will be tested through a series of experiments described in the next section.

4 Experiments

Once we have defined the methodology for the generation of IMS-LD level B, we present a set of experiments to show how this approach works in practical cases.

Description. The experiments use 4 representative examples of $\mathcal{SC}$ that show different scenarios for the application of our approach, e.g. Math or AI subjects. Each $\mathcal{SC}$ is composed of 8 basic tasks and 30 durative actions on average. These durative actions have an average duration of three hours.

We also considered students automatically classified in an average of 20 students' profiles according to 6 different features. Therefore, considering that we have divided scores into 3 different subsets, the planner has to solve an average of 60 goals for each problem.

Over every set $\mathcal{SC}$ we applied the algorithms for the automatic generation of domain and problem planning files, which were passed to our planner to obtain a fully standard IMS-LD level B able to be handled by tools like RELOAD.

Results. As we may see in Table 1, the average running time to generate every domain and problem file for all the sequences of the four sample sets was 1.17

seconds. Then, our planner generates all the plans for every $\mathcal{SC}$ in 0.93 seconds average. All these plans for every subset $\mathcal{SC}$ are joined into a standard IMS-LD level B document in about 2.1 seconds.

Discussion. In [1] the effort needed for an instructor to design a customized learning sequence is reported to take 20 hours of design for every hour of teaching. This effort does not consider the time needed to translate the design into standard IMS-LD.

In our case, a customized learning design, already expressed in standard IMS-LD level B, is obtained in 0.7 seconds average for every hour of teaching. Given that the labeling of learning resources with metadata and the design of intermediate tests is assumed in both cases and that there is a very small overload due to the introduction of the planner, the advantages of this approach are extremely clear.

However, there are other approaches that are intended to solve the same problem and that are commented in the next section.

5 Related Work

The standard IMS-LD first draft arose 6 years ago. During the first four years, there were no more than 40 courses encoded in IMS-LD, most of them written in the lowest level of expressiveness, the level A.

After this, several projects to ease the process of writing these learning designs at its level B[3,10] have been reported. One of them is found in [3] where an assistant is created to guide instructors in the writing of the different sections of a standard IMS-LD for given courses. However, this process is still very slow and requires the instructor to know the detailed syntax of the standard.

We also explored the possibility of using a conditional planning approach like in [11]. Our HTN planner does not support planning with conditional branches, but, since it introduces an unnecessary complexity in the planning process making the planner to branch the plan at every intermediate test, it was finally discarded since the approach presented in this paper is much easier, achieving the same results.

On the other hand, the continual planning approach of [12] can be an option for course sequencing with mid-term evaluations because sequencing could be done by generating plans for accomplish a goal at runtime according to the outcomes of the previous test. But, it is not a good approach for our problem because the IMS-LD must be completely generated before the course starts.

Finally, we also considered a plan fusion approach like in [13], but it was also discarded because the main complexity of plan fusion comes from the detection and bounding of subplans able to merge from one plan to another, and in our case they are very easily bounded by intermediate tests or well defined features of IMS-LIP.

6 Conclusions and Future Work

In summary, the approach presented in this paper is based on a former proposal [4] in which domain and problem files are automatically generated for an HTN planner to obtain customized learning designs for a given course expressed in the standard IMS-LD level A. Its main contribution is an extension of this approach able to handle with conditional sequences supported by IMS-LD level B, given the intermediate evaluations example, and the subsequent generation of conditions and branches able to adapt the learning design to a discrete set of possible outcomes of these evaluations besides the customization to the features of each student profile.

Thanks to the expressiveness and the efficiency of the planner [5] we have been able to automate the generation of the domain and problem files, needed to accomplish each goal, avoiding the need for a planning expert and increasing the independence of instructors to use the available AI planning technology.

However, there are a couple of issues that still remain open.

The first one, strictly falls within the scope of e-learning institutions and requires education institutions to increase the number of learning resources and courses described under the existing standards.

And the second one relates to the exploitation of the temporal planning capabilities of our HTN planner [5]. These capabilities would have permitted the parallelization and synchronization of sequences among different profiles, opening the door to the modeling of shared or collaborative activities, but this has been left for a future work.

References

1. Brusilovsky, P., Vassileva, J.: Course Sequencing Techinques for Large-Scale Web-Based Education. International Journal Continuing Engeenering Education and Lifelong Learning 13(1/2), 75–94 (2003)
2. OUNL: Ims Global Learning Consortium. Open University of the Netherlands (2003), `http://www.imsglobal.org/`
3. Heyer, S., Oberhuemer, P., Zander, S., Prenner, P.: Making Sense of IMS Learning Design Level B: From Specification to Intuitive Modeling Software. In: Duval, E., Klamma, R., Wolpers, M. (eds.) EC-TEL 2007. LNCS, vol. 4753, pp. 86–100. Springer, Heidelberg (2007)
4. Castillo, L., Morales, L., Fernández-Olivares, J., González-Ferrer, A., Palao, F.: Knowledge Engineering and Planning for the Automated Synthesis of Customized Learning Designs. In: Borrajo, D., Castillo, L., Corchado, J.M. (eds.) CAEPIA 2007. LNCS (LNAI), vol. 4788, pp. 40–49. Springer, Heidelberg (2007)
5. Castillo, L., Fernández-Olivares, J., García-Pérez, O., Palao, F.: Efficiently Handling Temporal Knowledge in an HTN Planner. In; ICAPS 2006, pp. 63–72 (2006)
6. Liber, O.: REusable eLearning Object Authoring and Delivery. University of Bolton (2007), `http://www.reload.ac.uk/`
7. Kontopoulos, E., Vrakas, D., Kokkoras, F., Bassiliades, N., Vlahavas, I.: An Ontology-based Planning System for e-course Generation. Expert Systems with Applications 35(1/2), 398–406 (2008)

8. Ullrich, C., Melis, E.: Pedagogically Founded Courseware Generation Based on HTN-planning. Expert Systems with Applications 36(5), 9319–9332 (2009)
9. Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J.W., Wu, D., Yaman, F.: Shop2: An HTN Planning System. Journal of Artificial Intelligence Research 20(1), 379–404 (2003)
10. Sicilia, M., Sánchez-Alonso, S., García-Barriocanal, E.: On Supporting the Process of Learning Design through Planners. CEUR Workshop Proceedings: Virtual Campus 2006 Post-Proceedings 186(1), 81–89 (2006)
11. Onder, N., Pollack, M.E.: Conditional, probabilistic planning: A unifying algorithm and effective search control mechanisms. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 577–584 (1999)
12. Myers, K.L.: Towards a framework for Continuous Planning and Execution. In: AAAI Fall Symposium on Distributed Continual Planning (1998)
13. Foulser, D., Li, M., Yang, Q.: Theory and Algorithms for Plan Merging. Artificial Intelligence 57, 143–181 (1992)