VSP International
Science Publishers
P.O. Box 346, 3700 AH Zeist
The Netherlands

*Lecture Series on Computer
and Computational Sciences*
Volume , 2006, pp. -

AI techniques for Automatic Learning Design

Maria D. R-Moreno[1] and David Camacho.
Departamento de Automática. Universidad de Alcalá.
Alcalá de Henares (Madrid), Spain. E-mail: mdolores@aut.uah.es

Computer Science Department. Universidad Autónoma de Madrid.
Madrid, Spain. E-mail:david.camacho@uam.es

Abstract: This paper presents a new approach to the problem of control and monitoring Learning Design courses. Our approach uses the integration of AI planning and scheduling as the main solving/reasoning process. These techniques are used to solve some problems in a particular Learning Design. Those problems will be detected from the educators/students interactions, and it will be necessary to map both, the metadata provided by the educators, and the results obtained from previous interactions, into an appropriate representation that could be used by any planner (and/or scheduler) to reason about plans. This paper describes how AI planning and scheduling techniques can help in detect and solve problems in the Learning Design.

Keywords: AI Planning&Scheduling, E-Learning, Learning Design, Automatic Monitoring.

1 Introduction

The development of new Information Technologies have originated new possibilities to develop pedagogical methodologies that provide the necessary knowledge and skills in the Higher Education environment. These technologies are built around the use of Internet and other new technologies, such as the Virtual education, Distance Learning or Longlife Learning. On one hand, many educators have seen them as a way to give flexibility and re-use previous courses stored in a database, or in other electronic formats. On the other hand, the increasing computing power and the actual network infrastructure allows to share and distribute these courses between public institutions and private corporations.

This new environment provides an interesting framework for different kind of researchers: Artificial Intelligence (AI) researchers that can experiment with their automatic problem solving algorithms, or to develop and design new algorithms in this complex domain; and educational researchers that can use a new kind of tools and techniques that could aid to detect, reason, and solve (automatically) their problems.

One of the AI areas most suitable to merge within this context is Automated Planning and Scheduling. Planning generates a plan (sequence or parallelization of activities) such that it achieves a set of goals given an initial state and satisfying a set of domain constraints represented in operators schemas. In scheduling systems, activities are organised along the time line having in mind the resources available. These systems can perfectly handle temporal reasoning and resource consumption, together with some quality criteria (usually time or resource usage) but they cannot produce the needed activities and their precedence relations given that they lack an

[1]Corresponding author.

expressive language to represent the activities. Traditionally, planning was first performed and the solution was given as an input to the scheduling systems.

These techniques have been applied in different real (and complex) environments such as, Industry, Robotics, or Information Retrieval with success. Of special attention in the last few years has been the development of an autonomous architecture [4] that can carry out a large number of functions such as planning activities, tracking the spacecraft's internal hardware or rover´s position, and ensuring correct functioning and repair when possible, without (or little) human intervention.

We present a metadata-based model that is combined with AI planning & scheduling techniques to monitor how is evolving a particular Learning Design (LD), and to propose solutions in those modules where the students encounter difficulties in the learning process. The paper is structured as follows. Section 2 provides a brief description the proposed metadata model for a LD representation. Section 3 describes how several AI techniques, like Planning and Scheduling, can be used to aid in the process of monitoring and quality assurance in the LD. Section 4 shows how the integration can be possible. Finally, Section 5 summarizes the conclusions and future work.

2 A Metadata-based model Representation for Learning Designs

The new extended representation of a classical course program will be used to define and monitor the LD. The LD structure represents all the information, or topics, related with the contents of the course. These topics represent the learning contents to be given to the students, the bibliography refers to these elements as *Unit of Learning* (UL), or *Unit of Study*. Only those new metadata information, necessary in the mapping process to generate the problem representation to the planning/scheduling module, will be described. The metadata information that have been included in our model is used to add the following information:

- *Dependencies.* This information is used to define logical, and semantic, relationships between the different UL. These dependencies are defined by the educators and represent the logical order among different UL in the course. We have defined two possible kind of dependencies: *Strong* and *Weak* dependencies.

- *Priority.* Any UL in the learning design will have assigned a priority between a maximum (represents any essential UL for the students) and a minimum value (represents a UL that can be relaxed from the course). These values are assigned by educators. Currently, we have defined a numerical value from 10 (maximum priority) to 0 (minimum priority).

- *Time duration.* This is a tuple: (min, med, max), that represents the minimum, medium and maximum time required to acquire the UL from a module of the LD. Each UL is divided into time units (we consider as a time unit = one hour), this tuple represents the effort from the educators to give the different units.

- *Pedagogical complexity.* This semantic parameter is defined by educators to represent the knowledge difficulty of a particular unit of the course. Currently it is possible to use the following values: (trivial, very low, low, medium, high, very high, extreme).

Therefore the metadata can be defined for each UL, or sub-units. The metadata definition will affect directly to other important processes, like planning and scheduling, because some of these information (time duration and priority) will be used in the mapping process.

3 AI Techniques & Learning Designs Monitoring

AI P&S consists of a set of techniques that enable efficient searching for solutions of problems with time and resources. Traditionally, there is a clear subdivision of techniques and roles that belong to planning and scheduling. Planning generates a plan (sequence or parallelization of activities) such that it achieves a set of goals given an initial state and satisfying a set of domain constraints represented in operators schemas. In scheduling systems, activities are organized along the time line having in mind the resources available. These systems can perfectly handle temporal reasoning and resource usage, together with some quality criteria (usually focused around time or resource consumption) but they cannot produce the needed activities and their precedence relations given that they lack an expressive language to represent the activities.

Nowadays it is being an increasing interest to integrate these two fields because of real domains needs. From this perspective, by combining scheduling and planning systems synergistically the weaknesses of both areas can be solved. Then, systems as IPSS [5] or IxTeT [2] are suitable candidate to solve this type of problems. Using a high level description, the inputs are:

- *Domain theory*: the STRIPS representation originally proposed by Fikes and Nilsson is one of the most widely used alternatives [1]. In the STRIPS representation, a world state is represented by a set of logical formulae, the conjunction of which is intended to describe the given state. Actions are represented by so-called operators. An operator consists of pre-conditions (conditions that must be true to allow the action execution), and post-conditions or effects (usually constituted of an add list and a delete list). The add list specifies the set of formulae that are true in the resulting state while the delete list specifies the set of formulae that are no longer true and must be deleted from the description of the state. Each course can be defined in terms of a set of learning activities that are performed by students. Therefore, there is a strong relation between operators in planning and learning activities to perform in order a student to successfully complete a course in learning environments.

- *Problem*: is described in terms of an initial state and goals. Those states are represented by a logical formula that specifies a situation for which one is looking for a solution. Examples of initial states would be: the previous knowledge of the students, the resources that the course uses and when they are available, the maximum number of student for each teacher, the priority of each module, etc. Goals are often viewed as specifications for a plan. In our case, that the student is able to apply critical thinking about a specific subject.

For scheduling systems, many techniques used in this area come from the Operational Research (OR) area (i.e., branch and bound, simulated annealing, lagrangian relaxation). Lately, Constraint Programming (CP) has been applied to the different scheduling problems with very good results. Scheduling techniques can be easily generalized and applied in the LD Scheduling Problem. In this case, instead of having machines (Job-Shop Scheduling problem) and jobs (RCPSP_{max} [3] problem), we have students and teachers, and UL in courses. Each unit (operation) needs to be processed during a period of time for a given student (machines), and the unit will be supervised by a learner. The course will also have a limited duration (deadline). Each learner will also have a maximum number of students (we consider a learner as a resource with a total resource capacity given by the number of students). We need to know the initial and end time of each UL considering precedences constraints among them. The variable values are imposed by the problem conditions: learning activity durations, course duration, number of learners, etc.... As a result of AI integrated planner & scheduler systems, they generate as an output a plan or set of plans if a solution exists for the given deadline. A plan can be seen as a sequence of operator applications (learning activities) with a specific duration that can lead from the initial state to a state in which the goals are reached with the resources available.

4 Automatic Planning Problem Generation

To generate an appropriate representation for IPSS, or other planner and scheduler system, it will be necessary to map both, the metadata stored in the LD and the statistical results obtained from the interaction (tests) with the students. Those tests must incorporate some metadata information when they are designed. When the educator builds a particular question (and their related answers), s/he defines what are the unit (or subunits) of learning that are related to. Therefore these metadata can be understood as the learning concepts that should be correctly learned by the students. Two metadata features provided by educators in the LD definition (*priority* and *complexity*) have been combined to generate *simulated values* for the students results. A simple function were used to obtain them, this function generates high fail values for those UL with a high complexity and low priority, high hit values for UL with high priority and low complexity, finally those UL with a high complexity and priority will obtain medium hits/fails values.

Previous information (LD course, and statistical results) are provided to the mapping module. This module generates automatically a problem to the planner module. The mapping process for our (e-Learning) planning domain it summarizes as follows:

1. First, information (represented in XML format) is parsed, the next attributes are extracted. The way we have used to represent them into IPSS is:

   (Attribute Class Value)

   The attributes of each unit or course, that is, their properties (complexity, name, etc) are parsed as the predicate names in IPSS. Them one of the parameters will be the class (that is, the unit/course) that the attributes belong to and the value associated (high, low, 5, etc) Figure 1 shows some of the initial conditions for a JAVA course program.

```
(duration Introduction (1 2 3))(complexity Introduction very_low)
(priority Introduction 3) (annontation Introduction none)
(availability educator1 30) (knows student1 basic-programming)
(interactivity-level learning_object1 3)
(learning-style student1 active)(learning-style student2 passive)
 ...
(goal (and (knows student1 Java) (knows student2 Java) ))))
```

Figure 1: A problem representation using the metadata-based model in the IPSS system.

2. Using previous data, the initial state problem template is fulfilled building the initial state of the problem using the general schema described above. From the statistical results the initial state of the students (in terms of number of hits) will be extracted, the LD structure is used to build the rest of the problem.

3. The goals of the problem are acquired from the interaction with the educator. Currently, and to allow an easily evaluation of our approach, this interaction has been substituted for an automatic goals generation process.

4. Using the initial state problem template fulfilled and the goals to achieve, a new (planning) problem is generated.

Once the problem is generated, it is given to the IPSS planner. IPSS will produce several solutions that try to achieve the proposed goals given a deadline (total course duration). The Figure 2 shows some of those solutions. As it can be seen on this figure, the solution is represented as a list of values with the following information: the start and end time of each activity and the

```
Start-Time     End-Time      Operator
        0         1.5         (LEARN_UNIT_OF_LEARNING U1 Introduction 1.5 DrRussel)
        1.5       5           (LEARN_UNIT_OF_LEARNING U2 Everything-an-object 3.5 DrRussel)
        5         10          (LEARN_UNIT_OF_LEARNING U3 Controling-flow 5 DrCamacho)
        0         1           (EXCLUDE_UNIT_OF_LEARNING  U4  Hiding-implementation)
       ...
```

Figure 2: Solutions generated by IPSS.

operator instantiated with the different values (unit name, duration and the teacher assigned) in case that the unit has beed included in the LD, otherways it can be excluded.

Finally, this solution (or solutions) is newly parsed into a user-friendly representation, this parsing process is currently made by an explanation module that simply modifies several attributes (i.e. start-time, end-time) from the old LD. This new LD is given to the educators as a new proposal, these modifications are analysed and evaluated by the educators, if these modifications are finally approved by them, this new LD will be deployed and used in the following cycle of the long-life educational process.

5 Conclusions and Future Work

This paper has presented two main contributions. On one hand, a metadata-based model to represent course programs in Higher (or other) Educational environment has been defined, on the other hand the paper shows how this model can be instantiated, and mapped, into and standard representation manageable for planners and schedulers systems. The paper provides an specific example of this instantiation in a particular planner/scheduler system (IPSS), and the analysis of the solutions proposed by this system.

In the next future the authors wish to deploy both, the metadata-based representation and the intelligent Planner/Scheduler module into an e-Learning platform (i.e. BlacBoard, WebCT, etc...) to allow the interaction with several groups of students.

Acknowledgments

This work has been funded by the Universidad de Alcalá project UAH PI2005/084.

References

[1] R. Fikes and N. Nilsson. STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2:189–208, 1971.

[2] M. Ghallab and H. Laruelle. Representation and Control in IxTeT, a Temporal Planner. In *Procs. of the Second International Conference on AI Planning Systems (AIPS-94)*, 1994.

[3] R. Kolisch and S. Hartmann. Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. *Project scheduling: Recent models, Algorithms and Applications*, pages 147–178, 1999.

[4] N. Muscettola, G.A. Dorais, C. Fry, R. Levinson, and C. Plaunt. IDEA: Planning at the Core of Autonomous Reactive Agents. In *Procs. of the Workshop On-line Planning and Scheduling, AIPS 2002, Toulouse, France*, pages 49–55, 2002.

[5] M. D. R-Moreno, A. Oddi, D. Borrajo, A. Cesta, and D. Meziat. IPSS: Integrating Hybrid Reasoners for Planning and Scheduling. In *Procs. of the 16th European Conference on Artificial Intelligence, ECAI04*, 2004.