CAMOU: A simple integrated eLearning and planning techniques tool

David Camacho† and María D. R-Moreno‡ and Unai Obieta‡

† Computer Science Department. Universidad Autónoma de Madrid. Madrid, Spain.
e-mail: david.camacho@uam.es
‡ Departamento de Automática. Universidad de Alcalá. Madrid, Spain.
email: mdolores@aut.uah.es

Abstract. In this paper we present an educational tool which has been designed to manage (learning) knowledge acquired from the interactions with the students, and to automatically aids educators in the complex process of course design and analysis. In the tool, only some essential learning knowledge will be translated (mapped) and provided to an automatic reasoning system, named IPSS. This system, which integrates Artificial Intelligence Planning and Scheduling, analyzes and detects problems in the current tested course, providing new solutions in form of new learning designs that can be approved (or rejected) by educators.

Keywords: e-Learning, AI Planning and Scheduling, Virtual Education Tools, Learning Designs adaptation.

1 Introduction

Most of the current Virtual Learning Environments (VLE) contain pre-fixed courses where the user navigates and learns the concepts that they have been planned for. Well known educational platforms are: First Class [1], LMS [2], WebCT [3], Moodle [4], or E-ducativa [5].

Those mentioned tools, and platforms, allow the instructors to get statistics as well as other information about the student progress. But there is still a lack of feedback among the previous users, the tool, the instructors and the future users. Among the tools that have worked in this direction we can mention the CourseVis system [11] that visualizes data from a java on-line distance course accessed through WebCT. The tool tracks the students evaluation and takes into account the instructors' requirements. This examination has to be done manually without any tool that can assist the instructor in the decisions that have to be made. Our approach can solve some of the deficiencies of eLearning

[1] http://www.softarc.com
[2] http://www.lotus.com/lotus/offering6.nsf/wdocs/homepage
[3] http://www.webct.com/
[4] http://moodle.org/
[5] http://www.e-ducativa.com/

courses and gives automatic solutions to the improving of existing courses by taking into account student interaction with them.

On the other hand, several (eLearning) standards and guides have been proposed related to learning object metadata, student profiles, course sequencing, etc. The IEEE Learning Technology Standards Committee (LTSC, 2006) has developed the Learning Object Metadata (LOM, 2006) standard which specifies the attributes required to describe a Learning Object (LO), where a LO is defined as any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning. Relevant attributes of learning objects to be described include type of object, author, owner, terms of distribution, format, and pedagogical attributes, such as teaching or interaction style. Another specification which allows the modeling of learning processes is the Learning Design (LD) information model (IMS LD, 2006) from the IMS Global Learning consortium. A learning design (LD) is a description of a method enabling learners to attain certain learning objectives by performing certain learning activities in a certain order in the context of a certain learning environment.

LD integrates other existing specifications. Among these, it is worth mentioning the IMS Content Packaging (IMS CP, 2006), which can be used to describe a learning unit (LU). A LU can have prerequisites which specify the overall entry requirements for learners to follow that unit. In addition, a LU can have different components such as roles and activities. Roles allow the type of participant in a LU to be specified. Activities describe the actions a role has to undertake within a specified environment composed of LO. LD also integrates the IMS Simple Sequencing (IMS SS, 2006), which can be used to sequence the resources within a LO as well as the different LO and services within an environment. Content is organized into a hierarchical structure where each activity may include one or more child activities. The learning process can be described as the process of traversing the activity tree, applying the sequencing rules, to determine the activities to deliver to the learner. However, the increasing interest, and research, in educational standards makes quite difficult to reuse them with other techniques such as Artificial Intelligence based. Currently complex mapping processes are hardly programmed to adapt different aspects from the eLearning standards (LOs, metadata, etc.) into an appropriate AI-based representation (i.e. PDDL planning representation language). Our approach, tries to simplify how to deal with these knowledge using only some statistical and educational interactions among students and educators, to integrate them into a reasoning module, to show how automatic reasoning techniques (i.e. planning and scheduling) can be used.

The paper is structured as follows. Section 2 provides a brief description about the related Artificial Intelligence techniques used. Next, Section 3 describes the learning tool developed to interact and test the educational courses. Then, Section 4 shows both how the integration among the AI reasoning system, and the educational system, has been done and provides a simple execution example. Finally, Section 5 shows the main conclusions and future work of the paper.

2 Automatic Reasoning in VLE

Although, the initial approach that we have followed in [4] integrates the IPSS [12] system in an adaptive (deployed) learning tool, named TANGOW [5]. TANGOW requires tasks and rules. The tasks define the units in the learning process, the rules specify the way of organizing tasks in the course along with information about the task execution (order among tasks, free task selection, prerequisites among tasks, etc.). However, in this work our main motivation is the study of the reasoning techniques to manage, and deal, with the educational problems, for this reason the approach presented in this paper does not need to define rules, since the new tool (CAMOU) does not perform any individual adaptation but the course per se. The new system is used for advising and fault detection and it is based on the statistic results of the students to replan the whole course.

2.1 Brief Introduction to AI Planning & Scheduling Techniques

In the last decades Artificial Intelligence (AI) Planning and Scheduling (P&S) has become a successful, and widely used techniques. It allows us to generate a sequence of activities that achieves a set of goals having in mind the time and resources available.

These techniques have been applied with success in different real (and complex) environments such as, Industry, Robotics, Space missions or Information Retrieval. Traditionally, there is a clear subdivision of techniques and roles that belong to Planning and Scheduling. Planning [2] 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. Scheduling has to face the problem of organizing tasks in time. The problem is to locate a set of tasks in time, each task needing one or several resources during its execution. Nowadays it is being an increasing interest to integrate AI P&S because of real domains needs. From this perspective, by combining them the weaknesses of both areas can be solved. In this direction, IPSS [12] has been built. Other approaches that have followed this approach are O-PLAN-2 [14], IxTeT [1] or EUROPA [8]. Using a high level description, the inputs to those kind of systems are:

- *Domain theory*: the STRIPS representation originally proposed by Fikes and Nilsson is one of the most widely used alternatives [7]. 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.

- *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.

As output, the planner generates a plan with the sequence (linear or parallel) of operators that achieves a state (from the initial state) that satisfies the goals.

For scheduling systems, many techniques used in this area come from the Operational Research (OR) area [13] (i.e., branch and bound, simulated annealing or lagrangian relaxation). Lately, Constraint Satisfaction (CSP) [6] has been applied to the different scheduling problems with very good results. A CSP problem has inputs:

- A set of variables.
- A set of domains values containing the possible values for the corresponding variable.
- A set of constraints for the variables.

The output of scheduling systems is a values assignment that fulfills all the constraints in the variables.

As a result of the integration, they generate as an output a plan or set of plans (if a solution exists) time and resource consistent. 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 (i.e. educators available).

In educational environments several works to automatically generate courses based on pedagogical tasks and methods has been performed.For instance, in [15] an AI hierarchical task network (HTN) planner called JSHOP [10] which assembles learning objects retrieved from one or several repositories to create a whole course has been used. The learning objects are linked by taking into account the user knowledge information and the learning goals that the user should achieve. Our approach not only can link learning objects, but also schedule them along a period of time and consider previous student results to generate different LDs.

2.2 Integrating Planning and Scheduling: IPSS

The IPSS system is divided in two blocks as shown in Figure 1. The Plan Reasoner (IPSS-P) composed of an heuristic planner and an a deorder algorithm [3]. The deorder algorithm transforms the sequence of activities given by the planner (Total Order plan) into a parallelization of activities, eliminating the inneccesary precedence constraints (Partial Order plan). And the Scheduler reasoner (IPSS-S), is represented as a Constraint Satisfaction Problem (CSP) partitioned in two sub-problems. A basic Ground-CSP to reason on temporal constraints and a Meta-CSP to reason on resource constraints. Like that, IPSS is able to manage not only simple precedence constraints, but also more complex temporal requirements and multicapacity resource usage/consumption.

Then, the reasoning is subdivided in two levels. The planner focuses on the actions selection (possibly in the optimisation of some quality metric different than time-resource usage), and the scheduler on the time and resource assignments. During the search process, every time the planner chooses to apply an operator, it consults the scheduler for the time and resource consistency. If the resource-time reasoner finds the plan inconsistent, then the planner backtracks. If not, the operator gets applied, and search continues until a solution is found.

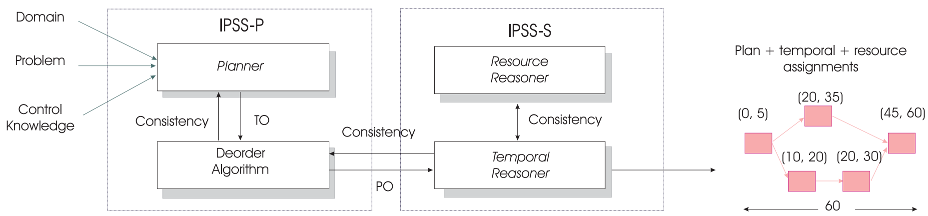

Fig. 1. Planning IPSS architecture.

3 Statistical course redesign based on planning techniques: CAMOU

Using our previous experience, we have designed and implemented a new learning tool which facilitates the definition of LDs and the acquisition of student interactions, both kind of data are later translated to be automatically analyzed by IPSS. The tool, named CAMOU, has been implemented using the following modules (Figure 2):

- *Learning Design Generation Module*. It allows (educators) to manage all the activities related to LDs generation and monitoring (i.e. create a new LD, modify, delete, or listing the stored LD), Figure 3 shows some screenshots of these functionalities, they can be summarized as follows:
  - *Learning Design management*. It allows to define the information related with a particular LD stored in the system (Figure 3 a) and b)), i.e. number of educators, groups...
  - *Unit of Learning management*. It is used to define the Unit of Learning (UL), and their associated pedagogical contents that defines the course. We use a meta-data representation, that can be used by other elements in our system (i.e. IPSS planner) to reason with the stored information (Figure 4 a) and b)). It is quite interesting to remark that some meta-data information related to the maximum and minimum duration for each UL should be provided by the educator (later this information will be used by the reasoner).

```
1- Learning Design Generation Module:
    - Learning Design management
    - Unit of Learning management
    - Dependencies management

2- Students & Educators Management Module:
    - Educators management
    - Student management

3- Exams & Tests Module:
    - Question generation module
    - Answers generation module
    - Exams & test management

4- Statistical Module:
    - Exams & Test statistics
    - Group & Students statistics
    - Questions statistics
```

Fig. 2. CAMOU Architecture.

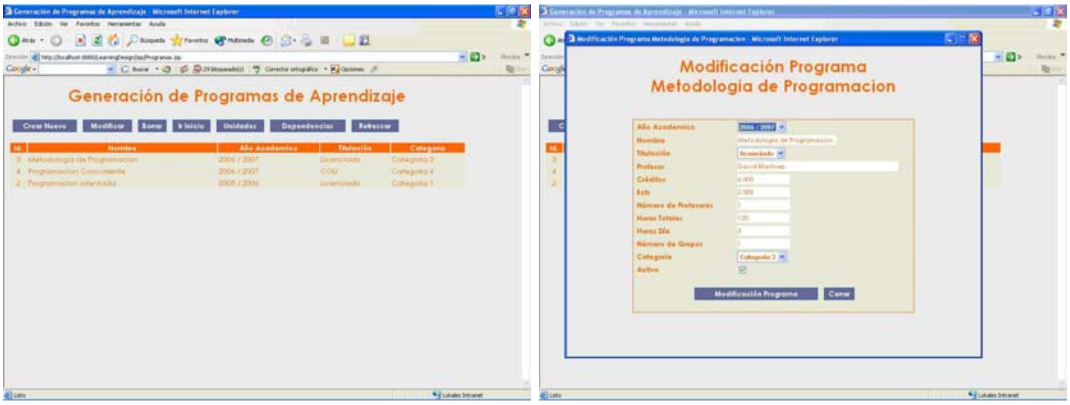

Fig. 3. a) LD listing; b) LD meta-data modification

- *Dependencies management*. This submodule allows to define (or modify) two different kind of dependencies (weaks and strongs) between the different UL that defines the course (Figure 5 a) and b).

– *Students & Educators Management Module*. This is module allows (using several Web interfaces) to manage the main actors in the system, educators and students. Figure 6 shows both how a particular educator is registry in the system, and the current list of students for a particular course.

– *Exams & Tests Module*. This module allows the educator to generate (modify or delete) both the questions and the related answers that will be used to make the exams and tests to our students. This module incorporates meta-data information related with both UL and LD. Figure 7 shows (a) several

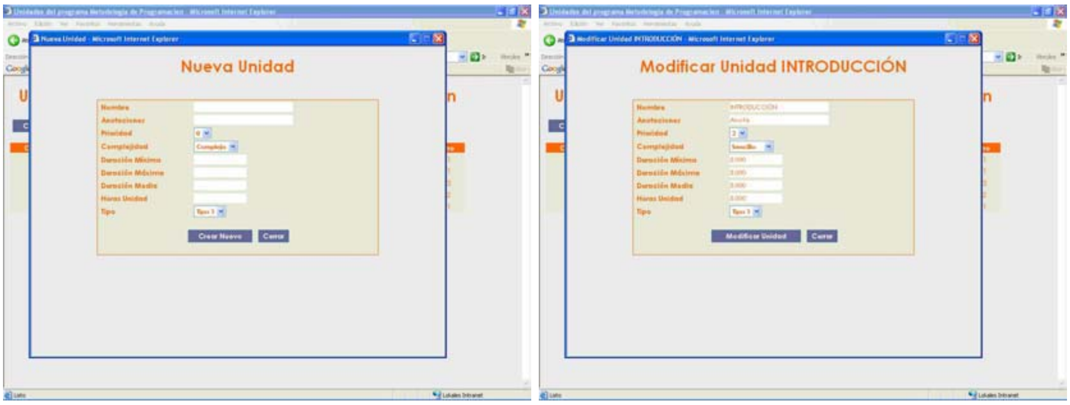

Fig. 4. a) UL definition; b) UL modification

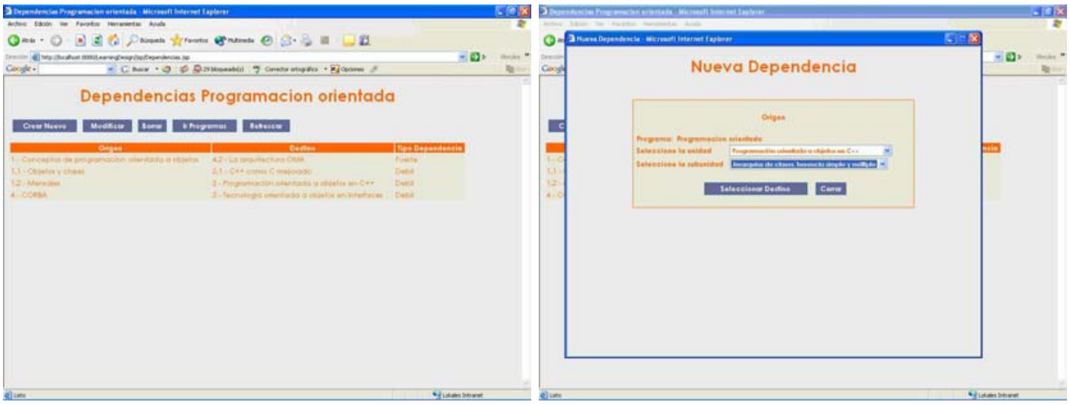

Fig. 5. a) Listing some existing dependencies for a particular LD definition; b) Dependency definition

questions and their UL related that have been created and stored in the system, and (b) how a new question is generated in the system.

- *Statistical Module*. Finally, this module generates a set of classical statistical values for different issues: groups, questions and persons. Figure 8 shows (a) several statistical results for each group, and (b) the statistics for each question.

4 How is the integration between CAMOU and IPSS done?

In this section we show the process that students and educators follow for a particular course, i.e. a LaTeX course [9]. We present this example to illustrate

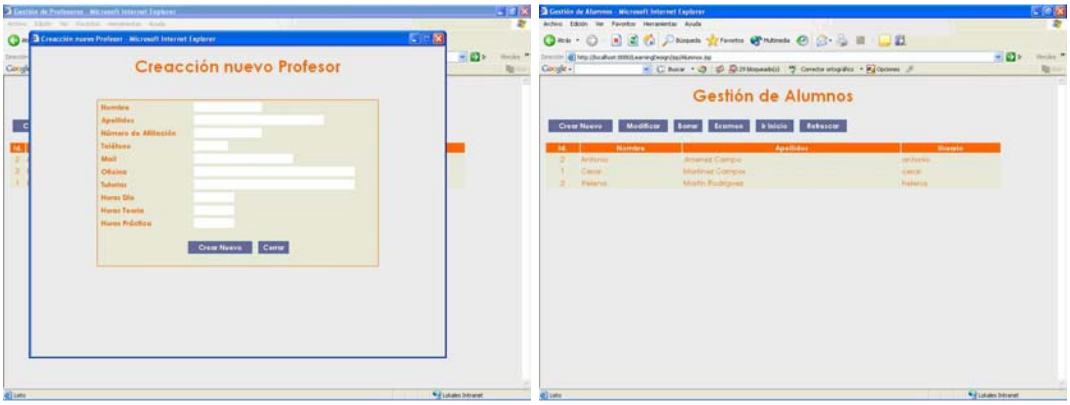

Fig. 6. a) Registration of a new educator; b) Listing some students

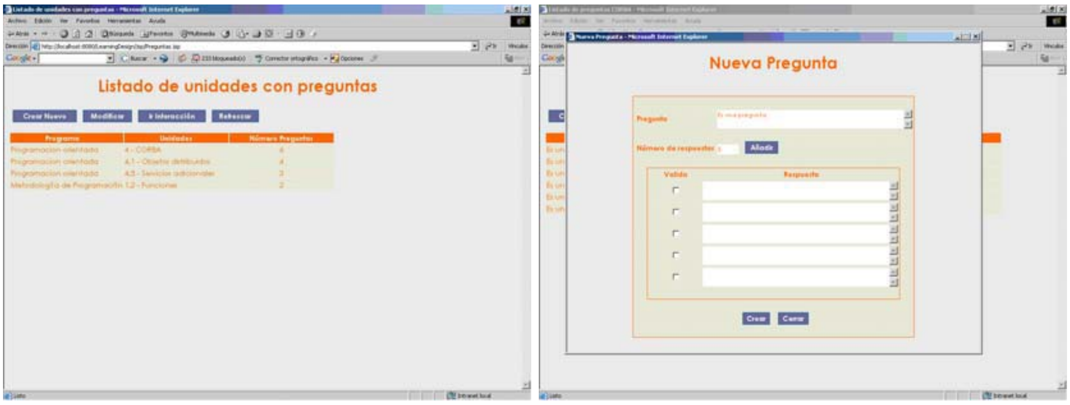

Fig. 7. a) Listing of Question for several UL; b) Generation of a new Exam/Test Question

how the integration is done. The first step is to define all the information about the units that are part of the course and associate to them the contents and exercises. This task will be done by the educators using the *Learning Design Generation Module* described on previous section.

Figure 9 shows the different units and subunits that compose the course, and some annotations such as the minimum and maximum duration, the priority or the complexity. The tool checks that the total course duration (known as a `makespan` in AI terminology) is equal to the sum of the units and subunits. If there is an inconsistency, a message is presented to the educator before the automatic module can be run.

Another information that we should provide is the dependencies (i.e. weak and strong). All this information is needed in order to translate it into IPSS. Units

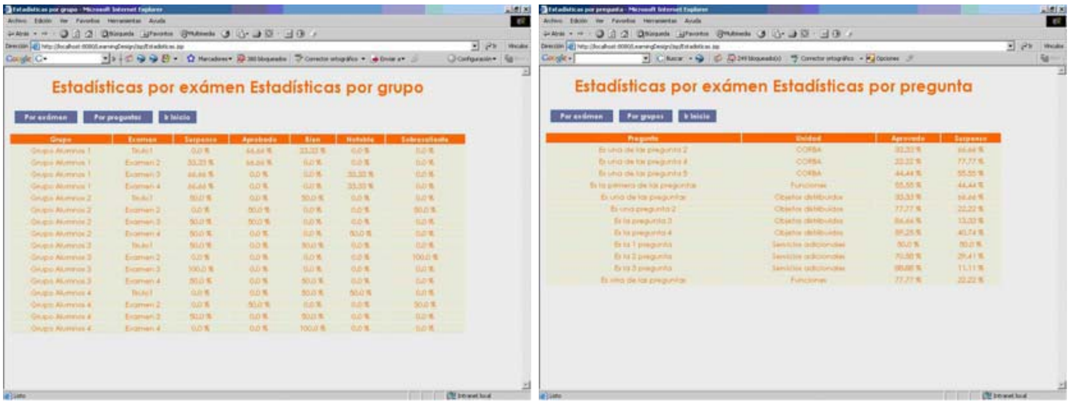

Fig. 8. a) Statistical results for different exams/groups; b) Statistical results for several questions and LD

```
1- Introduction (priority = 3, duration (2,4,6) Complexity = very low):
    1.1- History
    1.2- Components

2- Structure of a Document (priority = 8, duration (7,10,13) Complexity = medium):

3- Basic Formattin Tools (priority = 4, duration (3,4,5) Complexity = medium):

4- The Layout of the page (priority = 5, duration (6,8,10) Complexity = high):

5- Tabular Material (priority = 6, duration (12,14,16) Complexity = high):
    5.1- Tabbing
    5.2- array
    5.3- supertab and longtable
    5.4- Applications

6- Mastering Floats (priority = 4, duration (6,8,10) Complexity = high):
```

Fig. 9. An example of a LaTeXcourse.

with weak dependencies could be eliminated from the course in case some other units require more duration. IPSS will decide based on the dependencies, the minimum and maximum duration, and the priority. The units with high priority and weak dependencies are less probable to be eliminated than the units with low priority and weak dependencies. The base priority, that makes IPSS to decide which units can be part of the course or not, should be provided by the educators for a complete description).

Once the educators have introduced all the information, the students can start using the tool. It is now up to the educators to evaluate the student's knowledge and psychological model. This test that can be performed through the tool, will allow the educators to define and know the student profile. Actually, when a student starts a course, the student previous knowledge is uncertain and the educator does not know what can be the main difficulties that he/she has to face with. Thanks to the new Information Technologies and well made tests, this information can be known almost immediately and it can automatically be translated into the initial state of a planning problem and the preconditions of the operators.

At the beginning all the students will start the course with the first unit of learning: "Introduction". IPSS will assign to the course the minimum time duration that the educator has decided, due to the low priority and complexity values. Until know, there are not many options for the scheduler to plan for different solutions.

After one or several units, let us suppose that an exam is planned. The students are now in the "Tabular Material" unit of learning, and thanks to the tests, we have a personalized knowledge of the weak points of the already learnt subunits.

From the results we can know that 70% of the students have failed the "array" sub-unit. Then, a failure in the Learning Design (LD) has been detected. This information is saved for the future LD revisions. In this situation, the pedagogical responsible can decide to add more examples to this subunit, what implies the increase in the minimum, medium and maximum duration time. This increase of time in one of the modules will produce a reduction in other modules in order to keep consistency with the global course duration (deadline). And in more drastic cases, to eliminate one (or several) subunit(s). That decision will be made automatically by IPSS, but it is the responsibility of the pedagogue to check the consistency from the pedagogical point of view.

5 Conclusions

In this paper we have described both, a simple tool (CAMOU) that has been designed to manage educational knowledge acquired from the interactions with the students, and how it can be integrated with an automated reasoning system (IPSS) to help educators in the complex process of course design. Although there exist some current eLearning standards (i.e. IMS, LOM or SCORM) widely used by the e-Learning community, when these standards are combined, or integrated, with other techniques (i.e. Artificial Intelligence) it can be quite hard to represent, translate and manage the stored knowledge due the complexity of those standards. Our approach tries to simplify how to deal with this knowledge using only some statistical and educational interactions among students and educators, and integrate them into a reasoning module.

The CAMOU tool only uses some essential learning knowledge that is translated (mapped) and given to IPSS. Using only these knowledge we are trying to minimize the bias in the translation/mapping process, because if we try to map all the eLearning knowledge, possibly important semantic and syntactic knowledge will be missing by the reasoning system.

Acknowledgements

This work has been funded by the research projects: CYCIT project TSI 2006-12085 and Junta de Castilla-La Mancha project PAI07-0054-4397.

References

1. P. Albers and M. Ghallab. Context Dependent Effects in Temporal Planning. In *Procs. of the 4th European Conference on Planning, Toulouse, France.*, pages 1–12, 1997.
2. J. F. Allen, B. Hendler, and A. Tate. *Readings in Planning.* Morgan Kaufman, 1990.
3. C. Bäckström. Computational aspects of reordering plans. *Journal of Artificial Intelligence Research*, 9:99–137, 1998.
4. D. Camacho, A. Ortigosa, E. Pulido, and M. D. R-Moreno. *AI techniques for Monitoring Student Learning Process.* Information Science Reference, formerly Idea Group Publishing, Ed. by Francisco J. García, 2008 (to appear).
5. R. Carro, A. Ortigosa, and J. Schlichter. A rule-based formalism for describing collaborative adaptive courses. In V. H.-R. Palade and S.-V. L. Jai, vol. 2774, editors, *Knowledge-Based Intelligent Information and Engineering Systems. Proceedings of the International Conference KES'2003*, pages 252–259, 2003.
6. R. Dechter and F. Rossi. Constraint Satisfaction. *Survey ECS*, 2000.
7. R. Fikes and N. Nilsson. STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2:189–208, 1971.
8. J. Frank, A. Jónsson, and P. Morris. On Reformulating Planning as Dynamic Constraint Satisfaction. In *Procs. of the Symposium on Abstraction, Reformulation and Aproximation (SARA), Horseshoe Bay (Lake LBJ), Texas, USA*, pages 271–281, 2000.
9. M. Goossens, F. Mittelbach, and A. Samarin. *The Latex Companion.* Addison-Wesley, 1994.
10. O. Ilghami and D. S. Nau. A general approach to synthesize problem-specific planners. Technical Report Technical Report CS-TR-4597, Department of Computer Science, University of Maryland, 2003.
11. R. Mazza and V. Dimitrova. CourseVis: Externalising student information to facilitate instructors in distance learning. In U. Hoppe, F. Verdejo, and J. Kay, editors, *Proceedings of the International conference in Artificial Intelligence in Education*, pages 117–129, 2003.
12. M. D. R-Moreno, A. Oddi, D. Borrajo, and A. Cesta. Ipss: a hybrid approach to planning and scheduling integration. *IEEE Transactions on Knowledge and Data Engineering.*, 18(12):16811695, December 2006.
13. H. A. Taha. *Operations Research: An Introduction.* Prentice Hall. Eighth edition., 2006.
14. A. Tate, B. Drabble, and K. R. *O-Plan2: An Open Architecture for Command, Planning, and Control.* Morgan Kaufman, 1994.
15. C. Ullrich. Course Generation Based on HTN Planning. In *Proceedings of 13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems*, pages 74–79, 2005.