



UNIVERSITÀ DELLA CALABRIA



UNIVERSITA' DELLA CALABRIA

Dipartimento di Fisica

Scuola di Dottorato

“Archimede” in Scienze, Comunicazione e Tecnologie

Indirizzo

Fisica e tecnologie quantistiche

In co-tutela di tesi con Universitat Politècnica de València

Con il contributo di

POR CALABRIA FSE 2007/2013

CICLO

XXVI

TESI

**“INTELLIGENT PLANNING FOR CURRICULUM PERSONALIZATION
OF E-LEARNING CONTENTS. MOODLE AS A CASE STUDY”**

Settore Scientifico Disciplinare FIS/08 – Didattica e storia della fisica

Direttore: Ch.mo Prof. Pietro Pantano

Supervisor: Ch.ma Prof.ssa Assunta Bonanno

Ch.mo Prof. Antonio Garrido

Dottoranda: Dott.ssa Valentina Caputi

La presente tesi è cofinanziata con il sostegno della Commissione Europea, Fondo Sociale Europeo e della Regione Calabria. L'autore è il solo responsabile di questa tesi e la Commissione Europea e la Regione Calabria declinano ogni responsabilità sull'uso che potrà essere fatto delle informazioni in essa contenute.

Abstract

The use and enhancement of technology in the field of (distance) education has increased the adoption of e-learning. Nowadays, a simple connection to the Internet makes it possible to access any on-line content. Thus, e-learning platforms based on Learning Management Systems (LMSs) permits us to eliminate the barriers of time and space that usually limit traditional teaching. LMSs facilitate the administration, storage, delivery, visualization and monitoring of e-learning contents to both students and teachers in a simple and functional way.

There are many LMSs, such as Moodle, Docebo, Atutor, Ilias, .LRN, etc., widely used to support learning/teaching activities. For the best use of a LMS, it is fundamental not to consider it in an isolated way as a simple repository of learning contents, but as part of a larger system in which learning contents are aggregated for the construction of courses that can be fully personalized or adapted to the specific needs and abilities of each student.

Curriculum personalization is faced in several ways by using several techniques, such as evolutionary algorithms, data mining techniques, decision support systems, etc., with the main objective to offer the best content to the most adequate person. In this thesis, we focus on Artificial Intelligence planning as a valuable formalism to describe actions (learning contents) in terms of preconditions (precedence relationships) and causal effects to find plans (learning paths) that entirely fit the students' profiles. Thus, the integration of intelligent planning techniques into LMSs makes it possible to achieve the greatest learning benefits thanks to the automatic generation of customized learning paths. In particular, we focus on Moodle (Module Object-Oriented Dynamic Learning Environment) as a free, open-source PHP web application for producing modular Internet-based courses. Teachers and students interact in Moodle by means of activities (e.g. lessons, chats, SCORMs, forum, wikis, etc.) On the other hand, resources (e.g. text pages, web pages, links,

etc.) are optionally used to transmit additional information regarding the activities. Consequently, courses can be created by appropriately combining activities and resources to deliver learning contents to the students. Moodle is flexible enough to model from small to big courses and it can be easily used and modified in order to extend its functionalities.

The integration of planning in Moodle is not a straightforward task because Moodle and, in general, any LMS presents some limitations for this. The lesson is the most adequate Moodle's activity to define causal and precedence relationships. But the lessons' number and order must be fixed by the teacher. In other words, the execution sequence, i.e. learning path, may not take into consideration all the students' needs, as determined by their profiles (background, learning style and goals). Thus, we have faced and solved the main limitations in order to integrate our intelligent planning approach in Moodle. First, complex relationships that usually appear when planning very elaborate courses cannot be easily defined in the form of (Moodle) lessons. We have overcome this limitation by using intermediate dummy lessons that simulate virtual transitions between learning states. Second, the information that students input into the platform are not always sufficient to exhaustively define their background and learning goals, indispensable to define customized learning paths. Again, our solution implies the creation of dummy lessons that help specify initial/goal learning states. Third, compiling a planning model from Moodle's database is not intuitive because course properties are not easily available and, thus, we defined a detailed mapping. Fourth, once generated a plan by using a standard planner, a challenge arises to personalize the visualization and navigation of individual paths to each student. We coped with this by associating personal views to dynamically created groups of students.

This work offers a scalability evaluation to demonstrate the applicability of the proposed approach. In particular, by using two standard planners it is possible to solve a number of tests involving courses of different sizes (up to 64 lessons) and with different numbers of students (up to 500 students). Courses containing up to 64 lessons and with 300 students are customized in less than 900 seconds that is an excellent result. From a more practical perspective, we have include a real demonstration of the functioning of the presented approach by implementing and customizing in Moodle a real Physics course of the University of Calabria (Italy).

In conclusion, this thesis presents an AI planning approach to generate personalized learning paths within Moodle. In particular, it is explained how to generate and visualize personalized learning paths within Moodle. It is im-

portant to observe that in the presented system can use any standard planner and that the entire approach can be easily extrapolated to any other standard LMS. Furthermore, thanks to the flexibility of the approach, it is possible to adapt and customize our ideas to any type of learning course in any type of LMS.

Riassunto

L'utilizzo e la valorizzazione della tecnologia nel campo dell'istruzione a distanza ha incrementato l'adozione dell'e-learning. Al giorno d'oggi una semplice connessione ad Internet permette di accedere a qualsiasi contenuto didattico on-line. In tal modo, piattaforme e-learning basate su Learning Management Systems (LMSs) consentono di eliminare le barriere spazio-temporali che generalmente limitano l'insegnamento tradizionale. Un LMS agevola la gestione, l'immagazzinamento, la divulgazione, la visualizzazione e il monitoraggio dei contenuti di e-learning in modo semplice e funzionale.

Esistono molti LMSs (Moodle, Docebo, Atutor, Ilias, dotLRN, ecc.) ampiamente utilizzati per supportare le attività di insegnamento/apprendimento. Al fine di utilizzare al meglio un LMS, è fondamentale non considerare tale piattaforma in modo isolato come un semplice database di contenuti didattici, ma come parte di un sistema più ampio, in cui i tali contenuti vengono aggregati per la realizzazione di corsi che possono essere completamente personalizzati o adattati alle esigenze e capacità specifiche di ogni studente.

La personalizzazione di curricula viene affrontata in diversi modi e utilizzando svariate tecniche, come ad esempio gli algoritmi evolutivi, le tecniche di data mining, i sistemi di supporto alle decisioni, ecc, con l'obiettivo principale di offrire il contenuto didattico più adatto alla persona più adeguata. In questa tesi ci si focalizza sulla pianificazione in Intelligenza Artificiale come un formalismo prezioso per descrivere azioni (contenuti didattici) in termini di precondizioni (relazioni di precedenza) e di effetti causali per trovare piani (percorsi di apprendimento) che si adattano completamente ai profili specifici degli studenti. Pertanto, l'integrazione di tecniche di pianificazione intelligente negli LMSs consente di ottenere i maggiori benefici di apprendimento grazie alla generazione di curricula personalizzati. In particolare, questo lavoro si focalizza su Moodle (Module Object-Oriented Dynamic Learning Environment) che è un'applicazione web PHP free e open source per la realizzazione di corsi

elettronici modulari. Insegnanti e studenti interagiscono su Moodle per mezzo delle attività (lezioni, chat, SCORM, forum, wiki, ecc.) Le risorse invece (pagine di testo, pagine web, link, ecc.) sono moduli facoltativi utilizzati per trasmettere ulteriori informazioni riguardanti le attività. Di conseguenza, i corsi possono essere creati combinando opportunamente attività e risorse per fornire i contenuti didattici agli studenti. Moodle è una piattaforma abbastanza flessibile, consente di modellare da piccoli a grandi corsi e può essere facilmente utilizzata e modificata al fine di estendere le sue funzionalità.

L'integrazione della pianificazione in Moodle non è un compito semplice perchè Moodle e, in generale, qualsiasi LMS presenta alcune limitazioni in tal senso. La lezione è l'attività più adeguata per definire le relazioni causali e di precedenza. D'altro canto, però, il numero e l'ordine delle lezioni devono essere fissati dal docente. In altre parole, la sequenza di esecuzione, vale a dire il percorso di apprendimento, non può prendere in considerazione tutte le esigenze degli studenti dipendenti dagli specifici profili (background, stile e obiettivi di apprendimento). Abbiamo perciò affrontato e risolto le limitazioni principali che ci impedivano di integrare il nostro approccio di pianificazione intelligente in Moodle. In primo luogo, le relazioni complesse che compaiono generalmente quando si pianificano corsi molto elaborati non possono essere facilmente definite utilizzando il modulo lezione. Abbiamo superato questa limitazione utilizzando lezioni fittizie intermedi che simulano transizioni tra stati virtuali di apprendimento. In secondo luogo, le informazioni che gli studenti possono inserire nella piattaforma non sono sempre sufficienti per definire esaurientemente il loro background ed i loro obiettivi di apprendimento, indispensabili per definire percorsi di apprendimento personalizzati. Ancora una volta, la nostra soluzione implica la creazione di lezioni fittizie che consentono di specificare stati iniziali ed obiettivi di apprendimento. In terzo luogo, la compilazione di un modello di pianificazione dal database di Moodle non è intuitiva perchè le proprietà di un corso non sono facilmente disponibili e, quindi, abbiamo dovuto definire un mapping dettagliato in tal senso. In quarto luogo, una volta generato un insieme di percorsi di apprendimento utilizzando un pianificatore standard, sorge il problema della visualizzazione e della navigazione personalizzata per ciò che concerne i differenti percorsi. Abbiamo affrontato questo problema associando visualizzazioni personalizzate a gruppi di studenti creati in modo dinamico di volta in volta.

Questo lavoro offre una dimostrazione di scalabilità in cui si evidenzia l'applicabilità dell'approccio proposto. In particolare, utilizzando due pianificatori standard, è possibile risolvere una serie di test che includono corsi di

diverse dimensioni (fino a 64 lezioni) e con diversi numeri di studenti (fino a 500 studenti). Corsi contenenti fino a 64 lezioni e fino a 300 studenti vengono personalizzati in meno di 900 secondi (che è un ottimo risultato). Da un punto di vista più pratico, abbiamo illustrato una dimostrazione del funzionamento dell'approccio presentato grazie all'implementazione e alla personalizzazione in Moodle di un corso reale di Fisica dell'Università della Calabria.

In conclusione, questa tesi presenta un approccio di pianificazione intelligente per la generazione di percorsi di apprendimento personalizzati all'interno di Moodle. In particolare, si spiega come generare e visualizzare percorsi di apprendimento individuali all'interno della piattaforma. È importante osservare che nel sistema presentato si può utilizzare qualsiasi pianificatore standard e che l'intero approccio può essere facilmente estrapolato ad altri LMS standard. Inoltre, grazie alla flessibilità dell'approccio, è possibile adattare le nostre idee a qualsiasi tipo di corso di apprendimento e in qualsiasi tipo di LMS.

Tale attività di ricerca è coerente con lo specifico curriculum cofinanziato dal FSE, in linea con l'area di interesse specifica del Polo di Innovazione Regionale (Tecnologie dell'informazione e delle telecomunicazioni) indicato nel decreto di nomina per ciascun vincitore della borsa FSE.

Resumen

El uso y la mejora de la tecnología en el área de la educación a distancia ha hecho que incremente la incorporación del e-learning. Actualmente, una simple conexión a Internet hace posible el acceso a cualquier contenido didáctico on-line. Las plataformas de e-learning basadas en Learning Management Systems (LMSs) nos permite eliminar barreras temporales y espaciales que normalmente limitan el sistema tradicional de enseñanza. Los LMSs facilitan la administración, el almacenamiento, la provisión, la visualización y el control de los contenidos de e-learning tanto a los estudiantes como a los profesores de una forma simple y funcional.

Hay muchos LMSs tales como Moodle, Docebo, Atutor, Ilias, .LRN, etc., comunmente utilizados para dar soporte a las actividades de enseñanza/aprendizaje. Para un mejor uso de un LMS es fundamental no considerarlo de forma aislada como un simple repositorio de contenidos de aprendizaje, sino como una parte de un sistema más grande en el cual estos contenidos son agregados para la construcción de cursos que pueden ser completamente personalizados o adaptados a las necesidades y habilidades específicas de cada estudiante.

La personalización de los curricula ha sido tratada de diferentes formas utilizando varias técnicas como algoritmos evolutivos, minería de datos, sistemas de ayuda a la decisión, etc., con el principal objetivo de ofrecer el contenido más adecuado según la persona. En esta tesis nos centramos en el área de la Inteligencia Artificial y más concretamente en la planificación como un formalismo que describe acciones (contenidos de aprendizaje) en términos de precondiciones (relaciones de precedencia) y efectos para encontrar planes (rutas de aprendizaje) que se ajustan completamente a los perfiles de los estudiantes. Por lo tanto, la integración de técnicas inteligentes de planificación en los LMSs hace posible alcanzar mayores beneficios en el aprendizaje gracias a la generación automática de rutas de aprendizaje adaptadas. En particular, nos centramos en Moodle (Module Object-Oriented Dynamic Learning Environ-

ment) como una aplicación web en PHP libre y open-source para producir cursos modulares en Internet. Profesores y estudiantes interactúan en Moodle a través de actividades (ej. lecciones, chats, SCORMs, forum, wikis, etc.) Por otra parte, los recursos (ej. páginas de texto, páginas web, links, etc.) son opcionalmente utilizados para transmitir información adicional a las actividades. Por lo tanto, los cursos pueden ser creados combinando de manera apropiada actividades y recursos para proporcionar contenidos de aprendizaje a los estudiantes. Moodle es suficientemente flexible para modelar desde pequeños a grandes cursos y puede ser fácil de utilizar y modificar para extender sus funcionalidades.

La integración de técnicas de planificación inteligente en Moodle no es una tarea directa debido a que Moodle, y en general cualquier LMS, presenta algunas limitaciones. La lección es la actividad más adecuada en Moodle para definir relaciones causales y de precedencia. Pero el número de lecciones y el orden debe ser fijado por el profesor. En otras palabras, la secuencia de ejecución, la ruta de aprendizaje, puede no tener en consideración todas las necesidades de los estudiantes y sus perfiles (conocimientos previos, estilo de aprendizaje y objetivos). Por lo tanto, nosotros hemos tratado las principales limitaciones para integrar nuestra propuesta basada en planificación inteligente en Moodle. Primero, las relaciones complejas que normalmente aparecen cuando se planifican cursos muy elaborados no pueden ser fácilmente definidas en la forma de lecciones de Moodle. Nosotros hemos superado esta limitación utilizando lecciones intermedias ficticias que simulan transiciones virtuales entre estados de aprendizaje. Segundo, la información que los estudiantes introducen en la plataforma no es siempre suficiente para definir de manera exhaustiva sus conocimientos previos y objetivos de aprendizaje, indispensables para ayudar a especificar rutas de aprendizaje personalizadas. Nuestra solución implica la creación de lecciones ficticias que ayudan a especificar los estados de aprendizaje iniciales y los objetivos de aprendizaje. Tercero, compilar un modelo de planificación de la base de datos de Moodle no es intuitivo porque las propiedades del curso no están fácilmente disponibles y, por lo tanto, nosotros definimos un mapping detallado. Cuarto, una vez que se ha generado un plan utilizando un planificador estándar, surge la necesidad de personalizar la visualización y la navegación de las rutas individuales para cada estudiante. Nosotros tratamos este problema con la asociación de vistas personales a grupos de estudiantes creados de manera dinámica.

Este trabajo ofrece una evaluación de escalabilidad para demostrar la aplicabilidad de la propuesta. En particular, utilizando dos planificadores estándar

es posible solucionar un número de tests que incluyen cursos de diferentes tamaños (hasta 64 lecciones) y con diferente número de estudiantes (hasta 500 estudiantes). Cursos que contienen hasta 64 lecciones y con 300 estudiantes son personalizados en menos de 900 segundos lo que es un excelente resultado. Desde una perspectiva más práctica, hemos incluido una demostración real de el funcionamiento de nuestra propuesta implementando y personalizando en Moodle un curso real de Física de la Universidad de Calabria (Italia).

Esta tesis presenta una propuesta basada en planificación en Inteligencia Artificial para la generación de rutas personalizadas de aprendizaje en Moodle. En particular se explica cómo generar y visualizar rutas de aprendizaje personalizadas en Moodle. Es importante tener en cuenta que en el sistema presentado se puede utilizar cualquier planificador estándar y que la propuesta completa puede ser fácilmente extrapolada a cualquier LMS estándar. Además, gracias a la flexibilidad de la propuesta, es posible adaptar y personalizar nuestras ideas a cualquier tipo de curso en cualquier tipo de LMS.

Contents

Abstract	iii
Riassunto	vii
Resumen	xi
1 Introduction	1
1.1 E-learning	1
1.2 LMSs, Learning Management Systems in current education . .	2
1.2.1 Motivation of our work: the importance of curriculum personalization	3
1.2.2 A simple motivating example	4
1.2.3 How to personalize curriculums	4
1.3 Objectives	5
1.4 Structure	7
2 Background in e-learning systems' personalization	9
2.1 Introduction	9
2.2 Curriculum personalization in e-learning	10
2.3 Supporting learning paths: from creation to execution and mon- itoring	11
2.3.1 Supporting learning paths' creation	12
2.3.2 Supporting learning paths' execution and monitoring . .	13
2.3.3 Summary	14
2.4 Automatically building a learning path	15
2.4.1 Automatic learning path's creation without practical ap- plications	15
2.4.2 Automatic standalone learning path's creation	16

2.4.3	Automatic learning path's creation imported in existing LMS	18
2.4.4	Summary	19
2.5	Conclusions	20
3	Techniques to automatically customize e-learning systems	25
3.1	Introduction	25
3.2	Evolutionary algorithms	26
3.2.1	Genetic algorithms	27
3.2.2	Other evolutionary algorithms-based techniques	28
3.3	AI planning	29
3.3.1	What is AI planning	30
3.3.2	PDDL, a Planning Domain Definition Language	30
3.3.3	Planning vs. e-learning	31
3.3.4	A PDDL example	32
3.4	Other techniques	34
3.5	Conclusions	36
4	AI planning for curriculum personalization in Moodle	39
4.1	AI personalization within a LMS: general overview	39
4.2	Why Moodle	42
4.2.1	Moodle vs. other LMSs	42
4.2.2	Moodle's general description	44
4.2.3	Moodle in detail	46
4.3	Introduction to our mapping activity	49
4.4	General mapping and Moodle's limitations	49
4.4.1	General mapping	49
4.4.2	Moodle's limitations	51
4.4.3	Building a course by using Moodle's lessons	56
4.5	Moodle-dependent mapping	57
4.5.1	From Moodle's lessons to PDDL constructs	58
4.5.2	From Moodle's database to PDDL domain+problem files	59
4.5.3	Plan generation	61
4.5.4	From planning solutions to Moodle's elements	61
4.5.5	From planning solution to Moodle's database	62
4.6	Some notes on monitoring	62
4.7	Conclusions	63

5	Experimental evaluation	77
5.1	Introduction	77
5.2	A quantitative evaluation	78
5.2.1	Introduction	78
5.2.2	Small course	79
5.2.3	Medium course	82
5.2.4	Big course	87
5.2.5	Very big course	92
5.2.6	Summary	97
5.3	Conclusions	102
6	A real case study: personalizing a Physics course in Moodle	105
6.1	Students' difficulties in learning Physics concepts	105
6.1.1	The body motion	106
6.2	The Physics course: "Il moto dei corpi"	107
6.2.1	Course's structure and adaptation in Moodle	107
6.2.2	Plan generation and visualization	111
6.3	Conclusions	118
7	Conclusions	121
7.1	Contributions	121
7.2	Conclusions	123
7.3	Future lines of research	124
7.4	Pubblications	124
7.4.1	Submitted works	125
	References	127

Chapter 1

Introduction

In this Chapter we provide some interesting notions as introduction to the contents of the entire document. In particular, we illustrate the concept of e-learning and, therefore, we explain (thanks also to a short practical example) the importance of customization of educational on-line courses (which is the motivation of our work) and the way in which it is possible to achieve this objective. In conclusion, we provide both the objectives and the structure of the whole work.

1.1 E-learning

The increasing use of information technologies (IT) has made e-learning (i.e. the Internet-enabled learning [29]) more and more popular as a way to promote distance education. In an e-learning system, learning materials (or learning activities) are electronically provided to a remote learner (or student) through computer networks. Thus, e-learning has revolutionized the traditional learning paradigms by introducing new dynamic models [42] which take advantage of information, computing and telecommunication technology, together with a wide range of electronic multimedia uses. Furthermore, the application of multimedia tools have a great impact on education by supporting (and facilitating) learning, and their usage within e-learning makes the learning process friendly to students, who interact with teachers in a better way than in traditional classroom teaching [46].

E-learning permits us to remove the barriers of time and space, which are characteristic of traditional teaching worldwide, because the access to a course is now possible by a simple connection to Internet. Moreover, e-learning

proves to be a great equalizer because, in addition to eliminating the space-time barrier also eliminates the socio-economic status of the participants in the learning process [29].

In addition, e-learning makes it possible to better monitor the learning progress of the students thanks to the possibility to use online assessment methods (whose validity has already been demonstrated in [31]). Monitoring a learning process can result very useful for both students (who can constantly realize their progress) and teachers (who, by having a clear vision of the learning process of each student, are aided in the evaluation activity).

As well as making education more easily accessible, e-learning also brings great challenges to students (in terms of perception of the effectiveness of new tools to support the learning activity) and teachers (in terms of attitudes regarding new tools to support the teaching activity) [42].

Among the benefits achieved from e-learning we can identify a defined accountability for all participants in the learning process and a reduction in costs in the face of a faster learning process. Furthermore, e-learning increases productivity, profitability and loyalty of the learning process [29].

1.2 LMSs, Learning Management Systems in current education

E-learning requires two kinds of activities: communication activities (e-mail, forums, conferences, on-line blogs, etc.), and exploration activities (navigation of contents). These activities usually take place on a LMS (Learning Management System). LMSs, such as Moodle (<https://moodle.org/>), Claroline (<http://www.claroline.net/>), ATutor (<https://atutor.ca/>), dotLRN (<http://www.dotlrn.org/>), Dokeos (<http://www.dokeos.com/>), etc., are web-based software generally used to design, provide, re-use and manage learning activity usually combined as courses (or e-courses). A LMS offers the enrolled students a vast number of courses with highly customizable capabilities and also offers the teachers a number of combinations of possible modules to compose a didactic course in a suitable way. LMSs are increasingly being used by schools, universities and companies as powerful means to carry out educational activities and to improve teaching activities.

A LMS not only provides learning contents but it also manages user registration for courses and provides tools for the courses' administration, tracking, reporting and skills gap analysis [26]. In this way, the teacher is greatly aided in monitoring and evaluating students' activities. Moreover, thanks to the

communication tools (forum, chat, etc.) generally provided by a LMS, the learning activity of a student is facilitated by the possibility of continuous interaction with the teacher or with other students within a course.

1.2.1 Motivation of our work: the importance of curriculum personalization

In current society and education, the great risks in using LMSs are not to exploit them up to their full potential, but to simply use them as mere “repositories” of learning contents. For the best use of these contents, it is fundamental not to consider LMSs in an isolated way but as part of a much larger system in which learning contents are aggregated for the construction of courses that can be fully personalized or adapted to the specific needs and abilities of each student.

In fact, it is important to note that each student is characterized by his/her own traits (profile, learning style, prior background and learning goals). In other words, by considering a learning path as a set of learning activities that a student needs to perform to achieve a certain knowledge level (or learning goal), it is not enough to plan a general learning path for all the students enrolled in a course, but it is necessary to provide each student the most adequate learning path to attain his/her learning outcomes by starting from his/her initial knowledge level [24]; [54]. Thus, learning paths should be student-oriented, or planned to meet the individual characteristics of each student. Personalized learning is a critical factor not only to promote the learning performance of each student but also to protect him/her from cognitive overload and disorientation [54].

On the other hand, because of the massification of education, teachers’ attention is increasingly becoming a scarce resource [70] and consequently the autonomy of each student in performing his/her own learning path is fundamentally important. Thus, what is essential for an e-learning system is, first, to identify an individual learning path for each student enrolled in a specific course, and second, to provide him/her the maximum possible autonomy within the phase of the learning activities’ execution. In order to work at maximum potential, a LMS should [67]:

- help students and teachers to understand what are the most appropriate learning goals for each one;
- provide a means of testing student’s abilities and knowledge level;

- support collaboration and generate reports to provide useful information to the continuous improvement of the learning process;
- specify the proper learning path for each student;
- monitor the progress of each student.

1.2.2 A simple motivating example

We motivate the necessity of curriculum personalization by using a simple example. Let us imagine that two students, Paul and Kate, enroll on an Italian course. The course consists of three sequential modules (corresponding to three different learning levels): Elementary, Intermediate and Advanced. It is possible for a student to take the entire course or just a part of it. Let us suppose that Paul has a sufficient Italian knowledge and he only wants to improve his grammar. Kate, however, has already a good Italian level but she wants to learn to speak more fluently. Certainly, it makes no sense to design the same learning path for both students. It is necessary to plan for Paul a path that only includes the “Intermediate module” and for Kate a path that includes the “Advanced module”. Starting from these considerations, it is necessary to find (and to put it into practice in a LMS) the best learning path so that each student achieves his/her learning objective, starting from his/her initial characteristics to evaluate the progress of the student.

In addition, a good planning activity should be accompanied by a good monitoring activity of the learning process. In fact, while a student is following a certain learning path, such a path could eventually need to change because of discrepancies between expected and real results, updates on the learning objectives, etc., and a re-planning of the path (in part or whole) may be necessary.

1.2.3 How to personalize curriculums

The need for systems that automatically build student-oriented learning paths by combining appropriate learning contents has become more and more intense in the last years ([5]; [14]; [24]; [38]; [54]). Generally speaking, literature abounds with works to exploit techniques on nearly all aspects of e-learning.

There is a variety of studies that face the problem of curriculum personalization in different ways, without focusing on a specific LMS. For example, the study in [19] shows three different strategies to automatically detect and

exactly adjust students' learning styles, by taking into account students' performance. Another proposal ([40]) highlights that the use of digital technology in teaching is not always optimized and suggests the Learning Design Support Environment project as a way to enable the teachers to develop and test their learning ideas in terms of effective learning design. A number of works ([12]; [13]; [64]) apply a genetic algorithm approach to customize and personalize course generation. The results of these works are promising, but their application to standard LMSs can be difficult. From a perspective based on the design, analysis and scoring of tests, the personalization of e-learning systems has been approached by using the Item Response Theory (PEL-IRT) which, by considering the difficulty of the learning materials to be provided and the ability of the students, finds personalized learning paths [15]. Another work, based on the students' results of pre-tests, has led to a genetic-based customized e-learning system which conducts to a personalized curriculum sequencing [14]. Other authors combine a personalized multi-agent e-learning system based on item response theory with artificial neural networks and soft computing methods in order to customize learning systems [5]; [9]; [32].

Several works use also Artificial Intelligence (AI) methods in order to identify student-oriented learning contents. In particular, the prediction of the students' behavior in order to help in the decision-making teaching procedures in open and distance education has been considered by using Bayesian networks [68]. This work takes into consideration general students' behavior without focusing on specific learning contents. On the other hand intelligent planning has been used for learning paths' personalization ([24]; [38]). These works focus on creating a planning ontology from the e-learning information and use standard planners to solve the problem.

As a summary, several works face the problem of curriculums personalization in e-learning systems by using various techniques. None of these studied is really focused on the automatic customization of learning contents within an existing LMS. On the contrary this kind of personalization is the main objective of this study and we detail it in Chapter 3, where we provide a thorough review of these and other works.

1.3 Objectives

The idea that underpins this work builds on the general proposal of [24] and extends the results presented in [10] to offer now a thorough design, development, implementation and testing of intelligent personalization in Moodle (Module

Object-Oriented Dynamic Learning Environment, <https://moodle.org/>).

Particularly, in our work the personalization of an e-learning path is faced from the point of view of AI planning through the automated compilation of e-learning models. We have **fully adapted** the knowledge engineering planning mapping introduced in [24] to be **directly used in Moodle**. Moodle is a platform that includes a constructivist and social constructionist approach to education, emphasizing that students (and not just teachers) contribute to the educational experience. Consequently, Moodle facilitates the interaction among students in real time by permitting the exchange of views and the sharing of knowledge and difficulties while taking the courses.

We detail in our work an automated way to bridge the gap between the course's model implemented in Moodle and the planning model for supporting contents personalization, which means the generation of student-oriented learning paths. There are some goals that result essential to derive the greatest possible learning benefits. In particular, it is necessary to find:

- a transparent way to translate from the Moodle's e-learning insights (lessons, background and learning goals) to the planning ones (actions, initial state and goals), and from the plans to Moodle;
- a seamless procedure to run an intelligent planner to personalize as much as possible each learning path, depending on each particular student;
- the possibility to re-plan the learning paths to adapt them to new scenarios.

Once implemented in a Moodle a system for the automatic generation customized learning paths, it is fundamental to:

- evaluate the scalability of the system by trying to test it with courses of different sizes and a copious number of enrolled students;
- implement a real course in Moodle in order to generate personalized learning paths for students enrolled in it;
- provide a qualitative evaluation of our personalization system.

The explanation of how we face and achieve these goals is the main objective of this document, in which we also provide some experimental results to evaluate the scalability and feasibility of our system.

1.4 Structure

The structure of the document is incremental.

Once introduced the document's content, for reasons of convenience we organized the related work in two Chapters. In particular, in Chapter 2 we discuss in general the different levels of curriculum personalization that we can find in the literature, by highlighting strengths and weaknesses. We also identify in this Chapter the challenges that still remain in the field of customization of curriculums. Next, in Chapter 3 we illustrate in a more practical way (techniques, algorithms, etc.) the main ways to automatically generate personalized learning paths, by highlighting AI planning methods as the way that we have chosen in order to develop our own personalization system.

In Chapter 4 we present a way to automatically plan student-oriented learning contents in Moodle through intelligent planning. In particular, we explain what are the difficulties (that we faced and solved in the adaptation of our system within Moodle). Thus, we concretely illustrate how to automatically generate personalized learning paths within Moodle by using intelligent planning methods. We present a way to generate customized views in Moodle, depending on specific learning paths. In addition, we provide some notes on the monitoring activity which it is possible to perform in order to eventually re-plan the learning paths.

In Chapter 5 we carry out a quantitative evaluation of our personalization system. In particular, we provide a measure of the scalability of the system by generating personalized learning paths for various students' quantities enrolled within courses of various dimensions. By means of these experiments, we demonstrate the scalability and viability of our approach.

A real implementation of our system is presented in Chapter 6, in which it is possible to find personalized learning paths for students enrolled in a real physics course of the University of Calabria (Italy).

Finally, in Chapter 7 we present the contributions of our work and we also discuss about our future line of research.

Chapter 2

Background in e-learning systems' personalization

In this Chapter we identify the most recent approaches that we can find in the literature with respect to the problem of curriculum personalization within e-learning systems. In particular, here we explain which are the different personalization's levels that it is possible to find in the literature by highlighting strong points, weaknesses and any possible improvements.

2.1 Introduction

The investigation in the field of personalizing e-learning systems is relatively young. Nowadays, a lot of studies focus on developing learning systems with customized learning mechanisms in order to improve the learning performance of each student. The e-learning community is already starting to determine the latest wave of next generation e-learning system [53]. We individuated three personalization levels within an e-learning system:

- 0-personalization;
- supporting learning paths' creation, execution and monitoring;
- automatically building a learning path.

In particular, in an e-learning system it is possible to pass from a level of 0-personalization (where the background and the learning goals of each individual student are not considered at all) to a partial customization system

(in which students and teachers are supported, respectively, in the learning activities and in the teaching activities) up to an automatic learning paths' personalization system (where each student is provided a learning path tailored to his/her own specific needs).

In the following Sections we illustrate the three main different curriculums personalization's levels by detailing the works related to "supporting learning paths' creation, execution and monitoring" and "automatically building a learning path" that we found in the literature.

2.2 Curriculum personalization in e-learning

In an e-learning system we can identify three different levels of curriculum personalization. If all students enrolled in an e-course have to perform the same learning activities, it is possible to talk about 0-personalization. In this case, the teacher uses a LMS as if it was a simple repository of the most used learning materials for a specific course. Therefore, students with different needs are forced to perform the same learning path within a course and they are exposed to cognitive disorientation and information overload. The learning phase is not optimized in any way (the student is not supported as much in choosing the learning material to be taken as in its execution).

On the contrary, in a second level the curriculum personalization can be manual (as in the traditional teaching), that is structured by the teacher depending on the information he/she has about students and learning activities (see Fig. 2.1). Generally, a LMS provides tools (lessons, forum, chat, wiki, reports, etc.) in order to support the teacher to design and monitor the execution of the learning activities contained in a course and the student to perform all the courses in which he/she is enrolled. There are some proposals in the literature that focus on the useful support (also in the form of tools to be improved or optionally inserted in existing LMSs) that can be provided to a teacher (to properly design and monitor the execution of the e-courses) and to a student (in order to successfully execute his/her learning path). This kind of approach does not provide a real curriculum personalization but only suggestions to eventually achieve and execute it.

The customization of an e-course can be also automatically provided, as in a third level, by a system which considers as inputs the various learning activities of the course (in terms of timing, learning concepts associated with each activity and possible priority ordering between activities) and the own characteristics of each student (in terms of background, learning style, learning

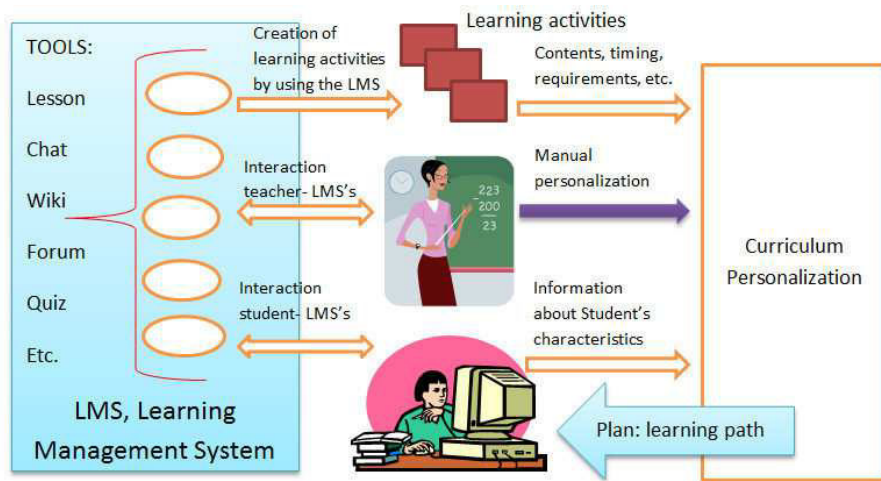


Figure 2.1: Curriculum personalization in e-learning systems.

goals, etc.) By using this information, a system can provide each student a proper learning path (see Fig. 2.1).

The most important contributions in the literature are focused on the automatic learning paths' generation. In particular, several studies (by using a number of different techniques) present a way to automatically detect and offer each student the most appropriate learning path for his/her needs.

2.3 Supporting learning paths: from creation to execution and monitoring

As already mentioned, there are a number of studies that deal with the support that can be supplied to the teacher regarding the design and monitoring of the learning paths, and to the student regarding the execution of the learning activities contained in his/her learning path. Ideally, these studies can be divided into two sub-categories: those that provide support in the learning paths' creation process, and those that offer support in its execution and motorization. By considering that, although they do not offer a way to generate an automatic personalization, these works offer valuable support to the student/teacher in the learning/teaching activities, we analyze some proposals present in both approaches.

2.3.1 Supporting learning paths' creation

Designing a learning path is not a simple activity because it requires a lot of information about the learning activities to propose (in terms of timing, difficulty, prior background required, etc.) and about the learning profile of each student (in terms of learning style, initial knowledge, learning goals, IT equipment, etc.) There are some approaches in the literature whose scope is to aid the teacher to achieve the most important information in order to properly create a learning path for each student enrolled in a course.

Data mining is one of the techniques proposed with the aim to support the teacher in learning paths' creation. In particular, it is possible to use a tool which applies data mining techniques to preprocess LMS's data (in terms of activities already performed by the students) in order to obtain information regarding course visualization, statistics, clustering, and classifications [56]. By using this kind of information properly combined with the information regarding to the learning activities' features, a teacher can correctly create personalized learning paths within a course. Data mining techniques have been also used to introduce a complementary technology to extract similar features (learning styles, intellectual traits, learning material preferences, etc.) of ex-students' enrolled in a course and use them to generate suggestions in order to design customized learning curriculums for current students [37]. This approach is totally theoretic and it does not provide a practical application within any LMS.

OLAP (Online Analytical Processing) techniques have been used to analyze LMS's data to highlight the attitudes of each student as valuable information to design adaptive learning paths [60]. Time spent in the various activities, student-student, student-teachers, and student-learning content interactions are part of the data analyzed by the automated scaffolding technique (OLAP-based) proposed. Also, a recommender system which identifies the groups of students with the same learning style has been proposed with the objective to aid the teacher to design different learning paths within a course [34]. Students have to complete a questionnaire that is used to identify common learning styles and with this information it is possible to carry out a students' clustering operation as a way to offer different learning contents to different learners' categories.

Reinforcement learning (RL) techniques have been used to propose a probabilistic, dynamic and automatic way to identify and adjust students' learning styles [19]. This approach takes into account students' performance in order to identify useful information to properly design customized curriculums.

Finally, neural networks have been proposed as a way to identify students' useful information to model their profiles in order to choose the adequate material to design personalized learning paths. For example, a multi-layer feed-forward (MLFF) neural network has been proposed to collect students' browsing behaviors with the objective to identify students' cognitive styles as a way to aid in designing an adaptive web based learning system [44]. In fact, once identified the students' cognitive styles, it is possible to use them with the objective to design adaptive web interfaces, containing a number of learning activities and interactive components. It is also possible to use neural networks to predict students' course selection as a critical factor to identify students' satisfaction in online higher education schools [36]. This approach suggests course difficulty, course characteristics, course grade, and teacher's characteristics among the critical factors in choosing courses.

Although all these approaches offer the possibility of identifying the useful factors to properly select the learning materials to include in students' learning paths, it is obvious that they do not offer a way to manage and monitor the execution of the learning paths and they do not propose a clear way to automatize the generation of customized curriculums, which is entirely delegated to the teacher.

2.3.2 Supporting learning paths' execution and monitoring

Several studies focus on the idea that it is possible to give assistance to both teacher and student during the phase of execution/monitoring of a learning path. In particular, a teacher can be assisted in evaluating the student's progress and a student can be assisted in properly performing the learning activities contained in his/her learning path. In addition, some other approaches focus on the importance of modularity and extensibility for a LMS as a way to offer the teacher and the student a comfortable and personalized environment where eventually generate, execute and monitor the execution of a learning path.

We present the most meaningfully proposals that it is possible to find in the literature with respect to both approaches. Various works offer tools to be integrated into the existing LMSs in order to assist students and teachers in executing and monitoring the learning process. An annotation system that allows students to annotate, search, share, and classify learning materials has been proposed for a direct implementation into a LMS in [16]. Also a personal recommender system has been proposed to suggest students the most suitable learning material to choose within a number of learning activities [21].

Another study proposes a recommender system to analyze students' learning performance in order to provide each student real-time information about his/her productivity and learning progresses [35]. This kind of information can result very useful to support the monitoring activity of the teacher. Other tools to provide the students the possibility to auto-evaluate their progresses are self-assessment quizzes based on a web-based authoring tool that can be integrated in Moodle [69] and a tool [70] to structure, answer and crossing evaluate (by blind peer review) questions in relation to a specific topic.

Among the approaches for the creation of personalized learning environments it is possible to find the theory of mobile learning [63] extended by [11] to make the learning process more student-centered. In particular, in this work the Moodbile Project is proposed, which is a way of redesigning some Moodle's modules to be integrated into mobile devices and therefore to make them usable everywhere by students and teachers.

Another way to create customized learning environment is the PLE (Personal Learning Environment) approach that proposes the possibility of using only the individually learning tools that are considered useful by the student/teacher. In this way, each student/teacher can build his/her own customized e-learning platform [61]. This approach has been extended by [53] to propose the possibility to install single learning tools (as LMS's single modules) on mobile devices and by [57] to present a third generation LMS as a totally modular software whose (also individually) modules can be imported and used into other LMSs.

These studies provide useful tools to support students and teachers, respectively, to execute and monitor curriculum learning paths and ways to create personalized learning environments as comfortable environments where, eventually, perform teaching and learning processes. Nevertheless, none of these proposals can be applied to the creation of personalized curriculums but they can only offer a support in the phase of execution/monitoring.

2.3.3 Summary

As pointed in Table 2.1, some of the analyzed proposals offer useful ways to help the teacher in the process of data analysis (identification of students' characteristics in relation to the learning material to be proposed) necessary to properly design personalized learning paths, while other works offer ways to support the student/teacher in the execution/monitoring of the learning paths. Other approaches, however, are focused on the creation of personalized learning environments (such as, for example, the possibility of using a mix of

modules originated from different LMSs) so that the student/teacher can work (execute/monitor a learning path) in the context that best pleases him/her. None of these studies propose a way to automatically detect learning paths. Therefore, we can consider them only as valid ways to support personalization but not as complete proposals for customizing curriculums. In addition, Table 2.1 shows that not all the analyzed proposals make use of existing standard LMSs (which are the most used tools by the whole world of distance education).

2.4 Automatically building a learning path

There are many approaches in the literature about the possibility of offering systems that can automatically build customized e-courses. The most important distinction that we can find among these proposals relies on the more or less theoretical nature that they can assume in addressing the problem of personalization. In fact, some studies present algorithms to derive personalized learning paths but have no practical application (that is, we cannot find a concrete use case on any internet platform), while other works experience the construction of personalized e-courses by creating online platforms or simply using existing LMSs. In this Section we discuss about the most recent examples that we found in the literature.

2.4.1 Automatic learning path's creation without practical applications

It is possible to identify several studies that detect the most appropriate set of activities to each student enrolled in a course only in a theoretical way, that is, without applying the curriculum customization in practice within any existing LMS or any other online platform.

Various simulation-based experiments have been proposed by using an approach based on improved particle swarm optimization algorithms along with digital pheromones in order to create a customized course composition [59]. On the other hand, another study [28] demonstrates with computer simulation the good quality performance and the minimum error in applying the shuffled frog-leaping algorithm in order to compose appropriating learning materials into e-courses. Both approaches give good results but provide a course composition without sequencing the learning activities.

Item response theory and artificial neuronal networks have been used with

the objective to define students' abilities as a way to build customized curriculums [5]. Also in this case, the personalized recommendation has a theoretical nature and has not application in existing software.

The approaches based on genetic algorithms are other ways widely used to derive the personalization of e-courses. In particular, it is possible to apply one of these studies by considering the difficulty of the learning material to provide the time spent by each student on each learning concept, the learning goals of each student and the possible changes in his/her performance by executing the learning path [64]. In this case, a generic course is defined and it is possible to customize it by considering the student-learning contents interactions. Another study based on the genetic algorithm in conjunction with forcing legality has been used in an adaptive learning system in order to offer each student the most appropriate learning activities' composition as a learning path by excluding all the inappropriate learning materials [12].

The adaptive content sequencing is a technique that evaluates student's knowledge level and determines if the concepts associated with the learning activities of a course (which are represented inside a directed graph) are suited to his/her own skills. By considering the student's initial learning state and his/her attitudes, a learning path it is recommended to him/her where he/she is guided, step by step, about the next task to perform. This approach has been tested only in a fictitious way (Mathlab)[62].

Despite the techniques just mentioned above can be very useful to obtain the correct set of learning activities to be presented to each student, the lack of a practical application makes these proposals difficult to be adopted massively in schools, universities, companies, etc. The practical application by means of an existing LMS would make this kind of works immediately usable by the entire world of education.

2.4.2 Automatic standalone learning path's creation

It is possible to find several works that focus on the personalized learning paths' creation and evaluation through the use of e-learning platforms specifically created in order to put into practice individual approaches. In this case, the various authors preferred implement their works on standalone platforms rather than in already existing LMSs.

INSPIRE is the prototype based on the Active Server Pages (ASP) technology designed to support a curriculum personalization's system which considers learners' knowledge level through a qualitative model [54]. This model is based on the learning ability level demonstrated by the student in relation to specific

concepts as a way to define his/her current learning state. The system finds a learning path that is a sequence of lessons to be proposed to each student by considering student's current learning state and the goals that he/she selects from a set of possible concepts to be acquired. The system also monitors the progress of each student and supports him/her during the whole learning process.

PASER is the system created to automatically design personalized learning paths by using AI planning methods and semantic web technologies [38]. PASER is a system that can store, manage and organize teaching materials by using its three main modules: a metadata repository, a deductive object-oriented knowledge base system and a planning system. At the time of registration in the platform, the student can input, in addition to the personal details and technical specifications (operating system used, kind of browser, etc.), his/her knowledge level about a given subject. The next step consists in choosing a set of learning goals to be achieved. Once in possession of such information, the system is able to derive a personalized learning path that can guide the student to reach the desired knowledge level by starting from his/her own background.

PELS is the tool used to implement genetic-based customized e-learning systems [14], which is focused on an approach based on genetic algorithms to design personalized curriculums. In particular, in order to create the appropriate curriculums composition, the system considers courseware difficulty and pre-tests results. Pre-tests, based on learning concepts randomly selected within a course, are previously performed by each student and serve to collect the incorrect learning concepts used as parameters in the application of genetics algorithms to structure each personalized learning path.

PerLE is a personal learning environment developed to apply an ontology-based methodology that decomposes teaching material into learning units and re-composes them dynamically depending on the specific needs with the objective to create personalized learning paths [2]. At the time of enrollment in the system, each student fills out a registration form where inputs his/her initial learning state and his/her learning goals. In addition, student has to complete another test that the system needs to trace his/her cognitive profile (perception and a way to process the learning material). Thus, the system is able to structure a personalized learning path for each student.

PLCP is a system based on e-learning Decision Support System (DSS) techniques and suggests each student the correct learning path to perform [33]. PLCP collects students' profile data, starting from initial learning prior-

ities (related to a specific course) expressed by each student at the moment of the registration in the system. Furthermore, the system considers useful information such as the results obtained by the student in performing the learning activities. These results are registered in the system as historical student's learning data and they are used along with the initial preferences expressed by the student to obtain a personalized learning path.

PCLS is based on the data mining technique of decision trees and it is a useful tool to provide customized curriculums in order to optimize the creativity performance of each student [43]. The system proposes each student enrolled in a course a random learning path to perform as a game. The student has to solve several problems in the game and the system collects a variety of information regarding his/her creativity in carrying out the learning activities. The decision tree algorithm is then used by the system to analyze this information and to create a personalized learning path.

These approaches can be considered very useful in order to derive personalized learning paths for students enrolled into e-courses. The teacher does not have to make efforts to analyze the specific learning traits of each student and then to model the learning activities according to the students' specific needs. On the contrary, by using this kind of approaches the system takes charge of storing students' learning data in order to automatically design customized curriculums. Despite this, standalone systems are not widely used by the whole community of distance education. In fact, distance education takes place primarily by using LMSs such as Moodle, ATutor, dotLearn, etc. Therefore, the utility of the analyzed systems is limited by the poor usability in the entire world of e-learning.

2.4.3 Automatic learning path's creation imported in existing LMS

Surely, the more interesting approach that it is possible to identify in the literature is related to the automatic construction of personalized learning paths and subsequent application into existing LMSs. There are several approaches that focus on this kind of personalization.

A lot of studies involve the use of techniques in order to derive the correct sequence of learning activities to be performed by each student. It is possible, for example, to use an approach based on evolutionary algorithms to generate customized learning paths to be imported (as pre-configured courses) into Moodle as SCORMs (Shareable Content Object Reference Model) e-learning standards [41], in order to investigate the satisfaction of the students. This

approach analyzes students' initial learning states and learning goals by using the maximum likelihood estimation, evaluating the learning time of each student and adjusting the difficulty of the learning activities to properly design the learning paths. Another way to obtain personalized e-courses, as mentioned above, is the application of techniques based on genetic algorithms. In particular, by implementing an e-course authoring tool based on an approximation based on genetic algorithms, it is possible to generate customized e-courses and to import them into Moodle (as SCORMs) in order to verify the learning achievements of each student [13]. This kind of approach has been compared with the proposal concerning the importation into Moodle of personalized e-courses generated by using the particle swarm algorithm (PC2PSO) in conjunction with an intelligent tutoring system [17]. In particular, this study investigates students' learning outcomes and students' satisfaction and, by making a comparison with genetic algorithms, shows an higher efficiency and stability.

The automatic generation of plans (i.e. learning paths) to be subsequently imported into an existing LMS has also been proposed by using AI existing domain-independent planners [23]. In particular, this approach proposes to collect all appropriate information from a LMS concerning the characteristics of the students (initial learning states and learning objectives) and the characteristics of the learning activities (associated concepts, any priorities, timing) and use them, respectively, to compile a problem and a domain to be provided to a standard planner in order to generate a plan (a set of learning paths, one for each student) to be imported back into the LMS.

All these proposals have the advantage of using approaches to find very valid solutions in relation to the personalization of learning paths and also to implement them into one of the LMSs platform most used in the world (Moodle) so they can get wide use in the field of distance education. Ideally, however, the best achievement would be the possibility of having a personalized learning paths' generation system directly within a LMS. In fact, in this way it would give the teacher the possibility to directly interact with only one piece of software that could process real-time data and then create, or eventually modify, the most adequate learning path for each student according to his/her own specific needs.

2.4.4 Summary

Table 2.2 shows the main differences among the contributions coming from the various works found in the literature regarding the automatic generation

of learning paths. In particular, it is possible to observe that, except for a few theoretical proposals, several works propose the creation of standalone LMSs that automatically generate customized curriculums or propose the importation of (previously created) personalized curriculums within an existing LMS. No proposal focuses on the possibility of creating a system for the automatic generation of learning paths within an existing LMS (which therefore has a large influx of users throughout the whole community of e-learning).

2.5 Conclusions

In this Chapter we have proposed a study of the literature related to the topic of personalization in e-learning, by identifying various proposals. Ideally, a personalized e-learning system should provide each student with the most appropriate combination of learning activities (i.e. learning path), depending on his/her specific needs (in terms of learning goals, initial learning state, learning style, preferences, etc.) We found a group of proposals that focus on the support that can be provided to a teacher/student in the process of design, execution and monitoring a learning path. These approaches, however, only give support to the teacher but leave to him/her the burden of designing the learning paths.

Therefore, we identified another group of works that focus on the automatic generation of learning paths. In particular, some studies are restricted to provide only theoretical contributions (without practical applications into any LMS), while others works offer the possibility to customize curriculums within standalone LMSs (created in order to prove specific studies). Finally, some studies aim at generating learning paths to be imported into existing LMSs (as Moodle, dotLearn, ATutor, etc.) that are the most used learning platforms by schools, universities and companies around the world. Despite this latter group of works is the most interesting as the most usable by the whole community of distance education, none of the proposals that we analyzed provides the possibility of integrating a system for the automatic generation of curriculums within an existing LMS.

As future proposals, it would be interesting to develop a learning system that effectively covers all the strong points of the works which we have analyzed. For example, the incorporation of a system for the automatic generation of personalized learning paths into an existing LMS would have a great impact in the world of e-learning and, in general, it would improve distance education (the system that we present in this paper has been designed to achieve

this objective). Furthermore, such LMS should be totally modular, in order to plug&play different modules (e.g. modeling, assessment, monitoring tasks, etc.) and to be simply executed in different devices (e.g. desktop computers, tablets and mobile devices).

22CHAPTER 2. Background in e-learning systems' personalization

Proposal	Using a standard LMS	Evaluation of students' features	Support to learning paths' execution/monitoring	Create personal learning environments
Casany, Alier, Mayol, Piguillem, Galanis, García-Peñalvo & Conde, 2012, [11]	✓			✓
Chen, Hwang & Wang, 2012, [16]	✓		✓	
Dorça, Lima, Fernandes & Lopes, 2013, [19]		✓		
Drachsler, Hummel & Koper, 2008, [21]	✓		✓	
Jyothi, Bhan, Mothukuri, Jain & Jain, 2012, [34]	✓	✓		
Kaklauskas, Zavadskas, Seniut, Stankevic, Raistenskis, Simkevičius, & Gribniak, 2013, [35]			✓	
Kardan, Sadeghi, Ghidary & Sani, 2013, [36]		✓		
Knauf, Takada, Sakurai, Kawabe & Tsuruta, 2012, [37]		✓		
Lo, Chan & Yeh, 2012, [44]		✓		
Organero, Kloos & Merino 2010, [53]	✓			✓
Romero, Ventura & García, 2008, [56]	✓	✓		
Ros Hernández, Robles-Gómez, Caminero, Tobarra & Ruíz, 2013, [57]	✓			✓
Sengupta, Mukherjee, Bhattacharya & Dasgupta, 2012, [60]	✓	✓		
Yuuichi, Toshihiro, Seisuke & Hiroshi, 2006, [69]	✓		✓	
Zenha-Rela & Carvalho, 2006, [70]	✓		✓	

Table 2.1: Proposals to support learning paths' creation, execution and monitoring

Proposal	Automatic learning paths' creation	Incorporation into a standalone LMS	Importation into an existing LMS
Altimari, Plastina, Cronin, Servidio, Caria & Pedrazzoli 2012, [2]	✓	✓	
Baylari, & Montazer, 2009, [5]	✓		
Chang, Hsieh & Li, 2010, [13]	✓		✓
Chang & Ke, 2013, [12]	✓		
Chen, 2008, [14]	✓	✓	
Chu, Chang & Tsai, 2011, [17]	✓		✓
Garrido, Fernández, Morales, Onaindía, Borrajo & Castillo, 2012, [23]	✓		✓
Gomez-Gonzalez & Jurado, 2012, [28]	✓		
Jeong, Choi & Song, 2012, [33]	✓	✓	
Kontopoulos, Vrakas, Kokkoras, Bassiliades & Vlahavas, 2008, [38]	✓	✓	
Li, Chang, Chu & Tsai, 2012, [41]	✓		✓
Lin, Yeh, Hung & Chang, 2013, [43]	✓	✓	
Papanikolaou, Grigoriadou, Magoulas, & Kornilakis, 2002, [54]	✓	✓	
Sarath Chandar, Dheeban, Deepak & Elias, 2010, [59]	✓		
Sharma, Banati & Bedi, 2012, [62]	✓		
Tan, Shen & Wang, 2012, [64]	✓		

Table 2.2: Proposals to automatically generate learning paths

Chapter 3

Techniques to automatically customize e-learning systems

In contrast to Chapter 2, where we have shown in a general way the different levels of customizing an e-learning system, in this Chapter we focus more specifically on the different techniques to automatically personalize curriculums that we found in the literature. In particular, we describe the operating mode of each technique by paying particular attention to the intelligent planning techniques.

3.1 Introduction

Although it is possible to face the personalization of e-learning systems at different levels, certainly the approach that provides greater utility (as already explained in Chapter 2) concerns the automatic learning paths' personalization. In fact, the most important objective to achieve in customizing an e-learning system is that a specific learning content can be automatically provided to the appropriate person at the most opportune time.

As already illustrated in Chapter 2, we found in the literature a number of works that face the e-learning systems personalization by proving methods for the automatic creation of personalized learning paths. In the following Sections, we detail these different techniques, such as evolutionary algorithms, neural networks techniques, intelligent planning techniques, data mining methods, etc.

More particularly, we detail the intelligent planning techniques (by illustrating the most important features and by also providing a simple example) as

the way we have chosen in order create a system to automatically personalize learning paths.

3.2 Evolutionary algorithms

One of the most widespread techniques to personalize curriculums in e-learning systems is based on evolutionary algorithms and on their characterizations. These algorithms are based on the concept that given a population of individuals, the suitability of the population evolves according to the natural selection caused by environmental pressures [3]. Each individual represents a candidate solution. By considering an objective function, a collection of candidate solutions is casually created and the objective function is used to determine the suitability of each candidate. Thus, once determined the suitability of each individual, the fitness values are used in order to compose the best set of candidate solutions.

In more detail, the technique of composing a personalized e-course by using evolutionary algorithms is developed in three steps:

1. population's initialization: a population is a set of learning contents (or individuals) and each learning content represents a potential candidate solution. Generally the initial population is create randomly;
2. evaluation of the fitness value of each individual: determination of a fitness function in order to evaluate each individual;
3. generation of the next population: the fitness function's values determine the choice of the most adequate individuals in order to compose the next population.

Once a student starts to cover the learning concepts included in a path, the system needs a feedback in order to eventually adjust the learning content's difficulty or re-set the student's learning goals.

The work in [41] based on evolutionary algorithms proposes a personalization composed by four phases: determination of the course's structure, adjustment of the learning material's difficulty, analysis of students' initial background and learning goals and generation of the learning paths. In particular, in order to determine the learning paths this work considers a four-objective function to be minimized, where the four objectives are:

1. the average difference between the learning contents that a student has covered in his/her personalized e-course and his/her learning objectives. This goal is necessary to assess whether educational content in the course covers the learning objectives of the student;
2. the average difference between the difficulty level of the learning contents present in the learning path and the student's ability. In this way it is possible to assess whether the didactic content's difficulty is appropriate for the student's abilities;
3. the total required learning time of the student by considering the learning time required for each learning concept. This objective let us consider the learning time limitation of each student;
4. the relevance of each learning content. This objective balances the weight of learning contents covered in a learning path.

The fitness function is the sum of the four objectives' values, each one multiplied by a weight that represents the importance of each objective. The weight factor of each goal is established by the teacher according to his/her own aim on the course. The total sum of the weights is 1. This is the fitness function generally adopted by the evolutionary based algorithms.

Among the particularizations of the evolutionary algorithm used to customize e-courses we can find the techniques based on genetic algorithms ([12]; [13]; [14]; [64]), the ant colony optimization ([62]), the shuffled frog-leaping algorithm ([28]) and the particle swarm optimization ([17]; [59]). We discuss about these particularizations in the following Sections.

3.2.1 Genetic algorithms

Genetic algorithms are the most used techniques based on the evolutionary algorithm and consider a chromosome as an individual to compose a population. The fitness value of each chromosome is defined by considering a crossover and a mutation operation [27]. In the case of e-course's personalization a chromosome is a learning content while a population is a learning path. These algorithms are used to solve optimization problems in a systematic way and they involve a number of steps:

1. chromosome encoding and population initialization: each candidate is a chromosome (i.e. a learning content) which consists of a number of

genes (binary, integer or real values). The population, learning path in our case, is randomly initialized as a set of chromosomes;

2. fitness evaluation: it is necessary to establish a fitness function to estimate the value of the generated chromosomes' sequence (or learning path). The fitness value is calculated by considering a number of factors (already mentioned in Section 3.2 and classically considered in the evolutionary algorithms);
3. selection: generally the chromosomes' sequence with the smaller fitness function value has a relatively high probability to be selected for the next generation;
4. crossover: this operation involves the combination of two parent chromosomes (learning contents) in order to generate a child chromosome (offspring) characterized by a better performance (obtainable by combining the good qualities of the two parents' chromosomes). It is a learning content sequence rearrangement and generally it is generated with a crossover probability;
5. mutation: this operation involves the randomly selection of two genes (or chromosomes' subparts) in order to exchange their position in a chromosome;
6. stop criterion: the algorithms cyclically repeat the reproduction, crossover and mutation operations until they meet a stop criterion. In general, after a number of repetitions the stop criterion is set and a near optimal solution (optimal learning path, in our case) can be determined by considering the value of the fitness function.

Experimental results (such as [13] and [64]) have shown that these techniques provide valuable support to the student (by increasing his/her performances in the learning process) as they offer the automatic choice of the only educational material relevant to execute.

3.2.2 Other evolutionary algorithms-based techniques

Among the other techniques based on the evolutionary algorithm we can find in the literature the particle swarm optimization, the shuffled frog-leaping algorithm and the ant colony optimization. The particle swarm optimization is inspired by the swarm behaviour of birds and fishes [41]. In particular, each

bird/fish (or particle) is considered as an individual; it is possible to establish if the particle is near to an optimal value by simply using a fitness function. In the case of e-learning, a particle is a learning content while a learning path is a set of particles. This technique involves a number of steps ([17]; [59]):

1. initial swarm generation (generally it is a random activity);
2. fitness evaluation of particles (by means of the generic fitness function already explained in Section 3.2);
3. determination of the best particles (learning contents) in order to generate the best vector (sequence of particles);
4. updating velocities, where the velocity of a particle is the probability that the dimension of a particles' vector is null, and a stop criterion that may be a certain number of executions.

The shuffled frog-leaping algorithm draws on the particle swarm optimization as a local search instrument: the objective is to obtain a global solution by starting from parallel local searches [28].

Finally, the ant colony optimization is a metaheuristic based on the behaviour of real ants [62]. In this case, a course is represented as an oriented graph where each node is a learning content. A student is randomly assigned to a node and it is verified if the learning concept is compatible with his/her needs and abilities. The objective of the system is to build a path within the graph by simulating student's abilities in performing the learning contents (graph's nodes). The probability of transition from a node to another node of the graph takes into account the current student's knowledge, the time spent in the previous contents and the performance in previous activities.

3.3 AI planning

AI planning methods have been used with success in order to properly assemble learning contents to be provided in a personalized way to the students of an on-line course ([23]; [24]; [38]). We have decided to use this perspective in order to face the problem of curriculums personalization through the automated compilation of e-learning models. In the following Sections we detail the most important features related to these techniques.

3.3.1 What is AI planning

Most of human activities involve some kind of planning of tasks to reach an objective. According to Cambridge dictionary, planning is “the activity of thinking about and deciding what you are going to do or how you are going to do something”. In other words, planning involves determining a sequence of actions that will achieve a goal [58]. Therefore, intuitively, planning is about taking decisions on what is the most adequate action to be executed in every moment.

A planning problem is characterized by an initial state, a goal to achieve and a set of actions. A state is the representation of the properties (and the relative values) of the objects that make up the problem while a goal is a clear specification of the values that the objects’ properties must take. Instead, we can define an action as an instance of a transition function that takes a system state as an input value and leads to a next state. In particular, every action is executable only if all its preconditions (i.e. a set of conditions that must occur in the input state) hold before its execution and leads to a specific effect (i.e. a new state determined from the action’s execution). An action is considered not applicable if at least one of its preconditions does not occur.

Intelligent planning involves the representation of world models, reasoning about the actions’ effects, and on the techniques for efficiently searching the space of possible plans. Thus, given a domain of possible actions, intelligent planning selects a subset of them (e.g. a plan where actions are ordered according to their causal-effect relationships) that, after their execution, allow us to reach an objective starting from an initial state [25].

3.3.2 PDDL, a Planning Domain Definition Language

Planning technology has witnessed incredible advances in the last decades. State-of-the-art planning algorithms deal with problems with hundreds (and even thousands) of actions in a few minutes. In order to unify the definition of planning problems and promote an interchangeable use of planners, a standard Planning Domain Definition Language, PDDL, was agreed by the planning community [25]. In particular, PDDL was developed in 1998 by Drew McDermott and his colleagues [48] and it became the standard language used by the AI planning community [30].

The implicit formalism behind PDDL is the separation of the domain data (to describe a family of similar problems and enhance reutilization) from the problem data. It requires two plain text files. First, the domain file contains

all the actions that could be applied. The semantics of each action is described in terms of:

- a name that, grounded with the values of the optional parameters, acts as a unique identifier;
- an optional duration to model problems where actions have different durations (otherwise all durations are considered unitary);
- a set of preconditions that must hold before the action's execution (i.e. causal precedents);
- a set of effects that are asserted once the action is executed.

Second, the problem file contains the initial state of the world, the goals that need to be achieved by using the actions of the domain and, optionally, a metric to be optimized (such as makespan, number of actions, cost, etc.)

Hence, a domain and an associated problem can be used as input of a planner software, which tries to solve the given planning problem by using some appropriate planning algorithm. The output of the planner is a plan: a set of (totally or partially) ordered actions, some of which may be eventually executed in parallel. Even if the plan's actions are characterized by an execution order, PDDL does not specify this order. The plan allows us to reach the objectives starting from the initial state in an optimal or suboptimal way.

In the following Section we explain how to adapt AI planning in e-learning systems and we provide a simple PDDL example in Section 3.3.4.

3.3.3 Planning vs. e-learning

Metaphorically speaking, the personalization of e-learning paths is analogous to the execution of a planning process. The main elements of e-learning are:

1. the background and student's preferences;
2. the learning outcomes to achieve;
3. the learning contents adapted to the student's profile;
4. the ordering relationships;
5. the specific learning path for each student.

Through an e-learning to PDDL mapping, which will be detailed in Chapter 4, these elements can match, respectively, with the next planning elements:

1. the initial state;
2. the problem goals;
3. the actions;
4. the causal links;
5. the solution plan.

Once a solution plan is generated (containing a set of action to perform for the various students enrolled into the course), it must be processed in order to extract a learning path for each student. Depending from the specific selected LMS, it is necessary to choose the adequate LMS's modules (lesson, text files, etc.) in order to properly represent the learning path for each student.

The optimization process that AI planning offers is also very interesting, because students and teachers often prefer a quality learning path in terms of time, resources usage and/or cost, and not yet another path. The possibility of having a detailed learning path to take makes it easy to perform it for the students and to monitor it for the teachers.

3.3.4 A PDDL example

In order to clarify how we can use PDDL, let us revisit the Italian course considered in our motivation example of Section 1.2.2 and let us imagine that the “Elementary module” is required for the execution of the “Intermediate module”. Let us imagine that the minimum time that a student must spend in “Intermediate module” is 5 hours (300 minutes). In a very general way, that is, without deepening into the specifics of individual activities that make up the entire course, the “Intermediate module” can be represented in as a simple PDDL action (executable by a given parameterized student), with its duration, preconditions and effects, as shown in Fig. 3.1. In particular, we can observe that the Intermediate module's duration is 300 minutes and that the conditions necessary for the student (?s) to start the execution of the module are the following (at start):

- the module must not have been previously performed by the student (not done);

- the student must have already performed the Elementary module (done).

Furthermore, the result (effect) that is caused from the execution of the Intermediate module (at end) resides in the execution (done) of the same module.

Let us assume that Paul and Kate need to take, respectively, the “Intermediate module” and the “Advanced module”. Imagine, in fact, that Paul has already taken the “Elementary module” whereas Kate has already taken the “Intermediate module”. We can represent this information as the initial state, the goals and metric to be optimized (total-time that stands for makespan of the plan) of a PDDL problem, as shown in Fig. 3.2. We can observe in the initial states (init) of the PDDL problem the modules that the two students have already performed while the problem’s objectives (goal) are related to the modules that Paul and Kate expressed as learning goals. Moreover, the problem suggests (metric) identifying the learning paths for which the total execution time is minimized.

```
(:durative action intermediate
:parameters (?s - student)
:duration (= ?duration 300)
:condition (at start (and
                (not (intermediate_done ?s))
                (elementary_done ?s)))
:effect ((at end (intermediate_done ?s))))
```

Figure 3.1: The representation of the “Intermediate module” as a PDDL action of the domain.

```
(:init
    (elementary_done Student_Paul)
    (intermediate_done Student_Kate))
(:goal (and
    (intermediate_done Student_Paul)
    (advanced_done Student_Kate)))
(:metric minimize (total-time))
```

Figure 3.2: The PDDL problem for Paul and Kate.

3.4 Other techniques

Among the other techniques used with the objective to personalize an e-learning system we can find the application of the item response theory and artificial neuronal networks [5]. By starting from the results of a test (based on item response theory), the system generates recommendations (by using an artificial neuronal network technique to analyze the collected data) in order to find the most appropriate sequence of learning contents for each student. The system involves three kinds of tests: pre-test, post-test and review-test. These tests contain multiple choice items and they can be re-used. The system architecture integrates a multi-agent system into distance learning and involves four kinds of agents:

- activity agent: collects some necessary information as regarding learning contents, students' on-line activities and learning time of each student and determines the student's profile;
- planning agent: considering various learning sessions (composed by a number of learning contents), at start of each of them, the planning agent asks the student (through testing) if he/she has already acquired some knowledge about the learning contents of the session. In the positive case, the planning agent decides which part of the session is necessary to submit to the student (learning content already known will not be displayed). At the end of each session, the student has to complete an appropriate post-test (in order to establish his/her new abilities) and after a number of sessions the planning agent may require the student to complete a review-test in order to repair any possible problems;
- test agent: is the responsible of the tests generation. Thus, by selecting the most adequate material from the system's database (information about student's abilities, learning activities, etc.), this agent presents the right test (pre-test, post-test or review-test) to each student;
- remediation agent: by analyzing the results of the review test, this agent detects any learning problems and recommends the student the most appropriate learning material to take. This agent acts as a teacher by using information contained in the system's database.

A decision support system is a technique used with the scope to customize curriculums [33]. In particular, a personal learning course planner (PLCP)

can provide an e-learning course organized according to the learning implicit requirements of each student. The PLCP involves various kinds of agents with different roles. In particular, the student can input his/her learning goals within an e-course through a priority input agent. Thus, a user profile management agent analyzes student's priorities that can come from the priority input agent or from the results analysis agent which collects the concepts learned by the student as a result of an e-course and stores them into a user profile database. Thus, the course contents management agent structures the learning course by analyzing information contained into the courseware database. Finally, the e-learning decision support system, by using the data coming from the course contents management agent and from the user profile management agent, can provide each student a personalized learning path (through a course recommendation agent).

Data mining techniques of decision trees are other ways to generate personalized learning paths within an e-course [43]. In this case the personalized e-course is based on self-perception of creativity, learning style, and other useful student's characteristics. Also this work is based on a multi-agent system. In particular, the system involves four agents:

- user interface agent: collects general information about the student who has to fill out a questionnaire by highlighting his/her name, gender and previous studies;
- creativity game agent: proposes to each student three game-based tasks (scenarios or game based tasks ordered as a path to accomplish) in order to evaluate his/her creativity. All the results regarding each student's performance are stored into a learning profile database;
- path agent: manages the order of the scenarios to be proposed to each student;
- questionnaire agent: analyzes the learning information by considering the behaviour (creativity) of each student in performing the scenarios and tries to identify potential features for decision variables.

It is expected that each student performs a certain number of steps:

1. registration of personal information;
2. logging into the system;

3. read an instruction screen in order to know well how to correctly complete the tasks of a learning path;
4. perform a learning path that is randomly generate by the system;
5. complete the learning path by performing various games the system proposes (this is necessary to collect information about student's creativity).

Hence, the information about the execution of the learning path together with the general information about the student are analyzed by the system (more specifically, by the questionnaire agent) and thanks to the application of a decision tree algorithm it is possible to propose an optimized learning path.

Adaptivity in educational hypermedia system is another proposal to personalize curriculums [54]. In particular, a technology of curriculum sequencing is used with the objective to generate a sequence of lessons that drives the student to achieve his/her learning goals by starting from his/her initial knowledge. Furthermore, the system delivers the content of each lesson in different ways according to the student's learning style and learning progresses. The system also informs student about his/her knowledge level in relation to his/her learning goal.

Finally, an ontology-based methodology for the automatic decomposition of learning contents into re-usable learning units to be dynamically assembled into personalized e-courses has been proposed by [2]. The system involves a number of behavioral modules in order to monitor student's activities.

At the moment of registration, a student has to fill out a module by indicating his/her general data but also his/her skills, motivation, previous knowledge and learning goals. Thus, the system identifies the student's cognitive style (preference and perception related to learning materials) by submitting the student an additional questionnaire and generates a primary cognitive profile and a learning path. All the information (learning time and progress, and learning strategies) related to the execution of this path is used by the system to enrich the student's profile so as to eventually change the structure of the customized course.

3.5 Conclusions

In this Chapter we have detailed the different ways to automatic generate learning paths within an e-course.

The main methods to automatically customize e-learning systems that we found in the literature are the evolutionary-based algorithms (as genetic algorithms, shuffled frog-leaping algorithms, particle swarm optimization and ant colony optimization) and AI planning. We also identified some other techniques that can be used in order to personalized curriculums within e-courses as item response theory with artificial neuronal networks, decision support systems, data mining techniques, etc.

In this Chapter we have summarily explained the functioning of every method, but we have focused on a more detailed explanation of the AI planning techniques, since we have chosen them as our way to achieve the learning paths' personalization. For this reason, we have provided a more detailed explanation of these techniques by highlighting the most important aspects. In the following Chapter we will detail the various phases that our own personalization system involves by also providing some practical examples in order to clarify how our system works.

Chapter 4

AI planning for curriculum personalization in Moodle

In this Chapter we explain how we face the problem of curriculum personalization, by defining all the steps that our learning contents' personalization system involves. First of all, we provide a general overview of our system and then we justify the choice of a specific LMS for implementing it. Then, we explain the mapping process that we had to carry out in order to identify the similarities/differences between the elements that a planning activity needs and the elements that the LMS offers. As we will explain later, our work has required a mapping with three different abstraction levels. Hence, we detail each phase that our system needs in order to correctly customize the learning management system that we chose (by highlighting the difficulties that we had to face and the problems that we solved).

4.1 AI personalization within a LMS: general overview

As already mentioned in Chapter 3, we have faced the problem of curriculum personalization from the point of view of AI planning. In particular, we have fully adapted the knowledge engineering planning mapping introduced in [24] to be directly used within a LMS. As shown in Fig. 4.1, once chosen the LMS platform on which to focus, the learning paths' personalization requires developing a number of activities:

1. Carry out a mapping of the different modules present in the platform. This activity includes the understanding of the relationships among the

different modules and the study of the way in which each student can input information into the platform about his/her background and his/her learning objectives.

2. Build a course by using the most appropriate modules the LMS offers. In particular, it is necessary to find a module that permits the teacher to establish priority relationships, in order to design a course containing one or more sequences of modules.
3. Enroll the students in the course. It is necessary that the information (regarding initial states and learning goals) that the students input into the platform is as detailed as possible, in order to properly determine the consequent learning paths.
4. Define PDDL compilations. In particular, the relationships between the course's activities have to be translated into actions of a PDDL domain, while the information about students' background and learning objectives has to be translated into initial states and goals of a PDDL problem.
5. Provide PDDL domain and problem files to a standard planner, in order to generate a plan. More specifically, once in possession of the course's structure and students' initial states and learning goals, a standard planner is able to define a plan, i.e. a set of learning paths, one for each student enrolled in the course.
6. Establish the correct visualization for each student enrolled in the course. By using the modules available into the platform, it is necessary to ensure that each student only visualizes and takes the portion of the course involved in his/her own learning path.
7. Develop a monitoring activity. In particular, it is required to take into account all the changes that can occur in the performance of each learning path and also the possible variations of the students' goals. A good monitoring activity, by considering the performance of each student makes the system able to eventually re-plan the paths (in line with the new needs of each student).

In the following Sections we detail all these activities, clarifying them with practical examples.

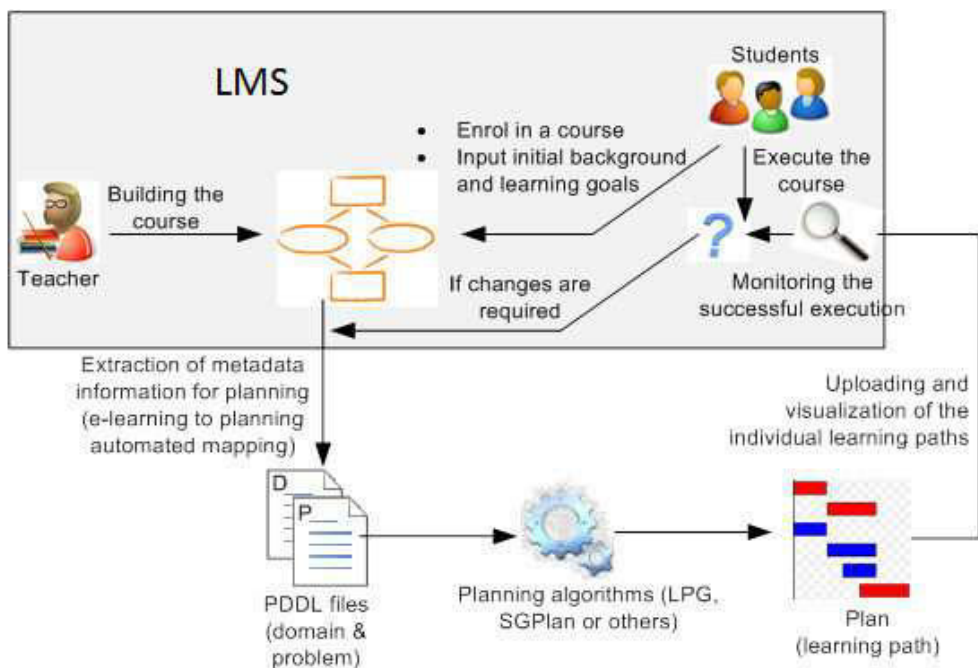


Figure 4.1: Activities required for personalized learning paths.

4.2 Why Moodle

A critical aspect in the field of distance education is the choice of the LMS platform to use. In fact, it is important to choose a platform that offers the greatest possible benefits for teachers and students in order to facilitate the learning activities. As already mentioned in Chapter 1, there are a lot of possible platforms to be taken into account and it is important to consider both positive and negative aspects of each LMS in order to properly make a choice.

4.2.1 Moodle vs. other LMSs

There are various studies in the literature that face the problem of LMSs' comparison. In particular, [55] offers an interesting comparison among free LMSs. Among the most used free LMSs we can find: Moodle (<https://moodle.org/>), ATutor (<http://www.atutor.ca>), Docebo (<http://www.docebo.org>), dotLRN (<http://www.dotlrn.org>), Ilias (<http://www.ilias.de>), Sakai (<http://www.sakaiproject.org>), Claroline (<http://www.claroline.net>), Fle3 (<http://fle3.uiah.fi/>) and OLAT (<http://www.olat.org/>).

For each platform it is possible to examine some crucial issues: supported languages, supported operative system and documentation publicly available. In particular, all these LMSs support more than one language (i.e. English, French, Italian, Spanish, etc.) and more than one operative system (Moodle, Sakai and ATutor can support up to four operative systems: Windows, Linux, Mac and Solaris). The only LMS that offers a multilingual documentation is Moodle. Claroline, Fle3 and Ilias offer English/Spanish documentation while ATutor, Docebo, OLAT, Sakai and dotLRN just provide English documentation. Furthermore, we can observe that all the analyzed LMSs except Fle3 and Sakai support SCORMs (Shareable Content Object Reference Model) e-learning standards. Furthermore, it is possible to make a more detailed LMSs' analysis ([55]) by considering:

- the communication tools (mail, forum, chat, agenda, etc.);
- the administrative tools (virus detection, access control and reports);
- the evaluation tools (quiz, tests, exercises, etc.);
- the facility of use for all the possible roles (administrator, teacher and student);

- the general offered documentation;
- the possible compatibilities (operative system supported, SCORM support, etc.)

By considering the presence or absence of the just mentioned elements, the work in [55] provides the following LMSs' completeness classification (by starting from the more exhaustive LMS):

1. Moodle;
2. Ilias;
3. Claroline;
4. Olat;
5. dotLRN;
6. Docebo;
7. ATutor;
8. Fle3.

In particular, the first and second platforms (Moodle and Ilias) provide all the mentioned elements but Moodle is simpler to use and offers a better documentation and a greater number of useful tools. Claroline and Olat are well documented and result very user friendly. dotLRN, Docedo and ATutor are not well documented and not very user friendly. Finally, Fle3 does not offer sufficient documentation in order to well structure an e-course.

It is possible to make another comparison among LMSs by considering features as:

- learner's tools: communication modules (forums, internal e-mail, chat, etc.), productivity modules (calendar, help module, progress module, etc.) and student involvement modules (e.g. the group work module or workshops);
- support tools: administration modules (registration and authentication modules, etc.), course delivery modules (quiz, exercises, grades, etc.) and curriculum design modules (course templates and personalization modules);

- technical specification tools: hardware/software tools and licensing tools;
- technical aspect of the LMS: interoperability, cost of ownership, accessibility, internationalization, etc.

The study in [39] analyzes this kind of elements for a number of LMSs, such as Moodle, Claroline, Dokeos (<http://www.dokeos.com/>), Blackboard (<http://www.blackboard.com/>), OLAT and Sakai and finds that Moodle is:

- the best LMS regarding the technical tools;
- one of the best LMSs (such as OLAT, Sakai or Claroline) with regard to the support tools;
- one of the best LMSs (together with Sakai and OLAT) concerning the technical specification tools;
- one of the best LMSs (such as ATutor and Ilias) for the technical aspect of a LMS.

By taking into consideration all the above mentioned factors, we decided to focus on Moodle as the best LMS to customize and to use in our work.

4.2.2 Moodle's general description

As already explained in the previous Section, we have decided to use Moodle to offer online personalized learning contents (Fig. 4.2). The developer of Moodle is Martin Dougiamas who created this LMS in order to perform a dissertation on a socio-constructivist approach to learning. According to the constructionist theory, the learning process becomes really effective when constructing knowledge for others through personal experience. The social constructionist theory is the application of constructivism in social contexts, that is, when we build the knowledge of each other within a group. Thus, Moodle has been designed to excel in terms of features that support the student in the learning process [20].

One of the biggest strengths of Moodle is that it is surrounded by a very wide user community of users and developers that involves over 1,315,892 registered people from 232 countries (<https://moodle.org/community/>, August 2013). This community (where it is possible to communicate in 78 languages)

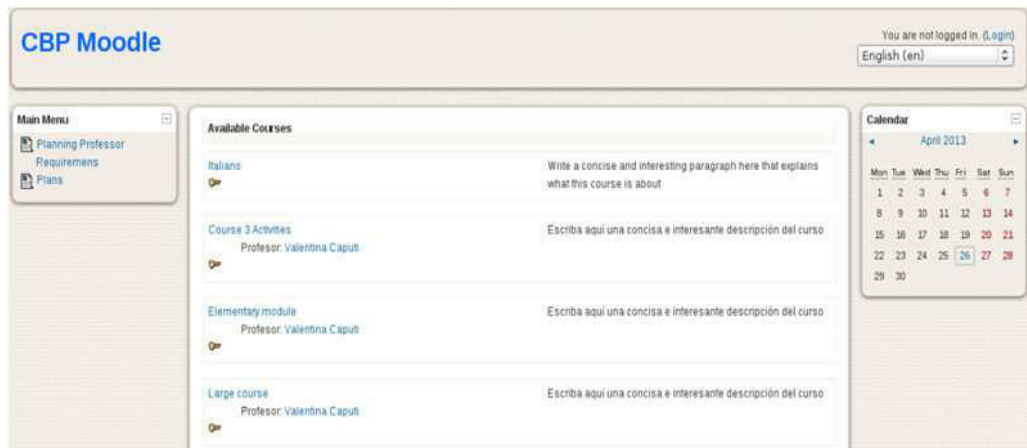


Figure 4.2: An example of Moodle’s main page containing four courses.

has grown up around the project and each member participates in active discussion forums by providing his/her own contributions and thus taking part in the continuous improvement of Moodle [1].

It is possible to identify Moodle’s advantages and disadvantages [1]. In particular, among the advantages of Moodle we find:

- the possibility of download it for free;
- the simplicity in creating high quality online courses;
- the extended community of users and developers supporting the software;
- the opportunity to run on any system that supports PHP language (such as Unix, Linux and Windows);
- the excellent documentation that is provided to users and developers.

On the other hand, Moodle’s limitations relate to:

- (as for other LMSs) the complexity of use for users who have a low IT experience;
- the possible difficulty of installation because of the many technical words that are contained in the instructions;

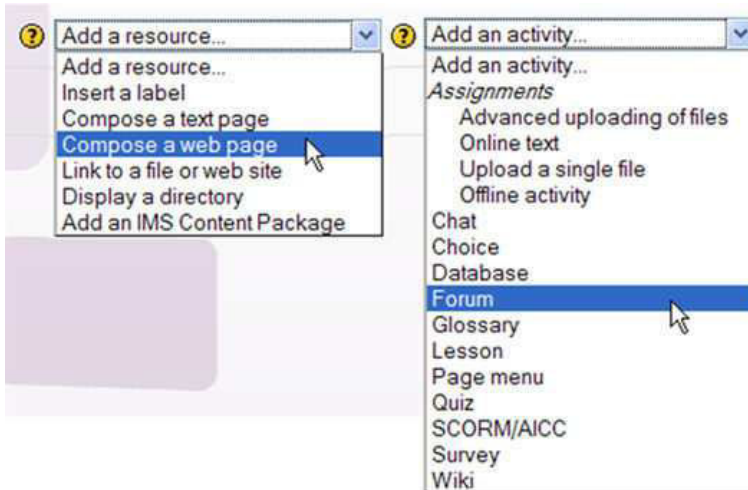


Figure 4.3: All Possible modules in a Moodle’s course.

- (as for the other LMSs) the inability of the LMS to design courses without the teacher’s help. Thus, the impossibility to automatically generate personalized courses for individual students¹.

4.2.3 Moodle in detail

As shown in Fig. 4.3, the main modules to use in order to build a course in Moodle are activities and resources. While activities are considered as the main way for the student-student and student-teacher interaction within a course, resources are used to transmit additional information to activities. Hence, the various contents of a course can be translated into the different activities available in Moodle. In more detail, when defining a course a teacher can select a number of activities such as:

- assignment: this module allows teachers to provide comments and grades on uploaded assignments. The student can execute an assignment on and off-line and, once uploaded, the file is only visible for the teacher or for a selected group of students;

¹This is one of the limitations that we had to solve in order to design our learning contents’ personalization system within Moodle.

- chat: it contains various features for managing discussions among users that want to exchange opinions in real-time with regard to a determined topic;
- choice: allows the teacher to build answers about a specific question. The teacher can set up a single or a multiple answer;
- database: allows participants (teacher and students) to generate and manage a series of records (containing, for example, images, URLs, text, etc.) related to a specified topic;
- external tool: enables the teacher and the students to interact with learning resources and activities on other web sites;
- feedback: permits users to generate and conduct analysis to collect feedback. It is used by the teacher in order to evaluate students through non-graded questions;
- forum: allows students and teachers to communicate, by posting comments, and exchange ideas with respect to a determined topic. The comments can be graded by the students or by the teacher;
- glossary: as a dictionary, it allows teacher and students to create and maintain a list of definitions to share with the other participants to a course;
- lesson: contains a number of HTML pages to be presented to the student. Each page ends with a question and the answer redirects the student to another specific page contained in the lesson. The main difference between a lesson and the other activities consists in the possibility to customize the way, and the sequence, to present the learning contents;
- quiz: enables the teacher to generate quizzes containing a number of a large variety of question types (true-false, multiple choice, etc.) It is possible to re-use the single questions for different quizzes;
- SCORM (Sharable Content Object Reference Model): permits the teacher to upload any SCORM into a course. It is a collection of specifications that enables interoperability, accessibility and reusability of web-based learning contents;
- survey: by collecting students' data, it helps the teacher to evaluate his/her own teaching way;

- wiki: it is a collection of collaboratively authored web documents. A wiki page can be created in collaboration among the teacher and the students without needing to know HTML. Each person can add pages to the wiki by simply creating a link to another web page;
- workshop: it is an activity of peer review. Students present their work through a text tool and they can simultaneously evaluate the work of other students.

As already mentioned, a resource is a medium for teaching which supports the activities. There is a variety of resources in Moodle that the teacher can add via drop-down menus:

- book: enables the teacher to design multi-page documents with a book-like format. It is also possible to import didactic material into the book module. The book generally contains chapters and sub-chapters, but sub-chapters cannot contain their own sub chapters;
- file: permits the teacher to share a file (e.g. a picture, a text document, a sound file, a video file, etc.) in order to enrich the course;
- folder: can contain other folders and enables the teacher to display together a number of course's resources;
- IMS (IP Multimedia Subsystem): helps users to define technical standards, generally for e-learning materials. The IMS (<http://www.imsglobal.org/content>) Content Packaging specification makes it possible collect portions of material in a standard format which can be re-used in different systems, without the necessity to execute a conversion into new formats;
- label: can be used to add text, images, multimedia, or code in a course's page. The label can contribute to improve the appearance of a course; however, the excessive use of multimedia in the labels may slow down the process of loading of a course's page;
- page: is used to create a link to a web page where the student can view the content created by the teacher. This content may be text, audio, video, images, etc.;
- URL (Uniform or Universal Resource Locator): is a link that redirects the student to a website or to an online file. The teacher can use the URL to prevent students from manually input another URL.

4.3 Introduction to our mapping activity

In order to correctly design our learning contents personalization system in Moodle, we had to perform two different mapping activities, corresponding to two different abstraction levels:

1. a general mapping;
2. a Moodle-dependent mapping.

In fact, first of all we had to consider, in general, how to adapt Moodle's elements to the planning elements. Once selected the most adequate elements to use in order to correctly define a PDDL problem and a PDDL domain, we carried out a most specific mapping activity by concretely considering how to switch from Moodle's standard elements to the planning elements and vice versa.

We illustrate the general mapping and the Moodle-dependent mapping, respectively, in Section 4.4.1 and in Section 4.5

4.4 General mapping and Moodle's limitations

In this Section we discuss about the general mapping that we had to perform in order to find the analogies/differences among Moodle and planning elements. Furthermore, we detail the difficulties that we had to face in order to implement our curriculums personalization idea within Moodle.

4.4.1 General mapping

As already mentioned, we had to carry out a general mapping by considering how to adapt Moodle's elements to the planning elements. In particular, in Table 4.1 we can observe the mapping of the elements generally used in learning paths' planning and the elements available in Moodle.

As Table 4.1 shows, there are no substantial differences as regards to the general definition of a course (and the tasks that are part of it) in a planning activity, and a course in Moodle. In other words, it is possible to use the Moodle's modules in order to build a course in an exhaustive way. As already mentioned, the main elements that Moodle offers teachers to define a course are activities and resources. By taking into account that activities have a role of primary importance with regard to the student-student and student-teacher

Planning in general	Planning in Moodle
Course	Course
Tasks	Activities: lesson, chat, forum, wiki, etc.
Initial background	Student's profile (very limited)
Learning goals	Student's profile (very limited)

Table 4.1: Mapping between planning terms and Moodle terms.

interaction they are considered the main modules to be used in the structuring of a course. Hence, the various contents of a general course can be translated into the different activities available in Moodle.

In particular, among all the possible activities to choose for the definition of a course we decide to only use the lesson because it is the only activity which permits us the definition of relationships and it is very simply to use for both students and teachers. Establish a priority order between the activities within a course is fundamental in order to define personalized learning paths, which, just as paths, include the execution of an ordered sequence of steps.

In general terms, a lesson in Moodle is composed by a sequential series of pages. The total number of pages depends, in general, on the content which is necessary to provide. The teacher can, at any moment, add or remove pages, depending on the specific educational requirements. At the end of each page there is a question with a number of possible answers. Teachers can set the answer mode in several ways: true/false, multiple-answer, numeric answer, etc. If the student answers correctly, then s/he will be able to continue with the next page in the lesson, or otherwise s/he will have to repeat the entire lesson or the single page (depending on how it is originally defined by the teacher). The student may not be obliged to complete the entire lesson in one session, but s/he can access it several times (each time the system will show him/her the point where s/he left it). At the end of the lesson, the teacher can optionally associate a final completion grade, i.e. a percentage obtained by considering the contribution of the individual pages' resolution. If nothing is specified by the teacher, then the completion percentage by default is 100%.

In addition to a correct definition of a course, the planning activity requires the identification of the student's profile. A correct student's profile should involve an initial state (or background) and the learning goals chosen by the student. In this sense, we found some difficulties in Moodle. In fact, as we will explain in more detail in the following Sections, a student can just input into the platform general information (name, e-mail address, country, etc.)

but it is not expected that s/he inputs detailed information about his/her background and learning goals, which results fundamental in learning path's planning. The problem that arises from this limitation is a lack of information for the subsequent phase of the PDDL problem's definition.

4.4.2 Moodle's limitations

The integration of planning into Moodle is not straightforward. Considering that Moodle, as the other LMSs, was initially designed to provide the same material (activities and resources) to all students enrolled in a course, we found some limitations in implementing our idea of learning contents' customization. In particular, complex relationships among activities that usually appear when planning elaborates courses cannot be easily defined within Moodle. Second, the information that students input into Moodle is not always sufficient to exhaustively define their background and learning goals. Third, another challenge arises to personalize the visualization and navigation of each individual learning path.

Hence, as we illustrate in this Section we had to face all these difficulties in order to embed a planning approach in Moodle [10].

Building a course: limitations in defining complex relationships

When defining a lesson in Moodle the teacher can set two kinds of links related to other activities: "dependency" and "activity link". The first is just a binding link while the second is considered as a simple suggestion. Let us clarify these concepts with an example. Let us imagine that the teacher is structuring the relationships for a certain lesson L2. As graphically represented in Fig. 4.4 we suppose that L2 cannot be executed before the completion of the lesson L1. In addition, it is necessary to suggest a lesson L3 to take after the completion of L2. As shown in Fig. 4.5 (and also suggested in Fig. 4.4) in order to correctly establish these relationships, the teacher has to set in the configuration page of L2:

- the dependence of L2 from L1;
- the activity link to the lesson L3.

The "activity link" and the "dependency" help us create only one-to-one relationships among lessons. In the configuration page of a lesson we can insert only one dependency and only one activity link. From the perspective of



Figure 4.4: Graphical representation of the relationships for the lesson L2.

Figure 4.5: Definition of relationships for Lesson L2 in its configuration page in Moodle.

planning, this it is very limiting because it requires the student to perform a single possible path within the course [10]. Consequently, to ensure the possibility to define multiple relationships within a course, we insert the concept of fictitious lesson (L_f). Unlike a real lesson (L_r), it represents the achievement of a certain learning level (or learning state) on which depends the performance of other lessons and that can be reached by taking one or more alternative lessons.

Let us elucidate the concept of fictitious lesson by another example. Let us consider the fictitious lesson L_c of Fig. 4.6. L_c can be reached by alternatively completing the lessons L_a or L_b (we have to set L_c in the field “activity link” of the two real lessons’ configuration pages). Only after L_c ’s completion, it is

possible to start the lessons Ld and Le. In particular, we can translate these constraints by setting Lc in the field “dependency” of Ld and Le’s configuration pages. In this case² it is not necessary to change anything regarding the field “dependency” and “activity link” of the fictitious lesson Lc. Thus, we can establish complex relationships within a course by simply using fictitious lessons to ensure that students can take more alternative learning paths³.

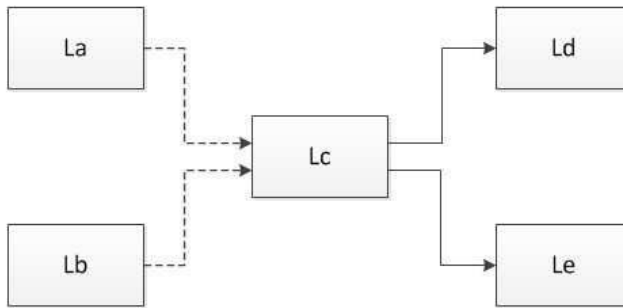


Figure 4.6: “Activity links” (as broken lines) and “dependencies” (as continuous lines) as relationships for the real lessons and the fictitious lesson Lc.

Students’ enrollment in the course: limitations in defining initial states and learning goals

In order to make sure that every student performs the most appropriate learning path to his/her needs, it is essential to know his/her background and his/her learning goals before the course starts. Unfortunately, Moodle (as well as other LMSs) was not initially designed to provide customized contents to the different students enrolled in a course. For this reason, when a student has to input into the platform his/her personal information (see Fig. 4.7), there is not a field in which s/he can express what is, for example, his/her own previous knowledge with respect to a certain topic and what is the learning level that s/he wants to achieve. Rather, s/he has only the ability to enter


²In the following Section we explain that, to correctly define on personalized course, all the fictitious lessons have to depend on a particular fictitious lesson that is an initial questionnaire.

³All the relationships among lessons will be translated into actions’ preconditions and effects in a PDDL domain (as we will show in Section 4.5.1).

fairly generic information such as e-mail address, preferred language, etc.

Kate Roberts

[Profile](#)
[Edit profile](#)
[Forum posts](#)
[Blog](#)
[Notes](#)
[Activity reports](#)
[Roles](#)


 City/town: London
 Email address: kate.roberts@gmail.com
 Courses: Italiano

General

Username*
 New password ? Unmask
 Force password change ?
 First name*
 Surname*
 Email address*
 Email display:
 Email activated:
 Citytown*
 Select a country*
 Timezone:
 Preferred language:
 Description ?

Figure 4.7: Kate’s profile in Moodle.

By using the concept of fictitious lesson, we found a way to define the initial state and goals of each student. In particular, we solved this Moodle’s limitation by creating an initial questionnaire (or L0 for short), which is a dummy lesson to be executed before the effective start of the course. L0 consists of two question pages, concerning respectively the initial state and the learning goals of the student. In particular, in the two question pages students can define their initial states (in terms of course’s modules already performed) and their learning goals (in terms of course’s modules to take).

All the course’s fictitious lessons will depend on the initial questionnaire (by setting L0 in the field “dependency” of their configuration pages). Hence, the execution of L0 makes it possible to know the necessary information which has to be properly translated into initial states and goals of a PDDL problem. We provide a PDDL problem’s example in Section 4.5.2.

Course navigation: limitations in defining personalized views

The first level mapping involves finding all the possible limitations in Moodle that prevent us from achieving a good personalization activity. Let us imagine we have a number of learning path, one for each student enrolled in the course. At this point it is necessary to create a personalized course's view for each student by considering only the specific lessons that belong to his/her learning path. This operation is not simple to be defined in Moodle. In fact, we found some limitations concerning the ability to create different groups of students in the various lessons of a course.

In general, if we wish to assign, and thus make it visible, one particular activity (that can be a lesson or any other activity) only to a selected cluster of students, it is necessary first of all to create in Moodle a group containing the selected students and then a grouping associated to the particular activity. Thus, we have to make the association between the group and the grouping. In particular, a grouping can contain one or more groups.

Let us consider the example in Fig. 4.8. Let us imagine a course consisting of three activities: A1, A2 and A3. Students enrolled in the course are Paul and Kate. Considering the specific learning paths of a supposed plan, it is necessary that Paul takes the activities A1 and A2 while Kate has to take the activities A1 and A3. Thus, it is necessary that the two students both visualize the activity A1. In addition, A2 must be just visible for Paul and A3 must be only visible for Kate.

These constraints in Moodle should be achieved by creating two groups, g1 and g2, each one associated to a student. Then, it is necessary to create the groupings GA1, GA2, and GA3 to be associated, respectively, to the three activities A1, A2 and A3. Furthermore, it is necessary to set for each activity the separate group mode. In fact, without setting this parameter, all the activities will be visible for each student enrolled in the course.

Finally, we have to make the associations between groups and groupings. Unfortunately, up to our knowledge about the structure of the platform, the desired results (prior to any modification) were not achieved. In fact, considering our simple example, Paul visualized only A2 while Kate visualized only A3. We had to modify in the code⁴ (as shown in Fig. 4.9) and we got the correct visibility of activities for groups of students.

Thus, once a plan is generated, every student can now visualize only the lessons which s/he has to perform. In section 4.5.4 we illustrate a practical ex-

⁴In particular, we modified a function in the file (`/var/www/moodle/user/index`).

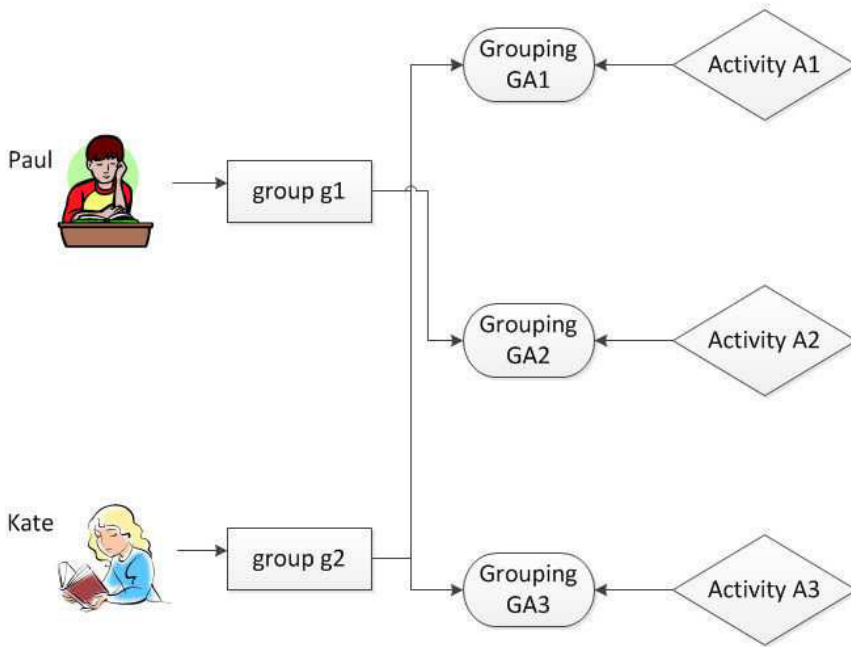


Figure 4.8: How to associate activities to students in Moodle.

ample of correct customized learning paths' visualization for students enrolled in a course in Moodle.

4.4.3 Building a course by using Moodle's lessons

Once solved the problems that prevented us from successfully implement our idea, our work in Moodle begins with the definition of a course. For example, we can structure the Italian's Elementary module as shown in Fig. 4.10. The module is composed by 10 lessons. As already mentioned in Section 4.4.1, we decided to only use this module because it facilitates the definition of relationships among activities and it is very simply to use for both students and teachers.

By considering the observations in Section 4.4.2, the Elementary module (as shown in Fig. 4.10 and in Fig. 4.11) can be composed by an initial questionnaire (L0), five real lessons (L1, L2, L3, L4 and L5) and four possible learning states, very bad, bad, almost sufficient and sufficient, to which we associate the fictitious lessons (corresponding to four knowledge levels) Lvb,

Lb, Las and Ls, respectively. The dependencies between lessons are represented as continuous lines while the suggestions as broken lines. Fig. 4.10 also shows the duration of each lesson. In particular, we have to assign a minimum execution time for each lesson. Thus we suppose that fictitious lessons have a duration of 1 minute, because null times are not allowed in Moodle.

Each learning state is also part of L0's multi-choice question pages to define the initial state and learning goals of each student (as shown in Fig. 4.12). Thus, it is important the consistency between the nomenclatures used in the definition of L0 and in the rest of the course.

The result that we want to achieve in our system is that once a teacher defines a course, each student (by simply completing an initial questionnaire) can already get a learning plan suited to his/her specific needs.

4.5 Moodle-dependent mapping

Once carried-out a general mapping, our work involves a Moodle-dependent mapping (as shown in Table 4.2) characterized by two directions:

- from Moodle to planning;
- from planning to Moodle.

Furthermore, for each mapping direction it is possible to define two different abstraction levels:

- Moodle's elements;
- Moodle's database.

In particular, the first direction mapping consists in analyzing how to adapt the Moodle's lessons in planning constructs. This mapping involves two abstraction levels: the former involves the definition of the correct use of the Moodle's lessons in order to reach a correct definition of the PDDL constructs (Section 4.5.1) while the latter involves the concrete selection of the Moodle database's relational tables necessary for defining the PDDL domain and problem (Section 4.5.2).

By providing a PDDL domain and a PDDL problem to a standard planner, we can obtain a plan (as shown in Section 4.5.3), that is a set of personalized learning paths.

The second direction mapping includes, in general, the transition from a planning solution (i.e. a set of actions that describe a number of learning paths, one for each student enrolled in the course) to a correct learning paths' visualization in Moodle. Also in this case it is possible to identify two abstraction levels: the first concerns the selection and the correct use of the Moodle's elements needed for setting the lessons' display mode (Section 4.5.4) while the second level relies on the concrete choice of the Moodle database's relational database needed in order to get a correct learning paths' visualization (Section 4.5.5).

Abstraction level	From Moodle to planning	From planning to Moodle
1. Moodle's elements	From Moodle's lessons to PDDL constructs	From planning solutions to Moodle's elements
2. Moodle's database	From Moodle's database to PDDL domain+problem files	From planning solutions to Moodle's database

Table 4.2: Mapping: from planning to Moodle, and vice versa.

4.5.1 From Moodle's lessons to PDDL constructs

Once structured a course, the next step consists in performing a more detailed study of the Moodle's features that allow us to generate the PDDL domain and problem.

In particular, first of all, it is necessary to analyze in a theoretical way how we can generate the PDDL domain by starting from the Moodle's lessons (Section 4.5.1). The PDDL domain's generation (jointly to the PDDL problem generation, once students enroll in the course) can be concretely obtained only implementing in Moodle some PHP code necessary to translate Moodle's information into PDDL information.

From Moodle's lessons to a PDDL domain

When designing a personalized learning path we consider both the information about the course and each student who takes it. Course's information has to be translated into PDDL actions, in order to define a PDDL domain. On the other hand, a PDDL problem is generated from the initial states and goals of each student. The PDDL domain does not depend on a specific PDDL problem, but it describes a family of similar problems. In order to generate a correct learning path for each student, it is necessary to organize of the course

in terms of dependency links and activity links between lessons. In particular, we have to establish these relationships so as to have the right analogies with the PDDL domain to generate.

As already mentioned, a course consists of a set of real lessons $L_r = L_{r_1}, \dots, L_{r_n}$, a set of fictitious lessons $L_f = L_{f_1}, \dots, L_{f_n}$, and an initial questionnaire (that we named L0), that is a particular fictitious lesson. The constraints to be respected when structuring a course are the following:

- L0 has not dependency constraints or activity links to any other lesson;
- each L_{r_i} can depend on a real or fictitious lesson. If L_{r_i} allows us to reach a certain learning state, it is necessary to set in the L_{r_i} 's configuration page an activity link that leads to the fictitious lesson representative of that state;
- each L_{f_j} has only a dependency on L0 and does not have activity links.

The features listed above become preconditions and effects for actions in the PDDL domain, as shown in Fig. 4.13.

4.5.2 From Moodle's database to PDDL domain+problem files

In order to generate the PDDL domain and problem files, it is necessary to implement in Moodle some PHP code to perform the PDDL translation. This code involves the selection of some required Moodle database's relational tables. In particular, Table 4.3 shows the relational tables that we selected from the Moodle database in order to generate, respectively, the PDDL domain and the PDDL problem. A graphical representation of how these relational tables are used for the generation of the PDDL domain+problem is provided in Fig. 4.14. We can notice that the PDDL translator is composed by a domain generator and a problem generator.

Thus, once structured a course, the Domain generator creates a PDDL domain by using the information coming from the Moodle database's relational tables. For example, Fig. 4.15 shows a real action and a fictitious action in the PDDL domain, which includes parameters, durations, preconditions and effect, of the Elementary module presented in Fig. 4.10.

After the PDDL domain is generated and once students enroll in the course, the Problem generator uses all the necessary information coming from the database's relational tables (Table 4.3) in order to generate a PDDL problem file. In particular, the choices made by the students when executing L0 will

Type of PDDL document to generate	Relational tables selected from Moodle's database
PDDL domain	course lesson course_modules
PDDL problem	course course_display context role_assignments user lesson lesson_pages lesson_attempts lesson_answers lesson_grades

Table 4.3: Mapping between Moodle's database relational tables and PDDL domain and problem generation.

be the initial state and goals of the PDDL problem, as already explained in Section 4.5.1.

Let us imagine that four new students, Mark, Laura, David and Polly just want to take the Elementary module and the results arising from the L0's execution are as shown in Table 4.4. For example, Mark has a background that includes the learning concepts related to Lvb and he wants to achieve a learning level corresponding to Ls. L0's results are translated by the Problem generator into the initial states and goals of the PDDL problem, illustrated in Fig. 4.16. As explained in section 4.5.1, L0's results can be translated into the initial states and goals of the PDDL problem, illustrated in Fig. 4.16.

Student	Initial states	Learning goals
Mark	Lvb	Ls
Laura	Lb, L2	Ls
David	Lb, L1, L2	L4, Ls
Polly	Lvb	L5, Ls

Table 4.4: Initial states and goals of the students enrolled in the Elementary module.

4.5.3 Plan generation

Once created the PDDL domain and problem, it is necessary to use them in order to generate a plan that contains a learning path for each student enrolled in the course.

As already mentioned in Section 3.3.2, PDDL is a standard planning language and it is supported by most state-of-the-art planners. In consequence, there are many planners that can we can use here. In our implementation, we have chosen LPG (<http://zeus.ing.unibs.it/lpg/>) because it is publicly available and shows a good tradeoff between running time and quality of solutions⁵. But it is important to note that we can use other planners without further modifications. We can see the resulting plan in Fig. 4.17, where we have omitted the representation of L0, which is obviously taken by every student when s/he defines his/her initial state (background) and learning goals.

4.5.4 From planning solutions to Moodle's elements

Once generated a plan, in order to create an efficient course's visualization it is necessary that each student only visualizes in Moodle the lessons that are included into his/her learning path. As explained in Section 4.4.2, if we want to associate a lesson to a specific cluster of students in the platform we have to create a group, involving the set of selected students and a grouping associated to the particular lesson. This is a Moodle weird feature. Thus, it is necessary to create an association between the group and the grouping. Generally speaking, a group can contain one or more students and a grouping can include one or more groups.

Thus, the steps to be performed to personalize the lessons' visualization are illustrated as follows (see also Fig. 4.18, where we show these steps to customize the learning paths' visualization within the Elementary module):

1. we create a group for each student (for simplicity, we name each group as the user's id of the student in the platform);
2. we create a grouping for each lesson;
3. we create the associations between the groups and the groupings depending on the results of the plan, i.e. the actions associated to each student (see Fig. 4.17 for the Elementary module).

⁵In Chapter 5 we will compare LPG with another standard planner and we will demonstrate that it represents the best planner to use within our personalization system

In particular, as shown in Fig. 4.18, in the specific case of the Elementary module we associate the groupings of the lessons L1 and L2 to Polly's and Mark's groups, the grouping of the lesson L4 to David's and Laura's groups and the grouping of the lesson L5 to Polly's group.

4.5.5 From planning solution to Moodle's database

As explained in Section 4.5.4, once generated a plan (containing a number of learning paths, one for each student enrolled within the course) it is necessary to get the correct learning paths' visualization in Moodle. Concretely, this result is obtained by implementing in Moodle some PHP code to automatically achieve the groups-groupings associations mentioned in Section 4.5.4. This code contains some references to determine Moodle database's relational tables which must be automatically updated each time a plan is generated. Fig. 4.19 shows how to obtain a correct learning path's visualization in Moodle. At this point, each student only visualizes the lessons included in his/her own learning path (as shown in Fig 4.20 for the specific case of the Elementary module).

4.6 Some notes on monitoring

Once the plan is generated and all its content, i.e. a set of lessons, are correctly shown to each student, it needs to be executed. By considering our initial knowledge in terms of the students' initial states and goals, we know that the learning path that we offer to each student is the most appropriate to his/her needs. But we cannot be sure that it is executable in its entirety. In fact, it is possible that at some time the expected results do not correspond to the real results achieved by the student (because a discrepancy appears).

For example, let us consider Polly's learning path within the Elementary module. Before executing L2, she has to spend 60 minutes in completing L1. Imagine that she has already spent 60 minutes in performing L1 without terminating it. A similar problem can occur for a variety of reasons: lack of appropriate equipment, loss of time due to external factors, error in the self-assessment of the initial state, etc. In the specific Polly's case, it is possible, for example, that she changed her learning goals over time. Thanks to how we have designed the system, Polly (and any student in general) can express a change of the initial state and/or learning goals by simply performing one more time L0 (see Fig. 4.21).

Thus, together with the planning activity, it is necessary to carry out a

monitoring activity that allows us to assess if the student's progress remains in line with his/her expected path. The monitoring activity, which is included in our system by simply invoking again the planner whenever it is deemed necessary, is indispensable to eventually (partially or completely) re-plan students' learning paths. Every time we invoke the planner, the lessons already carried out by the student and any changes in his/her learning goals are considered respectively as part of the initial state and goals in a new PDDL problem. This problem, together with the PDDL domain (which can also change if the teacher modifies the course structure), is considered by the planner to generate a new student-oriented plan, that is a new learning path, which is consequently shown to the student.

4.7 Conclusions

In this Chapter we have shown the implementation of an AI planning approach within a specific LMS. In particular, we explained how it is possible to generate, visualize, and monitor the execution of a plan in Moodle. It is important to observe that we implemented our system in Moodle, but our entire approach can be easily extrapolated to any other standard LMS.

To summarize, first of all we gave a general overview of curriculum's personalization within LMSs, by identifying all the required activities to develop to customize the LMS's learning contents.

Second, we determined that Moodle represents the best LMS consider in order to implement our learning contents' customization idea. We reached this conclusions by considering several studies aimed at comparing the most important standard LMSs' features. In particular, Moodle is considered the best standard LMS because it involves all of the following characteristics: user friendly interface, support for multiple languages, support for more than one operative system, multilingual and exhaustive documentation, support for SCORMs e-learning standards. Furthermore, Moodle offers a number of tools (evaluation tools, administrative tools, communication tools, etc.) greater than the other standard LMSs.

Third, we established that the lesson is the best Moodle's module to use to implement our customization idea because it is the only module which permits us to establish priority relationships, key feature in the planning constructs. On the other hand, we had to solve some limitations: impossibility of defining multiple relationships between lessons and difficulties in establishing the initial state and the learning goals of each student. In the Chapter we detailed the

resolutions that we provided for these problems.

Fourth, we offered a contribution concerning a Moodle-dependent mapping where we established how to switch from Moodle's elements to planning elements and vice versa. In particular, we established the use of the Moodle's lessons in order to reach a definition of the PDDL constructs. More in detail, we showed the selection of the Moodle database's relational tables necessary for defining the PDDL domain and problem. Providing these files to a standard planner, it is possible to obtain a plan, that is a set of personalized learning paths. Hence, we defined the mode of transaction from a planning solution (i.e. a set of actions that describe a number of learning paths) to a learning paths' visualization in Moodle. In particular, once individuated the Moodle's elements needed for setting the lessons' display mode, we determined the choice of the Moodle database's relational tables needed to get a correct learning paths' visualization.

Finally, we provided some notes on the monitoring activity that it is necessary to develop together with the planning activity, in order to eventually modify a learning path (for example, in case of change of the student's learning goals).

```
function groups_get_user_groups($courseid, $userid=0)
{
    global $CFG, $USER;
    if (empty($userid))
    {
        $userid = $USER->id;
    }
    $v = "Select gg.groupingid, g.id from ".$CFG->prefix."groups g, "
        ."$CFG->prefix."groups_members gm, "
        ."$CFG->prefix."groupings_groups gg
        where gm.userid=$userid and g.courseid=$courseid
        and gm.groupid=g.id and gg.groupid=g.id";
    $groups=get_records_sql($v);
    $result = array('0' => array_keys($groups));
    foreach ($groups as $group)
    {
        if (is_null($group->groupingid))
        {
            continue;
        }
        if (!array_key_exists($group->groupingid, $result))
        {
            $result[$group->groupingid] = array();
        }
        $result[$group->groupingid][$group->id] = $group->id;
    }
    return $result;
}
```

Figure 4.9: A PHP function modified (in particular, we changed the SQL query) in Moodle in order to get the correct course visibility.

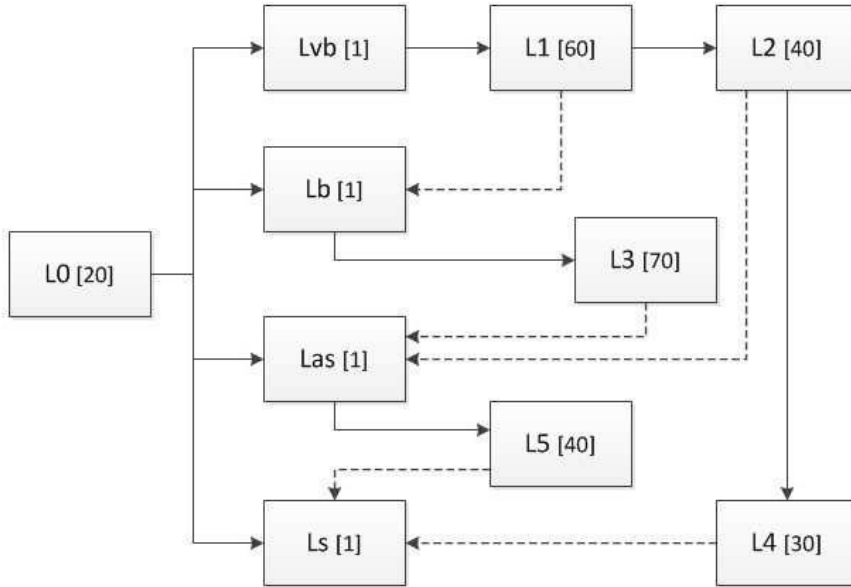


Figure 4.10: Structure of the Elementary module. Durations are in minutes between brackets, activity links are represented as broken lines and dependencies as continuous lines.



Figure 4.11: Elementary module in Moodle.

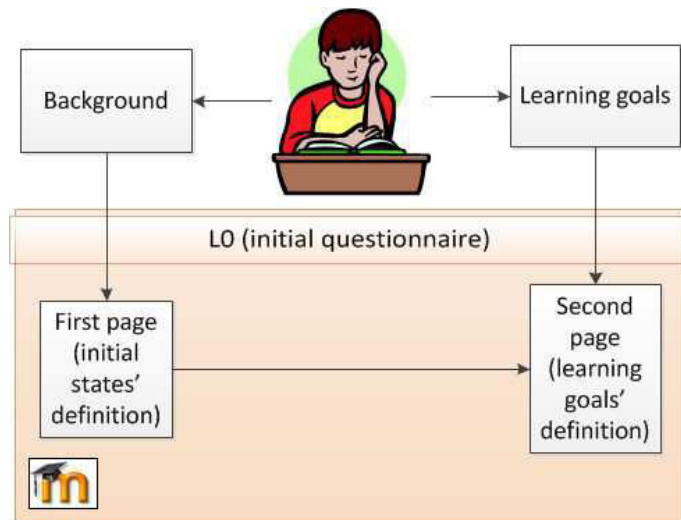


Figure 4.12: Student's initial background and learning goals modelled in Moodle by L0.

Moodle's lesson			PDDL's action	
kind	dependency	activity link	preconditions	effects
L0	-	-	not (L0_done ?s)	(L0_done ?s)
L _{r_i}	(allows to reach a L _{f_j})	to L _{f_j}	not (L _{r_i} _done ?s) (L0_done ?s) ((L _{r_h} _done ?s) or (L _{f_k} _done ?s))	(L _{r_i} _done ?s) and (L _{f_j} _done ?s)
	(does not allow to reach a L _{f_j})	-	not (L _{r_i} _done ?s) (L0_done ?s) ((L _{r_h} _done ?s) or (L _{f_k} _done ?s))	(L _{r_i} _done ?s)
L _{f_j}	L0	-	not (L _{f_j} _done ?s) (L0_done ?s)	(L _{f_j} _done ?s)

Figure 4.13: Mapping: from Moodle's lessons to PDDL's actions for a given student ?s.

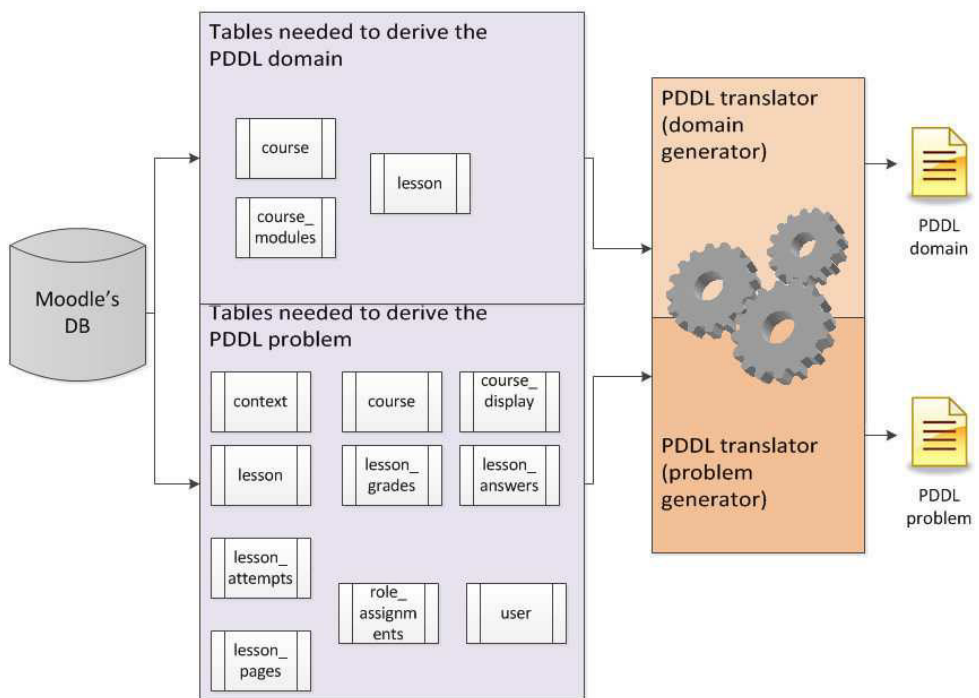


Figure 4.14: From the relational tables selected from Moodle's database to PDDL domain and problem.

```
(:durative-action L2
:parameters (?s - student)
:duration (= ?duration 40)
:condition (at start (and
                (not (L2_done ?s))
                (L1_done ?s)))
:effect (and
        (at end (L2_done ?s))
        (at end (Las_done ?s))))

(:durative-action Lvb
:parameters (?s - student)
:duration (= ?duration 1)
:condition (at start (and
                (not (Lvb_done ?s))
                (L0_done ?s)))
:effect ((at end (Lvb_done ?s))))
```

Figure 4.15: A real lesson (L2) and a fictitious lesson (Lvb) in the PDDL domain of the Elementary module.


```
(:init
  (Lvb_done Student_Mark)
  (L2_done Student_Laura)
  (Lb_done Student_Laura)
  (Lb_done Student_David)
  (L1_done Student_David)
  (L2_done Student_David)
  (Lvb_done Student_Polly))

(:goal (and
  (Las_done Student_Mark)
  (Ls_done Student_Laura)
  (Ls_done Student_David)
  (L4_done Student_David)
  (Ls_done Student_Polly)
  (L5_done Student_Polly))))
```

Figure 4.16: PDDL problem for the Elementary module.

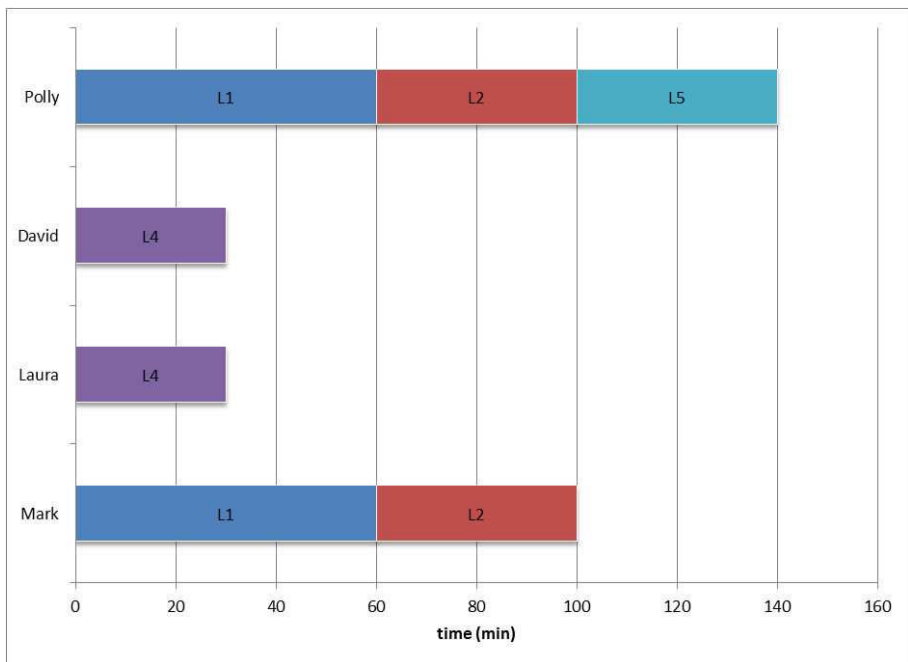


Figure 4.17: Resulting learning paths for the Elementary module.

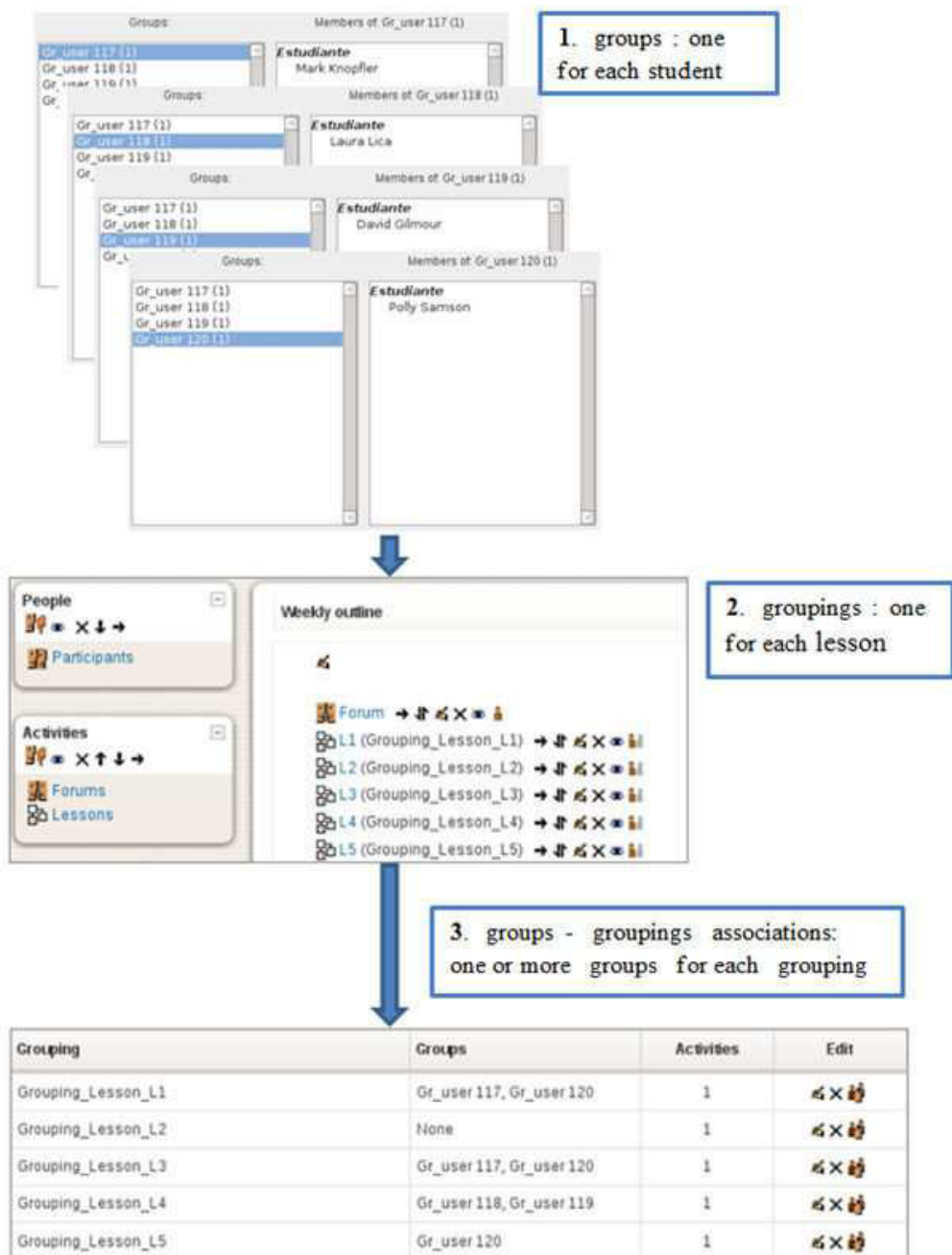


Figure 4.18: How to associate activities to students in the Elementary module in Moodle.

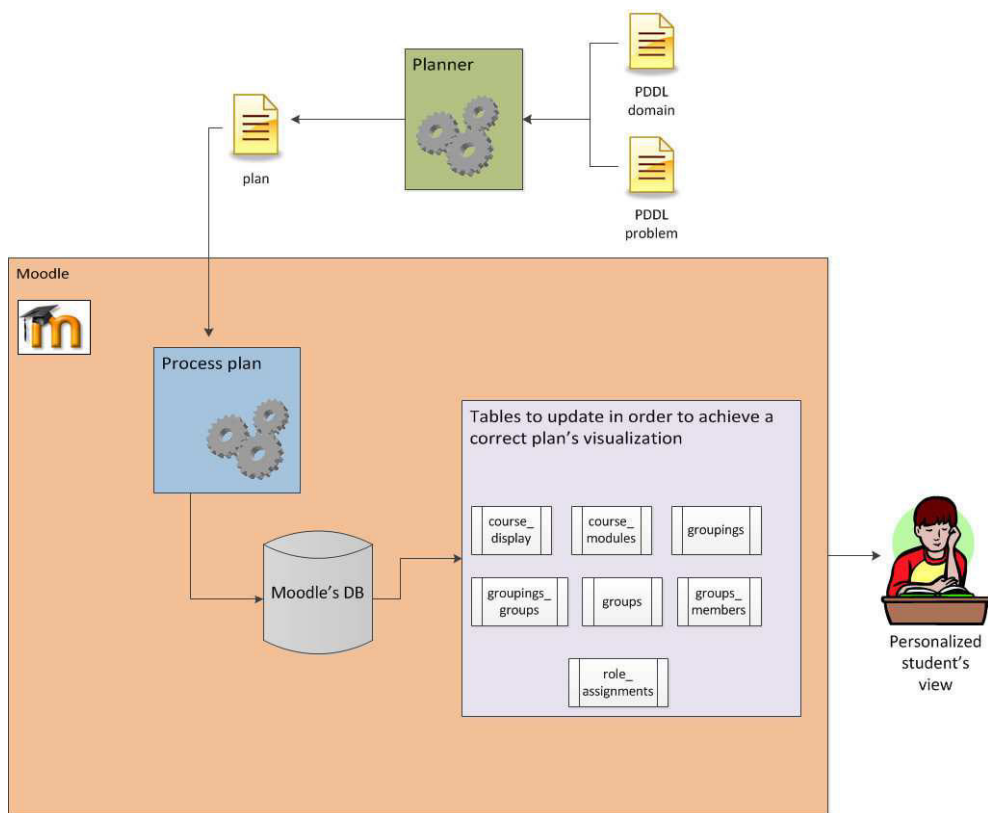


Figure 4.19: How to obtain the correct learning path's visualization.

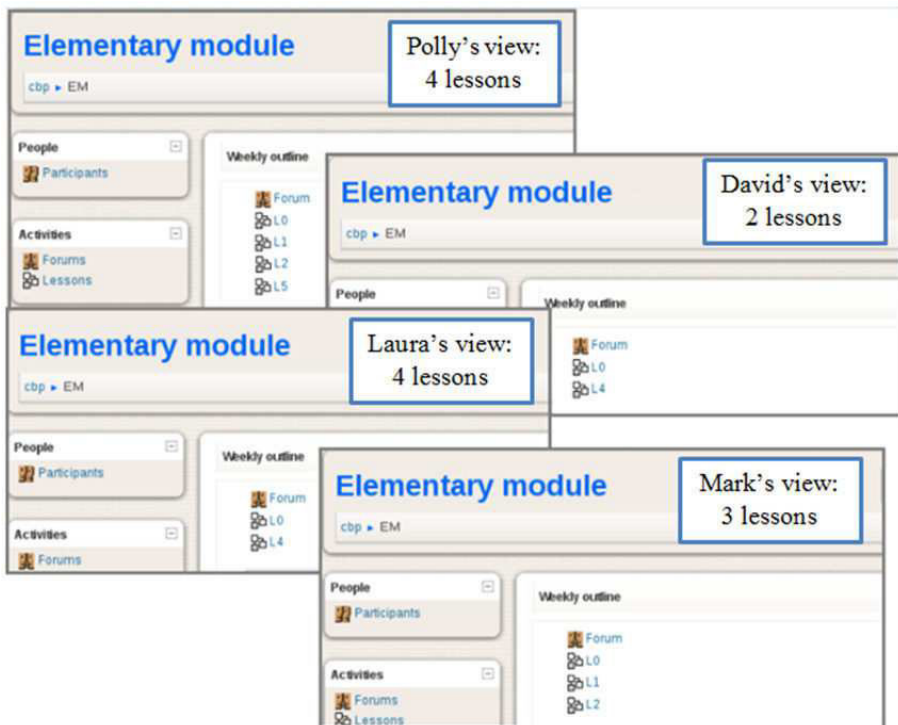


Figure 4.20: Personalized visualizations of the “Elementary module” in Moodle.

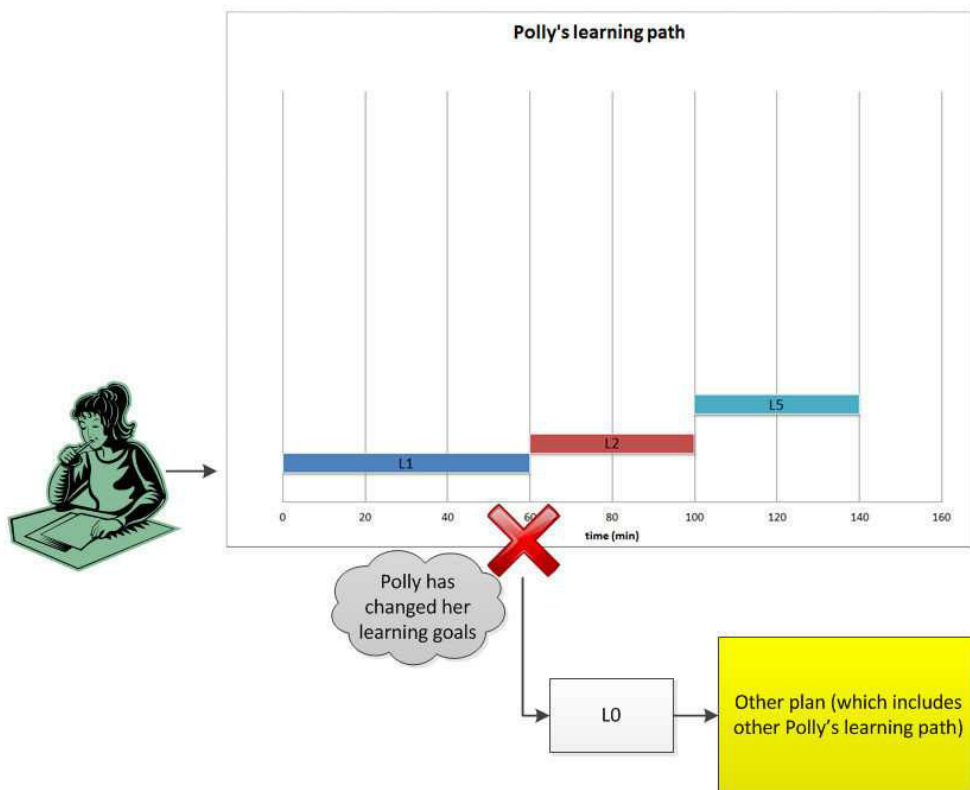


Figure 4.21: Monitoring and re-planning of Polly's learning path.

Chapter 5

Experimental evaluation

After completed the steps required to customize the learning contents in Moodle, we have carried out a quantitative evaluation of our work. In particular, in this Chapter we perform several tests to assess the scalability of our system (i.e. the ability of the system to increase its performance if a large amount of resources is provided to it, and thus in function of the total computing power required to it).

5.1 Introduction

Once created a new e-learning system, it is necessary to carry out an evaluation of its effectiveness before it is placed into the educational world.

In particular, the evaluation of the scalability of a system is a fundamental step to ensure the real potential of the system and, in our case, to check whether our system is worthwhile to propose it to the educational world. Considering that we wanted to test the system with a large amount of data and considering that (due to time restrictions) it would have been practically impossible to gather hundreds of students to test big courses, we performed a quantitative evaluation by using fictitious courses and fictitious students.

In particular, we created in Moodle a large number (up to 500) of fictitious students and we designed and implemented into the platform four courses having different sizes (from a minimum of 10 to a maximum of 64 lessons). Once generated a PDDL domain and a number of PDDL problems for each fictitious course, we used two standard planners in order to solve as many problems as possible. In this way, we could evaluate the behavior of the planners (required time, actions generated, percentage of solved problems,

etc.) in relation to the different courses and thus check the suitability of our approach.

5.2 A quantitative evaluation

5.2.1 Introduction

We have performed a quantitative evaluation in order to measure the scalability of the system by creating fictitious courses and fictitious students. In particular, we created four fictitious courses having different sizes: up to 9, 14, 41 and 64 lessons with durations named respectively “Small”, “Medium”, “Big” and Very big course.

Furthermore, we created up to 500 fictitious students whom we randomly assigned a number of initials states and learning goals for each course. In particular, we designed a vector of fictitious students containing the following elements: 1, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500. Thus, we tested each fictitious course for each vector’s element as the number of students virtually enrolled into the course.

Hence, our system automatically generates the PDDL problems by starting from all the possible combinations between two identical vectors representing respectively the number of possible initial states and goals to be assigned to each fictitious student. More in detail, each vector contains a number of elements (starting with the value 1 and incremented by two units at a time) that we set depending on the dimension of the course.

After the automatic generation of the PDDL domains+problems, we used two different standard planners to assess the viability of the solving process by current planning technology. In particular, we use LPG (<http://zeus.ing.unibs.it/lpg/>) and SGPlan (<http://www.sgplan.com/>) because they have traditionally shown very effective in the International Planning Competitions (IPCs, <http://ipc.icaps-conference.org/>) but we can use any other planner that supports PDDL. We run all our experiments on an Intel Core i5 CPU (dual core 2.27 GHz processor) with 4 GB of RAM. All experiments were censored after 900 seconds.

In the following Sections we detail the characteristics of each experiment in terms of PDDL domain and problem’s design and generated plans. Finally we provide a comparison between the results that we achieved for the various courses.

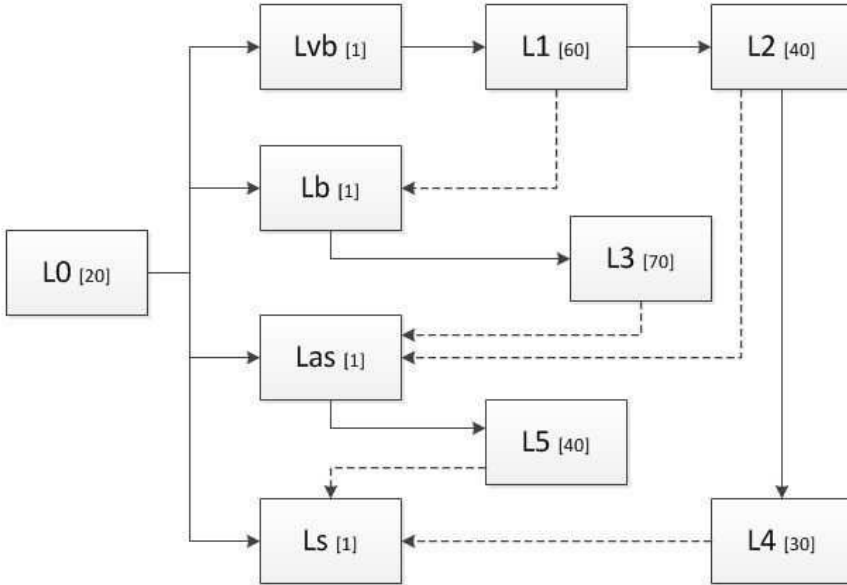


Figure 5.1: Representation of the Small course's structure (durations are in minutes between brackets, activity links are represented as broken lines and dependencies as continuous lines).

5.2.2 Small course

First of all we implemented in Moodle a course consisting of 9 lessons with durations: the Small course. The course's structure representation is provided in Fig. 5.1 (where we highlight the dependency links as continuous lines, the activity links as broken lines and the duration of each lesson between brackets) while in Fig. 5.2 it is possible to observe the Moodle's view of the course. Thus, our system automatically generates a PDDL domain containing 9 actions, one for each lesson.

In order to automatically generate the PDDL problems, we decided to randomly assign up to 6 initial states and up to 6 goals for each of the 500 generated fictitious students. In particular, the 2 identical vectors representing the number of initial states and goals contain the following 4 elements: 1, 2, 4, 6. By considering all the possible combinations among the vectors' elements, our system automatically generates 16 (4×4) PDDL problems by randomly choosing the students' initial states and goals among all the lessons of the

The screenshot displays the Moodle interface for a 'Small course'. At the top, the course name 'Small course' is shown in blue, along with the user's name 'Valentina Caputi' and a 'Logout' link. Below this, there are navigation links for 'ctbp' and 'smallc', and a 'Switch role to...' dropdown menu with a 'Turn editing on' button.

The main content area is divided into several sections:

- People:** A section with a 'Participants' link.
- Activities:** A section with 'Forums' and 'Lessons' links.
- Search Forums:** A search box with a 'Go' button and an 'Advanced search' link.
- Administration:** A list of administrative tasks including 'Turn editing on', 'Settings', 'Assign roles', 'Grades', 'Groups', 'Backup', 'Restore', 'Import', 'Reset', 'Reports', 'Questions', and 'Files'.
- Weekly outline:** A central section showing a list of activities and their dates:
 - 29 January - 4 February:** Includes 'Foro de Noticias'.
 - 5 February - 11 February:** (Empty)
 - 12 February - 18 February:** Includes a list of levels: L0, L1, L2, L3, L4, L5, Lvb, Lb, Lg, and Lvg.
- Latest News:** A section with an 'Add a new topic...' link and a message: '(No news has been posted yet)'. There is also a 'Go to calendar...' link.
- Upcoming Events:** A section with a message: 'There are no upcoming events' and a 'New Event...' link.
- Recent Activity:** A section showing 'Activity since Monday, 23 September 2013, 10:28 AM' and a 'Full report of recent activity...' link. Below this, there is a 'Course updates:' section with two entries: 'Deleted Chat'.

Figure 5.2: Moodle's representation of the Small course.

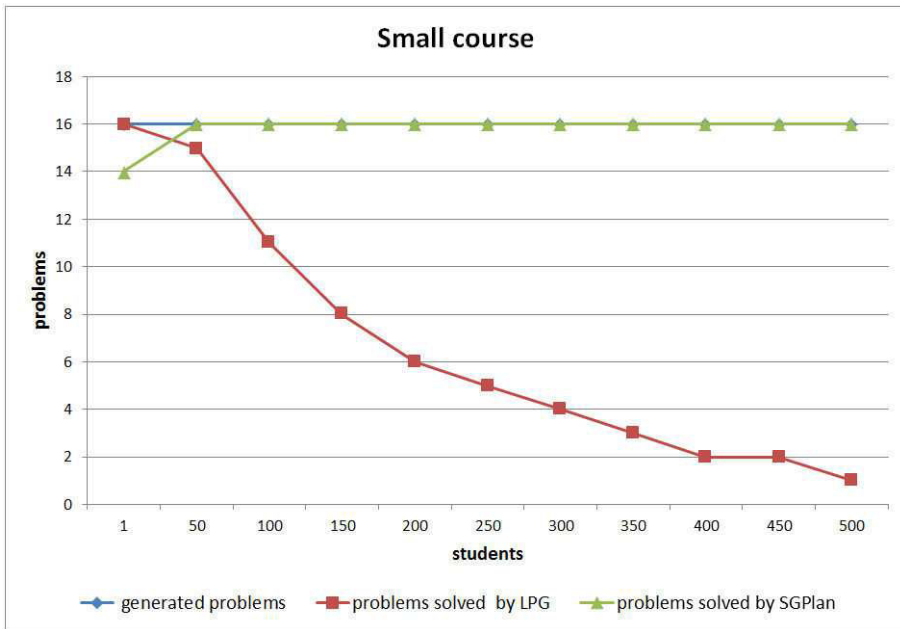


Figure 5.3: Number of solved problems for the Small course by the two planners.

course.

Once generated the PDDL domain and the PDDL problems, we used LPG and SGPlan to create the possible plans (or solved problems) illustrated in Fig. 5.3. It is possible to observe that both planners found solution for a number of problems for up to 500 fictitious students. We can also observe that LPG solves a greater number of problems to fewer than 50 students but, in general, SGPlan has a much greater yield (finding almost all the solutions for all the generated problems). In particular, although for up to 50 students SGPlan is not able to solve all the problems, by starting from this point the planner always solves all the 16 generated problems. More in detail, in Table 5.1 it is possible to observe the percentage of solved problems by each planner in relation to the total number of generated problems.

Table 5.2 shows the median value of actions generated by each planner when solving the plans. We can observe that LPG, whose relative median value of generated actions is irregularly shaped, reaches a maximum value for problems related to 450 students while SGPlan, whose median value of generated actions increases linearly (as the number of students), reaches the

Number of students	LPG	SGPlan
1	16(100%)	14(87.5%)
50	15(93.75%)	16(100%)
100	11(68.75%)	16(100%)
150	8(50%)	16(100%)
200	6(37.5%)	16(100%)
250	5(31.25%)	16(100%)
300	4(25%)	16(100%)
350	3(18.75%)	16(100%)
400	2(12.5%)	16(100%)
450	2(12.5%)	16(100%)
500	1(6.25%)	16(100%)
Total	73(41.5%)	174(98.8%)

Table 5.1: Percentage of problems solved by LPG and SGPlan for the Small course.

maximum value in correspondence of problems related to 500 students.

Finally, Fig. 5.4 shows the average time taken by each planner in order to solve problems related to 1, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 students respectively. It is possible to observe that, by solving fewer problems when the number of students increases, LPG obviously takes (in general) less time than SGPlan and, in particular, LPG takes time than SGPlan in problems for 50 students. LPG reaches the maximum time to solve the problems (approximately 2 seconds) when the number of students is 450. In contrast, considered that SGPlan solves a greater number of problems, it takes a time that grows in direct proportion to the number of students and reaches a maximum value of 18 seconds for problems involving 500 students.

5.2.3 Medium course

We decided to increment the complexity of our experiment by designing a course composed by 14 lessons with durations: the Medium course (as shown in Fig. 5.5). Thus, we structured the course in Moodle as illustrated in Fig. 5.6 and our system automatically generates a PDDL domain containing 14 actions corresponding to the various lessons of the course. We randomly assigned up to 10 initial states and up to 10 goals to each student for each of the 500 students. In this case, the initial states and learning goals' vectors contain the values: 1, 2, 4, 6, 8, 10. For each combination among the 6 values of the two vectors, the system automatically chose the initial states and the goals in a random way among the course's lessons to generate a total of 36

Number of students	LPG (for solved problems)	SGPlan (for solved problems)
1	3(16)	4(14)
50	136(15)	139(14)
100	179(11)	267(16)
150	254(8)	425(16)
200	284(6)	543(16)
250	288(5)	677(16)
300	282.5(4)	827(16)
350	322(3)	956(16)
400	300(2)	1072(16)
450	353(2)	1251(16)
500	243(1)	1385(16)

Table 5.2: Median value of actions generated by LPG and SGPlan in order to solve the problems in the Small course.

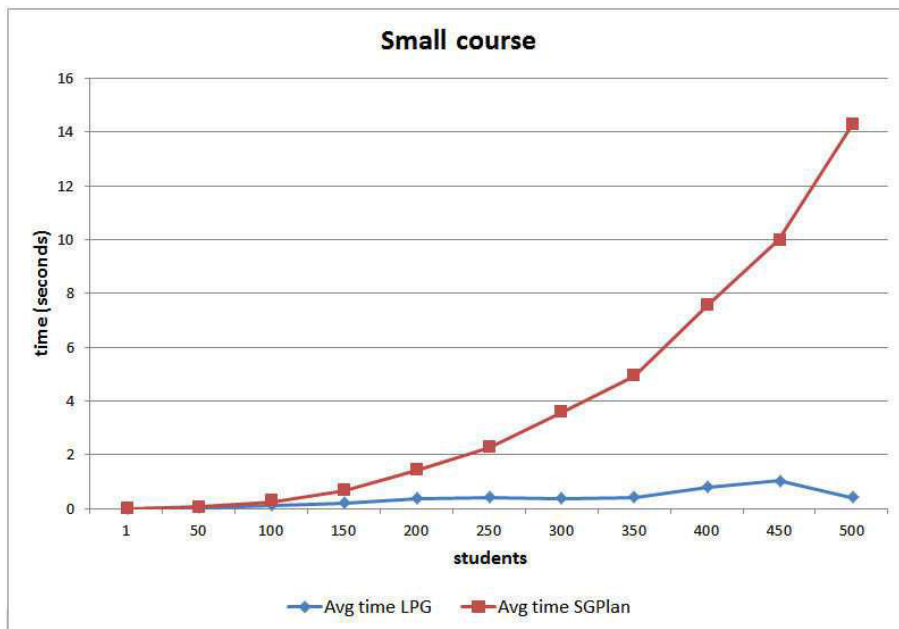


Figure 5.4: Average time (in seconds) to find plans for the Small course by the two planners.

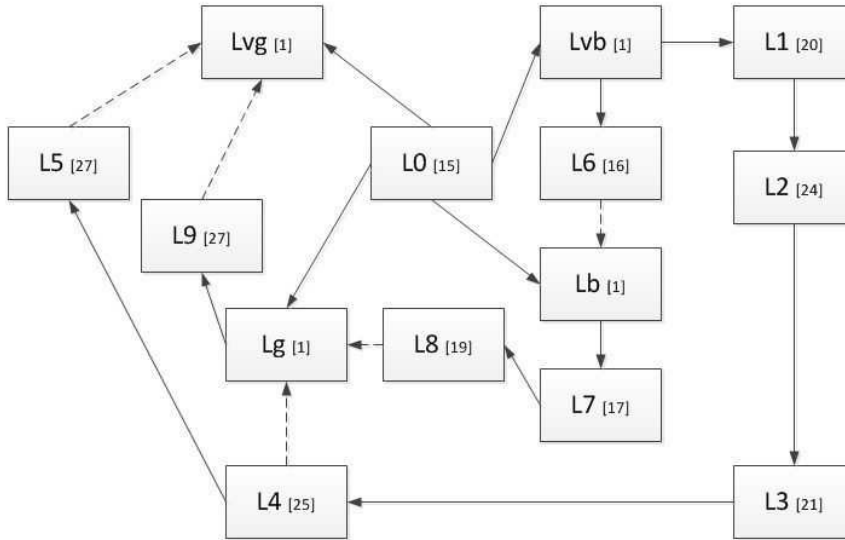


Figure 5.5: Representation of the Medium course's structure (durations are in minutes between brackets, activity links are represented as broken lines and dependencies as continuous lines).

(6*6) PDDL problems.

Hence, by using LPG and SGPlan we generated all the possible plans (or solved problems) illustrated in Fig. 5.7. It is possible to observe that both planners found solution for up to 500 fictitious students. More in detail, both planners generate the same plans (32) for problems related to 1 student but, from this point on, the curve representing the problems solved by SGPlan grows and reaches a constant value on 36 plans, corresponding to the total generated problem. On the contrary, the curve representing the plans generated by LPG decreases up to represent less than 5 plans for problems related to 500 students.

In order to provide a more detailed illustration of the difference between the two planners' behaviour, Table 5.3 shows the percentage of PDDL problems solved respectively by LPG and SGPlan for 1, 50, 100, 150, 200, 250, 300, 350, 400, 450 and 500 students.

On the other hand, Table 5.4 illustrates the median value of actions generated by each planner in order to solve the problems. It is possible to observe

The screenshot displays the Moodle course interface for a course titled "Medium course". At the top right, it indicates the user is logged in as "Valentina Caputi (Logout)". Below the course title, there is a breadcrumb trail "cbp > cmed" and a "Switch role to..." dropdown menu with a "Turn editing on" button.

The main content area is divided into several sections:

- People:** A section with a "Participants" link.
- Activities:** A section with "Forums" and "Lessons" links.
- Search Forums:** A search box with a "Go" button and an "Advanced search" link.
- Administration:** A list of administrative tools including "Turn editing on", "Settings", "Assign roles", "Grades", "Groups", "Backup", "Restore", "Import", "Reset", "Reports", "Questions", and "Files".
- Weekly outline:** A central section titled "Weekly outline" containing a "Foro de Noticias" forum and a list of lessons labeled L.0 through L.9, Lvb, Lb, Lg, and Lvg.
- 25 January - 31 January:** A section with a checkbox.
- 1 February - 7 February:** A section with a checkbox.
- Latest News:** A section with a link to "Add a new topic..." and a note that no news has been posted yet.
- Upcoming Events:** A section stating there are no upcoming events, with links to "Go to calendar..." and "New Event..".
- Recent Activity:** A section showing activity since Monday, 23 September 2013, 10:28 AM, with a link to "Full report of recent activity..." and a note that nothing new has occurred since the last login.

Figure 5.6: Moodle's representation of the Medium course.

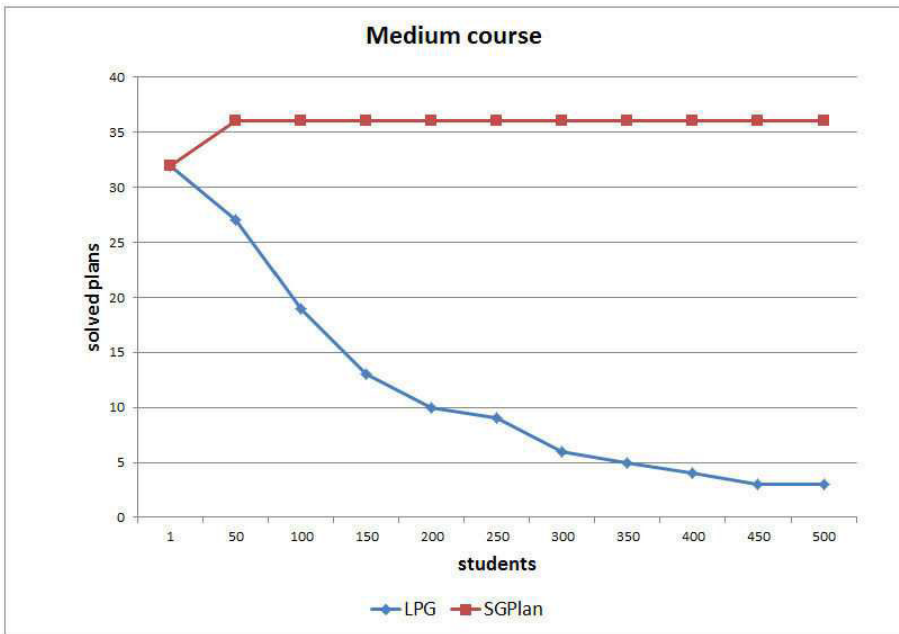


Figure 5.7: Number of solved problems for the Medium course by the two planners.

Number of students	LPG	SGPlan
1	32(88.8%)	32(88.8%)
50	27(75%)	36(100%)
100	19(52.7%)	36(100%)
150	13(36.1%)	36(100%)
200	10(27.7%)	36(100%)
250	9(25%)	36(100%)
300	6(10.6%)	36(100%)
350	5(13.8%)	36(100%)
400	4(11.1%)	36(100%)
450	3(8.3%)	36(100%)
500	3(8.3%)	36(100%)
Total	131(33%)	392(98.9%)

Table 5.3: Percentage of problems solved by LPG and SGPlan for the Medium course.

that the median value of actions generated by SGPlan, as in the case of the Small course, grows linearly and reaches a maximum value for problems with 500 students while the actions generated by LPG oscillates around the value 240 and reach a maximum value (303 actions) for problems with 250 students.

Finally, Fig. 5.8 represents the average time spent by each planner to find solutions for 1, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 students respectively. LPG takes for all experiments a time that we can approximate to 0.3 seconds and reaches a peak of 0.67 seconds in problems with 500 students for which the planner finds only 3 plans. On the other hand, the time required by SGPlan to solve the problems grows in line with the increasing number of the students and it reaches a maximum value (30.63 seconds) in solving the total of the problems (36) for 500 students.

5.2.4 Big course

To further increase the complexity of our evaluation, we have designed the Big course, composed by 41 lessons with durations and graphically represented in Fig. 5.9. Thus, as well as for the other courses we carried out the implementation in Moodle (Fig. 5.10). Our system automatically generates a PDDL domain containing 41 actions (as well as the number of lessons included in the course).

In order to generate a high number of PDDL problems, we randomly assigned up to 12 initial states and up to 12 goals to each student for each of the 500 fictitious students enrolled in the Big course. In particular, the two vectors

Number of students	LPG (for solved problems)	SGPlan (for solved problems)
1	4(32)	4(32)
50	138(27)	202(36)
100	209(19)	406(36)
150	227(13)	607(36)
200	252(10)	793(36)
250	303(9)	1005(36)
300	247(6)	1206(36)
350	229(5)	1399(36)
400	252(4)	1607(36)
450	299(3)	1818(36)
500	312(3)	2003(36)

Table 5.4: Median value of actions generated by LPG and SGPlan in order to solve the problems in the Medium course.

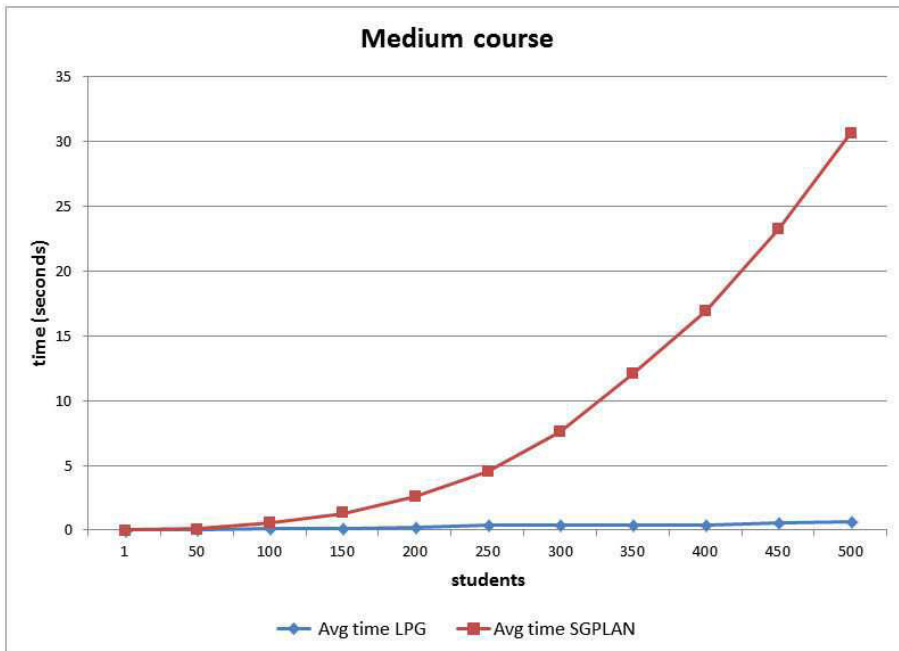


Figure 5.8: Average time (in seconds) to find plans for the Medium course by the two planners.

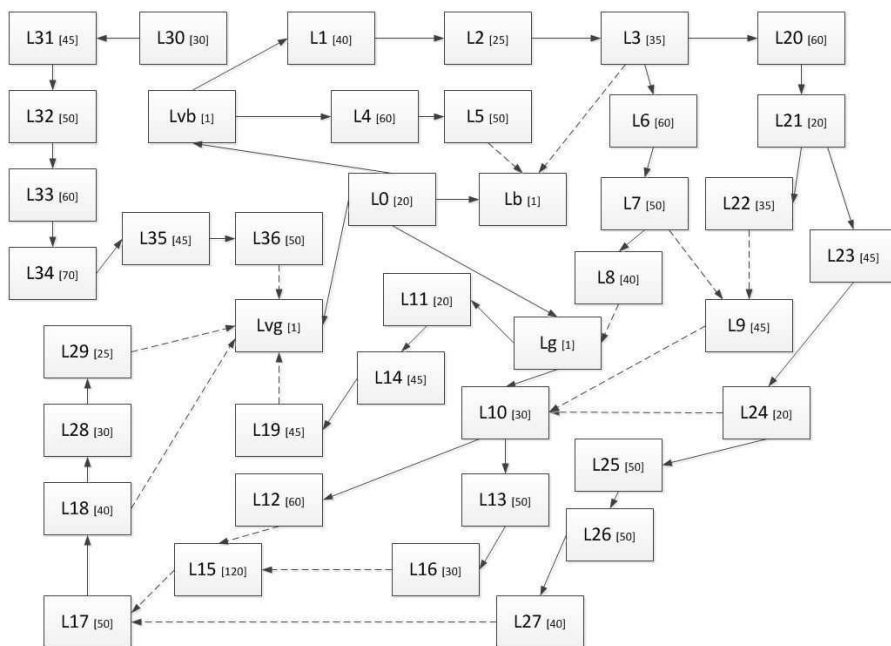


Figure 5.9: Representation of the Big course’s structure (durations are in minutes between brackets, activity links are represented as broken lines and dependencies as continuous lines).

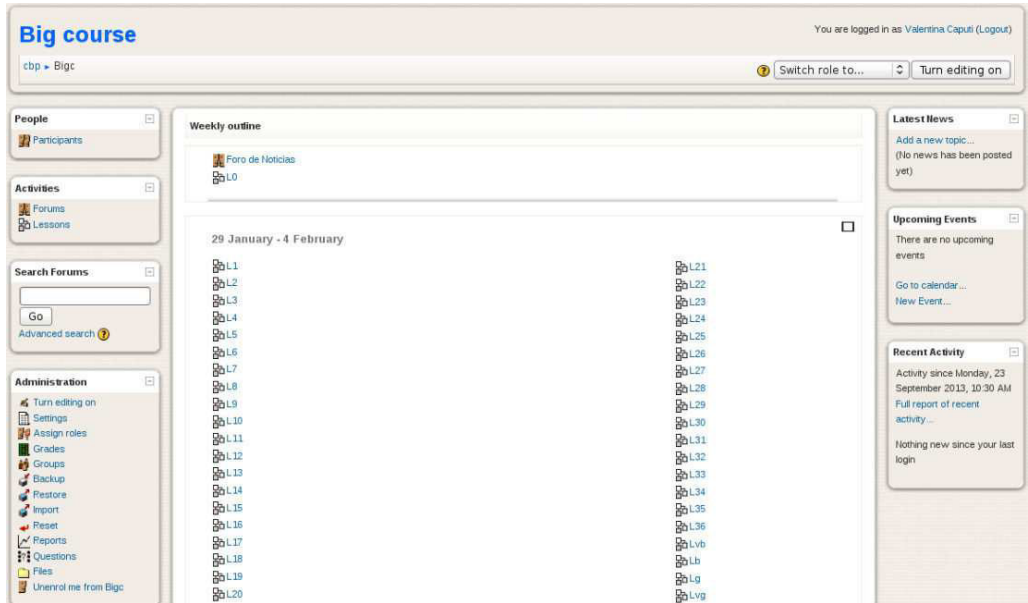


Figure 5.10: Moodle's representation of the Big course.

representing the possible number of initial states and goals are composed by 7 values: 1, 2, 4, 6, 8, 10, 12. By combining the elements of the two vectors the system randomly chose the corresponding numbers of initial states and goals to create 49 (7×7) PDDL problems for 1, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 students respectively.

Once created the PDDL domain and problems we generated the plans illustrated in Fig. 5.11. We can notice that LPG and SGPlan solve the same number of problems when only a fictitious student is enrolled in the course. Overtaken this value, LPG solves fewer and fewer problems until reaching problems with 300 students for whom the planner is no longer able to find solutions. On the contrary, SGPlan solves all the generated problems (49) until 400 students. From this value on, the planner decreases its performance to reach the minimum value of generated plans (43) in problems with 500 students. Table 5.5 shows the percentage of PDDL problems solved respectively by LPG and SGPlan for all the students in the test.

Table 5.6 shows the median value of actions generated by each planner. We can notice that LPG generates lower actions than SGPlan (except in problems for one student for which both planners generate the same number of actions),

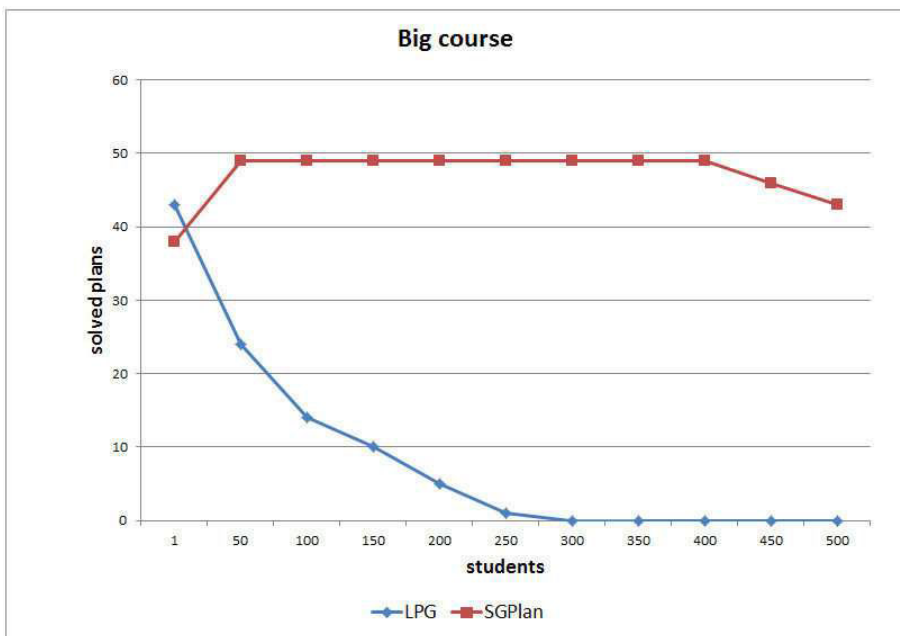


Figure 5.11: Number of solved problems for the Big course by the two planners.

Number of students	LPG	SGPlan
1	49(100%)	49(100%)
50	24(48.9%)	49(100%)
100	14(28.6%)	49(100%)
150	11(22.4%)	49(100%)
200	5(10.2%)	49(100%)
250	1(2%)	49(100%)
300	0(0%)	49(100%)
350	0(0%)	49(100%)
400	0(0%)	49(100%)
450	0(0%)	46(93.9%)
500	0(0%)	43(87.7%)
Total	104(19.2%)	530(98.3%)

Table 5.5: Percentage of problems solved by LPG and SGPlan for the Big course.

and this is easily explainable by considering that number of problems solved by LPG is always much lower than the number of problems solved by SGPlan.

It is possible to observe in Fig. 5.12 the average time spent by each planner to find solutions for problems that involve 1, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 students respectively. As observed in Fig. 5.11, LPG only solves problems with a maximum of 250 students, but it always takes a time lower than 6 seconds. On the contrary, SGPlan takes a time proportional to the total number of students enrolled in the course and it takes a maximum time of nearly 900 seconds for problems related to 500 students.

5.2.5 Very big course

The most complex experiment that we carried out involves a course consisting of 64 lessons with durations, as graphically illustrated in Fig. 5.13. Once designed the course, we implemented it in Moodle, as shown in Fig. (5.14). At this point, a PDDL domain consisting of 64 actions is generated.

We randomly assigned up to 20 initial states and up to 20 learning goals to the 500 fictitious students enrolled into the course in order to generate a number of PDDL problems. Each vector representing the possible number of initial states and learning goals is composed by the following 11 values: 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20. The system automatically generates 121 (11*11) PDDL problems (for 1, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 students, respectively) by combining the elements of the two vectors and randomly choosing the corresponding number of initial states and learning

Number of students	LPG (for solved problems)	SGPlan (for solved problems)
1	12(49)	12(49)
50	343(24)	596(49)
100	449(14)	1190(49)
150	640(11)	1845(49)
200	575(5)	2417(49)
250	415(1)	2999(49)
300	0(0)	3520(49)
350	0(0)	4173(49)
400	0(0)	4824(49)
450	0(0)	5268(46)
500	0(0)	5567(43)

Table 5.6: Median value of actions generated by LPG and SGPlan in order to solve the problems in the Big course.

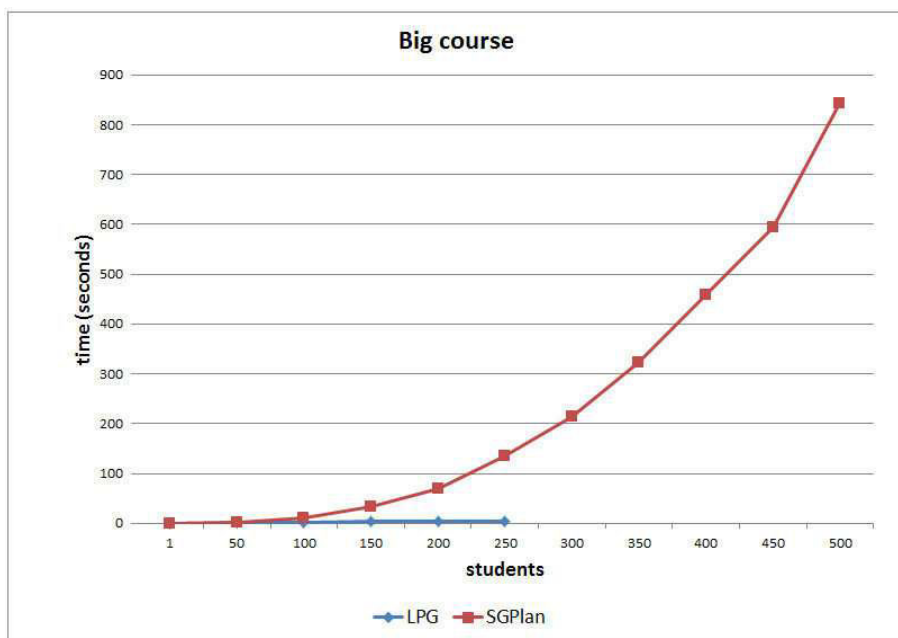


Figure 5.12: Average time (in seconds) to find plans for the Big course by the two planners.

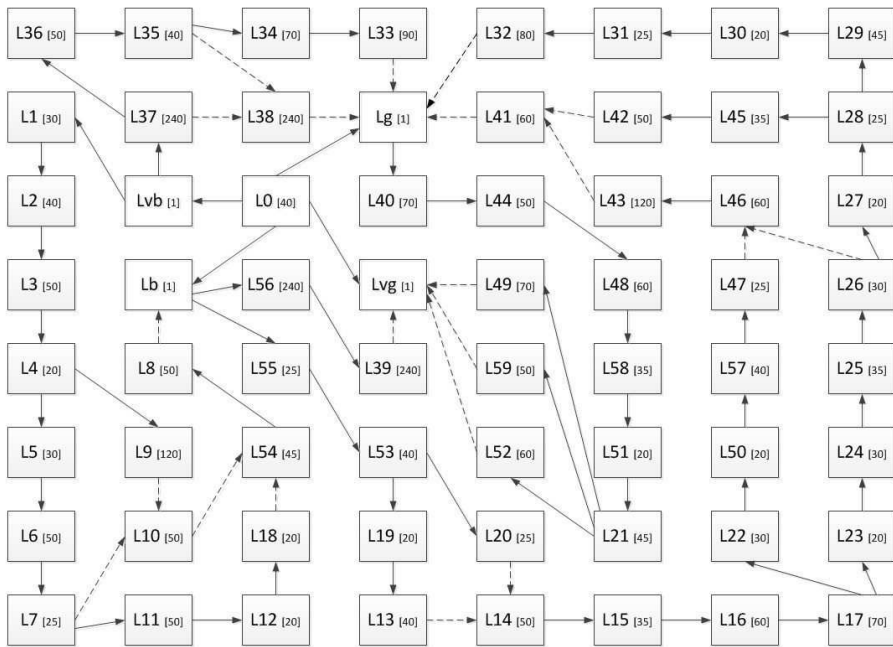


Figure 5.13: Representation of the Very big course's structure (durations are in minutes between brackets, activity links are represented as broken lines and dependencies as continuous lines).

The screenshot displays the Moodle interface for a course titled "Very big course". The user is logged in as "Valentina Caputi". The course breadcrumb is "cbp > vbc". The main content area shows a "Weekly outline" for a forum titled "Foro de Noticias", listing 49 topics (L0 to L48) arranged in three columns. The left sidebar contains several panels: "People" (Participants), "Activities" (Forums, Lessons), "Search Forums" (with a search box and "Go" button), and "Administration" (with options like Turn editing on, Settings, Assign roles, Grades, Groups, Backup, Restore, Import, Reset, and Reports). The right sidebar contains "Latest News" (with an "Add a new topic..." button), "Upcoming Events" (stating "There are no upcoming events"), and "Recent Activity" (showing activity since Tuesday, 8 October 2013, 07:33 PM).

Figure 5.14: Moodle's representation of the Very big course.

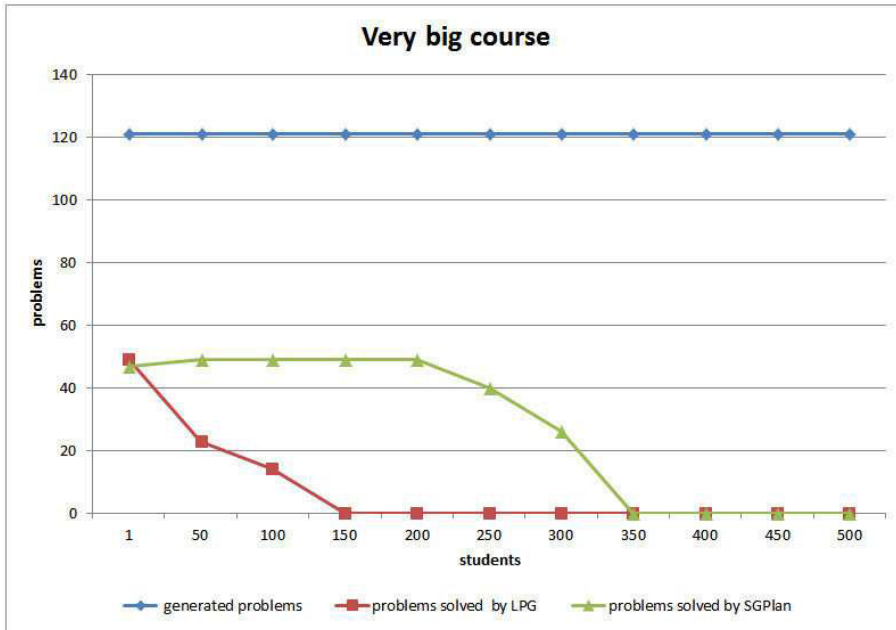


Figure 5.15: Number of solved problems for the Very big course by the two planners.

goals among the various lesson of the course.

Fig. 5.15 and Table 5.7 show that for problems with only one student LPG solves a greater number of problems than SGPlan. In particular, LPG solves 49 problems while SGPlan solves 47 problems (respectively, 40.5% and 38.8% of the total of generated problems). We can also notice that, overtaken this value, LPG decreases its performance with respect to SGPlan by always generating a lower number of plans. In particular, once reached problems involving 150 students, the planner is not longer able to solve the problems. On the contrary, SGPlan generates plans for problems involving up to 350 students. The maximum number of solved problems by SGPlan is 49 (corresponding to the 40.5% of the total of generated problems) and this value is reached for problems with 50, 100, 150 and 200 students.

Table 5.6 shows the median value of actions generated for the Very big course. We can observe that LPG generates a maximum number of actions (692) in correspondence of problems with 100 students while SGPlan (which solves much greater number of problems than LPG) achieves the maximum

Number of students	LPG	SGPlan
1	49(40.5%)	47(38.8%)
50	23(19%)	49(40.5%)
100	14(11.6%)	49(40.5%)
150	0(0%)	49(40.5%)
200	0(0%)	49(40.5%)
250	0(0%)	40(33%)
300	0(0%)	26(21.5%)
350	0(0%)	0(0%)
400	0(0%)	0(0%)
450	0(0%)	0(0%)
500	0(0%)	0(0%)
Total	86(6.4%)	309(23.2%)

Table 5.7: Percentage of problems solved by LPG and SGPlan for the Very big course.

number of generated actions (3942) for problems with 250 students.

Furthermore, we can observe in Fig. 5.16 the average time taken by each planner. LPG never uses more than about 13 seconds to solve the problems, value reached for 100 students. For the same number of students SGPlan takes a time of approximately 39 seconds while it reaches a maximum execution time (about 535 seconds) for problems with to 300 students.

5.2.6 Summary

Table 5.9 shows the vectors indicating the number of students' initial states and learning goals used depending on the specific tested course, the number of generated problem for each number of student and the total of PDDL problems generated for the 11 (1, 50, 100, 150, 200, 250, 300, 350, 400, 500) different numbers of students for each course.

In total, our system automatically generates 2442 problems that we tried to solve with LPG and SGPlan by setting a limit of 900 seconds per problem. Fig. 5.17, Fig. 5.18 and Table 5.10 show the total number of solved problems and the percentage of solved problems by the two planners for the four courses, respectively.

We can immediately observe that in general SGPlan is more able than LPG in solving problems with a high number of students. Specifically, LPG has some limitations when dealing with courses with more than 100-150 students because it can only solve problems that contain up to a maximum of 15000 predicates. It is also possible to notice in Fig. 5.17 that the courses that LPG

Number of students	LPG (for solved problems)	SGPlan (for solved problems)
1	20(49)	21(47)
50	502(23)	978(49)
100	692(14)	1908(49)
150	0(0)	2856(49)
200	0(0)	3844(49)
250	0(0)	3942(40)
300	0(0)	3227(26)
350	0(0)	0(0)
400	0(0)	0(0)
450	0(0)	0(0)
500	0(0)	0(0)

Table 5.8: Median value of actions generated by LPG and SGPlan in order to solve the problems in the Very big course.

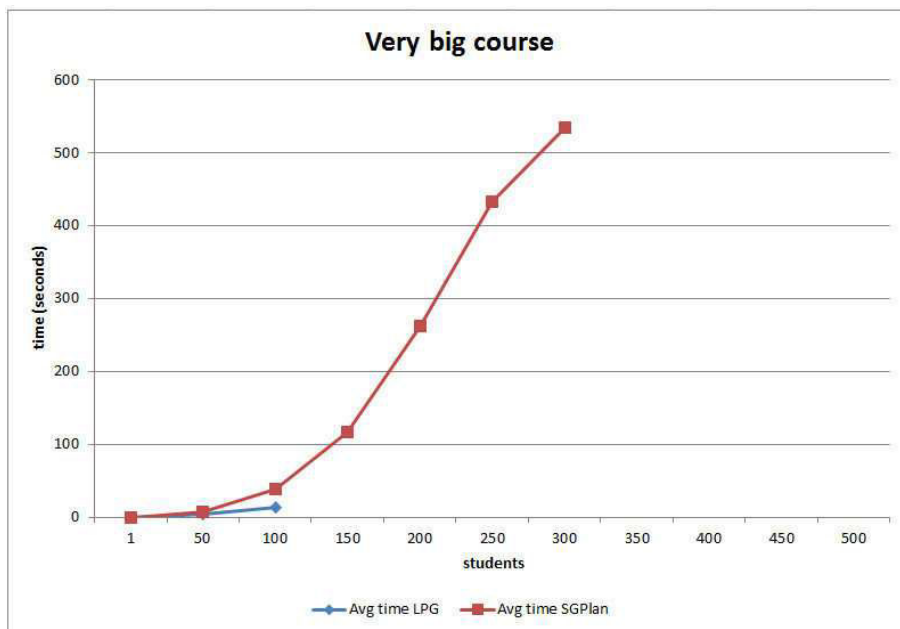


Figure 5.16: Average time (in seconds) to find plans for the Very big course by the two planners.

Course	Possible number of students' initial states/learning goals	Generated problems for each number of students	Total generated problems for all the 11 different numbers of students
Small	1, 2, 4, 6 (4 values)	16(4*4)	176(16*11)
Medium	1, 2, 4, 6, 8, 10 (6 values)	36(6*6)	396(36*11)
Big	1, 2, 4, 6, 8, 10, 12 (7 values)	49(7*7)	539(49*11)
Very big	1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 (11 values)	121(11*11)	1331(121*11)

Table 5.9: Possible number of students' initial states/learning goals in the various courses and total generated problems for each course.

Course	# generated problems	LPG	SGPlan
Small	176	73(41.48%)	174(98.85%)
Medium	396	131(33.08%)	392(98.99%)
Big	539	104(19.28%)	530(98.33%)
Very big	1331	86(6.46%)	309(23.21%)
Total	2442	394(16.13%)	1405(57.53%)

Table 5.10: Percentage of problems solved by LPG and SGPlan for the four courses.

is more able to solve are the Small course and the Medium course. Thus, we can certainly affirm that this planner can be used in a profitable way for customizing courses containing approximately 15 lessons and involving a maximum of 100-150 enrolled students. On the contrary, SGPlan shows a very scalable behavior and has little problems in finding plans for all students in the Small, Medium and Big courses and a few more problems in generating plans for the Very big course (Fig. 5.18). In particular, we can observe that for the Very big course SGPlan can just solve problems with up to 350 students (by considering the limit of 900 seconds that we set).

On the other hand, Fig. 5.19 and Fig. 5.20 show the average time to solve problems depending on the total number of students. The plots show that LPG is very fast in the problems it manages to solve. In particular, it takes less than 5 seconds in finding plans for the Small and "Medium" courses, even for up to 500 students. The maximum time taken by LPG for the resolution

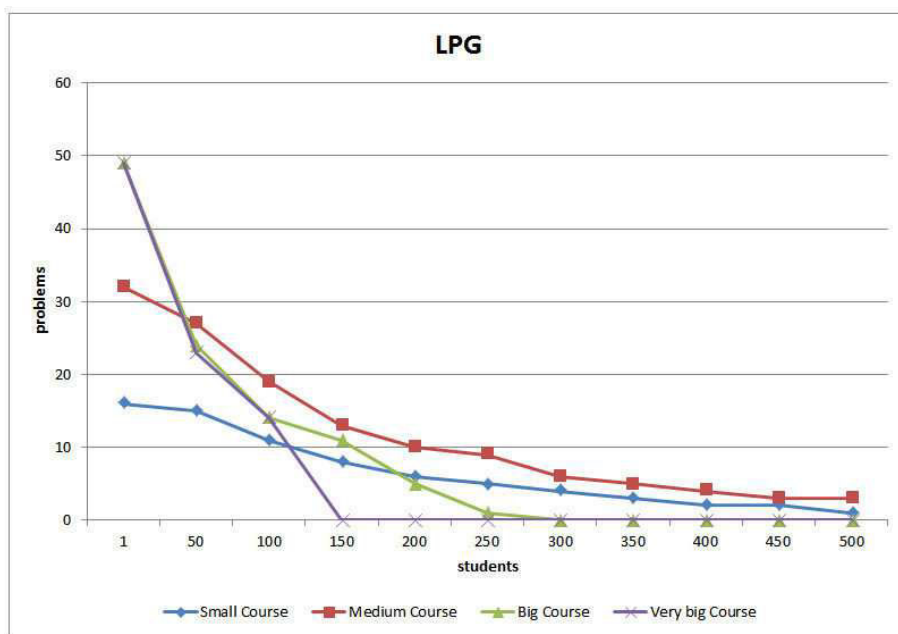


Figure 5.17: Number of solved problems by LPG in all four courses.

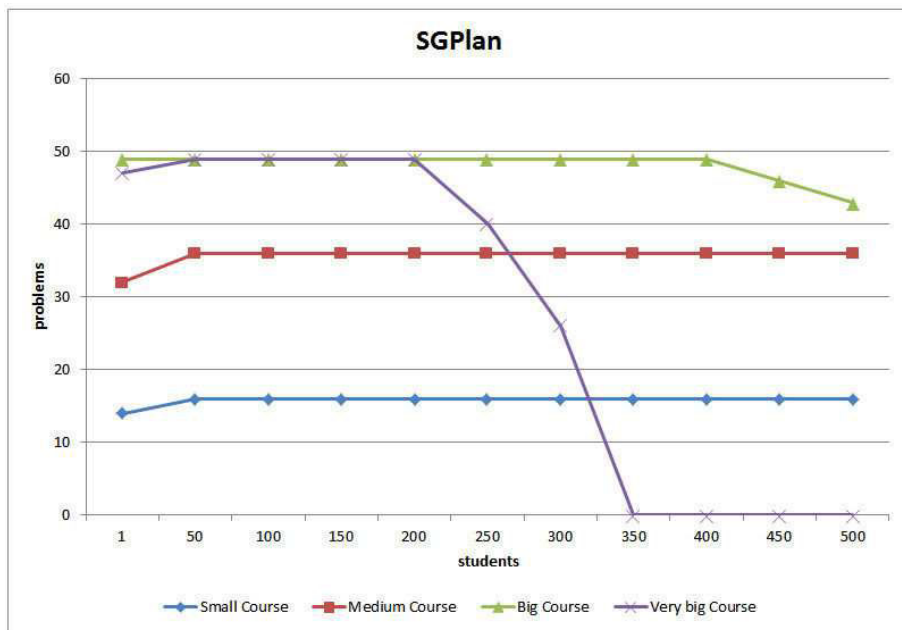


Figure 5.18: Number of solved problems by SGPlan in all four courses.

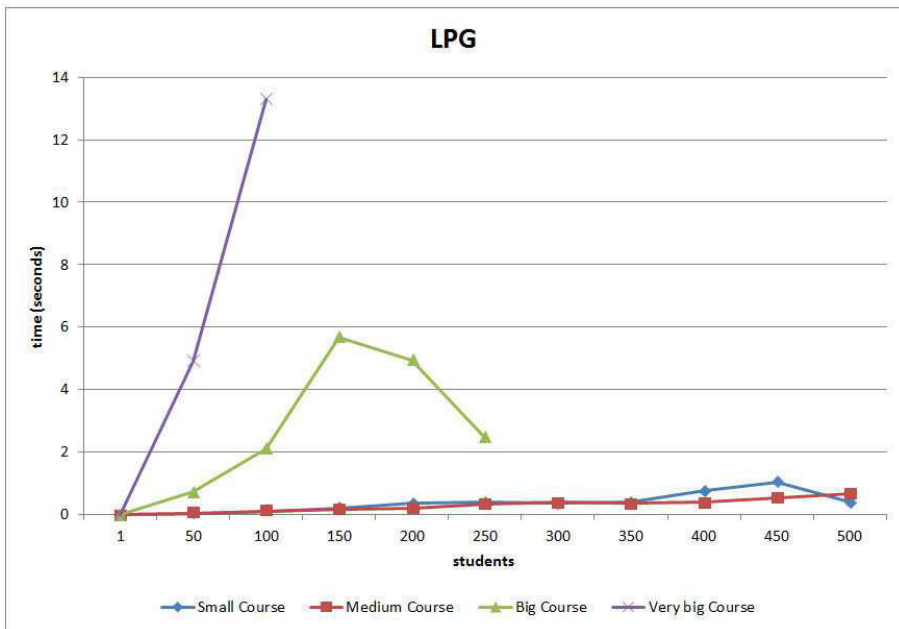


Figure 5.19: Average time (in seconds) to find plans by using LPG.

of problems related to the Big course is 5.67 seconds, for problems with 150 students. Furthermore, in the generation of plans for the Very big course, LPG takes a maximum execution time of 13.28 seconds for 100 students.

On the contrary, SGPlan takes in general more time, but it also solves more problems. In particular, problems of the Big course even with 500 enrolled students are solved by SGPlan in less than 900 seconds, which is an excellent result. The only course for which SGPlan is not able to find all the solutions in 900 seconds is the Very big course. In this case, SGPlan only solves problems with 350 taking 534.69 seconds.

The obtained results demonstrate that current planning technology is sufficient to solve the personalization planning (also very big) problems we apposite create in our evaluation approach in order to test the scalability our system.

5.3 Conclusions

In this Chapter we have given evidence of the scalability of our system. In particular, we performed a number of tests by using courses of different sizes

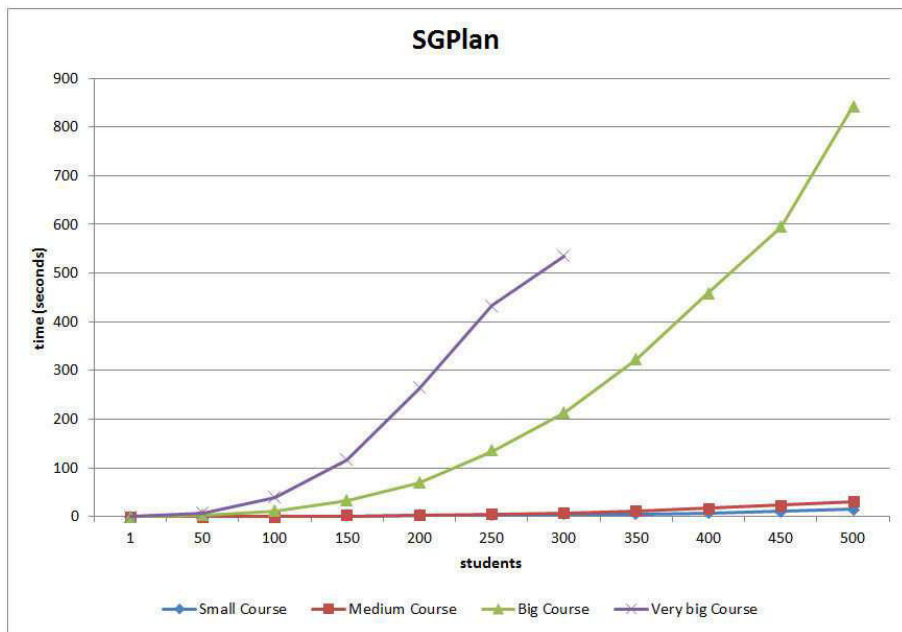


Figure 5.20: Average time (in seconds) to find plans by using SGPlan.

and with different numbers of students. It would have been practically impossible to gather hundreds of students to test on-line courses of big dimensions. Thus, we performed the scalability evaluation by using fictitious courses and fictitious students.

The results of our tests have shown that for a reasonable time it is possible to find plans that include personalized learning paths even for a large number of students (up to 350 students) and for courses containing a large number of lessons (up to 64 lessons). In order to solve these problems, we used two specific standard planners (LPG and SGPlan) but it is important to highlight that our system can use any standard planner. The results show that, for example, given a course consisting of 64 lessons, LPG creates personalized learning paths in less than 14 seconds for a maximum of 100 students. On the contrary, SGPlan permits us to find plans also for a large amount of students in very big courses. Courses containing up to 64 lessons and with 300 students are solved by SGPlan in less than 900 seconds, which is an excellent result.

We noticed that LPG is faster than SGPlan in solving small and medium-size problems (containing less than 15000 predicates). That is, in planning personalized learning paths within small/medium courses with a reduced number of students. On the other hand, SGPlan shows better results in terms of scalability, by solving in general a bigger number of problems. Clearly, learning paths' personalization does not usually need to involve such a high number of students because independent paths can be generated for much smaller groups of students, which means having many different problems but with no more than 20-50 students each. This means our experiments significantly exceed the usual requirements and, therefore, we prove that planning technology can successfully cope with very demanding courses.

Given all the foregoing considerations, we decided to use LPG in our system because we are oriented to the resolution of problems concerning small/medium courses with a limited number of students and LPG is the faster planner to use in this case.

Chapter 6

A real case study: personalizing a Physics course in Moodle

In this Chapter we present a practical application of our system within Moodle and using a real course. We implement in Moodle a real Physics course intended for students of the Faculty of Physics, University of Calabria (Italy). We provide personalized learning paths for ten students enrolled in the course.

6.1 Students' difficulties in learning Physics concepts

Many studies have discussed about the problems that the students have to face in order to learn Physics concepts ([6]; [51]). In particular, the main problem in learning Physics topics arises in the passive role that students have in the learning process. That is, what the teachers believe about the concepts assimilated by the students is not always what corresponds with what the students have truly learned ([6]; [51]; [52]; [65]). Too often, students solve exercises without understanding the laws and concepts that are the basis of the argument that they are studying ([4]; [49]; [51]). Therefore, learning Physics can be very boring for the students if they have a static view (without connections to the real world) of the concepts that they have to study [6]; [47]; [66].

On the contrary, it is fundamentally important that students have a con-

structive role in the learning process, that is, they have to be driven by spontaneous reasoning in the building of their knowledge. Thus, it is necessary that students are an active part in building their own knowledge [18]; [22]; [45]; [52]; [65]; [66]. Physics experiments have to be seen by students as something concrete and, above all, connected to the real world. A way to achieve this result is the use of the personal computer combined with laboratory tools [52]. In particular, on-line experiments represent valid ways in order to learn Physics topics because the student can follow on-line (in real time) the whole evolution of the phenomena he/she is studying [6]; [7]; [8].

In the following Section we detail the difficulties that students found in studying the body motion, topic of the course that we implemented in Moodle.

6.1.1 The body motion

The majority of the students find difficulties in the analysis of the body motion in the graphic representation of position, velocity and acceleration with respect to the time [49]. These difficulties are not exclusively attributable to the students' mathematics gaps, because even those who have acquired good knowledge of mathematics topics demonstrate difficulties in applying the mathematical concepts in physical analysis. In particular, a concept that is often misunderstood by the students concerns the relationship between force and acceleration. In fact, students tend to confuse the relationship between force and acceleration with the relationship between force and velocity, and they are driven by the spontaneous idea that a body subject to constant force is subject to constant speed instead of constant acceleration [50]. Thus, in order to structure a course concerning the body motion, it is fundamental to achieve two many objectives:

1. propose learning contents that drive the students in a correct way to the right comprehension of the concepts related to Newton's second law (i.e. a sum of non-zero constant forces acting on a body corresponds to a uniformly accelerated rectilinear motion);
2. design the course in a personalized way that aids the student to construct his/her knowledge, by assuming an active role in the learning process.

In the following Sections we illustrate the course that we designed in order to achieve the first objective and the implementation in Moodle to personalize the course's concepts for each student in order to achieve the second objective.

6.2 The Physics course: "Il moto dei corpi"

In order to achieve the objectives mentioned in the previous Section, we decided to implement in Moodle the Physics course "Il moto dei corpi"¹ (the body motion in English). The course has been structured in conjunction with Giacomo Bozzo, PhD in "Physics Education" at the University of Calabria (Italy). It is a mainly experimental course which is arranged in such a way that the experiments covered in each lesson are characterized by progressive difficulty. In this way, new physical quantities are introduced one by one and students do not feel disoriented because they are not put in front of too many variables at a time. We present the course's structure (lessons' contents, precedences and timing) and the subsequent implementation in Moodle. Furthermore, we illustrate the automatic learning paths' generation and personalized visualization for ten students enrolled in the course.

6.2.1 Course's structure and adaptation in Moodle

"Il moto dei corpi" is composed by 16 lessons (12 real lessons and 5 fictitious lesson). The course is structured as illustrated in Fig. 6.1 (where we highlighted the dependency links as continuous lines, the activity links as broken lines and the lessons' durations between brackets). To facilitate the graph's reading, we provide the extended lessons' names in Table 6.1.

We can detail each real lesson, by also providing some graphical examples, as follows:

- L0 (initial questionnaire): contains all the initial states and goals that a student can choose in order to take the personalized course. In particular, it is possible to choose a number of lesson among all the lesson present in the course representing the initial states and the learning goals.
- L1 (ability to foresee the motion of a body): contains a test (as shown in Fig. 6.2) that is used to check if the students know and have fully understood the concepts associated with uniform motion, uniformly accelerated rectilinear motion and parabolic motion;
- L2 (motion on a horizontal smooth plane): is used to check the students' ideas on the space-time relationship in a rectilinear motion (see Fig. 6.3);

¹We use the Italian terminology within the course because it is addressed to students belonging to the University of Calabria.

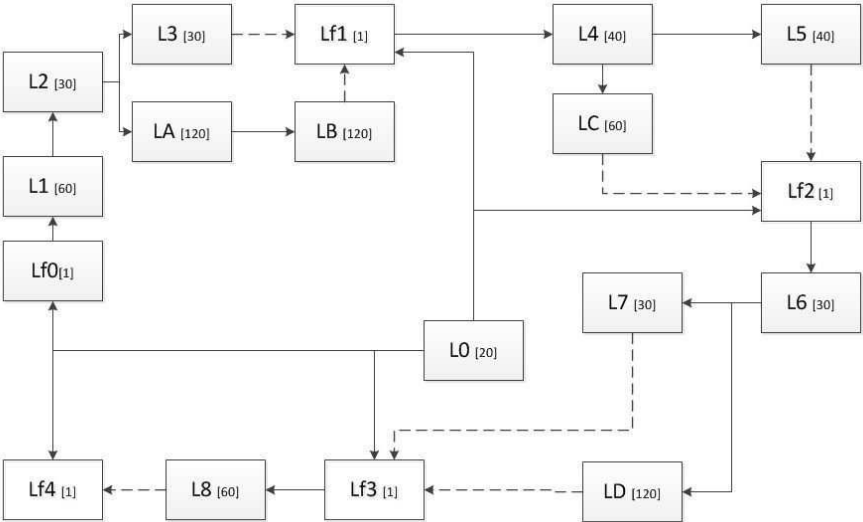


Figure 6.1: Physics course’s structure. Durations in minutes between brackets, activity links as broken lines and dependencies as continuous lines. Lessons’ extended names in Table 6.1.

Lesson	Lesson's name
L0	Questionario iniziale
Lf0	Nessuna nozione previa riguardo al moto dei corpi
Lf1	Conoscenza del moto rettilineo uniforme e del moto rettilineo uniformemente accelerato
Lf2	Conoscenza di un moto parabolico
Lf3	Conoscenza della relazione spazio tempo nel moto rettilineo uniforme e uniformemente accelerato.
Lf4	Conoscenza dei moti unidimensionali e parabolici
L1	Capacità di previsione del moto di un corpo
L2	Moto di un corpo su un piano orizzontale liscio
L3	Moto di un corpo su un piano inclinato liscio
L4	Moto di un proiettile
L5	Moto rettilineo uniforme consecutivo a moto rettilineo uniformemente accelerato
L6	Rappresentazione grafica spazio tempo nel moto rettilineo uniforme
L7	Rappresentazione grafica spazio tempo nel moto rettilineo uniformemente accelerato
L8	Test di valutazione finale
LA	Esperimento reale sul moto rettilineo uniforme
LB	Esperimento reale sul moto rettilineo uniformemente accelerato
LC	Simulazioni sul moto parabolico
LD	Analisi e conoscenza dei testi didattici sui moti rettilinei uniformi e uniformemente accelerati


Table 6.1: Initial questionnaire (L0) for the Physics course.

- L3 (body motion on an inclined smooth plane): is used to verify the student's knowledge about the space-time relationship in a uniformly accelerated rectilinear motion (see Fig. 6.4);
- L4 (projectile motion): contains a test that permits us to verify the students knowledge about the parabolic motion (see Fig. 6.5);
- L5 (uniform rectilinear motion consecutive to uniformly accelerated rectilinear motion): is used to verify the student's knowledge on the space-time relationship in a combination of uniform motion and uniformly accelerated rectilinear motion (as shown in Fig. 6.6);
- L6 (graphical representation of space-time in uniform rectilinear motion): involves the explanation of the graphical representation of space-time in uniform rectilinear motion (Fig. 6.7);
- L7 (graphical representation of space-time in the uniformly accelerated rectilinear motion): contains the explication of the graphical representation of space-time in the uniformly accelerated rectilinear motion (Fig. 6.8);
- L8 (final evaluation test): is a final test of the course to check whether the student has acquired all the learning concepts contained into the course.
- LA (real experiment on uniform rectilinear motion): contains an experiment about the uniform rectilinear motion;
- LB (real experiment on uniformly accelerated rectilinear motion): includes an experiment about the uniformly accelerated rectilinear motion;
- LC (simulations on the parabolic motion): contains various simulations in order to show the different motions' compositions;
- LD (analysis and knowledge of educational texts about uniform and uniformly accelerated rectilinear motions): involves the analysis of textbooks on the uniform and uniformly accelerated rectilinear motions.

On the other hand, each fictitious lesson represents a knowledge state:

- Lf0: none prior notion about the body motion;

LEZIONE 1 (test con esercizio proposto): **Analisi di un esperimento reale – osservazione e discussione**



Si somministrano le seguenti domande a risposta aperta:

- 1) Descrivi il fenomeno osservato.
- 2) Cosa pensi che succeda alla pallina?

Figure 6.2: A portion of the learning content of L1.

- Lf1: knowledge about the uniform motion and uniformly accelerated rectilinear motion;
- Lf2: knowledge about the parabolic motion;
- Lf3: knowledge about the space-time relationship in uniform motion and uniformly accelerated;
- Lf4: knowledge about the uniform motion and uniformly accelerated rectilinear motion.

Once designed the course by using the lesson module, we implemented it in Moodle, as shown in Fig.6.9 that represents the teacher's view of the course. By using all the course's information (dependencies and activity links of the various lessons), our system automatically generated a PDDL domain involving as many actions as the total number of course's lessons.


6.2.2 Plan generation and visualization


After the implementation of the course in Moodle, ten real students (that, for privacy we call Student 1, ..., Student 10) took the course and completed the initial questionnaire L0. The students' answers about initial states and learning goals are shown in Table 6.2.


Our system used the information about the students' initial states and learning goals to automatically generate a PDDL problem and invoked the

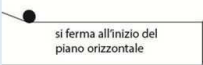
LEZIONE 2 (test delle conoscenze): Il piano orizzontale

DOMANDA:
La posizione della pallina lungo il tratto orizzontale viene memorizzata su una macchina fotografica in istanti successivi temporali equivalenti. Quali delle seguenti rappresentazioni è a tuo giudizio realistica? (inserire anche domande a risposta multipla con obiettivi concettuali)



A. 

B. 

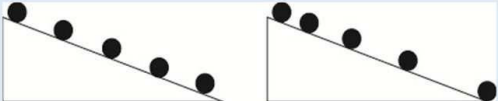
C. 

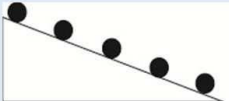
D. Non so rispondere

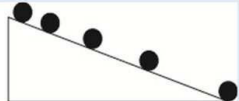
Figure 6.3: A portion of the learning content of L2.

LEZIONE 3 (test delle conoscenze): Il piano inclinato

DOMANDA:
La posizione della pallina lungo il piano inclinato viene memorizzata su una macchina fotografica in istanti successivi temporali equivalenti. Quali delle seguenti rappresentazioni è a tuo giudizio realistica? (inserire anche domande a risposta multipla con obiettivi concettuali)



A. 

B. 

C. Non so rispondere

Figure 6.4: A portion of the learning content of L3.

LEZIONE 4 (test delle conoscenze): Il moto nell'aria

DOMANDA:
 Individua la traiettoria a tuo giudizio più realistica dopo che la pallina ha lasciato il tratto orizzontale. (inserire anche domande a risposta multipla con obiettivi concettuali)

The figure shows five diagrams labeled A through E, each depicting a ball on a horizontal surface. A red arrow indicates the direction of motion after the ball leaves the surface. Diagram A shows a straight vertical line pointing downwards. Diagram B shows a straight diagonal line pointing downwards and to the right. Diagram C shows a curved path that starts horizontally and then curves downwards. Diagram D shows a horizontal line followed by a vertical line pointing downwards. Diagram E shows a horizontal line followed by a curved path that curves downwards.

Figure 6.5: A portion of the learning content of L4.

LEZIONE 5 (test delle conoscenze): Combinazione di un piano inclinato e di un piano orizzontale

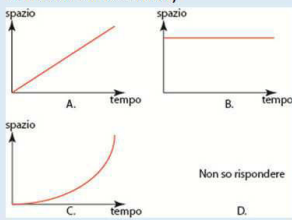
DOMANDA:
 Sulla macchina fotografica vengono ora memorizzate contemporaneamente le posizioni della pallina lungo il tratto obliquo e orizzontale. Quali delle seguenti rappresentazioni è a tuo giudizio realistica? (inserire anche domande a risposta multipla con obiettivi concettuali)

The figure shows four diagrams labeled A through D, each depicting a ball on an inclined plane and a horizontal plane. The ball's position is marked with a black dot. Diagram A shows the ball on the incline, then on the horizontal plane. Diagram B shows the ball on the incline, then on the horizontal plane. Diagram C shows the ball on the incline, then on the horizontal plane. Diagram D shows the ball on the incline, then on the horizontal plane.

Figure 6.6: A portion of the learning content of L5.

LEZIONE 6 (test delle conoscenze): Grafico spazio tempo nel moto di un corpo lungo un piano orizzontale

DOMANDA:
In un grafico vengono ora rappresentate le distanze percorse dalla pallina lungo il piano orizzontale in funzione del tempo. Quale grafico si otterrà? (inserire anche domande a risposta multipla con obiettivi concettuali)

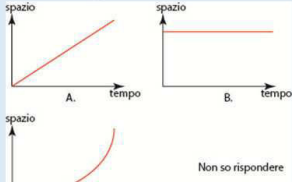


The figure shows four separate coordinate systems, each with a vertical axis labeled 'spazio' (space) and a horizontal axis labeled 'tempo' (time).
Graph A: A straight red line starts at the origin (0,0) and extends upwards and to the right at a constant slope.
Graph B: A horizontal red line is drawn at a constant positive value on the vertical axis.
Graph C: A red curve starts at the origin (0,0) and curves upwards, becoming steeper as it moves to the right.
Graph D: The text 'Non so rispondere' is written in the center of the graph area.

Figure 6.7: A portion of the learning content of L6.

LEZIONE 7 (test delle conoscenze): Grafico spazio tempo nel moto di un corpo lungo un piano inclinato

DOMANDA:
In un grafico vengono ora rappresentate le distanze percorse dalla pallina lungo il piano obliquo in funzione del tempo. Quale grafico si otterrà? (inserire anche domande a risposta multipla con obiettivi concettuali)



The figure shows three separate coordinate systems, each with a vertical axis labeled 'spazio' (space) and a horizontal axis labeled 'tempo' (time).
Graph A: A straight red line starts at the origin (0,0) and extends upwards and to the right at a constant slope.
Graph B: A horizontal red line is drawn at a constant positive value on the vertical axis.
Graph C: A red curve starts at the origin (0,0) and curves upwards, becoming steeper as it moves to the right.
The text 'Non so rispondere' is written in the center of the graph area.

Figure 6.8: A portion of the learning content of L7.



Figure 6.9: Implementation in Moodle of the Physics course.

planner LPG that, by starting from the information contained into the PDDL domain and problem, created a plan containing a customized learning path for each student. In particular, LPG generates in 0,03 seconds a number of actions corresponding to the lessons that each student has to take. We summarize the generated learning paths in Table 6.3. In Fig. 6.10 we can observe the temporal actions' sequence of each generated customized learning path. We omitted the representation of L0, which is obviously taken by every student in every moment s/he needs to define his/her initial state/learning goals.

Once in possession of all the information about the various students' customized learning paths, our system automatically generated the correct visibility associations between students and lessons so that each student only visualizes the lessons belonging to his/her own learning path. In particular, in Fig. 6.11 we can observe the course's personalized views from the point of view of the various students. Obviously, as shown in Fig. 6.11, each student can always visualize the initial questionnaire L0 so that at every moment he/she can obtain a new learning path adapted to his/her new eventual needs/difficulties, such as change in the learning objectives, difficulties in executing the current learning path in the expected time, etc.

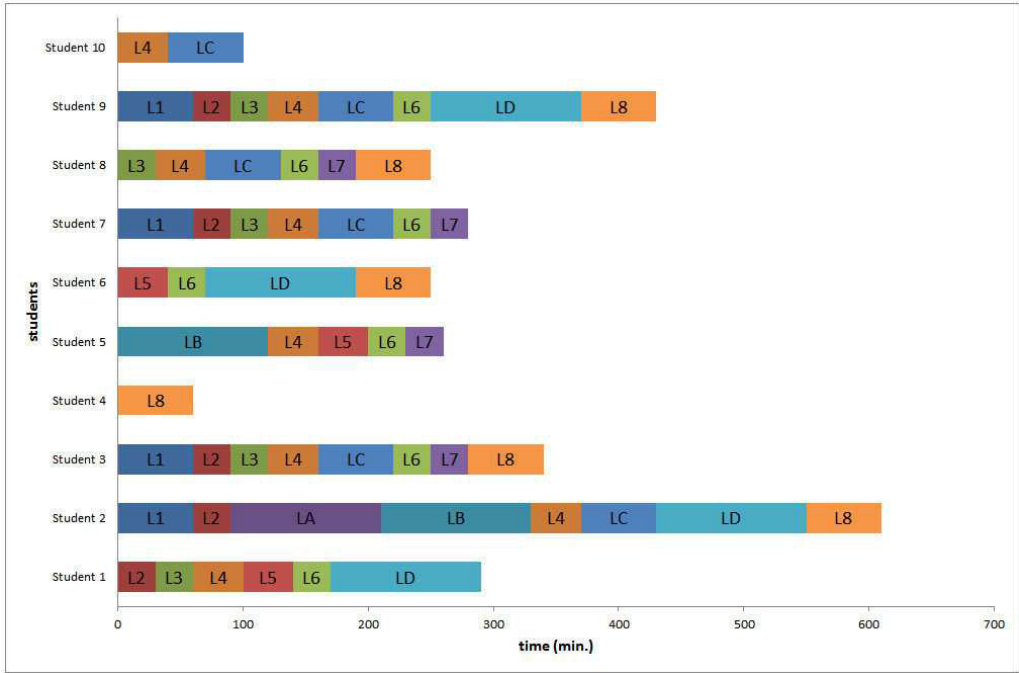


Figure 6.10: Graphical view of the generated students' learning paths for the Physics course.











































































<p>Weekly outline Student 1</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.2  L.3  L.4  L.5  L.6  L.D 	<p>Weekly outline Student 2</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.1  L.2  L.A  L.B  L.4  L.C  L.8
<p>Weekly outline Student 3</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.1  L.2  L.3  L.4  L.C  L.6  L.7  L.8 	<p>Weekly outline Student 4</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.8
<p>Weekly outline Student 5</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.B  L.4  L.5  L.6  L.7 	<p>Weekly outline Student 6</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.5  L.6  L.D  L.8
<p>Weekly outline Student 7</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.1  L.2  L.3  L.4  L.C  L.6  L.7 	<p>Weekly outline Student 8</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.3  L.4  L.C  L.6  L.7  L.8
<p>Weekly outline Student 9</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.1  L.2  L.3  L.4  L.C  L.6  L.D  L.8 	<p>Weekly outline Student 10</p> <p>Compila il questionario iniziale L.0 per personalizzare il tuo percorso di apprendimento</p> <p> Foro de Noticias</p> <ul style="list-style-type: none">  L.0  L.4  L.C

Figure 6.11: Physics course's view from the points of view of the students.

Student	Initial states	Learning goals
Student 1	Lf0	L2, Lf3, L5, LD
Student 2	Lf1, Lf2, L6	Lf4, LB, LC, LD
Student 3	Lf0	Lf3, Lf4, LC, L7
Student 4	Lf3	Lf4, L8
Student 5	L3, LA	Lf3, L7
Student 6	L4, LC, LA	Lf3, Lf4, LD
Student 7	L1, L2, L3	Lf2
Student 8	Lf0	Lf3, L6, LC
Student 9	L1, L2	Lf3, Lf4, L7
Student 10	Lf0	Lf4, LC, LD

Table 6.2: Students' initial states and learning goals.

Student	learning path
Student 1	L2, L3, L4, L5, L6, LD
Student 2	L1, L2, LA, LB, L4, LC, LD, L8
Student 3	L1, L2, L3, L4, LC, L6, L7, L8
Student 4	L8
Student 5	LB, L4, L5, L6, L7
Student 6	L5, L6, LD, L8
Student 7	L1, L2, L3, L4, LC, L6, L7
Student 8	L3, L4, LC, L6, L7, L8
Student 9	L1, L2, L3, L4, LC, L6, LD, L8
Student 10	L4, LC

Table 6.3: Students' personalized learning paths within the Physics course.

6.3 Conclusions

In this Chapter we have illustrated the practical application of our learning paths' customization system in the context of a real Physics course. In particular, together with Giacomo Bozzo (PhD in Physics education at the University of Calabria) we implemented in Moodle a Physics course addressed to students of the University of Calabria. We justified this choice by explaining, as observed in the literature, that a personalized course composed by experiments may facilitate students in learning Physics concepts.

Once implemented the course in Moodle, ten real students enrolled in it and each student expressed his/her preferences in terms of background and learning goals. Therefore, our system automatically generated a plan containing a personalized learning path (or a sequence to lessons to take) for each student. In addition, the system automatically generated also a customized course'

view in Moodle for each student depending on the specific learning path to be executed.

Chapter 7

Conclusions

This Chapter summarizes the main contributions of our work and identifies our future lines of research in order to extend these contributions.

7.1 Contributions

In this work we have performed a number of tasks that lead to the five contributions detailed as follows.

First, we have proposed a classification and evaluation of the techniques related to the personalization in e-learning. We analyzed various proposals that we classified into different groups, depending on the level of customization they offer. In particular we identified a first group of proposals that focuses on the support that can be provided to a teacher/student in the process of design, execution and monitoring a learning path. On the other hand, we grouped in a second set all proposals on the automatic generation of learning paths as the most appropriate combination of learning activities (i.e. learning path), depending on his/her specific needs (in terms of learning goals, initial learning state, learning style, preferences, etc.) We analyzed the techniques proposed in this second group, such as evolutionary algorithms, AI planning, data mining techniques, etc, as the most interesting techniques because they provide the possibility of automating the customization process, freeing the teacher from the burden of manually designing learning paths. However, we realized that none of the analyzed proposals provides the possibility of integrating a system for the automatic generation of curriculums within an existing LMS. Thus, we showed the challenges that still remain in the field of customization of curriculums.

Second, we have suggested a general mapping to switch from e-learning elements to planning elements. In particular, by analyzing a number of studies that proposes LMSs' comparisons, we established that Moodle is the best LMS (user friendly interface, support for multiple languages, support for more than one operative system, multilingual and exhaustive documentation, etc.) in order to implement our AI planning personalization system. Thus, we focused our general mapping by considering how to adapt Moodle's elements to the planning elements. After a detailed analysis of the platform, we have decided that the lesson is the most adequate module to correctly perform the personalization activity because, unlike the other modules, it offers the possibility to establish priority relationships. Furthermore, we have detected and solved some limitations in Moodle that prevented us from properly performing the customization tasks.

Third, we have provided a Moodle-dependent mapping to explain how to switch from Moodle's elements to planning elements and vice versa. In particular, we identified how to adapt the Moodle's lessons in planning constructs (by also identifying the selection of the Moodle database's relational tables necessary for defining the PDDL domain and problem). Once implemented in Moodle a way to automatically generate customized learning paths, we identified the Moodle's elements needed to establish a customized course's view in Moodle and we showed the Moodle database's relational tables needed to achieve this objective. Furthermore, provided some notes on the monitoring activity that has to be carried out together with the planning activity, in order to eventually re-plan a learning path (if, for example, the student changes his/her learning objectives or he/she is not able to correctly perform his/her personalized learning path).

Fourth, we have demonstrated a measure of applicability of our system. We used two standard planners to solve a number of tests involving courses of different sizes and with different numbers of students. We demonstrated that in a short time it is possible to find learning paths even for a large number of students and for very big courses. For example, courses containing up to 64 lessons and with 300 students are solved in less than 900 seconds that is an excellent result. By considering that learning paths' personalization does not usually need to involve such a high number of students, we noticed that our experiments significantly exceed the usual requirements. On the other hand, they demonstrate that planning technology can successfully cope with very demanding courses.

Fifth, we have put into practice our system by implementing in Moodle a

real Physics course addressed to students of the University of Calabria (Italy). We designed the course together with Giacomo Bozzo, PhD in Physics education at the University of Calabria. Once enrolled ten real students in the course, they expressed their preferences in terms of background and learning goals. Thus, our system automatically generates a personalized learning path for each student. In addition, the system generates personalized course's views in Moodle for the various students.

All in all, we achieved a way to automatically generate personalized learning paths within Moodle. We demonstrated the scalability of our system and, furthermore, we tested it by personalizing a real Physics course.

7.2 Conclusions

In our work, we have faced the learning paths' customization, from an AI planning perspective, in Moodle.

We have found that Moodle is an e-learning platform valid to incorporate the automatic personalization of learning paths. Although Moodle is a LMS that allows us to manage and to deliver courses' material in a simple and functional way, in order to get the maximum benefit from this platform, we provided the design of student-oriented learning paths (according to students' initial background and learning goals) to offer the best contents to the adequate person.

AI planning techniques have proved their worth in achieving the personalization of learning paths. In particular, we have carried out a mapping to switch from e-learning elements to planning elements and vice versa. By contextualizing this mapping to Moodle, we identified the Moodle' elements needed to generate the planning constructs and, therefore, to create personalized learning paths. Furthermore, once generated a plan, we shown the way to create personalized views in Moodle.

We gave a proof of the scalability of our system by testing it with two standard planners: LPG and SGPlan. We created a total of 2442 tests by using fictitious courses and fictitious students and we demonstrated that in 900 seconds it is possible to solve problems even for a large number of students (up to 350).

We noticed that LPG is faster than SGPlan in personalizing small-medium courses with a small number of students. On the other hand, SGPlan demonstrates a higher scalability by personalizing courses containing up to 64 lessons and with 300 enrolled students in less than 900 seconds. However, consider-

ing that we are focused on the personalization of small-medium courses, we decided to use LPG in our system.

7.3 Future lines of research

As part of our future work, we want to firstly evaluate the learning paths' execution within the Physics course that we have implemented in Moodle. In addition, we are oriented to the implementation in Moodle of more courses to be customized, monitored and evaluated. In fact, thanks to the flexibility of our system, it can be adapted in a straightforward way to any type of educational content and used by a wide variety of users.

Another line of future work involves the introduction of shared resources among the various students. For example, a lesson may represent a laboratory activity executable only by a determined group of students at a time. In this case, it is necessary to establish the manner of automatically allow only a specific group of students at a time to access the lesson.

Timing restrictions are another challenge that we want to address. In particular, a lesson may represent an activity (for example, the use of a machine to make experiments) that a student can only use for a specified time. In this case, it is necessary that the system automatically determines to accept or expel (if time has expired) the student from the lesson.

Considering that our system can be easily extrapolated to any other standard LMS, we are also interested in analyzing the main characteristics of the other standard LMSs in order to find ways to implement the system within them.

Finally, it would be interesting to develop a learning system that effectively covers all the strong points of the works which we analyzed in the literature. Thus, in addition to the automatic generation of personalized learning paths within a standard LMS (goal achieved in this work), it would be advisable that such LMS was totally modular, in order to plug&play different modules (e.g. modeling, assessment, monitoring tasks, etc.) and to be simply executed in different devices (e.g. desktop computers, tablets and mobile devices).

7.4 Publications

- Valentina Caputi and Antonio Garrido. Experiences on using intelligent planning for curriculum personalization in Moodle. In *EDULEARN13 Proceedings* (2013): 168-176.

7.4.1 Submitted works

- Valentina Caputi and Antonio Garrido. Towards curriculum personalization in e-learning systems. Submitted to *Journal of Educational Computing Research*.
- Valentina Caputi and Antonio Garrido. Student-oriented planning of e-learning contents for Moodle. Submitted to *Journal of Network and Computer Applications*.

References

- [1] Ajlan Al-Ajlan and Hussein Zedan. Why Moodle. In *Future Trends of Distributed Computing Systems, 2008. FTDCS'08. 12th IEEE International Workshop on*, pages 58–64. IEEE, 2008.
- [2] Francesco Altimari, Anna Franca Plastina, Michael D Cronin, Rocco Servidio, Maria Caria, and Attilio Pedrazzoli. Authoring tutored, adaptive e-courses in a personal learning environment: a dynamic syllabus and dynamic assembly approach. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, pages 236–242, 2012.
- [3] Thomas Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, Oxford.
- [4] Esther Bagno and Bat-Sheva Eylon. From problem solving to a knowledge structure: An example from the domain of electromagnetism. *American Journal of Physics*, 65:726, 1997.
- [5] Ahmad Baylari and Gh A Montazer. Design a personalized e-learning system based on Item Response Theory and Artificial Neural Network approach. *Expert Systems with Applications*, 36(4):8013–8021, 2009.
- [6] Assunta Bonanno, Giacomo Bozzo, Michele Camarca, and Peppino Sapia. An on-line experiment on electromagnetic induction. In *MPTL 14 proceedings*, 2009.
- [7] Assunta Bonanno, Giacomo Bozzo, Michele Camarca, and Peppino Sapia. Using a pc and external media to quantitatively investigate electromagnetic induction. *Physics Education*, 46(4):385, 2011.

- [8] Stefania Bosio, Vilma Capocchiani, Marisa Michelini, and Lorenzo Santi. Computer on-line to explore thermal properties of matter. *Teaching the Science of Condensed Matter and New Materials*, 1996.
- [9] Peter Brusilovsky and Julita Vassileva. Course sequencing techniques for large-scale web-based education. *International Journal of Continuing Engineering Education and Life Long Learning*, 13(1):75–94, 2003.
- [10] Valentina Caputi and Antonio Garrido. Experiences on using intelligent planning for curriculum personalization in Moodle. In *EDULEARN13 Proceedings*, pages 168–176. IATED, 2013.
- [11] María José Casany, Marc Alier, Enric Mayol, Jordi Piguillem, Nikolas Galanis, Francisco Javier García-Peñalvo, and Miguel Angel Conde. Extending Moodle services to mobile devices: the Moodbile project. In *UBICOMM 2012, The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 24–28, 2012.
- [12] Ting-Yi Chang and Yan-Ru Ke. A personalized e-course composition based on a genetic algorithm with forcing legality in an adaptive learning system. *Journal of Network and Computer Applications*, 36(1):533–542, 2013.
- [13] Yi-Chun Chang, Shu-Ming Hsieh, and Jian-Wei Li. Application of the genetic algorithm in customization personalized e-course. In *System Science and Engineering (ICSSE), 2010 International Conference on*, pages 221–226. IEEE, 2010.
- [14] Chih-Ming Chen. Intelligent web-based learning system with personalized learning path guidance. *Computers & Education*, 51(2):787–814, 2008.
- [15] Chih-Ming Chen, Hahn-Ming Lee, and Ya-Hui Chen. Personalized e-learning system using Item Response Theory. *Computers & Education*, 44(3):237–255, 2005.
- [16] Yu-Chien Chen, Ren-Hung Hwang, and Cheng-Yu Wang. Development and evaluation of a web 2.0 annotation system as a learning tool in an e-learning environment. *Computers & Education*, 58(4):1094–1105, 2012.
- [17] Chih-Ping Chu, Yi-Chun Chang, and Cheng-Chang Tsai. Pc2pso: personalized e-course composition based on particle swarm optimization. *Applied Intelligence*, 34(1):141–154, 2011.

- [18] John Clement. Students' preconceptions in introductory mechanics. *American Journal of physics*, 50(1):66–71, 1982.
- [19] Fabiano A Dorça, Luciano V Lima, Márcia A Fernandes, and Carlos R Lopes. Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: an experimental analysis. *Expert Systems with Applications*, 40(6):2092–2101, 2013.
- [20] Martin Dougiamas. Moodle: A virtual learning environment for the rest of us. *TESL-EJ*, 8(2):1–8, 2004.
- [21] Hendrik Drachsler, Hans Hummel, and Rob Koper. Navigation support for learners in informal learning environments. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 303–306. ACM, 2008.
- [22] Reinders Duit. Science education research—an indispensable prerequisite for improving instructional practice. *ESERA Summer School, Braga*, 2006.
- [23] Antonio Garrido, Susana Fernández, Lluvia Morales, Eva Onaindía, Daniel Borrajo, and Luis Castillo. On the automatic compilation of e-learning models to planning. *The Knowledge Engineering Review*, pages 1–16, 2012.
- [24] Antonio Garrido and Eva Onaindía. Assembling Learning Objects for personalized learning. an AI planning perspective. *IEEE Intelligent Systems*, 28(2):64–73, 2013.
- [25] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning: theory & practice*. Access Online via Elsevier.
- [26] Kym Gilhooly. Making e-learning effective. *Computerworld*, 35(29):52–53, 2001.
- [27] David E Goldberg, Bradley Korb, and Kalyanmoy Deb. Messy genetic algorithms: motivation, analysis, and first results. *Complex systems*, 3(5):493–530, 1989.
- [28] Manuel Gomez-Gonzalez and Francisco Jurado. Personalized e-learning using shuffled frog-leaping algorithm. In *Global Engineering Education Conference (EDUCON), 2012 IEEE*, pages 1–7. IEEE, 2012.

- [29] Angappa Gunasekaran, Ronald D McNeil, and Dennis Shaul. E-learning: research and applications. *Industrial and Commercial Training*, 34(2):44–53, 2002.
- [30] Malte Helmert. Concise finite-domain representations for PDDL planning tasks. *Artificial Intelligence*, 173(5):503–535, 2009.
- [31] Claire Hewson. Can online course-based assessment methods be fair and equitable? Relationships between students’ preferences and performance within online and offline assessments. *Journal of Computer Assisted Learning*, 28(5):488–498, 2012.
- [32] Norsham Idris, Norazah Yusof, and Puteh Saad. Adaptive course sequencing for personalization of learning path using neural network. *Int. J. Advance. Soft Comput. Appl*, 1(1):49–61, 2009.
- [33] Hwa-Young Jeong, Cheol-Rim Choi, and Young-Jae Song. Personalized learning course planner with e-learning DSS using user profile. *Expert Systems with Applications*, 39(3):2567–2577, 2012.
- [34] Nava Jyothi, Kaveri Bhan, Uday Mothukuri, Sandesh Jain, and Dhanan-der Jain. A recommender system assisting instructor in building learning path for Personalized Learning System. In *Technology for Education (T4E), 2012 IEEE Fourth International Conference on*, pages 228–230. IEEE, 2012.
- [35] Arturas Kaklauskas, Edmundas K Zavadskas, Mark Seniut, Tomas Stankevic, J Raistenskis, C Simkevičius, T Stankevic, A Matuliauskaite, L Bartkiene, L Zemeckyte, et al. Recommender System to analyze student’s academic performance. *Expert Systems with Applications*, 40(15):6150–6165, 2013.
- [36] Ahmad A Kardan, Hamid Sadeghi, Saeed Shiry Ghidary, and Mohammad Reza Fani Sani. Prediction of student course selection in online higher education institutes using neural network. *Computers & Education*, 65:1–11, 2013.
- [37] Rainer Knauf, Kouhei Takada, Yoshitaka Sakurai, Takashi Kawabe, Set-suo Tsuruta, et al. Personalized and adaptive curriculum optimization based on a Performance Correlation Analysis. In *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, pages 655–660. IEEE, 2012.

- [38] Efstratios Kontopoulos, Dimitris Vrakas, Fotis Kokkoras, Nick Bassiliades, and Ioannis Vlahavas. An ontology-based planning system for e-course generation. *Expert Systems with Applications*, 35(1):398–406, 2008.
- [39] Sheo Kumar, Anil Kumar Gankotiya, and Kamlesh Dutta. A comparative study of Moodle with other e-learning systems. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference*, volume 5, pages 414–418. IEEE, 2011.
- [40] Diana Laurillard, Patricia Charlton, Brock Craft, Dionisios Dimakopoulos, Dejan Ljubojevic, George Magoulas, Elizabeth Masterman, Roser Pujadas, Edgar A Whitley, and Kim Whittlestone. A constructionist learning environment for teachers to model learning designs. *Journal of computer assisted learning*, 29(1):15–30, 2013.
- [41] Jian-Wei Li, Yi-Chun Chang, Chih-Ping Chu, and Cheng-Chang Tsai. A self-adjusting e-course generation process for personalized learning. *Expert Systems with Applications*, 39(3):3223–3232, 2012.
- [42] Shu-Sheng Liaw, Hsiu-Mei Huang, and Gwo-Dong Chen. Surveying instructor and learner attitudes toward e-learning. *Computers & Education*, 49(4):1066–1080, 2007.
- [43] Chun Fu Lin, Yu-chu Yeh, Yu Hsin Hung, and Ray I Chang. Data mining for providing a personalized learning path in creativity: an application of decision trees. *Computers & Education*, 68:199–210, 2013.
- [44] Jia-Jiunn Lo, Ya-Chen Chan, and Shiou-Wen Yeh. Designing an adaptive web-based learning system based on students’ cognitive styles identified online. *Computers & Education*, 58(1):209–222, 2012.
- [45] David P Maloney, Thomas L O’Kuma, Curtis J Hieggelke, and Alan Van Heuvelen. Surveying students’ conceptual knowledge of electricity and magnetism. *American Journal of Physics*, 69:S12, 2001.
- [46] Teresa Martín-Blas and Ana Serrano-Fernández. The role of new technologies in the learning process: Moodle as a teaching tool in Physics. *Computers & Education*, 52(1):35–44, 2009.
- [47] Eric Mazur and Robert C Hilborn. Peer instruction: a user’s manual. *Physics Today*, 50(4):68–69, 1997.

- [48] Drew Mc Dermott, Malik Ghallab, Adele Howe, Craig A Knoblock, Ashwin Ram, Manuela M Veloso, Daniel Weld, and David Wilkins. PDDL: The planning domain definition language (technical report, C VC TR-98-003/DCS TR-1165). *New Haven, CT: Yale Center for Communicational Vision and Control*, 1998.
- [49] Lillian C McDermott. Guest comment: How we teach and how students learn—a mismatch? *American Journal of Physics*, 61:295, 1993.
- [50] Lillian C McDermott, Paula RL Heron, Peter S Shaffer, and MacKenzie R Stetzer. Improving the preparation of k-12 teachers through physics education research. *American Journal of Physics*, 74:763, 2006.
- [51] Lillian C McDermott and Peter S Shaffer. Research as a guide for curriculum development: an example from introductory electricity. part I: investigation of student understanding. *American journal of physics*, 60:994–994, 1992.
- [52] Lillian Christie McDermott. Millikan lecture 1990: what we teach and what is learned—closing the gap. *American Journal of Physics*, 59:301–315, 1991.
- [53] Mario Muñoz Organero, Carlos Delgado Kloos, and Pedro Munoz Merino. Personalized service-oriented e-learning environments. *Internet Computing, IEEE*, 14(2):62–67, 2010.
- [54] Kyparisia A Papanikolaou, Maria Grigoriadou, George D Magoulas, and Harry Kornilakis. Towards new forms of knowledge communication: the adaptive dimension of a web-based learning environment. *Computers & Education*, 39(4):333–360, 2002.
- [55] Juan José Sánchez Peña, Jesús Cáceres Tello, and Santiago Pérez de la Cámara. A comparative study between free e-learning platforms. In *IADIS international conference e-learning*, volume 1. IEEE, 2011.
- [56] Cristóbal Romero, Sebastián Ventura, and Enrique García. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368–384, 2008.
- [57] Salvador Ros, Roberto Hernández, Antonio Robles-Gómez, Agustín C Caminero, Llanos Tobarra, and Elio Sancristobal Ruiz. Open service-oriented platforms for Personal Learning Environments. *Internet Computing, IEEE*, 17(4):26–31, 2013.

- [58] Stuart Russell and Peter Norvig. *Artificial Intelligence: a Modern Approach*. Pearson Education, 2010.
- [59] AP Sarath Chandar, SG Dheeban, V Deepak, and Susan Elias. Personalized e-course composition approach using digital pheromones in improved particle swarm optimization. In *Natural Computation (ICNC), 2010 Sixth International Conference on*, volume 5, pages 2677–2681. IEEE, 2010.
- [60] Souvik Sengupta, Bulbul Mukherjee, Sudipta Bhattacharya, and Ranajan Dasgupta. OLAP based Scaffolding to support Personalized Synchronous e-learning. *International Journal of Managing Information Technology*, 4(3):73, 2012.
- [61] Charles Severance, Joseph Hardin, and Anthony Whyte. The coming functionality mash-up in Personal Learning Environments. *Interactive Learning Environments*, 16(1):47–62, 2008.
- [62] Richa Sharma, Hema Banati, and Punam Bedi. Adaptive content sequencing for e-learning courses using ant colony optimization. In *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011*, pages 579–590. Springer, 2012.
- [63] Mike Sharples, Josie Taylor, and Giasemi Vavoula. Towards a theory of mobile learning. *Proceedings of mLearn 2005*, 1(1):1–9, 2005.
- [64] Xiao-hong Tan, Rui-min Shen, and Yan Wang. Personalized course generation and evolution based on genetic algorithms. *Journal of Zhejiang University SCIENCE C*, 13(12):909–917, 2012.
- [65] Laurence Viennot. Analyzing students’ reasoning: tendencies in interpretation. *American Journal of Physics*, 53(5):432–436, 1985.
- [66] Stella Vosniadou. *International handbook of research on conceptual change*. Routledge, 2008.
- [67] William R Watson and Sunnie Lee Watson. What are Learning Management Systems, what are they not, and what should they become? *TechTrends*, 51(2):29, 2007.
- [68] Michalis Xenos. Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks. *Computers & Education*, 43(4):345–359, 2004.

- [69] Sugi Yuuichi, Kita Toshihiro, Yasunami Seisuke, and Nakano Hiroshi. Web-based Rapid Authoring tool for LMS quiz creation. In *Information Technology Based Higher Education and Training, 2006. ITHET'06. 7th International Conference on*, pages 617–620. IEEE, 2006.
- [70] Mário Zenha-Rela and Rafael Carvalho. Work in progress: Self evaluation through monitored peer review using the moodle platform. In *Frontiers in Education Conference, 36th Annual*, pages 26–27. IEEE, 2006.