

**GENERACIÓN AUTOMÁTICA DE DISEÑOS DE APRENDIZAJE:
DIFERENTES ENFOQUES DE PLANIFICACIÓN**

por

Lluvia Carolina Morales Reynaga

Una Memoria de Tesis Doctoral Presentada como
Requerimiento para optar al Grado de
Doctor en Ciencias de la Computación y Tecnología Informática

Directores:

Luis Castillo Vidal, Chief Technology Officer, IActive Intelligent Solutions.

Juan Fernández Olivares, Profesor Titular, Universidad de Granada.

UNIVERSIDAD DE GRANADA

Marzo 2011

Editor: Editorial de la Universidad de Granada
Autor: Lluvia Carolina Morales Reynaga
D.L.: GR 3137-2011
ISBN: 978-84-694-3556-4

**GENERACIÓN AUTOMÁTICA DE DISEÑOS DE APRENDIZAJE:
DIFERENTES ENFOQUES DE PLANIFICACIÓN**

por

Lluvia Carolina Morales Reynaga

ha sido aprobada

Marzo 2011

APROBADA:

, Presidente

Comité Supervisor

ACEPTADA:

Luis Castillo Vidal, Chief Technology Officer,
IActive Intelligent Solutions.

Juan Fernández Olivares, Profesor Titular,
Universidad de Granada.

ÍNDICE GENERAL

CAPITULO 1. Introducción	1
1. Objetivos	12
2. Descripción por Capítulos	13
CAPITULO 2. Preliminares	16
1. Un Poco de Historia	16
2. Plataformas de Enseñanza por Ordenador	18
2.1. Sistemas Tutores Inteligentes	18
2.2. Sistemas Gestores de Aprendizaje	23
2.2.1. Sistemas Gestores de Aprendizaje Comerciales	24
2.2.2. Sistemas Gestores de Aprendizaje Libres	25
3. E-Learning: Herramientas y Conceptos de Apoyo a la Generación de Diseños Instruccionales	29

3.1.	E-Learning	30
3.2.	Objetos de Aprendizaje	33
3.3.	Estándares	34
3.3.1.	ANSI/IEEE Standards	34
3.3.2.	IMS Standards	34
3.4.	Diseños de Aprendizaje	36
3.5.	Perfil del Estudiante	37
3.6.	SCORM	37
4.	Planificación y Scheduling Inteligentes	38
4.1.	Paradigmas de Planificación	40
4.1.1.	Basado en Estados	40
4.1.2.	Jerárquico (HTN)	44
4.1.3.	Comparativa y Selección de Planificadores	46
4.2.	Técnicas de Planificación para Entornos Dinámicos	49
4.2.1.	Re-Planificación	50
4.2.2.	Planificación Contingente o Condicional	51
4.2.3.	Planificación Continua	51
CAPITULO 3. Procesos de Ingeniería del Conocimiento		54
1.	Los Objetivos y la Estructura del Curso	56
1.1.	Metadatos Relacionales	59
1.1.1.	De Orden	60

1.1.2.	De Jerarquía	61
1.2.	Metadatos Tipo Atributos	61
1.2.1.	Generales	62
1.2.2.	Técnicos	63
1.2.3.	Educativos	64
2.	El Perfil del Estudiante	65
3.	Los Cuestionarios	70
4.	Los Diseños Instruccionales	71
CAPITULO 4. Generación Automática de los Ficheros de Dominio y Problema de Planificación		76
1.	Preprocesamiento y Preliminares	80
2.	Dominios y Problemas para Planificadores Jerárquicos	86
2.1.	Dominio Jerárquico	87
2.1.1.	Definición de Acciones Primitivas	89
2.1.2.	Definición de Tareas Compuestas	90
2.2.	Problema Jerárquico	97
3.	Dominios y Problemas de Planificación para Planificadores Basados en Estados	99
3.1.	Dominio Basado en Estados	100
3.1.1.	Estructuras Secuenciales de Objetos de Aprendizaje Ordenados	100
3.1.2.	Acciones Primitivas Durativas	103
3.1.3.	Acciones Primitivas NO Durativas	106
3.2.	Problema Basado en Estados	109

4.	Planes Generados (Diseños Instruccionales)	110
4.1.	Jerárquico	112
4.2.	Basado en Estados	114
5.	Comparativa	116
5.1.	Procesamiento Estructural	116
5.2.	Representación de las Restricciones Temporales	117
5.3.	Análisis de Complejidad	118
5.4.	Resumen Comparativo	119
CAPITULO 5. Planificación de Diseños Instruccionales en Entornos Dinámicos		122
1.	El Problema de los Entornos Educativos Dinámicos	123
2.	Pre-Planificación Condicional para la Generación de un IMS-LD nivel B	126
2.1.	El manifiesto IMS-LD	127
2.2.	Modificación a los Algoritmos de Generación Automática de Dominios y Problemas de Planificación	129
2.3.	El Proceso de Pre-Planificación Condicional	131
3.	Planificación Continua y Re-planificación para la Interacción con un LMS	132
3.1.	Planificación Continua en Entornos e-Learning Dinámicos	133
3.2.	Re-planificación en Entornos e-Learning Dinámicos	134
CAPITULO 6. Integración con el LMS		136
1.	Características Deseables en los LMS's	138
2.	Arquitectura del Sistema	139

2.1.	El Cliente de Planificación	141
2.2.	El Servidor de Planificación	145
2.3.	El Proceso de Comunicación	147
3.	La Vista Final del LD en Moodle	151
CAPITULO 7. Experimentación en Dominios Reales		154
1.	Casos de Uso	155
1.1.	Grupo Tadel: Planificación	156
1.2.	Seminario de Inteligencia Artificial y Educación: Planificación Conti- nua y Re-planificación	157
2.	Validación a Posteriori de los Diseños Instruccionales	159
CAPITULO 8. Conclusiones y Trabajo Futuro		163
1.	Conclusiones	164
1.1.	Publicaciones Obtenidas	167
2.	Trabajo Futuro	169
2.1.	Estrategias de What-iffing y/o Iniciativa Mixta	169
2.2.	Dominios e-Learning Colaborativos	170
2.3.	Sistema de Identificación de Precondiciones Inconsistentes	170
2.4.	Dominios Diferentes de e-Learning (Otras Aplicaciones)	171
2.5.	Generalización las Reglas para los Estilos de Aprendizaje y las Pre- condiciones Derivadas del Perfil de los Alumnos	171
2.6.	Búsquedas sobre Repositorios Distribuídos de Objetos de Aprendizaje	172

CAPITULO A. Manual del Usuario para Integrar y Utilizar el Cliente de Planificación en Moodle	1
1. Primeros Pasos	1
1.1. La generación y etiquetado de un objeto SCORM	1
1.2. La Instalación y Configuración del add-on dentro del Módulo SCORM de Moodle	6
1.3. Uso del add-on de Adaptación	10
2. Preguntas Frecuentes	13
3. Notas para el Desarrollo	13
CAPITULO B. Aplicación y Evaluación del cuestionario CHAEA	14
Bibliografía	26

CAPÍTULO 1

Introducción

Según el Diccionario de la Real Academia Española la *Enseñanza* es el sistema y método de dar instrucción. El sistema educativo tradicional desarrolla sus métodos y sistemas de instrucción dentro de un *Ambiente de Aprendizaje* que tiene como escenario principal el aula o algún otro ambiente físico propicio para desarrollar los métodos de enseñanza y tiene a un profesor/tutor como facilitador de las condiciones para que dicha enseñanza sea efectiva.

El sistema educativo tradicional se basa, también, en una serie de *Objetivos de Aprendizaje* que describen lo que el estudiante va a ser capaz de aprender, entender o hacer como resultado del desarrollo de una lección o tema que involucra un conjunto de *Actividades de Aprendizaje* por medio de las cuales se obtienen nuevos conocimientos, habilidades o aptitudes.

Los expertos en Ciencias de la Educación son conscientes de que las actividades de aprendizaje y el medio en el que se imparten son la clave para que el proceso de enseñanza se desarrolle de una manera mas efectiva. Tanto es así que las mayores inversiones monetarias de los ministerios de educación en el mundo suelen asignarse, aún, a obras de infraestruc-

tura que permitan construir y acondicionar aulas adecuadamente, así como a la formación y actualización del personal docente, pues son ellos los encargados de convertir de forma “innata” las actividades de aprendizaje en un diseño instruccional.

Un *Diseño Instruccional* integra un conjunto de actividades de aprendizaje con el ambiente de aprendizaje y permite estructurar dichas actividades en base a teorías pedagógicas que les permitan mejorar su eficiencia, eficacia y atractivo dependiendo de las características específicas de cada grupo de estudiantes o cada estudiante particular (en caso de tener grupos pequeños).

El sistema educativo tradicional ha ido evolucionando constantemente. El número de estudiantes ha crecido exponencialmente en todos los niveles haciendo insuficiente el número de aulas. Las empresas ya no se conforman con la preparación con la que egresan los titulados superiores o técnicos sino que requieren de ellos una capacitación y actualización continua que no coincide con los ciclos y horarios de las escuelas tradicionales. Los gobiernos intentan llevar la educación de calidad en todos los niveles a cada rincón de su territorio para que sus ciudadanos no tengan la necesidad de emigrar a las ciudades para recibirla y, sobre todo, las nuevas generaciones ya no conciben las actividades de aprendizaje sin el uso de la tecnología.

Científicos, gobiernos y educadores, conscientes de estas situaciones aprovecharon las facilidades y el uso promisorio que proyectaba la Web y, a principios de la década de los noventa, surgió el concepto y las primeras herramientas de educación en línea.

Así como con el ordenador los científicos de la computación, y más específicamente de la inteligencia artificial, intentamos emular las capacidades del cerebro humano, los creadores de estas herramientas en línea han intentado emular, no solo las aulas a través de *Ambientes Virtuales de Aprendizaje* (VLE del inglés Virtual Learning Environments) cada vez mas intuitivos, sino también las actividades que se llevan a cabo en las mismas. Por ejemplo, ahora los profesores de cursos en línea imparten su clase a través de videoconferencias, los enunciados de los problemas y ejercicios ahora se cuelgan la red y el alumno se los entrega al profesor través del VLE, los foros de discusión ahora se llevan a cabo textualmente y de manera asíncrona por medio de la herramienta de foro, si se deseara hacerlos de forma síncrona se cuenta con el chat, la aplicación de exámenes con diferentes estilos de preguntas

y límites de tiempo también ya es posible vía web, etc.

Han pasado casi 20 años desde que la primera herramienta de enseñanza en línea, para la educación a distancia, salió a la luz. El uso de la Web se ha extendido sobremanera, sobre todo en los países desarrollados, y eso ha facilitado que cada vez mas y mas personas se familiaricen y adapten fácilmente a las nuevas y complejas herramientas de enseñanza actualmente conocidas como *herramientas de e-learning* o e-aprendizaje. Grandes empresas y centros de gobierno del ramas aeroespacial, petroquímica, eléctrica e incluso deportiva, como por ejemplo la NASA¹ en Estados Unidos, la ESA² en Europa y la National Geographic Foundation³ en Reino Unido, entre otras, nos solo divulgan su actividad, se comunican y planifican proyectos a través de estas herramientas, sino que también las utilizan para mejorar el rendimiento de su personal entrenándolos y capacitándolos con ellas.

Hasta ahora, y durante los años de integración de estas nuevas tecnologías en el proceso de enseñanza-aprendizaje, lo que mas ha costado es formar y actualizar a los profesores para que se adapten al nuevo paradigma de enseñanza que se ha abierto camino a través de la enseñanza en línea. En muchos casos se han mostrado reticentes por que les resulta complicado percibir los comportamientos, respuestas y actitudes de los estudiantes, debido a las limitaciones inherentes a las interfaces de entrada y salida del ordenador, y generar con ello un diseño instruccional eficaz. Eso, aunado al gran número de estudiantes que hoy en día pueden matricularse en un curso en línea (se estima por lo menos una centena o mas, a diferencia de la capacidad de aproximadamente 50 estudiantes por aula) puede resultar, no solo aterrador sino también, imposible de procesar para uno o varios profesores que intenten general diseños instruccionales adaptados a todos y cada uno de los estudiantes del curso.

Cabe resaltar que hasta hace dos años la suma de todos los estudiantes matriculados en las tres principales Universidades en línea Europeas; las de Gran Bretaña, Países Bajos y España; llegaba a ser de poco mas de 350 mil estudiantes. Y tal y como ocurrió, y sigue ocurriendo, con las enseñanzas tradicionales, esta cantidad se duplicará en al menos 10 años.

¹National Aeronautics and Space Administration: www.nasa.gov

²European Space Agency: www.esa.int

³National Geographic Foundation: www.nationalgeographic.com/foundation/

Así pues, tiene sentido plantearse la siguiente pregunta:

¿Será posible aprovechar las capacidades de procesamiento del ordenador para manejar la gran cantidad de información asociada a todos y cada uno de los estudiantes de un curso, así como a las actividades de aprendizaje?

Y, por otro lado, también lo siguiente:

¿Existe la tecnología para crear una herramienta que apoye o emule las capacidades básicas de un profesor y genere diseños instruccionales adaptados las necesidades de cada estudiante?

Es preciso tener en cuenta que generalmente la información de los estudiantes se encuentra distribuida y desorganizada en distintos sitios web. Y algo similar pasa con infinidad de actividades de aprendizaje que se encuentran distribuidas en distintos servidores alrededor del mundo.

Como era de esperarse, hace poco mas de diez años y percatándose de esta problemática surgieron organizaciones como por ejemplo ADL⁴, AICC⁵ o IMS⁶ que se propusieron estandarizar los métodos de educación en línea y que han diseñado guías o estándares detallados para describir gran parte de los componentes de un ambiente de aprendizaje en línea, entre ellos:

- La información del estudiante, a la cual llamaron *Perfil del Estudiante* y que se almacena en recursos digitales llamados portafolios;
- La información de las actividades o recursos de aprendizaje, a las cuales llamaron *Objetos de Aprendizaje* que será el término que utilizaremos a partir de este momento y a lo largo de toda esta memoria para referirnos a ellas;

⁴Advanced Distributed Learning: www.adlnet.gov

⁵Aviation Industry Computer Based Training Committee: www.aicc.org

⁶IMS Global Learning Consortium: www.imsglobal.org

- Las evaluaciones del conocimiento del estudiante, que aunque podrían considerarse objetos de aprendizaje, se hizo especial hincapié en un estándar para modelarlas y recabar información acerca de ellas y les llamaron *Cuestionarios*;
- Y, sobre todo, estas organizaciones han dedicado un enorme esfuerzo al intentar generalizar toda la información para modelar diseños instruccionales y de esta manera guiar, mas no por ello facilitar, el trabajo del profesor al generarlos.

Estas organizaciones llamaron *Diseños de Aprendizaje* a la especificación de los diseños instruccionales estándar, de manera que podríamos considerarlos sinónimos, solo que este último se utilizará a lo largo de la memoria solo para referirnos a una especificación estándar y el de diseño instruccional para hablar del término educativo general.

Los principales VLE's que se utilizan para impartir educación a distancia [Inc., 2010a, dotLRN Consortium, 2009, Caroline, 2010, Dougiamas,] han integrado estos estándares para representar gran parte de sus componentes, lo cual ha dado lugar a un nuevo ambiente lleno de conocimiento acerca de los objetos de aprendizaje, las capacidades del estudiante, además de su contexto e interrelación. Y existe una comunidad en constante aumento que, al percatarse del gran potencial comercial de los objetos de aprendizaje ha dado lugar al surgimiento de empresas y centros de desarrollo que no solo generan grandes cantidades de objetos de aprendizaje que venden y/o intercambian a distintas instituciones educativas alrededor de mundo, sino que también mantienen y promueven su integración y evolución en estos VLE's que hoy en día son también conocidos como *plataformas de e-learning*.

Sin embargo, no todos los estándares propuestos han sido adoptados fácilmente por la comunidad académica dedicada a impartir cursos a distancia. Para casi todos ellos existen herramientas, externas o embebidas dentro de las plataformas de aprendizaje que permiten recabar información del perfil del estudiante, aplicar cuestionarios o enriquecer la información sobre los objetos de aprendizaje. La especificación de diseños de aprendizaje, por el contrario, **no cuenta hasta el momento con ninguna herramienta que realmente facilite su adopción y la haga mas intuitiva** para expertos de todas las áreas del conocimiento.

Existen trabajos como el de [Luckin et al., 2006] que sostienen que: “el diseño de metodologías educativas centradas en el alumno tiene en cuenta la heterogeneidad de los estudiantes, sus necesidades y rendimiento, la colaboración entre ellos para reforzar el aprendizaje y estudios previos que ayudan a las plataformas actuales de e-learning a personalizar los diseños de aprendizaje, de manera que el estudiante pueda aprovechar de manera óptima los contenidos de un curso dado a través de una experiencia de aprendizaje adaptada a sus necesidades específicas.”

Para que lo comentado en el párrafo anterior sea llevado a la práctica de manera efectiva, es preciso que un individuo pueda tomar en cuenta toda la información disponible del curso y los estudiantes y sea capaz de procesarla para que, siguiendo metodologías educativas centradas en el alumno, pueda adaptar los contenidos del curso a cada uno de los participantes del mismo y posteriormente plasmar dicho conocimiento dentro de una especificación que permita a cualquier plataforma de e-learning comprender y desplegar todos y cada uno de los contenidos (objetos de aprendizaje) correspondientes.

Por lo tanto, las cualidades deseables que se esperan de un profesor para que sea capaz de crear diseños instruccionales adaptados a cada uno de los estudiantes de un curso, son por lo menos las siguientes:

1. La posibilidad de reconocer las relaciones que existen entre los objetos de aprendizaje de un curso y reconstruir, en base a ellas, una secuencia de actividades coherente.
2. La capacidad de distinguir las restricciones técnicas y/o formativas impuestas a cada uno de los objetos de aprendizaje y representar dichas restricciones para su posterior procesamiento.
3. La capacidad de permitir la inclusión, dentro de un itinerario formativo, de objetos de aprendizaje con características específicas que refuercen el conocimiento de cada estudiante. La decisión de incluirlos, o no, podría depender de la disponibilidad temporal o capacidad académica del estudiante.
4. La aptitud de manejar redundancia en los nombres de los objetos de aprendizaje. Pues pueden existir objetos de aprendizaje que cumplan con los mismos objetivos y se

llamen igual, pero tengan diferentes restricciones; por ejemplo, si se desea introducir al concepto de *suma* por medio de un ejemplo, esto se puede hacer en inglés y por medio de una imagen con una serie de pasos enumerados o en español y por medio de una animación multimedia, lo cual supone tener dos objetos con el mismo nombre, el primero con una restricción en el nivel de inglés del estudiante y en segundo que restringe el equipo en el cual podrá reproducirse el objeto a uno que tenga instalado “flash”. También hay objetos que tienen el mismo nombre pero objetivos distintos, por ejemplo aquellos objetos cuyo nombre es *introducción* pero introducen a diferentes objetivos del curso, por ejemplo, uno a *suma* y otro a *multiplicación*.

5. La posibilidad de manejar duraciones sobre las actividades del curso ya sea definidas por los alumnos o sobre el profesor, y en base a ellas decidir la longitud de un curso (número de objetos de aprendizaje) para cada estudiante.
6. La capacidad de manejar la incertidumbre que conlleva la aplicación de evaluaciones intermedias, de manera que sea posible adaptar la secuencia de aprendizaje posterior a una evaluación, a los posibles resultados de la misma.
7. La posibilidad de detectar cambios en el perfil del estudiante y/o los objetos de aprendizaje, y re-adaptar la secuencia de aprendizaje que se vea afectada por dichos cambios.

Como se puede apreciar, muchas de estas cualidades surgen de forma innata en un profesor entrenado y con experiencia en la impartición de cursos a distancia. Sin embargo, en un ambiente lleno de información es muy difícil implementar las metodologías centradas en el alumno que permitan hacer mas efectivo el proceso de aprendizaje y mucho menos traducirlas a un lenguaje de especificaciones estándar tan nuevo y extenso con el de los diseños de aprendizaje.

Resulta impensable que con un número ingente de objetos de aprendizaje con distintas cualidades, pero que en ocasiones sirven para enseñar el mismo objetivo de aprendizaje; con una gran cantidad de alumnos de distintas procedencias (por tanto, distintos usos horarios), edades, historial académico, habilidades, intereses, capacidades intelectuales, disponibilidad horaria, etc; y con un ambiente de aprendizaje en constante cambio dado sobre todo por las

aportaciones y progreso de los estudiantes (manifestadas particularmente a través del cambio en su perfil y en sus conocimientos iniciales evaluados por medio de los cuestionarios), sea posible para una persona adaptar un diseño de aprendizaje para cada uno de ellos.

Es por eso que el uso de técnicas de inteligencia artificial que “imiten” la capacidad de un profesor para generar diseños instruccionales que consideren las metodologías centradas en el usuario, se está volviendo una necesidad. Pues ya no solo basta con emular el ambiente y los recursos de un ambiente de aprendizaje tradicional, sino también al profesor que difícilmente puede enfrentarse con una cantidad y diversidad de los estudiantes poco común en la enseñanza tradicional.

Las técnicas de inteligencia artificial (IA) que pueden satisfacer de una forma más robusta las necesidades de generación de diseños instruccionales son las que permiten la síntesis de soluciones, sobre todo aquellas que se utilizan en la rama de la IA denominada Planificación y Scheduling (AIP&S). Esta rama se basa fundamentalmente en:

- Recabar una gran cantidad de información acerca del mundo circundante para definir un *estado inicial* del entorno del problema con el cual habrá que enfrentarse.
- Definir un conjunto de reglas de actuación en base a determinadas precondiciones que una vez que se cumplen permiten llevar a cabo una o varias acciones que generan efectos sobre el estado del mundo, es decir, definir el *dominio* por medio del cual será posible cambiar el estado inicial de un problema para llegar hasta un estado que lo solucione.
- Establecer una *meta* que suele representarse como una acción o conjunto de acciones que se quieran llevar a cabo, además de una serie de condiciones que debe cumplir el mundo circundante una vez que se hayan ejecutado dichas acciones meta.

Con toda esa información, descrita en alguna de las variantes y versiones del lenguaje PDDL [Long and Fox, 2003] y organizada en un par de ficheros (fichero de dominio, que describe las reglas de actuación y fichero de problema, que describe el estado del mundo circundante y las metas), el o los algoritmos de búsqueda que conforman el motor de planificación se encargan de buscar la secuencia de acciones que lleve a alcanzar la meta definida.

en el fichero del problema. Siendo, por lo tanto, esta secuencia de acciones a la cual se le denomina comúnmente *plan*, la salida proporcionada por un proceso de planificación.

Gracias a la centralización y manejo estructurado de la información de cada uno de los componentes del ambiente de aprendizaje, se puede apreciar como el uso de las técnicas de AIP&S inteligentes se ha convertido en una opción clara para el diseño de metodologías de educación centradas en el alumno, no solo para cursos de posgrado, sino también para la capacitación de profesionales y la formación continua.

En el contexto del e-learning el *estado inicial* del problema se extrae a partir de la información del perfil del estudiante, los objetivos o *metas* del mismo se obtienen a partir de los objetivos de aprendizaje de un curso y el *dominio* se construye, para el propósito específico de esta memoria, utilizando un repositorio de objetos de aprendizaje en el cual cada objeto se traduce en una acción cuyas precondiciones y efectos se obtienen de la información expresada en sus metadatos o meta-información representada por medio de etiquetas.

En base a estos tres componentes, el resultado esperado de un planificador para un conjunto de perfiles de estudiantes, es un conjunto de secuencias ordenadas de acciones tales que permiten alcanzar los objetivos de aprendizaje para todos y cada uno de los estudiantes del curso. Cada secuencia de acciones es conocida como un *plan*, que posteriormente puede traducirse a una secuencia de actividades de aprendizaje en todas sus variantes de acuerdo al diseño instruccional específico para cada estudiante.

La necesidad de adaptar diseños de aprendizaje no es un tópico nuevo, trabajos recientes [Baldiris et al., 2008, Castillo et al., 2010, Garrido and Onaindía, 2010, Morales et al., 2009, Ullrich and Melis, 2009] prueban el enorme potencial de las de técnicas de AIP&S para resolver el problema de generación de diseños instruccionales en cursos a distancia (mas no de representación de diseños de aprendizaje basados en especificaciones estándar de alto nivel), ya que permiten explorar el conjunto de objetivos de aprendizaje, la enorme oferta de objetos de aprendizaje y las conocidas necesidades y limitaciones de los alumnos, de manera que sea posible crear un diseño de aprendizaje o plan adaptado a cada uno de ellos, facilitando así el trabajo de los profesores.

Sin embargo, para poder incorporar este enorme volumen de conocimiento en un dominio de planificación se requiere de un considerable esfuerzo de ingeniería de conocimiento para tener éxito, sobre todo en la etapa de representación de las actividades de aprendizaje que se encuentran en el repositorio de objetos de aprendizaje (lo que se denominaría el archivo de dominio de planificación) y para la representación del perfil del estudiante y sus objetivos (lo cual es llamado comúnmente archivo de problema de planificación) que posteriormente suelen ser interpretados por el planificador para obtener un plan.

Por lo general este esfuerzo de ingeniería de conocimiento lo llevan a cabo personas con gran habilidad en el dominio de lenguajes de planificación tales como PDDL [Long and Fox, 2003] (el lenguaje estándar usado en la planificación) o alguna de sus variantes, en este caso HPDL [Pérez, 2007], una extensión jerárquica desarrollada por el grupo de investigación de Sistemas Inteligentes de la Universidad de Granada⁷. Estos lenguajes son el esquema de representación de conocimiento que se usará a lo largo de toda esta memoria; el primero para representar el conocimiento utilizado por los planificadores basados en estados y el segundo por los planificadores jerárquicos que se describirán detalladamente en el siguiente capítulo.

Lo anterior puede ser una desventaja para los profesores que no tienen la habilidad para introducir las tecnologías de AIP&S en las plataformas de e-learning que se encuentren utilizando dado que tampoco para esto existe una herramienta que lo haga de forma automática. Ésta es la razón por la cual resultaría muy eficaz una herramienta que permita a los usuarios finales (instructores y estudiantes) que han adoptado los estándares del e-learning, no solo diseñar un diseño de aprendizaje utilizando las técnicas de AIP&S a un costo muy bajo comparado con el esfuerzo que se realiza actualmente, sino también recibirlo de forma oportuna y adaptada inclusive a los cambios constantes a los que están sujetas todas las variables del perfil de un usuario.

Dada la experiencia del grupo de investigación del grupo de Sistemas Inteligentes de la Universidad de Granada en técnicas de planificación y scheduling inteligentes (AIP&S), se ha decidido utilizar estas técnicas para resolver un problema que tiene todas las características

⁷Grupo de Sistemas Inteligentes: <http://isg.ugr.es/>

de un problema de planificación y scheduling, y llevar a cabo un estudio comparativo a lo largo de esta memoria el cual que tiene en cuenta los diversos trabajos de investigación que tratan de abordar el problema de la secuenciación automática y adaptada de contenidos para cursos en línea, en su mayoría implementados sobre plataformas de e-learning denominadas Sistemas Tutoriales Inteligentes (ITS's por sus siglas en inglés) y utilizando técnicas de AIP&S.

Por lo tanto, para este trabajo se parte de la base de que la información de los estudiantes y de los objetos de aprendizaje del curso ya se encuentra disponible y documentada en base a un conjunto de estándares dentro de la plataforma de e-learning (situación mas común actualmente). Teniendo dicha información como base, habría que llevar a cabo un proceso de extracción y organización del conocimiento para que posteriormente las técnicas de inteligencia artificial, en particular las de planificación y scheduling inteligentes, generen distintos diseños instruccionales para cada perfil de usuario y que dichos diseños se vayan adaptando a distintos aspectos dinámicos de un entorno de aprendizaje en línea, en particular los cambios en el perfil del estudiante y las variables de incertidumbre asociadas a los resultados de evaluaciones intermedias del curso.

Los aspectos que, aunque son de suma importancia durante la preparación y desarrollo de un curso en línea, quedan fuera del ámbito de esta memoria, son los siguientes:

- La búsqueda e implementación de técnicas de inteligencia artificial que faciliten el proceso de búsqueda y recuperación de objetos de aprendizaje, con características de adaptación predefinidas, de entre la infinidad de repositorios existentes alrededor del mundo.
- Los métodos de recuperación y centralización de la información de los estudiantes de un curso. Información que normalmente se encuentra distribuida en diferentes aplicaciones de, inclusive, diferentes instituciones.
- El diseño y presentación de los objetos de aprendizaje para facilitar las tareas de índole colaborativo o de navegabilidad.
- Las técnicas de inteligencia artificial que resuelvan problemas de sincronización de

planes adaptados de diversos estudiantes para fomentar el aprendizaje colaborativo.

Es importante mencionar que, tal y como lo se analizará mas a detalle en capítulos posteriores, ninguna otra propuesta anterior que haya utilizado estándares para la generación de diseños instruccionales, ha generado especificaciones de los mismos basadas en el estándar de diseños de aprendizaje y que, además, resuelva cuestiones de incertidumbre y dinamismo de un curso en línea.

1. Objetivos

El principal objetivo de este trabajo es generar automáticamente diseños de aprendizaje adaptados a las necesidades principales y específicas de cada estudiante dentro de un curso en línea y que, gracias a dicha automatización, el tutor se evite el engorroso proceso de diseñarlo a mano o tener que solicitar la ayuda de un experto en AIP&S cada vez que sea necesario generar otro curso o modificar uno ya adaptado.

Para llevar a cabo esta automatización será preciso cumplir con los siguientes sub-objetivos:

1. Definir las pautas mínimas para que el tutor pueda etiquetar correctamente los atributos y relaciones existentes entre cada uno de los objetos de aprendizaje que componen un curso e-learning. Así como también la información mínima que debe contener el perfil de un estudiante (capítulo 3).
2. Diseñar e implementar un algoritmo completo y eficiente que realice un proceso de ingeniería del conocimiento sobre el etiquetado completo y correcto de los metadatos de los objetos de aprendizaje, y que a través de dicho proceso sea posible generar de forma completamente automática un dominio de planificación capaz de guiar al planificador en el proceso de generación del diseño de aprendizaje adaptado (capítulo 4 secciones 2.1 y 3.1).
3. Diseñar e implementar un algoritmo completo y eficiente que realice un proceso de ingeniería del conocimiento sobre la información del perfil de cada uno de los estu-

diantes del curso, y que a través de dicho proceso sea posible generar un problema de planificación capaz de guiar al planificador en el proceso de generación del diseño de aprendizaje adaptado (capítulo 4 secciones 2.2 y 3.2).

4. Complementar el proceso de planificación con otros métodos de planificación que permitan manejar los cambios continuos del perfil del estudiante y la incertidumbre con respecto al resultado de la aplicación de evaluaciones intermedias. Es decir, varios aspectos dinámicos y de incertidumbre inherentes dentro del proceso de impartición de un curso en línea (capítulo 5).
5. Integrar en una plataforma de e-learning las funciones de planificación mencionadas. Es decir, que una vez generado un plan como resultado del proceso de planificación, se genere de forma automática el documento de especificación del diseño de aprendizaje adaptado y sea posible visualizarlo dentro de la plataforma e-learning mediante el despliegue secuencial de los objetos de aprendizaje que lo conforman, para posteriormente llevar a cabo pruebas de satisfacción y coherencia de dichas secuencias (capítulo 6).

2. Descripción por Capítulos

Para poder describir el desarrollo de la presente investigación y los objetivos que se han alcanzado gracias a la aplicación de la misma, esta memoria se ha estructurado de la siguiente manera.

En el capítulo 1 se ha llevado a cabo un resumen y justificación general de nuestro proyecto, además de definir los objetivos generales y específicos de la investigación que se abordará en esta memoria.

El capítulo 2, *preliminares*, permitirá concluir con la justificación general de este trabajo a partir de la especificación de los planificadores y métodos de planificación que serán utilizados a lo largo de la investigación, así como de las herramientas de e-learning que justificadamente se habrán de seleccionar para ayudar a resolver el problema planteado.

El capítulo 3 permitirá describir a detalle la propuesta de etiquetado sobre los diferentes estándares de e-learning. Propuesta que servirá como base para una adecuada recuperación

de la información por parte de los algoritmos descritos en los capítulos posteriores y, además, para la representación adecuada de la información que resulte del proceso de planificación, de manera que pueda ser interpretada por las herramientas de e-learning que permitirán su aplicación real. Al proceso de especificación y organización de información se le llama *Ingeniería de Conocimiento*, por lo cual ese será el título del capítulo.

En el capítulo 4, *Generación Automática de Dominios e-Learning*, se describirá el etiquetado mínimo que deben tener los objetos de aprendizaje, así como las diferentes formas de recabar la información sobre el perfil del estudiante y el conjunto de información mínima que deben contener dichos perfiles para llevar a cabo el proceso de adaptación. Posteriormente, se detallarán los algoritmos de ingeniería de conocimiento creados para generar los archivos de dominio y problema necesarios para el funcionamiento de los dos planificadores seleccionados y, finalmente, se mostrarán algunos de los planes (diseños de aprendizaje) obtenidos a partir de la ejecución de los planificadores y se hará una comparativa entre ellos.

A partir del capítulo 5 titulado *Planificación de Itinerarios Formativos en Entornos Dinámicos*, nos iremos adentrando en la naturaleza dinámica de los diseños instruccionales en entornos de e-learning, problemática que normalmente suelen pasar desapercibidos en un entorno tradicional. Y también enunciaremos una primera propuesta de aplicación de diversas técnicas de planificación sobre un conjunto reducido de variables dinámicas e inciertas.

El capítulo 6 llamado *Integración con el LMS*, se centrará en definir la arquitectura que permitirá integrar el proceso de generación automática de diseños de aprendizaje con un Sistema Gestor de Aprendizaje (LMS) genérico que soporte diversas restricciones. También abordará el comportamiento que deberá tener dicha arquitectura para poder soportar los diferentes métodos de planificación que complementarán la generación automática de secuencias de aprendizaje para su mejor integración en un dominio real de e-learning.

Para demostrar que nuestra propuesta funciona en un entorno real, en el capítulo 7 *Experimentación en Dominios Reales* nos enfocaremos en los aspectos técnicos de la implementación de la arquitectura descrita en el capítulo anterior dentro del LMS Moodle. La forma en que dicho LMS debe adaptarse para soportar los diferentes procesos de planificación necesarios para la adaptación y la evaluación cualitativa de la respuesta de los

estudiantes al trabajo con el LMS y las secuencias de adaptadas, serán también parte de este capítulo.

Finalmente, en el último capítulo, discutiremos nuestras conclusiones principales respecto a este estudio, daremos un breve repaso a la difusión que se ha hecho del mismo en diversas revistas y congresos nacionales e internacionales, además de abordar las posibilidades de futuras mejoras y/o aplicaciones a otros campos.

CAPÍTULO 2

Preliminares

En este capítulo se describirán los conceptos básicos que serán utilizados a lo largo de este trabajo. Tanto los conceptos relativos a las tecnologías, herramientas y estándares e-learning, como los que tienen que ver con la Planificación y Scheduling Inteligentes (AIP&S del inglés Artificial Intelligence Planning and Scheduling), serán tratados de una manera clara y resumida en el marco de la historia de la inteligencia artificial en la educación y del proceso de investigación seguido para concretar el rumbo y las aportaciones de esta memoria.

1. Un Poco de Historia

Los ordenadores han estado íntimamente ligados a la historia y evolución de los ambientes educativos. Aunque en un principio la comunidad académica se mostró renuente al uso de los ordenadores, poco a poco ésta se fue adaptando hasta convertir el ordenador en una herramienta indispensable para el desarrollo del ejercicio académico.

Los primeros intentos de introducir ordenadores en las escuelas se dieron en USA en el año de 1956. En 1959 dió inicio en este país el primer programa nacional pa-

ra implantar ordenadores en las aulas de escuelas primarias y secundarias denominado PLATO[Bitzer et al., 1961] (del inglés, Programmed Logic for Automated Teaching Operations) y 4 años más tarde se instaló en estos ordenadores el primer programa para la enseñanza de las matemáticas y la lectura. Sin embargo, no fue hasta el surgimiento de LOGO en 1967 cuando los profesores se dieron cuenta del gran potencial que tenían los programas computacionales como herramientas para la enseñanza de las ciencias.

En la década de los 80's surgieron los primeros Sistemas Tutoriales Inteligentes (ITS, del inglés Intelligent Tutoring Systems) y paralelamente se desarrollaron los primeros Sistemas Gestores de Aprendizaje (LMS, del inglés Learning Management Systems). Estas plataformas de e-learning, apoyadas posteriormente por el uso del Internet, han revolucionado la educación, ya que a partir de ellas se ha creado toda una nueva metodología de educación denominada Educación a Distancia.

La principal ventaja de esta nueva forma de enseñar radica en que los estudiantes pueden acceder a un tutor y material educativo sin necesidad de desplazarse a las grandes ciudades o a las instalaciones educativas, lo cual permite llegar a una mayor cantidad de personas que pueden radicar en cualquier parte del mundo y que tienen características muy diferentes entre sí.

Por otro lado, estas herramientas también permiten una comunicación asíncrona con el tutor y acceder a la información en cualquier momento, lo cual las hace más interesantes para las personas trabajadoras que desean organizar su tiempo según su variado número de actividades diarias.

Los ITS's fueron más útiles que los LMS's para probar las técnicas de inteligencia artificial que tenían auge en aquella época, dado que las técnicas de programación para la web no estaban tan desarrolladas como hoy en día. Es por esta razón que los ITS's se centraron en la automatización del diseño, despliegue y adaptación de cursos. Generalmente el ITS desplegaba un curso o un conjunto limitado de cursos con características similares, los cuales no necesariamente podrían mostrarse en la web, mientras que los LMS's se enfocaron en la gestión de cursos y creación de contenidos generalizables para la web.

Pero, ¿qué es un sistema tutorial inteligente? ¿qué es un sistema gestor de aprendizaje? y ¿por qué se ha decidido utilizar un sistema gestor de aprendizaje como parte de la arquitectura de este trabajo? Estas y muchas otras preguntas se irán contestando a lo largo de los siguientes capítulos.

2. Plataformas de Enseñanza por Ordenador

Como se ha mencionado en párrafos anteriores, existen dos plataformas principales para la enseñanza por ordenador que han ido evolucionando de maneras distintas a partir de 1980 y hasta nuestros días: los *Sistemas Tutores Inteligentes* y los *Sistemas Gestores de Aprendizaje*.

2.1. Sistemas Tutores Inteligentes

Los Sistemas Tutores Inteligentes son sistemas que utilizan técnicas de la inteligencia artificial para proporcionar formación personalizada en un área o tema particular, adaptando dicha información y el proceso de enseñanza de la misma a las características particulares de cada estudiante. Se centra en imitar las características de un “buen” tutor particular humano.

Estos sistemas facilitan el proceso de enseñanza/aprendizaje haciéndolo más efectivo, correcto y también más agradable. Algunas de sus características más importantes desde el punto de vista educativo según Maite Urretavizcaya [Urretavizcaya, 2001] son:

1. El conocimiento del dominio está acotado y claramente articulado.
2. Poseen conocimiento del estudiante que les permite dirigir y adaptar la enseñanza.
3. La secuencia de enseñanza no está predeterminada por el diseñador.
4. Realizan procesos de diagnóstico más adaptados al estudiante y más detallados.
5. La comunicación Tutor-Alumno mejora, permitiendo además que el alumno realice preguntas al Tutor.

Existen muchos y variados ejemplos de sistemas tutores inteligentes, algunos desarrollados en el mundo universitario y otros creados para fines comerciales. Entre los mas conocidos se pueden mencionar Andes, desarrollado por el equipo de Kurt VanLehn de la Universidad de Pittsburg y distribuido de forma gratuita, y Read-on! en el ámbito comercial, los dos citados en [Sanchez and Lama, 2007]. Mientras que Andes proporciona ayuda cuando lo solicitan los estudiantes que están resolviendo problemas o ejercicios y su forma de hacerlo es aportando pistas para avanzar en la solución o indicándoles sus fallos, Read-On! es un producto que enseña habilidades de comprensión lectora a los adultos analizando y diagnosticando cada una de sus deficiencias específicas y adaptando su formación en función de ellas.

Como algunos ejemplos de ITS's relacionados con el problema planteado en esta memoria se encuentran LEACTIVEMATH[Ullrich, 2005], PASER[Vrakas et al., 2007], CAMOU [Camacho et al., 2007], DCG y CoCoA [Brusilovsky and Vassileva, 2003]. Los primeros tres utilizan técnicas de planificación y scheduling inteligentes utilizando diferentes criterios y tecnologías de programación para ordenar los contenidos de un tema o curso, mientras que los dos últimos se basan en estrategias de razonamiento dirigidas por agentes para ordenar sus contenidos.

El siglo pasado existía una marcada distinción entre sistemas de enseñanza por ordenador tales como los micromundos, ambientes colaborativos de aprendizaje, sistemas de aprendizaje automático, aprendizaje basado en juegos y sistemas tutores inteligentes[Lewis, 1992, Sanchez and Lama, 2007], sin embargo, en la última década todos estos sistemas y las respectivas técnicas de inteligencia artificial que cada uno utilizaba, se han ido integrando para dar lugar a sistemas tutores inteligentes mas robustos pero normalmente diseñados para un dominio o área del conocimiento específica y con una comunidad de usuarios sumamente reducida, con sus contadas excepciones.

Por otro lado, no todos los ITS's consideran el paradigma del uso de Internet como base de su tecnología de desarrollo, por lo tanto, muchos de ellos tienen que ser instalados en cada ordenador donde vayan a utilizarse, requiriendo un software especial y sin permitir necesariamente la comunicación síncrona con el resto de los participantes del curso.

A grandes rasgos, los sistemas tutores inteligentes se basan generalmente en cinco

módulos principales que serán descritos a continuación y que, para una mayor comprensión, se encuentran resumidos gráficamente en la figura 1:

ESTRATEGIAS PEDAGÓGICAS

Estas estrategias están íntimamente relacionadas con las Metodologías de Enseñanza-Aprendizaje. Son actividades que se encargan de reforzar y apoyar los procesos de instrucción-aprendizaje que se lleven a cabo mediante nuevas tecnologías.

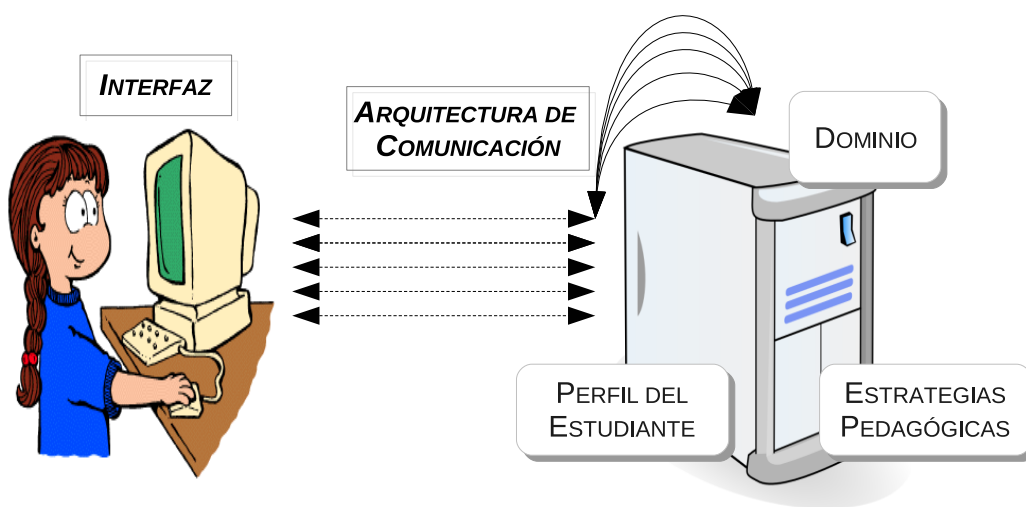


Figura 1. Arquitectura de un Sistema Tutor Inteligente. Contempla los cinco módulos principales de: interfaz, arquitectura de comunicación, perfil del estudiante, dominio y estrategias pedagógicas.

Por ejemplo, comparación de imágenes o textos, abstracción de las ideas principales de un texto, visualización de transparencias, vídeos o demostraciones en flash, identificación, clasificación y ordenación de elementos de un ambiente (ya sea una lista de conceptos u objetos en flash), crítica o discusión de un tema a través de foros, el enunciado de un trabajo de campo, etc.

Estas y muchas mas estrategias pedagógicas son utilizadas por distintas metodologías de enseñanza tales como: aprendizaje colaborativo, basado en juegos, basado en problemas, basado en el descubrimiento del conocimiento, individualizado, etc.

DOMINIO

El dominio del problema, es decir, el conocimiento que se tiene acerca de los contenidos del tema o curso que deberá estudiar el alumno.

Normalmente este dominio se encuentra estructurado de forma jerárquica como en un libro, asociando a cada objetivo de aprendizaje general un tema, y a los objetivos específicos diversos sub-temas y actividades que permitan adquirir, reforzar y evaluar el conocimiento sobre los mismos.

PERFIL DEL ESTUDIANTE

El perfil del estudiante comprende todas las características del estudiante, desde su nombre y correo electrónico, pasando por su historial académico, sus preferencias, su comportamiento dentro del sistema, etc.

Una característica pedagógica del perfil del estudiante que se ha tomado muy en cuenta dentro de estos sistemas en los últimos años ha sido el estilo de aprendizaje.

El término **estilo de aprendizaje** se refiere al hecho de que, cuando queremos aprender algo, cada uno de nosotros utiliza su propio método o conjunto de estrategias; percibiendo, interactuando y respondiendo al ambiente de aprendizaje. Aunque las estrategias concretas que utilizamos varían según lo que queramos aprender, cada uno de nosotros tiende a desarrollar unas preferencias globales. Esas preferencias o tendencias a utilizar más unas determinadas maneras de aprender que otras constituyen nuestro estilo de aprendizaje.

Existen distintas teorías de estilos de aprendizaje que categorizan al individuo según determinadas características, un resumen de estas teorías lo presenta Cazau [Cazau, 2001] por medio de la tabla 1.

Aunque, según los expertos, la teoría mas completa es la de Felder y Silverman dado que trabaja con casi todas las dimensiones de aprendizaje posibles; nosotros utilizaremos en este trabajo la teoría de Honey-Alonso por las dos razones siguientes:

- Categoriza la forma de aprender según el modo de procesar la información. Y dado que

Teoría Según:	Categorías
El hemisferio cerebral	Lógico Holístico
El cuadrante cerebral (Herrmann)	Cortical Izquierdo Límbico Izquierdo Límbico Derecho Cortical Derecho
El sistema de representación (PNL)	Visual Auditivo Kinestésico
El modo de procesar la información (Honey-Alonso)	Activo Reflexivo Pragmático Teórico
La categoría bipolar (Felder y Silverman)	Activo/Reflexivo Sensorial/Intuitivo Visual/Verbal Secuencial/Global
El tipo de inteligencia (Gardner)	Lógico-Matemático Lingüístico-Verbal Corporal-Kinestésico Espacial Musical Interpersonal Intrapersonal Naturalista

Tabla 1. Teorías de Estilos de Aprendizaje y sus Categorías.

en los ambientes de e-learning gran parte (si no es que todo) el proceso de aprendizaje se realiza a través de procesamiento e intercambio de información vía Internet, esta relación marca una pauta importante para su elección.

- Existen artículos de investigación importantes que relacionan cada categoría de dicho estilo con los tipos de recursos soportados por las herramientas de generación de objetos de aprendizaje y los estándares que pueden categorizarlos[Rojas and Defude, 2008] o que prueban que el uso de estilos de aprendizaje para la adaptación es bastante útil por ejemplo, [Ruiz et al., 2008], [Cobos et al., 2007], entre otros.

Aunque también se toman en cuenta otro tipo de características que definen niveles de conocimiento, tales como el nivel de inglés o de rendimiento académico en cursos previos.

ARQUITECTURA DE COMUNICACIÓN

Un ITS precisa una arquitectura que permita comunicar y diferenciar claramente los distintos módulos que se ha ido mencionando. Esta comunicación puede llevarse a cabo a través de servicios web, agentes o cualquier otro tipo de paso de mensajes.

Para ello también es necesario que se defina un protocolo de comunicación entre los diferentes módulos, así como la información estructurada que cada módulo tiene la posibilidad de enviar y/o recibir.

INTERFAZ

También debe contemplarse el diseño y desarrollo de una interfaz intuitiva, usable, dinámica, interactiva y/o adaptada que no solo permita a los alumnos explorar su diversas opciones de aprendizaje, si no también una modificación rápida y efectiva por parte del motor inteligente del sistema.

Estos cinco módulos se están empezando a considerar dentro de la comunidad de desarrollo de los Sistemas Gestores de Aprendizaje ya que son la base de todo ambiente virtual de aprendizaje que se considere inteligente, por lo tanto, serán tomados muy en cuenta como un paso muy importante dentro del proceso de desarrollo del trabajo descrito en esta memoria.

2.2. Sistemas Gestores de Aprendizaje

Un Sistema Gestor de Aprendizaje, por su parte, es un sistema especializado en tecnologías de aprendizaje basado en Internet que se utiliza para dar información y entrenamiento, todo ello siguiendo un paradigma abierto de aprendizaje a distancia [Avgeriou et al., 2003]. En la década de los noventa eran mas conocidos como Ambientes Hipermedia [Lewis, 1992].

Aunque en un principio su desarrollo era relativamente sencillo, mas no ampliamente

extendido. Hoy en día el diseño e implementación de este tipo de sistemas no es una tarea fácil ya que son sistemas complejos que incorporan una gran variedad de componentes organizacionales, administrativos, instruccionales y tecnológicos.

Los LMS's mas populares cuentan actualmente con un sistema gestor de contenidos integrado que permite crear una estructura de soporte (framework) para la creación y administración de contenidos por parte de los participantes, principalmente de páginas web. Un gestor de contenido consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio; y generalmente permite manejar de manera independiente el contenido y el diseño del mismo. Esta característica hace que los LMS's sean las plataformas mas utilizadas por las universidades e instituciones de enseñanza para impartir cursos por internet, mas aún que cualquier ITS.

2.2.1. *Sistemas Gestores de Aprendizaje Comerciales*

Los primeros LMS's empezaron a desarrollarse en la década de los 80's y surgieron a raíz de proyectos académicos que finalmente se convirtieron en empresas pequeñas. Con el paso de los años el gobierno y otras empresas, dedicadas a la transferencia de información en la web, apoyaron a estas pequeñas empresas y se fueron fusionando hasta crear los robustos sistemas gestores de aprendizaje comerciales.

Dos de los sistemas gestores de aprendizaje comerciales mas conocidos y utilizados en las instituciones educativas y empresas dedicadas a la formación, son:

- Blackboard Academic Suite[Inc., 2010a]. Ha sido creado por la compañía de software Blackboard Inc. que en un principio se dedicaba a dar consultoría a la organización sin fines de lucro IMS Global Learning Consortium y que, después de fusionarse con compañías pioneras y/o líderes en el ramo de los sistemas de aprendizaje en línea como CourseInfo LCC y WebCT, ha llegado a dominar este campo.

Cuenta con un entorno de manejo de cursos, comunidades y sistemas de portales en línea y un sistema de manejo de contenidos integrado. Es uno de los sistemas que ha ido adaptándose a los estándares IMS[IMS-GLC, 2003] (descritos en las siguien-

tes secciones) con mayor celeridad y ganando terreno en las comunidades educativas cubriendo mas de 2200 instituciones a lo largo de 60 países.

- Desire2Learn[Inc., 2010b]. Combina Sistemas de Gestión de Aprendizaje, Sistemas de Administración de Contenido, un Depósito de Objetos (Base de Datos) de Aprendizaje (del inglés LOR, Learning Objects Repository) y un número de otras herramientas para la educación en línea. La plataforma de aprendizaje consiste de un paquete de herramientas de enseñanza y aprendizaje para la creación, calificación, comunicación, manejo y entrega de cursos.

El LOR es un Depósito basado en estándares para almacenaje, etiquetado, búsqueda y reutilización de objetos de aprendizaje; eso permite a las organizaciones manejar y compartir contenidos a través de múltiples programas, cursos y secciones.

Si bien estos sistemas cuentan con una gran cantidad de usuarios satisfechos, en gran medida por el soporte técnico que ofrece la misma empresa, suelen tener alianzas con otras compañías estratégicas y tener utilidades muy novedosas; la principal desventaja general de este tipo de sistemas es su costo en licencias que suele ser muy elevado y su incapacidad de adaptarse a necesidades concretas de los clientes, lo que finalmente hace que los clientes terminen conformándose a las herramientas que ellos les ofrecen sin siquiera poder pensar en como adaptar éstas herramientas, integrarlas a otras o crear las suyas propias.

2.2.2. *Sistemas Gestores de Aprendizaje Libres*

También en el seno de las Universidades y/o con el apoyo de organizaciones sin fines de lucro, han surgido y se han establecido LMS's tan robustos como los comerciales, pero cuyos fundadores han decidido registrar bajo licencias libres. La difusión libre de los LMS's a permitido a la comunidad educativa tener un acceso mas fácil a los contenidos asociados a los mismos, además de que cualquier persona con bases de programación web puede visualizar, estudiar y modificar el código fuente de los mismos para adaptarlos a las necesidades de su institución educativa.

Los sistemas de aprendizaje que hemos decidido estudiar en este trabajo, debido a su amplia difusión, soporte de estándares y/o facilidad de uso, son los listados a continuación:

- *Moodle*[Dougiamas,]. Este sistema gestor de aprendizaje cuyo significado en español es Entorno Modular de Aprendizaje Dinámico Orientado a Objetos (Modular Object-Oriented Dynamic Learning Environment, por sus siglas en inglés) fue creado en 2002 por Martin Dougiamas quien basó su diseño en las ideas del constructivismo en pedagogía que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas, y en el aprendizaje colaborativo.

El código de este sistema está escrito en PHP¹ bajo la licencia GNU GPL² y soporta gran variedad de bases de datos y plataformas. Hasta Agosto de 2010, la base de usuarios registrados incluye más 36 millones, cuenta con 50,228 sitios que se encuentran distribuidos en 213 países y está traducido a más de 75 idiomas. Su acelerado crecimiento y la adaptación a nuevas tecnologías, que es posible gracias a la modularidad del mismo, lo han llevado a establecer alianzas estratégicas con empresas y universidades como el IMS Global Learning Consortium o la Open Source Initiative; esto le ha permitido, no solo continuar con su desarrollo, si no también comprometerse con el soporte de estándares y tener acceso a repositorios de objetos de aprendizaje en todo el mundo.

La principal desventaja de Moodle es que falta mejorar su interfaz y que algunas actividades pueden ser un poco mecánicas dado que dependen del diseño instruccional generado por las instituciones y grupos de profesores. Sin embargo, se espera que en su versión 2.0, que saldrá a la luz a finales del 2010, se mejoren al menos los aspectos de interfaz y su compatibilidad con las nuevas tecnologías de la web semántica.

- *dotLRN*[dotLRN Consortium, 2009]. Es un completo programa informático de tipo LMS empresarial que soporta recursos de e-learning y comunidades digitales, fue desarrollado por el Massachusetts Institute of Technology (MIT) y es utilizado por mas de medio millón de personas alrededor del mundo. Su arquitectura se basa en el kit de herramientas de código abierto Open Architecture Community System (OpenACS³) y cuenta con un sofisticado sistema de portales que integra herramientas para ad-

¹Fuente: <http://www.php.net/>

²Fuente: <http://www.gnu.org/licenses/gpl.html>

³<http://www.openacs.org/>

ministrar cursos, contenidos y herramientas de colaboración. Es escalable, robusto, extensible y cumple el estándar SCORM[SCORM, 2004]. Su principal característica y ventaja son las herramientas colaborativas que ha integrado a lo largo de los años.

- *Ilias*[of Cologne, 2000]. Es un LMS desarrollado en código abierto bajo la licencia GPL. El nombre de ILIAS proviene del alemán Integriertes Lern-,Informations- und Arbeitskooperations-System, y en español se traduciría como Sistema de Cooperación, Información y Aprendizaje Integrado. Ha sido desarrollado con la idea de reducir los costes de utilización de las nuevas tecnologías en la educación, teniendo en cuenta, siempre y en todo momento, las ideas de los usuarios del sistema de enseñanza. Fue uno de los primeros en integrar dentro de su plataforma la generación y etiquetado de objetos de aprendizaje bajo los estándares SCORM-RTE3 y LOM, además de soportar cuestionarios bajo el estándar IMS-QTI y actualmente usuarios de todo el mundo contribuyen en el proyecto bajo la coordinación de un equipo de la la Universidad de Colonia en Alemania. Es utilizado por instituciones como la NATO Headquarters Supreme Allied Command Transformation (SACT), Partnership for Peace - Consortium of Defense Academies and Security Studies Institutes y la International Organization for Standardization (ISO). Ha sido instalado y utilizado en diversas instituciones educativas, públicas y privadas en 20 países alrededor del mundo.

Cabe mencionar que en los inicios de la realización de este proyecto, ILIAS era el LMS utilizado por el Centro de Enseñanzas Virtuales de la Universidad de Granada(CEVUG).

- *Caroline*[Caroline, 2010]. Es una plataforma abierta de e-learning e e-working del tipo LMS que permite a los profesores crear cursos en línea y manejar actividades colaborativas y de aprendizaje en la web. Ha sido traducido a 35 idiomas, lo cual le permite ser utilizado por cientos de organizaciones de 93 países.

Esta plataforma es capaz de soportar fácilmente a un gran número de usuarios, es compatible con los sistemas operativos Windows, Linux y Mac, se basa en tecnologías como PHP y MySQL⁴ y utiliza estándares como SCORM e IMS-QTI para intercambiar

⁴Fuente: www.mysql.com

contenidos.

Al principio del desarrollo de este trabajo se hicieron pruebas con ILIAS, sobre todo para la recuperación de los objetos de aprendizaje y sus metadatos y posterior generación del plan, mas no se completó la etapa de integración. Todo esto por que, como ya se ha dicho, el Centro de Enseñanzas Virtuales de la Universidad de Granada (CEVUG) utilizaba este LMS y por que en ese momento era, si no el único, de los pocos LMS's que permitía la edición y recuperación de los objetos de aprendizaje y sus metadatos.

Posteriormente, y después de tomar en cuenta las descripciones y consideraciones anteriores acerca de cada uno de los sistemas gestores de aprendizaje y los sistemas tutores inteligentes estudiados se ha decidido decidido optar por un LMS libre por las siguientes razones:

1. Los ITS's no cuentan con un framework robusto para la administración de cursos, estudiantes y profesores. A diferencia de lo que ocurre con la mayor parte de los LMS's.
2. Muchos de los ITS's requieren la instalación de software específico en los ordenadores donde vaya a ser utilizado. Software que en ocasiones no puede ser distribuido fuera de las instalaciones educativas por cuestiones de licencias o complejidad y que limita su uso al no poder acceder por entero a él a través del navegador. Y algo similar ocurre con las licencias con los LMS's comerciales.
3. La mayoría de los ITS's existentes están limitados a un área del conocimiento o curso específico. Lo que no ocurre con un LMS que permite generar material educativo para cursos de muchas y muy variadas áreas del conocimiento. Esta característica de los LMS's permite generalizar la adaptación de diseños instruccionales para cualquier tipo de curso de cualquier área del conocimiento.
4. Un LMS comercial no permite acceder a la información de sus bases de datos mediante código externo, debido a que no existe una documentación al respecto y tampoco permite su modificación por personas ajenas a la empresa.

5. El LMS mas utilizado y que se equipara a Blackboard en el soporte de los estándares que serán descritos mas adelante, es libre y se llama Moodle.

Por lo tanto, la experimentación del modelo propuesto en este documento se ha llevado a cabo en Moodle debido, sobre todo, a que es el mas utilizado de todos los LMS's libres, a que soporta la mayor parte de los estándares que se requieren para resolver nuestro problema y a que, en caso de que no los soportara, sería posible modificarlo de una forma relativamente simple debido a su arquitectura basada en la modularización de contenidos y a su completa documentación.

Es importante resaltar que Moodle es actualmente el sistema gestor de aprendizaje utilizado por el CEVUG y plataforma de intercambio de información dentro de los cursos del Máster en Soft Computing y Sistemas Inteligentes de la Universidad de Granada⁵. Por otro lado, es el LMS que utiliza Grupo Tadel, empresa con la cual se ha probado en un entorno real una de las etapas de la propuesta de esta memoria, gracias a una beca de colaboración y experiencia empresarial.

Sin embargo, aunque la elección de plataforma de e-learning haya sido un LMS, somos conscientes de que la mayor parte de las aplicaciones inteligentes para la educación han sido implementadas sobre ITS's. Por otro lado, debemos mencionar que uno de los problemas mas comunes y complejos que se presenta a gran parte de los educadores experimentados en la impartición de cursos en línea, es el de la adaptación a las diferentes necesidades de los estudiantes de las secuencias de contenidos que conforman los diversos temas de un curso, como lo comentábamos en la introducción de esta memoria.

3. E-Learning: Herramientas y Conceptos de Apoyo a la Generación de Diseños Instruccionales

Para poder resolver el problema de la generación automática de diseños instruccionales para plataformas e-learning es preciso profundizar en la terminología que la comunidad de educadores en línea utiliza, los estándares que usan para representar la información de los

⁵Máster en Soft Computing y Sistemas Inteligentes de la Universidad de Granada: <http://doctosi.ugr.es/master/scsi/index.php>

componentes principales de un curso en línea, las herramientas que usa para acceder a y utilizar dichos estándares, entre otros tantos conceptos que, aunque ya han sido descritos de manera general en la introducción de esta memoria, se detallarán en los siguientes apartados.

3.1. E-Learning

La terminología relativa al e-learning ha sido propuesta por personas que en su mayoría trabajan en el área de educación, por esta razón es muy diferente a la utilizada por los estudiosos de la planificación y scheduling inteligentes y, sin embargo, las dos tienen una equivalencia que se ve con claridad una vez que es posible entender sus conceptos principales.

En [Rosenberg, 2001] Rosenberg define al e-learning como: *el uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento, el cual está basado en tres criterios fundamentales:*

- 1. El e-learning trabaja en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información.*
- 2. Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet.*
- 3. Se enfoca en la visión más amplia del aprendizaje que van más allá de los paradigmas tradicionales de capacitación.*

Las plataformas de e-learning más utilizadas, tanto en universidades como en centros de formación continua que se dedican a llevar práctica este concepto son, como se ha descrito al principio de este capítulo, los sistemas gestores de aprendizaje.

En los últimos tiempos este tipo de sistemas ha empezado a adoptar, como parte de los contenidos de sus cursos, los objetos de aprendizaje. Las capacidades de reusabilidad y de transferencia de información adicional, que de estos objetos se obtiene gracias a los estándares que se han empezado a usar sobre ellos, serán esenciales para el desarrollo de este trabajo.

Por otro lado, este mismo conjunto de estándares junto con diversas teorías pedagógicas

como la relativa a los estilos de aprendizaje de los estudiantes, permitirán ir mas allá de los paradigmas tradicionales de capacitación creando con los objetos de aprendizaje una secuencia personalizada llamada diseño instruccional, o LD (Learning Design) si se traduce a la especificación estándar que lo describe dentro de una plataforma de e-learning y que se analizará mas a detalle en los apartados siguientes.

Para la definición de éste LD también se toman en cuenta otras características de cada uno de los alumnos del curso, incluido su historial pedagógico y su interacción con el LMS y el resto de los alumnos, cosa que suele complicar al profesor (quien es el encargado de construirlo) la definición del mismo, sobre todo cuando existe un gran número de estudiantes y/o son demasiados los contenidos del curso.

Es importante, antes de ahondar en la descripción de nuevos conceptos y características de dominios e-learning, hacer énfasis en las capacidades de interacción y manipulación que son necesarias para llevar a cabo un proceso de personalización sobre los objetos de aprendizaje que lo componen; sobre todo en aquellas características que se esperan de un profesor (o de cualquier herramienta que lo apoye, en el caso de esta memoria una herramienta de AIP&S) para que sea capaz de crear diseños instruccionales adaptados a cada uno de los estudiantes de un curso. Dichas características, tal y como se ha descrito en la introducción de esta memoria son por lo menos las siguientes:

1. La posibilidad de reconocer las relaciones que existen entre los objetos de aprendizaje de un curso y reconstruir, en base a ellas, una secuencia de actividades coherente.
2. La capacidad de distinguir las restricciones técnicas y/o formativas impuestas a cada uno de los objetos de aprendizaje y respresentar dichas restricciones para su posterior procesamiento.
3. La capacidad de permitir la inclusión, dentro de un itinerario formativo, de objetos de aprendizaje con características específicas que refuercen el conocimiento de cada estudiante. La decisión de incluirlos, o no, podría depender de la disponibilidad temporal o capacidad académica del estudiante.
4. La aptitud de manejar redundancia en los nombres de los objetos de aprendizaje.

Pues pueden existir objetos de aprendizaje que cumplan con los mismos objetivos, se llamen igual, pero tengan diferentes restricciones; u, objetos que tengan el mismo nombre, pero objetivos distintos.

5. La posibilidad de manejar restricciones temporales, tanto por parte de los alumnos como de las actividades del curso, y en base a ellas decidir la longitud de un curso (número de objetos de aprendizaje) para cada estudiante.
6. La capacidad de manejar la incertidumbre que conlleva la aplicación de evaluaciones intermedias, de manera que sea posible adaptar la secuencia de aprendizaje posterior a una evaluación, a los posibles resultados de la misma.
7. La posibilidad de detectar cambios en el perfil del estudiante y/o los objetos de aprendizaje, y re-adaptar la secuencia de aprendizaje que se vea afectada por dichos cambios.

Existen, además, algunos términos que son utilizados comúnmente en la literatura educativa y que para efectos de esta investigación serán definidos de la forma siguiente:

PLATAFORMAS DE E-LEARNING. Son los sistemas computacionales que nos permiten acceder, modificar, estructurar y utilizar la información completa de uno o varios cursos en línea, así como la de sus instructores, moderadores, diseñadores y estudiantes, según sea el caso. Las principales plataformas e-learning son conocidas como sistemas gestores de aprendizaje o sistemas tutores inteligentes.

AMBIENTE DE APRENDIZAJE. Es el escenario donde existen y se desarrollan las condiciones favorables para el aprendizaje.

AMBIENTES VIRTUALES DE APRENDIZAJE. Es el ambiente de aprendizaje que proporciona la plataforma e-learning.

OBJETIVO DE APRENDIZAJE. Es aquel que describe lo que el estudiante va a ser capaz de aprender, entender o hacer como resultado del desarrollo de una lección o tema.

HERRAMIENTAS DE E-LEARNING. Son todos aquellos componentes de un curso en línea que nos permiten acceder, compartir, experimentar, ejercitar, etc., la información necesaria para comprender y aprender los objetivos de aprendizaje de un curso en línea.

ACTIVIDADES DE APRENDIZAJE. Son aquellas herramientas de e-learning por medio de las cuales se obtienen nuevos conocimientos, habilidades o aptitudes. Una descripción detallada de las mismas será expuesta en la sección de objetos de aprendizaje, ya que ese es el nombre oficial que adoptan las actividades de aprendizaje cuando se les estandariza y comparte.

DISEÑO INSTRUCCIONAL. Es aquel que integra un conjunto de actividades de aprendizaje con el ambiente de aprendizaje, y permite estructurar dichas actividades en base a teorías pedagógicas que les permitan mejorar su eficiencia, eficacia y atractivo dependiendo de las características específicas de cada grupo de estudiantes o cada estudiante particular (en caso de tener grupos pequeños). A un diseño instruccional también suele llamársele *Itinerario Formativo*.

En las siguientes secciones se dará una breve introducción al concepto de objetos de aprendizaje y los estándares en general, haciendo una mención especial para aquellos que se utilizan sobre la información de los objetos de aprendizaje. Se dedicarán secciones completas a los estándares relativos a la definición de diseños de aprendizaje y del perfil del estudiante.

3.2. Objetos de Aprendizaje

Un objeto de aprendizaje fue definido en sus inicios por Wiley [Wiley, 2000] como un recurso digital que puede ser rehusado para ayudar en el aprendizaje. Una definición mas amplia y orientada al ámbito del e-learning se puede encontrar en [ANSI/IEEE, 2007], en donde se definen como cualquier entidad, digital o no digital, que puede ser usada, reutilizada o referenciada durante el aprendizaje soportado por tecnología.

Los objetos de aprendizaje que se busca manipular en este trabajo son aquellos a los que hace referencia la segunda definición. Comprenden una amplia gama de actividades de aprendizaje tales como lecturas, ejercicios, experimentos, visualización de transparencias, vídeos o imágenes, participación en foros de discusión asíncrona o de chat, etc.

Como es posible intuir, los objetos de aprendizaje son cualquier tipo de actividad de aprendizaje soportada principalmente por un LMS y que aunque no tenga asociada ninguna herramienta digital de e-learning si pueda al menos ser referenciada por el estándar IEEE-LOM o el IMS-MD, estándares que se describirán en el siguiente apartado.

3.3. Estándares

En esta sección se presenta una breve relación de los estándares utilizados en el desarrollo del proyecto, la relación entre ellos y una breve introducción al uso que se les dará a lo largo de esta memoria.

3.3.1. *ANSI/IEEE Standards*

IEEE corresponde a las siglas de The Institute of Electrical and Electronics Engineers, el Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Dentro de ésta asociación se ha integrado un comité encargado de los estándares técnicos, buenas prácticas y guías recomendadas para el aprendizaje de las tecnologías.

Uno de los estándares que este comité a desarrollado es el LOM (Learning Object Metadata), el cual define la sintaxis y semántica de los metadatos asociados a los objetos de aprendizaje. Estos metadatos son atributos que describen a un objeto de aprendizaje con el objetivo de ser localizado, gestionado y evaluado fácilmente.

En la presente memoria se utilizarán algunos de estos metadatos con el fin de obtener información pedagógica, técnica y relacional de cada objeto, que después traduciremos a precondiciones, anidaciones y efectos. La especificación de los metadatos a utilizar y el uso adecuado de los mismos se abordará en la sección 1 del capítulo 4. Por lo pronto, en la figura 2 se muestran todas las categorías en las que se subdividen estos metadatos para una mejor comprensión, y las sub-categorías y/o metadatos(campos) que las componen.

3.3.2. *IMS Standards*

En 1997 la iniciativa estadounidense de infraestructura de aprendizaje llamada Educa-se inició un proyecto para crear un conjunto de estándares que pudiesen ser ampliamente adoptados para el intercambio colegial de contenidos de aprendizaje. Con el paso de los años dicho proyecto tomó el nombre de Instructional Management Systems Global Learning Consortium, mejor conocido como IMS, cuya misión es apoyar en todo el mundo la adopción y uso de las tecnologías en la enseñanza y cuya principal actividad es desarrollar

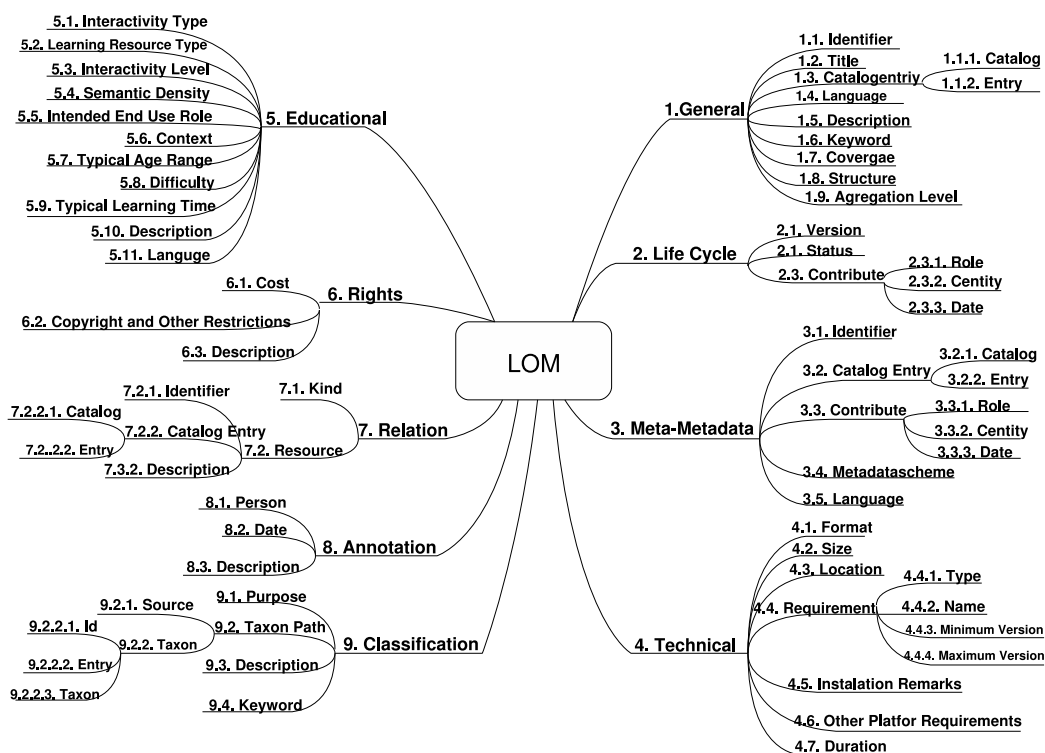


Figura 2. Categorías del estándar LOM y sus metadatos asociados. Con algunos de estos metadatos se etiquetarán los objetos de aprendizaje y jerarquías del curso que se pretenda definir.

especificaciones que cubran todos los elementos del aprendizaje distribuido.

Entre las especificaciones que ha desarrollado se encuentra la IMS-MD que se basa en el estándar LOM, la especificación IMS-CP que define un formato estándar para desplegar, gestionar, distribuir y agregar paquetes de contenidos (en este caso, objetos de aprendizaje); lo cual ayudará al profesor a organizarlos de una forma mas visual y es la base para la integración en el LMS que usaremos; IMS-LIP que recolecta información acerca del perfil del estudiante o el productor de contenidos de aprendizaje y no se puede dejar de mencionar la especificación IMS-LD a la cual se dedicará el siguiente apartado.

Por otro lado, la gran mayoría de los sistemas gestores de aprendizaje soportan una especificación llamada SCORM (del inglés Sharable Content Object Reference Model) la cual permite crear objetos pedagógicos estructurados. En esta especificación están integradas las especificaciones IMS-MD e IMS-CP; la IMS-LD está en proceso de integrarse.

- Metadatos educativos como IEEE-LOM [ANSI/IEEE, 2007] o IMS-MD [IMS-GLC, 2003] que permiten a los instructores clasificar recursos de aprendizaje de acuerdo a un conjunto de variables que pueden ser descritas, reconocidas y recuperadas de una forma mas eficiente.
- Los perfiles de los estudiantes (IMS-LIP [IMS-GLC, 2003]) son representados de manera que se pueda recoger información sobre sus características.
- Los diseños de aprendizaje (IMS-LD [IMS-GLC, 2003]) permiten a los instructores adaptar la secuencia de aprendizaje y utilizar los recursos de aprendizaje de acuerdo a las características y capacidades de cada estudiante. También permiten definir ciertos roles y grados de interacción entre ellos. Sería, por lo tanto, el equivalente en estándares al concepto de diseño instruccional definido en la sección 3.1.
- La interoperabilidad de cuestionarios y exámenes (IMS-QTI [IMS-GLC, 2003]) que permiten aplicar cuestionarios a los estudiantes y recabar toda la información necesaria para su posterior evaluación de acuerdo a un baremo o el análisis puntual de sus resultados.

3.4. Diseños de Aprendizaje

También conocido como Learning Design, en inglés, un diseño de aprendizaje es la especificación para un metalenguaje que permite modelar un proceso de aprendizaje. A través de este modelado ha sido posible generar, a mano por el momento, diseños instruccionales adaptados al perfil de un estudiante.

Los principales elementos de un LD suelen poner una mayor atención en ambientes con varias dimensiones del “contexto de aprendizaje”, una visión del e-learning con mas “actividad” (en lugar de “absorción”) y mayor atención en el rol de “multi-alumno” (pues no se limita solo a un alumno).

En el estándar IMS-LD existen tres niveles de representación de un diseño de aprendizaje:

A Implica la representación de los diferentes roles que tomarán parte del ambiente de

aprendizaje y de las acciones que se llevarán a cabo por los mismos en y sobre dicho ambiente.

B Toma en cuenta la reusabilidad de los objetos de aprendizaje, el conocimiento previo de cada estudiante y sus preferencias para trabajar sobre la definición de unidades de aprendizaje personalizadas, de acuerdo a diferentes pedagogías.

C Permite la comunicación entre componentes y roles, acepta cambios en el perfil del estudiante que implique una redefinición de las unidades de aprendizaje y el envío de mensajes a otros roles. Estas acciones se ejecutan en tiempo real y de forma colaborativa.

3.5. Perfil del Estudiante

IMS-LIP o paquete de información acerca del estudiante, es un metalenguaje que sirve para el intercambio de información entre sistemas web, especialmente gestores de aprendizaje.

Las estructuras de información principales de esta especificación se basan en: la accesibilidad, actividades, afiliaciones, competencias, metas, intereses, relaciones, llaves de seguridad, transcripciones, títulos, certificaciones y licencias de un estudiante o creador de contenidos.

En este caso se ha decidido utilizar solo aquellas que proporcionan información del estudiante soportada por un LMS. Tal es el caso de las estructuras de competencias, metas, accesibilidad e intereses, cuya información se detallará en la sección 2 del capítulo 3. Sin embargo, existe un dato dentro de esta definición llamado estilo de aprendizaje que, como se ha detallado en la descripción del módulo del perfil del estudiante de los ITS's, forma una parte muy importante dentro del proceso de adaptación del LD.

3.6. SCORM

SCORM no es un estándar, es mas bien una especificación que sirve como modelo de referencia a los programadores para poder utilizar y combinar adecuadamente un conjunto de estándares técnicos, como por ejemplo los IMS, de manera que sea posible establecer

con ellos un proceso de comunicación y de intercambio de conocimiento entre distintas plataformas de e-learning.

Su nombre viene del inglés Sharable Content Object Reference Model que se traduce a *Modelo de Referencia para el Contenido de Objetos Intercambiables*, por lo cual podemos darnos cuenta que esta especificación solo incluye estándares que hagan referencia a objetos de aprendizaje y las relaciones entre ellos.

SCORM fue creado por el Advanced Distributed Learning (ADL), un grupo de investigación patrocinado por el departamento de defensa de Estados Unidos. Su última versión conocida se remonta a finales del año 2004 y, sabemos que es la especificación que mayor número de herramientas tiene para facilitar su uso y definición, lo cual ha facilitado su aceptación dentro de la comunidad de expertos en contenidos e-learning convirtiéndose así en la más utilizada y soportada por las plataformas.

Cuando se habla de un *paquete SCORM*, se hace referencia a un fichero en el formato zip que contiene, no solo el fichero XML que describe los objetos de aprendizaje y sus metadatos, si no también todos los ficheros “estáticos” de tipo pdf, html, doc, etc., que permiten al los estudiantes acceder llevar a cabo las actividades para las que fueron creados.

Con esta especificación se terminan de describir todas las herramientas y conceptos de e-learning que serán utilizados a lo largo de esta memoria para describir la solución al problema de generación automática de diseños instruccionales.

En las siguientes secciones se desarrollará una breve descripción y análisis de las técnicas y métodos de AIP&S, para que posteriormente sea posible definir las técnicas y herramientas que serán utilizadas para resolver el problema de la adaptación de secuencias de actividades de aprendizaje.

4. Planificación y Scheduling Inteligentes

La planificación y scheduling en la inteligencia artificial se centra en la generación automática de un plan, que implique asignación de recursos y restricciones temporales, para resolver un problema dentro de un dominio particular. A los sistemas que se encargan de

llevar a cabo el proceso de planificación para generar un plan automáticamente se les llama planificadores.

Las primeras aproximaciones, según [Castillo, 1998], enuncian un problema de planificación en base a los siguientes componentes:

1. Una descripción del mundo circundante, los objetos que lo componen, sus características y las relaciones que existen entre ellos. Esta descripción también es conocida como *Estado Inicial* del problema.
2. Una descripción de la meta que se desea alcanzar, también conocida como *Objetivo*, es decir, una descripción de las relaciones y de las características que se desean alterar en los objetos que componen el estado inicial.
3. Una descripción de todas las posibles acciones que se pueden realizar, cómo afecta cada una de ellas a los objetos del mundo, a sus características y a sus relaciones. Esta última descripción también se conoce como *Teoría del Dominio*.

En el contexto del e-learning el estado inicial del problema se extrae a partir de la información del perfil del estudiante, los objetivos o metas del mismo se obtiene a partir de los objetivos de aprendizaje de un curso y la teoría del dominio se construye, para nuestro propósito específico, utilizando un repositorio de objetos de aprendizaje en el cual cada objeto se traduce en una acción cuyas precondiciones y efectos se obtienen de la información expresada en sus metadatos o meta-información representada por medio de etiquetas.

En base a estos tres componentes, el resultado esperado de un planificador es un conjunto de secuencias ordenadas de acciones tal, que permiten alcanzar los objetivos de aprendizaje para todos y cada uno de los estudiantes del curso. Esta secuencia de acciones es conocida como un *plan*, que posteriormente puede traducirse a una secuencia de actividades de aprendizaje en todas sus variantes de acuerdo al diseño instruccional específico para cada estudiante.

4.1. Paradigmas de Planificación

Existe una gran variedad de planificadores y de formas de clasificarlos. Para el propósito de esta memoria se tendrán en cuenta las dos formas de representar las acciones del dominio y de resolver el problema de planificación que utilizan la mayor parte de los planificadores existentes. Estas son, representación lineal basada en estados y representación jerárquica basada en redes jerárquicas de tareas, es decir, HTN.

Las definiciones formales de las representaciones HTN y basada en estados, así como los planificadores que las utilizan, serán estudiados brevemente en las siguientes secciones. Sin embargo, solo los planificadores cuyo código fuente y/o binarios estén disponibles bajo licencias de código abierto y que cuenten con documentación suficiente para poder aprovechar todas sus capacidades serán considerados en este estudio.

4.1.1. *Basado en Estados*

Los planificadores basados en técnicas de planificación basadas en estados utilizan alguna de las variantes de la representación STRIPS (Stanford Research Institute Problem Solver), desarrollada aproximadamente en 1970 en el SRI International, para referirse al estado del mundo y los cambios que ocurren en el durante la resolución de un problema de planificación, así como a los operadores utilizados para resolverlo.

STRIPS[Fikes et al., 1998] representa modelos del mundo como fórmulas para el cálculo de predicados de primer orden. En este contexto, un problema se resuelve al encontrar una secuencia de operadores en el espacio de modelos del mundo que puedan transformar el modelo inicial en otro modelo en el cual se pueda probar que una fórmula dada, llamada meta, pueda ser verdadera.

La descripción original del lenguaje STRIPS se basa en tres representaciones básicas[Russell and Norvig, 2003]:

- *Representación de Estados*. Los planificadores descomponen el mundo en términos de condiciones lógicas y representan un estado como una secuencia de literales positivos conectados. Los literales pueden ser proposicionales o de primer orden simples, es decir,

sin dependencias funcionales.

Se asume la hipótesis de un mundo cerrado, por lo que todas las condiciones que no son mencionadas en un estado se asumen falsas.

- *Representación de Objetivos.* Un objetivo es un estado parcialmente especificado representado como una secuencia de literales positivos y simples.

Un estado proposicional s satisface un objetivo g si s contiene todos sus elementos en g .

- *Representación de Acciones.* Una acción es especificada en términos de precondiciones que deben cumplirse antes de ser ejecutada y de las consecuencias que se siguen cuando se ejecuta.

Para entender mejor como funciona la representación de una acción es preciso describir la definición de un *esquema de acción*, el cual representa un número de acciones diferentes que pueden ser derivadas mediante la instanciación de las variables proporcionadas durante la descripción del nombre de la acción. Por lo tanto, se puede decir que un esquema de acción consta de tres partes.

- El nombre de la acción y la lista de parámetros de los que depende, los cuales sirven para identificar la acción.
- La *precondición*, que es la unión de literales positivos sin dependencia funcional. Ésta establece lo que debe ser real en un estado antes de que una acción sea ejecutada.
- El *efecto* es la unión de literales sin dependencia funcional describiendo como el estado cambia cuando la acción es ejecutada. Un literal positivo P en el efecto se espera que sea verdadero en el estado resultante de la acción, mientras que un literal negativo $\neg P$ se espera que sea falso.

Para hacer mas sencilla la legibilidad algunos sistemas de planificación dividen el efecto en dos listas de literales.: los positivos en la *lista Añadir* y los negativos en la *lista Borrar*.

Lo descrito anteriormente es la sintáxis de la representación de problemas de planificación según la representación STRIPS clásica. La semántica se puede describir por medio de la

descripción del cómo las acciones afectan a los estados. Diremos que una acción a es aplicable en cualquier estado que satisfaga sus precondiciones; en otro caso, la acción no tendrá efecto. La aplicación para un esquema de primer orden se reduce a la sustitución θ de las variables en la precondición.

Comenzando en el estado s , el resultado de ejecutar una acción aplicable a nos lleva a otro estado s' , que es el mismo que s excepto que cualquier literal positivo P en el efecto de a es añadido a s' , y cualquier literal negativo $\neg P$ es eliminado de s' . Si un efecto positivo está ya en s , no se añade dos veces, y si un efecto negativo no está en s , entonces esa parte del efecto es ignorada. Esta definición expresa la llamada hipótesis STRIPS: cada literal no mencionado en el efecto permanece sin modificar.

Finalmente, la solución de un problema de planificación en su forma mas sencilla es simplemente una secuencia de acciones que, ejecutada en el estado inicial, da como resultado un estado final que satisface el objetivo.

Para conocer la versión final del lenguaje y planificador STRIPS [Nilsson, 1982] publicada por primera vez por el SRI International, tuvieron que pasar mas de diez años. A partir de entonces se hicieron muchos intentos por mejorar el algoritmo original, entre ellos el de [Herve, 1994] que despertó a la comunidad de planificación del letargo innovador que se había vivido por mas de 10 años, es decir, desde 1982 hasta 1994.

Todos estas mejoras a los planificadores no tuvieron gran repercusión dentro de la bibliografía, por lo que según [Nau et al., 2004] gran parte de estas propuestas podrían quedar resumidos y, en ciertos casos, mejorados por tres publicaciones de finales del siglo XX [Bacchus and Kabanza, 2000];[Bonet and Geffner, 1999];[Nau et al., 1999] que marcaron la pauta para las actuales representaciones de los dominios basados en estados. HSP, el planificador propuesto en [Bonet and Geffner, 1999] fue uno de los mas innovadores hasta ese entonces según lo acredita la competición internacional de planificación del año 1999, sin embargo con la propuesta de [Hoffmann and Nebel, 2001] y la competición previa a su publicación en el año 2000, quedó claro que era posible integrar tiempo y recursos en un dominio de planificación basado en estados y aún así mejorar el rendimiento del proceso de planificación.

Así pues, según los resultados de las competiciones internacionales de planificación de los años 2002⁶ y 2004^{7, 8}, los planificadores que eran mas eficientes y con mas capacidad de representación temporal y numérica, son los siguientes:

LPG-TD

Del inglés **L**ocal Search for **P**lanning **G**raphs - **T**imed initial literals and **D**erived predicates, LPG-td es un planificador basado en estados que trabaja con cantidades numéricas y restricciones temporales [Gerevini and Serina, 2002]. También permite expresar hechos que pueden llegar a ser falsos o verdaderos en ciertos períodos de tiempo conocidos de antemano, llamados “timed initial literals”; y predicados derivados que no pueden ser alterados por ninguna de las acciones disponibles en el plan si no que sus valores de verdad están determinados por un conjunto de reglas del dominio.

A pesar de que puede lidiar con restricciones numéricas y temporales, además de soportar literales iniciadas por el tiempo, el principal problema con este planificador en particular es que con el no es posible representar condiciones abstractas, ni solapamiento de actividades, entre otras características deseables en el planificador a elegir para los experimentos de esta memoria, dada la naturaleza del problema que hemos descrito en la sección 1.

METRIC-FF

Por su parte, el planificador Metric-FF (fast forward)[Hoffmann, 2003], extensión del FF, es un planificador independiente del dominio desarrollado por Jörg Hoffmann. Trabaja con variables de estado numéricas como las descritas en el lenguaje PDDL 2.1 nivel 2. Aunque este planificador es uno de los mas rápidos(si no es que el más rápido hasta el momento) e innovadores según los resultados de las competiciones de planificación(International Planning

⁶IPC2002: <http://planning.cis.strath.ac.uk/competition/>

⁷IPC2004: <http://ipc.icaps-conference.org/>

⁸Las IPC's de 2002 y 2004 son las que estaban vigentes al inicio del proyecto de la tesis, por lo que sus resultados son los que se han tomado en cuenta para obtener el estado del arte de los planificadores basados en estados.

Por otro lado, cabe mencionar que en las IPC's de los años 2006 y 2008 no hubo track temporal por lo que no se han tomado en cuenta sus resultados dentro de esta memoria. Sin embargo, en la sección de trabajo futuro se propone probar el algoritmo descrito en esta memoria con los planificadores que ganaron el track determinista de dichas IPC's.

Competitions), para problemas reales en los que están involucradas variables abstractas y tiempos límite, entre otros, no es una buena opción y es un claro ejemplo del por que se han descartado muchos planificadores parecidos a este cuyo objetivo principal es el de soportar la semántica mas actual del PDDL y generar planes lo mas rápido posible.

4.1.2. Jerárquico (HTN)

Los planificadores que utilizan redes jerárquicas de tareas o HTN (por sus siglas en inglés Hierarchical Tasks Networks), son una familia de planificadores que han mostrado que pueden ser muy poderosos en aplicaciones prácticas de dominios muy variados, por ejemplo, en dominios aeroespaciales[Muscettola et al., 1998], emergencias civiles[Bienkowski, 1995, Castillo et al., 2006] o campañas militares [Wilkins and Desimone, 1992].

Este paradigma está basado, al igual que el de planificadores basados en estados, en la representación STRIPS. Sin embargo, la diferencia radica en que los dominios de planificación HTN están diseñados en términos de jerarquía de actividades compuestas, que para ser representadas utilizan los siguientes elementos básicos:

- El nivel mas bajo de actividades, llamadas *acciones primitivas* u operadores primitivos, son actividades que no se pueden descomponer y que básicamente codifican cambios en el ambiente del problema. En el transcurso de esta memoria se representarán estos operadores primitivos como acciones durativas, las cuales serán codificados en el lenguaje PDDL 2.1[Long and Fox, 2003].
- Por otro lado, las acciones de alto nivel llamadas *tareas*, son acciones compuestas que pueden ser descompuestas en actividades de bajo nivel, otras acciones compuestas o una mezcla de ambos.
- Cada tarea puede ser descompuesta siguiendo diferentes esquemas, o *métodos* que permiten definir una serie de condiciones previas a la aplicación de las distintas secuencias de sub-actividades correspondientes. Estas sub-actividades pueden ser, a su vez, tareas que pueden tener alguna descomposición, o acciones primitivas.

Las tareas y sus descomposiciones, posiblemente múltiples, codifican las reglas que dependen

del dominio de aplicación para poder obtener un plan que únicamente puede estar compuesto por operadores primitivos.

A diferencia de los planificadores basados en estados, las metas de los planificadores jerárquicos no se especifican con una fórmula bien estructurada que puede llegar a ser válida para el plan a partir de un estado inicial. En vez de eso, las metas se describen por medio de un conjunto de tareas parcialmente ordenadas que necesitan ser llevadas a cabo.

Finalmente, el algoritmo principal de planificación HTN toma el conjunto de tareas que deben ejecutarse, explora el espacio de posibles descomposiciones que deben realizarse reemplazando una tarea dada por las actividades que la componen, hasta que el conjunto de tareas dadas es sustituido por un conjunto formado por acciones primitivas que componen el plan.

En la bibliografía existen muy pocos planificadores basados en redes jerárquicas de tareas, uno de los más conocidos es

SHOP2

Proviene del nombre en inglés Simple Hierarchical Ordered Planner 2 [Nau et al., 2003]. Es un planificador independiente del dominio basado en la descomposición ordenada de tareas.

Un planificador jerárquico que utiliza la descomposición ordenada de tareas es aquel que planifica las tareas en el mismo orden en que serán ejecutadas lo que reduce la complejidad de tener que lidiar con la incertidumbre acerca del estado.

A diferencia de otros planificadores HTN, SHOP2 puede hacer uso de axiomas, utilizar condiciones simbólicas y numéricas de forma indiferente y llamar a funciones externas. Además, permite la descomposición de los métodos en un conjunto de subtareas parcialmente ordenadas y la creación de planes que suministran subtareas a partir de tareas distintas.

Para los propósitos de la generación de diseños instruccionales, la ventaja más importante que este planificador ofrece es el uso de condiciones simbólicas, ya que ciertas condiciones para el despliegue de diversos objetos de aprendizaje de un curso a distancia suelen requerir

la representación de conceptos abstractos; el hecho de que las subtarear de los métodos puedan estar parcialmente ordenadas podría ser útil en ciertos casos también.

Sin embargo, no permite definir restricciones temporales que son realmente importantes dentro de un dominio lleno de actividades de aprendizaje, que deben incluirse o no en un plan dependiendo de las restricciones temporales definidas por el un estudiante.

Por otro lado, podemos encontrar también el planificador

SIADEx

Este planificador, creado inicialmente para ser la base de un Sistema Inteligente para la Ayuda a la Decisión en la EXtinción de incendios forestales, sigue el formalismo jerárquico HTN, “Hierarchical Task Network” [Schwalb and Dechter, 1991] basado en un proceso de búsqueda hacia adelante que supone el mantenimiento del estado del problema, al igual que SHOP2.

La principal diferencia de SIADEx con respecto a SHOP2 radica en que las acciones primitivas que se definen para él son completamente compatibles con el nivel 3 del lenguaje PDDL 2.2, con acciones temporizadas y capacidades de razonamiento numérico (fluents) [Castillo et al., 2006].

Además, los creadores de este planificador han definido una extensión de PDDL para representar tareas y métodos temporizados que heredan sus restricciones temporales a las acciones que los componen. También permite determinar plazos mínimos y máximos de tiempo para la ejecución de los planes, la definición de condiciones simbólicas y acciones primitivas con el mismo nombre.

4.1.3. *Comparativa y Selección de Planificadores*

Como se puede observar en las secciones anteriores, se han considerado para su estudio solo aquellos planificadores de los dos paradigmas de planificación que son open source, que sobresalen por su eficiencia y que abordan en mayor o menor grado algunas de las necesidades planteadas por el problema de generación de diseños instruccionales.

Se sabe que existen otros planificadores en el entorno comercial que cumplen con características específicas que superan o que pueden resultar interesantes para fines de esta investigación, sin embargo, no ha sido posible comprobar sus características y/o utilizarlos para pruebas concretas de del dominio de e-learning debido a cuestiones de licencias.

Sin embargo, esta sección pretende clarificar las diferencias entre unos y otros planificadores, además de exponer también las ventajas que cada uno supondría si fuera utilizado para apoyar en la solución del problema en cuestión.

Por un lado tenemos LPG-td que permite planificar teniendo en cuenta cantidades numéricas y restricciones temporales, también permite expresar “timed initial literals” y predicados derivados. El principal problema con este planificador en particular es que con el no es posible representar condiciones abstractas, ni solapamiento de actividades.

Por otro lado se encuentra Metric-FF que utiliza variables de estado numéricas y que, hasta 2006, era uno de los planificadores más rápidos e innovadores. Sin embargo, cuando se le presentan problemas reales en los que están involucradas variables abstractas y tiempos límite, entre otros, no es una buena opción ya que su eficacia disminuye sustancialmente o, simplemente, no le es posible manejar estas características.

También tenemos a SHOP2 un planificador jerárquico que puede hacer uso de axiomas, utilizar “condiciones simbólicas” y numéricas de forma indiferente y llamar a funciones externas. Además, permite la descomposición de los métodos en un conjunto de subtareas parcialmente ordenadas y la creación de planes que suministran subtareas a partir de tareas distintas. Sin embargo, no permite definir restricciones temporales.

SIADEx es compatible con el nivel 3 del lenguaje PDDL 2.2, cuenta con acciones temporizadas y capacidades de razonamiento numérico, además de una extensión de PDDL para representar tareas y métodos temporizados que heredan sus restricciones temporales a las acciones que los componen. Por si fuera poco, se ha continuado su desarrollo en la empresa IACTIVE[S.L., 2010], la cual mantiene estrechas relaciones con el personal del departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada (lugar de desarrollo de esta memoria), y gracias a ello se tiene la posibilidad de conocer las mejoras

a corto plazo que se vayan originando sobre el y que sean útiles para este trabajo.

Las características principales que pueden ser útiles en un planificador que pretenda aportar novedad en la generación de diseños instruccionales son las siguientes:

1. Capacidad y rapidez para resolver problemas en dominios reales, no solo problemas académicos. Ya que el dominio de la generación de diseños instruccionales es muy complejo y real.
2. Capacidad para soportar el uso de restricciones temporales, es decir, duraciones de las acciones, paralelización de las mismas (solapamiento), plazos límites de tiempo para el cumplimiento de ciertas metas; y en caso de planificadores HTN, duraciones y solapamiento también entre tareas. Ya que el tiempo para llevar a cabo las actividades
3. Permisión del uso de axiomas o predicados derivados que nos permitan definir algunas de las reglas pedagógicas a seguir.
4. Capacidad para hacer llamadas a funciones externas que se encuentren programadas en otro lenguaje de planificación ajeno a PDDL. Esto sería muy útil en caso hipotético del cálculo de calificaciones o de cambio en los valores de funciones en “tiempo real”.
5. Capacidad de enriquecer las acciones con información externa. En teoría todas las acciones podrían enriquecerse con información nueva a través de los predicados incluidos en sus precondiciones, pero son muy pocos los planificadores que permiten que esta, u otra información exclusiva de la acción, forme parte de la salida del proceso de planificación, cosa que facilitaría sobremanera la generación directa del diseño instruccional en el formato del estándar de un diseño de aprendizaje, sin tener que pasar por un proceso intermedio.

En la Tabla 2 se intenta resumir algunas de las características descritas anteriormente y proyectarlas a los cuatro planificadores cuyas características han sido resumidas en los párrafos anteriores.

En base al estudio anterior se ha decidido utilizar un planificador de cada paradigma, LPG-td y SIADEX respectivamente, para integrarlos dentro de esta propuesta. Estos plani-

CARACTERÍSTICAS	PLANIFICADORES			
	<i>LPG-td</i>	<i>Metric-FF</i>	<i>SIADEx</i>	<i>SHOP2</i>
<i>Rapidez para resolver problemas reales.</i>	Media	Poca	Mucha	Mucha
<i>Maneja restricciones temporales.</i>	Si	No	Si, incluso entre tareas y plazos en las metas.	No
<i>Permite el uso de axiomas.</i>	Si	Si	Si	Si
<i>Permite inclusión de funciones externas.</i>	Si	No	Si	Si
<i>Permite enriquecer las acciones con información externa.</i>	No	No	Si, y mostrar dicha información en el plan resultante.	Si

Tabla 2. Comparación de las características de los planificadores estudiados que son deseables para la generación de diseños instruccionales.

planificadores pueden ser utilizados de la forma tradicional, es decir, proporcionando dos ficheros que describan el dominio y el problema de planificación respectivamente y ejecutándolos para que, en base a esta información, generen un plan a seguir para que un estudiante siga la secuencia adaptada de actividades de aprendizaje necesaria para adquirir los conocimientos de un curso y, por tanto, son la vía más estándar para integrarlos en cualquier herramienta genérica que pueda surgir como resultado de este trabajo de investigación.

4.2. Técnicas de Planificación para Entornos Dinámicos

Es interesante saber que los diferentes paradigmas de planificación utilizados por los planificadores descritos en la sección anterior no solo pueden ser utilizados de la manera tradicional, es decir, obteniendo información de ficheros de dominio y de problema previamente definidos y estáticos, para posteriormente generar un único plan que cumpla con la meta definida en el problema; si no que también existen una gran variedad de técnicas de planificación que contemplan el uso de los planificadores de una manera diferente a la habitual y que pudieran ser útiles en caso de que, por ejemplo, la información inicial de uno o varios estudiantes cambiara en diferentes ocasiones a lo largo del curso, que existieran evaluacio-

nes intermedias del curso que repercutieran en la información que se debiera proporcionar al estudiante en los temas posteriores a la evaluación, entre otras situaciones que no solo involucran cierto grado de incertidumbre a lo largo del desarrollo del plan, si no también una constante interacción con el LMS.

Por estas razones es que en los siguientes apartados se presentarán las técnicas de planificación mas comunes en la bibliografía académica y su utilidad para solucionar algunas partes del problema planteado en la memoria.

4.2.1. *Re-Planificación*

La re-planificación es un técnica que aprovecha la observabilidad del entorno para actuar ante los cambios que se produzcan en el mismo.

El proceso de re-planificación se puede llevar a cabo de dos formas, mediante la reparación o adaptación del plan a las nuevas circunstancias o mediante la planificación desde cero de un nuevo plan.

La reparación o adaptación del plan[Fox et al., 2006, Knoblock, 1995] se lleva a cabo cuando se parte de la suposición de que la mayor parte del plan actualmente calculado es válido, por lo cual es una pérdida de tiempo calcularlo de nuevo. Esta técnica se utiliza generalmente para planes muy largos con baja probabilidad de fallo y que son generados a priori por planificadores basados en estados.

La re-planificación desde cero se lleva a cabo cuando el coste de reparar supera al de planificar desde cero. El coste suele medirse en base al número de actividades que se añadirán al plan para que sea válido o el tiempo que tarda el algoritmo en adaptar. Por lo tanto, si el planificador genera planes en un tiempo notablemente inferior al que tarda el algoritmo en adaptar el plan que ya no funciona, o agrega demasiadas acciones para reparar el plan, entonces se prefiere el método de re-planificar desde cero. Esto suele ocurrir cuando para la planificación se utilizan planificadores HTN que suelen ser sustancialmente mas rápidos en el cálculo de planes largos que los basados en estados.

En el entorno de scheduling de actividades de aprendizaje esta técnica podría resultar

muy útil dados los constantes cambios que se producen en el perfil del estudiante; cambios que, al no ser sustanciales, no debieran afectar radicalmente el plan generado a priori y, por tanto, permitirían la re-planificación utilizando cualquiera de los dos métodos mencionados anteriormente.

4.2.2. *Planificación Contingente o Condicional*

La planificación contingente es un técnica que intenta establecer de antemano ramas de actuación alternativas a tomar en caso de contingencia[Onder and Pollack, 1999]. Normalmente esta técnica se aplica introduciendo acciones que permitan parar en un punto determinado del plan para reevaluar el estado actual del mundo y/u obtener mas información acerca del mismo[Rintanen, 1999]. De acuerdo al nuevo estado detectado, se escoge la rama mas apropiada a seguir. Si no pudieran existir este tipo de acciones, se escoge de antemano la rama con más probabilidad de éxito. De ésta manera se logra conseguir un plan condicional que, aunque a un alto coste computacional, permite predecir a priori todas las posibles contingencias.

En el caso de la planificación de secuencias de aprendizaje, existe un entorno de despliegue de dichas secuencias que permite especificar, no solo las secuencias, si no también las condiciones pedagógicas para la ejecución de las distintas ramificaciones de las mismas. El problema es que dicho “entorno” necesita conocer a priori todas y cada una de las ramas del plan, por lo que la técnica de planificación condicional, adaptada a las necesidades específicas del problema de generación de diseños instruccionales podría resultar de gran ayuda.

4.2.3. *Planificación Continua*

En la planificación continua (del inglés continual planning) se intenta no planificar con demasiada antelación, ya que se asume que el plan puede fallar en cualquier momento. Existen varias alternativas para implementar esta técnica. La primera consiste en planificar y ejecutar de forma alternada, de manera que la ejecución se para cuando es necesario planificar la siguiente “rama” del plan[desJardins et al., 1999, Miksch and Seyfang, 2000]. La segunda requiere de una ejecución paralela de la planificación y la ejecución del plan. Cuando existe un fallo en la ejecución del plan, se continúa con la ejecución de aquellas

partes del plan que no se vean afectadas por el fallo y a la vez se repara aquella en la que se ha manifestado el error [Ambite and Knoblock, 1997, Myers, 1998].

La primera alternativa para implementar esta técnica de planificación resulta muy útil en entornos de e-learning en donde se puede interactuar con el entorno de despliegue de las secuencias de aprendizaje adaptadas a la necesidades del estudiante, ya que permite “parar” la planificación de la secuencia de un curso en cada evaluación intermedia aplicada al estudiante o en cada actividad grupal y planificar la siguiente rama del plan en base a los resultados recabados de dichas actividades.

De esta lista de análisis de técnicas de planificación útiles para los propósitos de la presente memoria se han descartado algunas como, por ejemplo:

- *Razonamiento Reactivo* [Georgeff and Lansky, 1987, Schoppers, 1987] que entrelaza planificación y ejecución basándose en respuestas cortas a estímulos del entorno, lo que podría traducirse a seleccionar la siguiente actividad de un plan en base a los resultados obtenidos de la ejecución de la actividad anterior. Pero este esquema no es compatible con un entorno de aprendizaje a distancia por dos razones. Primera, que la mayoría de las actividades no tienen embebido un método de evaluación de las mismas, método que a su vez las relacione con el resto de las actividades del curso. Segundo, que las llamadas constantes a los métodos de planificación podrían reducir la eficiencia inherente a dicho proceso, dependiendo del proceso elegido para comunicar la plataforma de e-learning y el planificador.
- *Iniciativa Mixta* [Allen et al., 1999, Ferguson et al., 1996, Myers, 1996, Smith et al., 1996, Veloso, 1996] que permite al usuario colaborar con el planificador de manera que éste pueda decidir el rumbo a seguir en caso de que no exista suficiente información del entorno como para hacerlo. En algunos casos, los enfoques de iniciativa mixta inclusive proporcionan al usuario la opción de definir precondiciones sobre algunas acciones en tiempo de ejecución. Este enfoque no es adecuado para implementarse actualmente sobre la plataforma de un sistema gestor de aprendizaje ya que implica el diseñar una interfaz de despliegue de planes con un alto grado de interactividad, cosa que actualmente no permiten la mayor parte de los

LMS's.

- *Mezcla de Planes* [Foulser et al., 1992, Gravot and Alami, 2001, Yang et al., 1992] es una técnica de planificación que se basa en la identificación de sub planes que pueden ser ejecutados de forma paralela para alcanzar la meta principal de una manera mas eficiente. Esta técnica podría considerarse interesante para que un estudiante con capacidades extraordinarias pudiera realizar a la vez las actividades de diferentes objetivos de un curso y así terminarlo mas rápidamente. Sin embargo, ésta técnica no es útil para el problema de generación de diseños instruccionales por dos razones principales. La primera es que los sub planes de un curso suelen estar perfectamente delimitados, teniendo en cuenta que hay un subconjunto de actividades “exclusivas” para comprender un objetivo o tema específico del mismo. Por lo tanto no es necesario identificar sub planes, es decir, llevar a cabo el primer paso de la técnica. La segunda razón es que este enfoque no funciona para la generalidad de estudiantes que están matriculados en curso a distancia, ya que el perfil de la mayoría de ellos implica una administración limitada del tiempo.

A través de herramientas como SCORM y los estándares que implementa, la especificación de diseños de aprendizaje y del perfil del estudiante, los sistemas gestores de aprendizaje, así como los planificadores y las técnicas de planificación para entornos dinámicos que se han descrito, estudiado y seleccionado, será posible, no solo proponer una solución viable al problema planteado, si no también llevar a la práctica dicha solución en el entorno de uno o varios cursos a distancia reales.

Es por eso que en el próximo capítulo se detallará la forma en que algunas de las especificaciones anteriormente descritas deben ser definidas, de manera que la información que se proporcione en dicha definición, pueda ser reutilizada y/o generada posteriormente por un *planificador inteligente*.

CAPÍTULO 3

Procesos de Ingeniería del Conocimiento

Como se ha dicho en la sección anterior, la definición de las entradas y la interpretación de las salidas de un planificador es una tarea que actualmente se lleva a cabo por expertos en planificación inteligente que, después de ser capacitados en el dominio específico sobre el cual se requiera planificar - en este caso un dominio de e-learning -, deben dedicarse a extraer de todos los medios (electrónicos y humanos) posibles la información que se debe suministrar al planificador para que genere un plan, que posteriormente deberán traducir a un diseño instruccional en el formato que les sea requerido.

Como se puede apreciar en la descripción anterior del trabajo que debe llevar a cabo un experto en planificación, el proceso de extracción de conocimiento, al no tener fuentes concretas y bien establecidas, suele llegar a ser muy lento y costoso.

En esta sección se propone un método claro, sencillo y específico de estandarizar la descripción de los datos principales sobre los cuales se llevaría a cabo el proceso de ingeniería de conocimiento anteriormente mencionado. El método consiste, a grandes rasgos, en definir sobre algunas de las principales especificaciones de e-learning, descritas en la sección 3 del

capítulo 2, un conjunto de metadatos (etiquetas en XML) básicos y suficientes como para que los expertos de planificación - o un programa de ordenador, como se propone en el siguiente capítulo - puedan llevar a cabo sobre ellos el proceso de ingeniería de conocimiento que les corresponde; permitiendo así contar con información centralizada de gran parte de la información de un curso y sus estudiantes.

La labor de captura de esta información puede ejecutarse de forma simultánea, y utilizando las herramientas visuales e intuitivas dispuestas para ello, por dos actores principales: los tutores y/o generadores de contenido y los estudiantes.

Mientras que los tutores y generadores de contenidos de aprendizaje se encargarían del diseño estructural del curso, de los objetivos de aprendizaje del mismo y del etiquetado de los contenidos necesarios para alcanzar dichos objetivos. Los estudiantes que tomarían dicho curso habrían de cumplimentar los datos de su perfil, llegando en ocasiones a contestar cuestionarios que permitan obtener información implícita del mismo, como es el caso de los cuestionarios para definir estilos de aprendizaje.

En las siguientes secciones se detalla el método propuesto. Cada sección se refiere a una parte del curso de e-learning en particular que generalmente es tratada por una especificación determinada, aunque es posible ver excepciones a esta regla al momento de definir de forma inicial los objetivos y la estructura del curso, así como los apoyos pedagógicos que se utilizarán para enseñarlo. Además, en el anexo A se incluye un manual del usuario que pretende servir de documentación para profesores y alumnos que deseen llevar a cabo este proceso de definición de datos de una manera adecuada y simplificada.

1. Los Objetivos y la Estructura del Curso

Dentro de un ambiente de aprendizaje tradicional, se considera que es labor del docente definir el diseño estructural del curso, así como los objetivos de aprendizaje del mismo. Sin embargo, cuando se trata de definir dicha información para un curso en línea, habría de apoyarse en un experto en generación de contenidos para que éste le ayudara a reflejar esta estructura y objetivos utilizando todos y cada uno de los objetos de aprendizaje que se diseñen para el curso.

Una vez que se han creado todos los objetos de aprendizaje del curso, lo primero que será necesario llevar a cabo es la definición de la estructura del curso. La figura 3 puede servir de ejemplo para este propósito.

Generalmente un curso tiene un título u objetivo principal, que en este caso es *Discrete Maths* y al cual llamaremos *Tarea Compuesta Principal* para empezar a relacionarlo con la terminología de la planificación inteligente. Existen también *tareas compuestas principales* que no son parte del objetivo principal del curso, si no que pertenecen a cursos (ya sea como objetivo principal o intermedios) o repositorios distintos del que estamos trabajando¹ y que podríamos requerir a lo largo del curso, como el ejemplo del objeto *Logic and Sets* marcado totalmente de negro en la figura 3 y que se explicará en las secciones siguientes.

Después se suelen definir otros objetivos intermedios mejor conocidos como temas o subtemas, que en la figura 3 están representados por el resto de los recuadros marcados en gris obscuro y a los cuales llamaremos *Tareas Compuestas*. Todos los tipos de tareas mencionados anteriormente se conocen también como objetos de aprendizaje abstractos, ya que no cuentan con un fichero físico visible relacionado a ellos, pero su nombre, descripción y relaciones como parte de la estructura del curso si deben ser descritos.

Finalmente se diseñan los objetos de aprendizaje concretos, que están representados por el color gris claro y son tareas a ejecutar por el estudiante para aprender el objetivo específico de cada tema o subtema de “penúltimo nivel”. Estos objetos de aprendizaje deben presentarse

¹El objetivo de esta memoria no es encontrar las mejores técnicas para extraer los mejores objetos de aprendizaje de diferentes repositorios al rededor del mundo, sin embargo, se hace mención y un uso hipotético de los mismos pues son una tecnología que promete emerger con fuerza en un futuro próximo.

de una forma tan variada como sea posible para cada concepto u objetivo específico y, una vez creados por los diseñadores de contenidos, podrán ser incluidos en la jerarquía del curso. A estos objetos de aprendizaje les llamaremos también *Acciones Primitivas*.

Es así como, dado un conjunto de objetos de aprendizaje \mathcal{O} , es posible etiquetar cada uno de ellos bajo el estándar IEEE-LOM[ANSI/IEEE, 2007]. El estándar define un conjunto de metadatos agrupados en distintas categorías, como se puede ver en la figura 2, con los cuales es posible describir las características principales del objeto.

Estos metadatos pueden ser editados gracias a algunas herramientas de software como por ejemplo RELOAD[Liber, 2007] o eXeLearning[eXe eXeLearning, 2010]. Para este caso particular se recomienda utilizar la herramienta RELOAD.

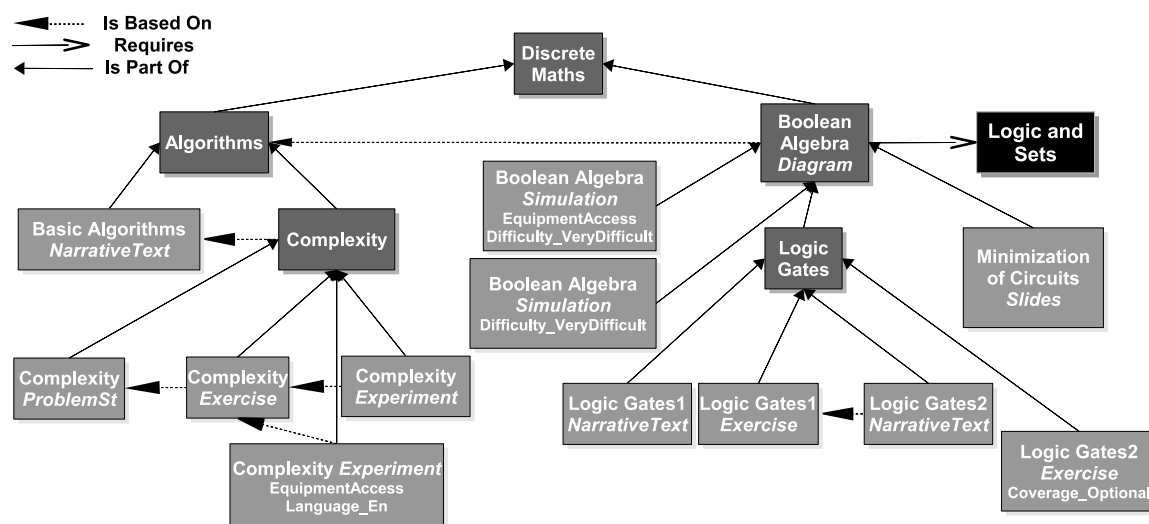


Figura 3. Diagrama del curso de ejemplo de matemáticas discretas en el cual se describen las relaciones, tanto de orden como jerárquicas, entre sus diversos componentes. Las figuras de un color gris mas claro corresponden a los objetos de aprendizaje que incluyen: en cursivas, el tipo de recurso de aprendizaje y, en letra mas pequeña, algunos de los metadatos tipo atributo que los caracterizan.

Al crear el curso dentro de RELOAD, las *tareas compuestas principales* se denominan **organizations** y el resto de los temas, subtemas y objetos de aprendizaje se llaman **items**. En la figura 4 podemos ver el curso de matemáticas discretas cuyo esquema principal hemos presentado en la figura 3. Se ha utilizado RELOAD para estructurar el curso de una forma gráfica que integra automáticamente, dentro del paquete SCORM representado por un IMS-

CP[IMS-GLC, 2003], todos los estándares que se requieren.

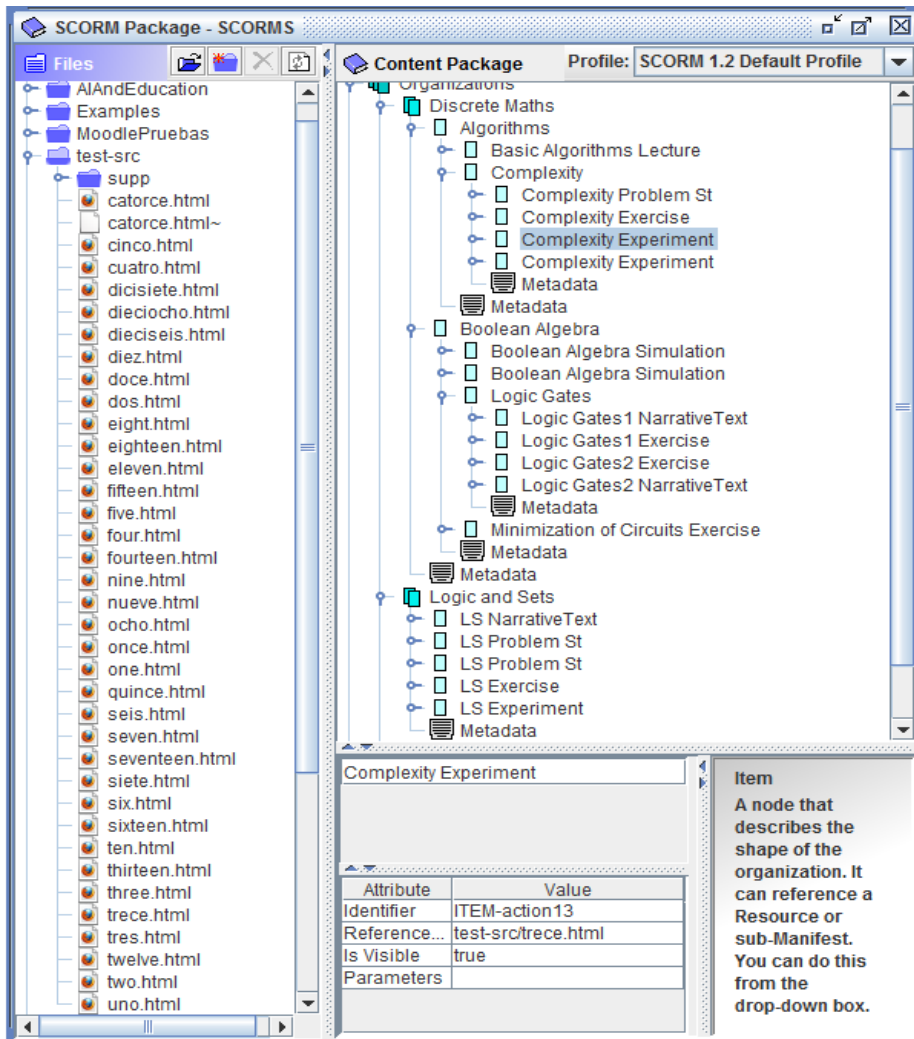


Figura 4. Curso de Matemáticas Discretas Definido en la herramienta RELOAD. En la parte izquierda se encuentran los archivos de definición del paquete; abajo se describe el item seleccionado, su nombre, identificador y el recurso asociado (que en este caso es nulo dado que *Complexity* es una *tarea compuesta*); en la parte principal se puede ver la representación jerárquica del curso.

Dentro del estándar LOM, tanto el título del curso como los temas, subtemas y objetos de aprendizaje son llamados **recursos**; también dentro de este estándar existe la categoría *Relation* que servirá para asignar los metadatos relacionales a cada uno de los elementos del curso.

Utilizando otras categorías del mismo estándar será posible etiquetar cada elemento del

curso según una serie de metadatos tipo atributo. Estos metadatos son esenciales para facilitar el posterior trabajo de extracción de conocimiento que deben llevar a cabo los expertos en planificación pero, como es posible observar, existen muchos otros metadatos que pudieran ser utilizados para describir atributos de los objetos que serían útiles a otros propósitos.

Debido a que el etiquetado del curso resulta ser también una tarea relativamente compleja, se ha intentado reducir esa labor al mínimo y facilitarla organizando la información en los conjuntos de metadatos que se explicarán a continuación. Para seguir las recomendaciones siguientes es posible basarse en la figura 2 y después introducir los datos tal y como se muestra en la figura 6 para el ejemplo del objeto de aprendizaje *Complexity Experiment*.

1.1. Metadatos Relacionales

Los metadatos relacionales permiten definir tanto las relaciones de orden como las jerárquicas que suelen establecerse implícitamente durante el proceso de diseño de un curso.

En la categoría *Relation* del editor de metadatos de RELOAD debemos especificar:

- El tipo de relación, eligiendo la relación correspondiente en el campo *Kind*, y
- El recurso asociado, rellenando el campo *Identifier* de la subcategoría *Resource* con el identificador único del objeto que asociemos al que actualmente estamos etiquetando.

Es posible, y en algunos casos hasta necesario, agregar más de una categoría tipo *Relation* a cualquier recurso.

En los siguientes puntos se explicará el significado de cada uno de los tipos de relaciones, es decir, los valores del campo *Kind* posibles y necesarios dentro del contexto de trabajo actual. Para ello se dividen estos valores en *Valores relacionales de orden* y *Valores relacionales de jerarquía*.

1.1.1. De Orden

Estos valores permiten definir un orden lineal entre las acciones o tareas internas o externas necesarias para la satisfacción de los objetivos del curso.

IsBasedOn. Proporciona relaciones de orden entre objetos de aprendizaje o temas, ya que un tema “*x*” puede necesitar de las bases que proporciona un objeto de aprendizaje “*y*” para poder cumplir con los objetivos de aprendizaje específicos del mismo.

Por ejemplo, en la figura 3 el tema *Complexity* **IsBasedOn** el objeto *Basic Algorithms Narrative Text*; esto quiere decir que para poder llevar a cabo todas las acciones que forman parte de *Complexity*, será necesario hacer primero la actividad *Basic Algorithms Narrative Text*, dado que es preciso conocer primero qué es un algoritmo para poder después calcular su complejidad.

Requires. Reporta dependencias entre contenidos. Se utiliza normalmente cuando un tema o subtema necesita de las bases de otro tema u objeto que proviene de un curso o repositorio externo al definido para el curso sobre el cual se está trabajando actualmente. Como por ejemplo, el objeto *Boolean Algebra Diagram* en la figura 3 que requiere, a través del metadato de orden **Requires**, el tema *Logic and Sets* cuya descripción se hace en la figura 5 dado que es un tema que debería haberse visto y aprobado en un curso anterior.

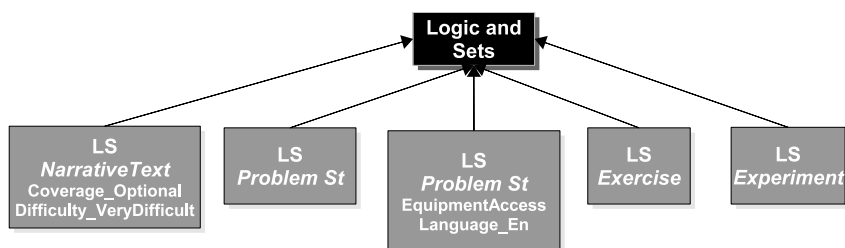


Figura 5. Diagrama del objeto *Logic and Sets* que es requerido por el objeto *Boolean Algebra Diagram* de la figura 3. En esta figura se especifica la estructura jerárquica del objeto, la cual es muy sencilla, y los objetos de aprendizaje que lo componen.

1.1.2. De Jerarquía

IsPartOf. Este metadato describe una estructura jerárquica entre los elementos del curso, estructura que es semejante a la de un árbol teniendo a la meta, tema u objetivo principal del curso como raíz. Así, los temas forman parte del tema principal del curso por medio del metadato **IsPartOf**, los subtemas son parte de los temas y así sucesivamente hasta llegar a los objetos de aprendizaje que no tienen ningún objeto como parte de los mismos ya que son el último nivel de cada rama particular del árbol.

En el ejemplo de la figura 6 estamos rellenando los metadatos del objeto *Complexity Experiment* y ya se ha agregado la categoría tipo Relation, en la subcategoría Kind hemos puesto el Value **IsPartOf** y para la subcategoría Resource hemos puesto el Identificador del objeto *Complexity* dentro del campo Identifier. No se alcanza a ver, pero se ha agregado una categoría Relation mas; para su subcategoría Kind y consecuente campo Value se utilizó el valor **IsBasedOn** y en el campo Identifier de la subcategoría Resource colocamos el identificador del objeto *Complexity Exercise* tomando en cuenta las relaciones de orden que se observan en la figura 3.

1.2. Metadatos Tipo Atributos

Dentro del estándar LOM existen una gran variedad de metadatos del tipo atributo que se encargan de describir las características principales de cada elemento del curso. Para los propósitos de esta memoria, básicamente serán necesarios los que se describen en los apartados siguientes y que se dividen para una mejor comprensión en las categorías del estándar a las que pertenecen, de manera que sea mas fácil localizarlos dentro del editor de metadatos.

Algunos metadatos de tipo atributo ayudan a definir precondiciones para los objetos de aprendizaje del curso durante la generación del dominio de planificación. Este tipo de metadatos se encuentran representados con letras pequeñas en las figuras 3 y 5 después del título y el tipo de recurso de aprendizaje (en cursivas) de cada uno de los objetos de aprendizaje.

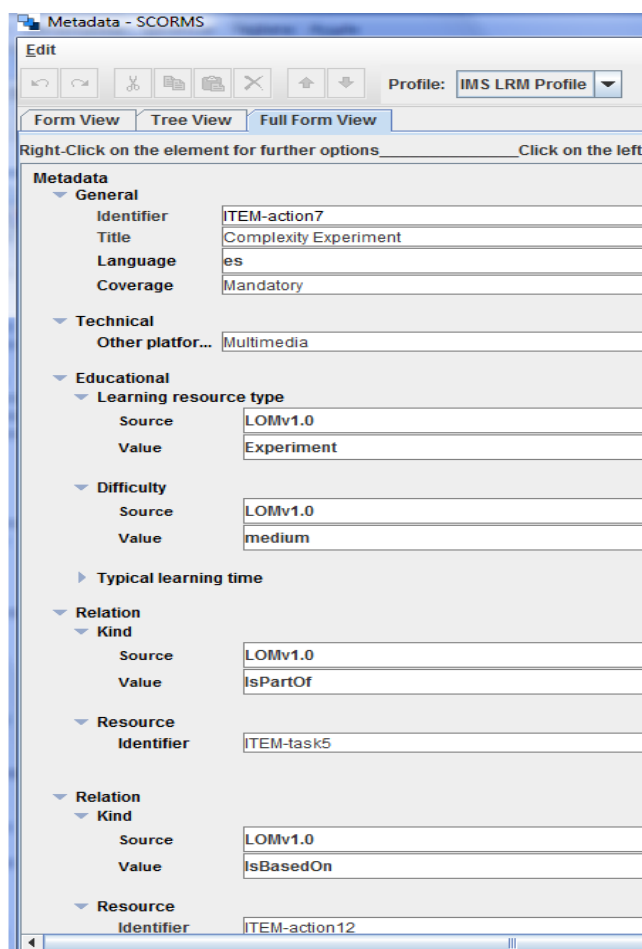


Figura 6. Etiquetado de metadatos del recurso *Complexity Experiment* utilizando la herramienta RELOAD.

Por otro lado, es importante resaltar que para el título del curso, así como temas y subtemas del mismo, no será necesario definir todos estos atributos, así como no es necesario definir metadatos relacionales para los objetivos principales del curso o los temas que representen dependencias externas del curso, lo cual simplifica en gran medida el trabajo de etiquetado que corresponde al tutor. Estas excepciones se especificarán en cada uno de los siguientes apartados para su mayor comprensión.

1.2.1. *Generales*

Los metadatos generales se cumplimentarán en la sección de metadatos de RELOAD que se muestra en la figura 6 y son los siguientes:

Identifier. Generalmente se refiere al identificador único que suelen generar de forma predeterminada las herramientas de edición, que suele ser un alfanumérico largo que va precedido por la palabra ÍTEM. Este identificador se encuentra junto a la descripción del título del recurso y solo será cuestión de copiar dicha cadena y pegarla en el campo Identifier de los metadatos correspondientes al recurso con el que se esté trabajando. Se sugiere cambiar el alfanumérico que sigue a la palabra ITEM por el sufijo *action* para los objetos “hoja” del árbol jerárquico del curso y *task* para los objetos del resto de los niveles.

Title. El título se define según el nombre del objetivo específico que se vaya a reforzar con dicha acción. Debe especificarse, no solo en los campos principales del recurso en la ventana principal de RELOAD (fig. 4), si no también en el metadato que recibe dicho nombre (fig. 6).

Language. Es el metadato en el cual se deberá especificar que idioma se requiere para poder entender y trabajar con el objeto de aprendizaje que se esté etiquetando. Por ejemplo, si el idioma es español su valor será **es**, si es inglés su valor será **en**, etc.

El metadato *Language* solo se definirá para los objetos de aprendizaje.

Coverage. En el se definirá el tipo de cobertura del tendrá el recurso. Es decir, si debe ser cubierto obligatoriamente o no dentro del curso. Sus valores pueden ser **Mandatory** u **Optional**.

Si el metadato *Coverage* tiene el valor de **Optional** para algún tema o subtema, éste valor será heredado por los objetos o subtemas que estén relacionados al mismo por medio de **IsPartOf** a cualquier nivel de profundidad.

1.2.2. *Técnicos*

Los metadatos técnicos, entre otras cosas, se refieren a las necesidades específicas de hardware y software que puede tener un objeto de aprendizaje.

Para simplificar su etiquetado, se utilizará solo uno de los metadatos que describen esta sección.

Other Platform Requirements. Si, por ejemplo, un objeto utiliza un recurso de flash, el paquete de Microsoft Office, altavoces, etc. su valor será **Multimedia** y en caso de que no requiera más que un monitor con las capacidades mínimas, un ratón y un teclado, su valor será **None**.

La figura 3 hace referencia al valor **Multimedia** de este metadato con la propiedad **EquipmentAccess** representada en letras pequeñas en uno de los recursos **Boolean Algebra Simulation**.

Este atributo será asignado únicamente a los objetos de aprendizaje.

1.2.3. *Educativos*

Todos los atributos pertenecientes a esta categoría serán necesarios solo en caso de estar etiquetando un objeto de aprendizaje del tipo acción primitiva.

Learning Resource Type. Es un metadato que describe el tipo de recurso de aprendizaje que se esté etiquetando. El valor de este campo debe de ser alguno de los siguientes: **Problem Statement**, **Simulation**, **Experiment**, **Exercise**, **Example**, **Table**, **Slide**, **Diagram** y **Narrative Text** cuyas traducciones literales son descripción de un problema, simulación, experimento, ejercicio y texto narrativo.

Difficulty. La dificultad se refiere al nivel de rendimiento que debe tener un alumno para ser capaz de cumplir con el objetivo cubierto por este objeto. Sus valores son: **VeryEasy**, **Easy**, **Medium**, **Difficult** y **VeryDifficult**. Si un alumno tuviera un rendimiento menor al requerido, el objeto que se estuviese evaluando no debería ser parte de su recorrido de aprendizaje.

Typical Learning Time. Se refiere al tiempo típico de aprendizaje que toma el llevar a cabo la acción que se esté etiquetando. Dicho tiempo debe ser el mínimo suficiente, de acuerdo a la experiencia empírica del tutor, como para cumplir cubrir el objetivo específico de aprendizaje que satisface el objeto y debe asignarse en minutos como unidades principales.

Una vez etiquetadas todas las actividades con este valor, será necesario sumar los valores de todas ellas para asignar un typical learning time global con formato *[máximo de minutos, mínimo de minutos]* a la organization a la que pertenezcan.

2. El Perfil del Estudiante

Dentro de los Sistemas Gestores de Aprendizaje existen una gran variedad de opciones que pueden ser utilizadas por el estudiante para describir su perfil.

La opción mas viable, a futuro, es rellenar ciertos metadatos de la especificación IMS-LIP. Pero para ello es preciso conocer a grandes rasgos la estructura general de esta especificación estándar que tiene once clases principales:

- *Identification*: se refiere a los datos biográficos y demográficos relevantes para el aprendizaje.
- *Goal*: detalla los objetivos de aprendizaje del alumno, su carrera u otros objetivos y aspiraciones.
- *QCL*: del inglés Qualifications, Certifications and Licences, se refiere a las cualificaciones, certificaciones y licencias que ha obtenido el estudiante y que han sido garantizadas por autoridades reconocidas.
- *Activity*: se refiere a cualquier actividad de aprendizaje que se haya completado o no. Incluye educación formal e informal, entrenamiento, experiencia laboral, servicios sociales o militares, etc.
- *Interest*: Información que describe las aficiones y actividades recreativas del estudiante.
- *Competency*: son las habilidades, conocimientos y capacidades intelectuales del estudiante.
- *Accessibility*: son las opciones de acceso general a la información de un ambiente virtual de aprendizaje, como por ejemplo, preferencias de aprendizaje cognitivas, físicas y tecnológicas, idiomas, discapacidades, etc.

- *Transcript*: es un registro que utilizado para proveer información histórica obtenida de registros académicos institucionales
- *Affiliation*: es la membresía a organizaciones profesionales a las que el estudiante pertenece.
- *Securitykey*: es el conjunto de claves o contraseñas asignadas a ese usuario para las transacciones con otros sistemas de información de estudiantes y otros servicios.
- *Relationship*: codifica las relaciones entre los componentes anteriormente mencionados.
- *Contenttype*: clase abstracta que permite contener información detallada sobre algunos elementos de las clases anteriores.

Sin embargo, dado que actualmente los sistemas gestores de aprendizaje no suelen integrar este estándar entre sus opciones y a que en cualquier caso la captura de la información del estudiante se llevará a cabo a través de las interfaces de las que disponen los LMS's para posteriormente almacenarla en el IMS-LIP, ha sido preciso buscar otras formas de representar esta información dentro de un LMS sin la necesidad de utilizar directamente los metadatos del estándar IMS-LIP.

Los LMS's suelen tener opciones tanto para incluir nuevos campos dentro del perfil del estudiante, como para establecerlos como obligatorios. Este tipo de opciones son las que se utilizarán para solicitar al estudiante la información necesaria para la personalización de sus cursos; por que, cabe recordar, que puede estar matriculado en uno o varios cursos dentro del mismo LMS.

Un ejemplo de la integración de campos obligatorios para la personalización de cursos dentro de un LMS se puede encontrar en la figura 7 donde se han integrado dichos campos dentro del LMS Moodle.

En los párrafos siguientes se describirán cada uno de los campos que le serán requeridos al alumno, así como su representación dentro del estándar IMS-LIP y sus posibles valores. Toda esta información servirá para que estos campos puedan ser definidos adecuadamente por el

administrador del LMS y cumplimentados correctamente dentro del perfil del estudiante, para que posteriormente puedan extraerse fácilmente por los expertos en planificación.

The image shows a form titled "Adaptation Fields" with the following fields and values:

Field Name	Value
English Level	Low
Learning Style	Reflexive
Performance Level	High
Equipment	NoOne
Available Time	Much
Previous Courses Level	40

Figura 7. Presentación de los campos que se le muestran al estudiante para que cumplimente su perfil. Éste modelo está diseñado en el LMS Moodle.

Short Name. Es el nombre de usuario que se elige el alumno al darse de alta en el sistema y que suele ser único dentro del ambiente del sistema gestor de aprendizaje. De esta manera será posible saber que no se recibirá ningún nombre de alumno repetido. El metadato, dentro de IMS-LIP donde se guarda dicha información se encuentra en la ruta: `identification → name → contenttype → referential → indexid`.

Cuando se llena el formulario de registro dentro del sistema, también se solicita el nombre completo del alumnos, cuya información (además de en las bases de datos del sistema) se guarda en la ruta `identification → formname → text`.

English Level. Este campo es específico del problema planteado, pero podría requerirse el nivel de cuantos idiomas sean soportados por los objetos de aprendizaje que utilice la plataforma. *English level* se refiere al nivel de inglés que tiene el alumno; los valores que puede tomar son: `Excelent`, `Acceptable`, `Low` y `Poor`, haciendo referencia a excelente, aceptable, bajo y pobre en español.

Para definir el idioma sobre el cual se requiere un cierto nivel en IMS-LIP se utiliza el metadato `accessibility → language → typename → typevalue → lang-value`; y el nivel del mismo se guarda en La ruta en XML en la que podemos definirlo es la siguiente: `accessibility → language → proficiency` que debe contener un atributo llamado `profcode` cuyo valor por default será igual a `Read`.

Equipment. Sirve para especificar si el alumno cuenta con requerimientos especiales de hardware o software requeridos en el curso. Si cuenta con ellos, éste campo debe tomar el valor `Multimedia`, en caso de que no los tenga disponibles tomará el valor `None`.

Dentro de la clase accesibilidad del IMS-LIP se deberá especificar un campo con el valor estático de `Equipment` como el siguiente `accessibility → preference → typename → tyvalue → Equipment` y posteriormente el valor de dicho campo en la siguiente ruta `accessibility → preference → prefcode → profile-type.value`.

Previous Courses Level. Esta será una categoría dentro del perfil y podrá contener un campo o campos cuyo valor será la nota obtenida en cada uno de los cursos (campos) donde se hayan evaluado los temas que estén asociados por medio de la relación `Required` al curso o cursos a los que el estudiante se haya matriculado.

El nombre del curso referenciado se guardará en la ruta LIP XML `activity → typename → typevalue → task → learningactivityref → text`, la ruta `activity → evaluation → result → score → fieldlabel → typename → tyvalue → Total` será estática ya que nos indicará que se guardará el resultado de la evaluación total de este curso o tema en la siguiente ruta `activity → evaluation → result → score → fielddata`.

Performance Level. Es el nivel de rendimiento que el alumno ha tenido en sus cursos anteriores. El promedio de las notas de los mismos puede ser asociado automáticamente por la plataforma a los valores `High`, `Medium` y `Low` o definido por el alumno según su criterio.

Estos valores se guardarán en la ruta LIP XML `competency → contentype → referential → indexid → performance-level`.

Available Time. Se refiere al tiempo total disponible que tiene el estudiante para dedicarlo a los cursos que se adaptarán a su perfil. Generalmente se debieran especificar las horas promedio a la semana. Este dato es muy significativo en este tipo de ambientes debido a que los estudiantes suelen combinar su estudio con el trabajo o las actividades del hogar o, dado que no son cursos presenciales, pueden elegir

el tiempo que les resulte mas cómodo. En la ruta dentro de las metas del estándar LIP `goal → typename → tyvalue` se especifica el nombre del curso o cursos que deben llevarse a cabo, después se dice que esa meta debe tener otra meta temporal llamada `time_availability goal → contenttype → temporal → temporalfield → fieldlabel → typename → tyvalue`, finalmente se proporciona de manera dinámica, es decir con la información que proporcione el estudiante, el valor que debe tener dicha disponibilidad en la ruta `goal → contenttype → temporal → temporalfield → fielddata`.

Para simplificar su uso se le han asignado los posibles valores de `Much` y `Little`, de manera que el estudiante pueda decir si tiene mucho o poco tiempo para realizar las actividades asociadas a los objetos de aprendizaje del curso.

Learning Style. Este campo indica el estilo de aprendizaje del alumno y puede ser rellenado directamente por el estudiante o automáticamente a través de la realización del cuestionario correspondiente dentro de la misma plataforma.

Dentro del estándar IMS-LIP existe una preferencia dentro de la clase `accessibility` que puede ser llamada `Learner Style Processing` siguiendo la ruta XML `accessibility → preference → typename → tyvalue` y cuyos valores particulares se pueden rellenar en la ruta `accessibility → preference → prefcode → profile-type.value`, en donde se le asigna un valor a la preferencia definida anteriormente.

Para este caso particular se ha decidido utilizar el estilo de aprendizaje `Honey-Alonso` cuyos valores pueden ser `Active`, `Reflexive`, `Theoretical` o `Pragmatic` pero, cabe resaltar, que es posible utilizar los valores de algún otro estilo de aprendizaje, siempre y cuando cuente con una relación explícita entre sus diversos valores y el orden en que debieran mostrarse los objetos de aprendizaje para cada uno de ellos de acuerdo a su *Learning Resource Type*.

Scores. Se refiere tanto a las calificaciones de las evaluaciones intermedias presentadas por el estudiante en todos y cada uno de sus cursos, como a las calificaciones totales de dichos cursos. Esta información no se detallará dentro de los campos del perfil del estudiante del LMS como el resto de las variables mencionadas anteriormente; por el

momento se obtendrá de los campos del estándar QTI que se describen en la siguiente sección o de la información que se encuentre en las bases de datos del LMS para ese alumno en particular.

El por qué la se ha incluido en esta sección radica en el hecho de que para cada alumno, la calificación de cada una de sus evaluaciones intermedias deberá almacenarse en la ruta LIP XML `activity → evaluation → result → score → fielddata`, mientras que la relación que identifica dicha calificación con su evaluación correspondiente se guardará en la ruta `activity → evaluation → evaluationid`. Por lo tanto, por cada evaluación intermedia que el estudiante lleve a cabo, se agregará a su LIP una nueva tag el tipo `evaluation` para la cual se cumplimentarán, al menos, los dos campos anteriormente mencionados.

Como podemos concluir, después de haber descrito toda la información del perfil del estudiante que es necesario cumplimentar, en un LMS ésta información no se encuentra centralizada, si no mas bien distribuida en un conjunto de tablas del mismo. Por lo que, al centralizarla toda en los diferentes metadatos del estándar IMS-LIP, ahorraremos un trabajo importante al experto en planificación.

3. Los Cuestionarios

Cuando se está tratando con dominios de e-learning dinámicos, lo primero en que se puede pensar es en la incertidumbre que conlleva la aplicación de evaluaciones intermedias o cuestionarios, específicamente en las calificaciones que pudieran obtener los estudiantes y que pedagógicamente pudieran implicar cambios en el diseño instruccional de los temas que soporten objetivos posteriores a dicha evaluación.

Aunque actualmente los LMS's ya tienen una forma de almacenar dentro de sus bases de datos las calificaciones de evaluaciones intermedias, los LMS's que están integrando el estándar IMS-QTI están almacenando en cada fichero XML los resultados de una evaluación particular. Estos resultados pueden posteriormente ser compartidos dinámicamente con su equivalente en el estándar IMS-LIP o el IMS-LD.

Los metadatos de la especificación IMS-QTI que para propósito de esta memoria se pretende compartir, son los siguientes:

- `metadata` → `general` → `identifier` que permite saber cual es el identificador de la evaluación.
- `outcomes_processing` → `outcomes` → `decvar` con los atributos `varname = "SCORE"`, `vartype = "Integer"` y `defaultval = "0"` que permite compartir la calificación real de la evaluación en un instante de tiempo. En este caso el instante de tiempo en el que se esté evaluando a un estudiante en particular.

4. Los Diseños Instruccionales

Por su parte, los diseños instruccionales pueden ser representados, en el ámbito de las especificaciones e-learning, por medio del estándar IMS-LD o diseño de aprendizaje.

El estándar IMS-LD, dentro de su estructura XML, tiene una gran variedad de metatags relacionadas con conceptos de e-learning que pueden ser equivalentes a un número considerable de conceptos de planificación inteligente. Dentro de su nivel B de representación, es posible utilizar información ya procesada de los estándares que se han utilizado en las secciones anteriores para definir distintas partes del curso en línea. Además, es posible desplegar de manera ordenada la salida de un planificador, especialmente si éste es capaz de proporcionar información extra relacionada con las acciones que deberán ejecutarse en el plan.

Para su representación XML se estructura por medio de dos etiquetas o tags principales: las `organizations` y los `resources`. La tag de recursos se limita incluir el identificador, tipo y ruta en la que se encuentra cada actividad u objeto "físico" de aprendizaje. Mientras que la de organizaciones permite integrar otros dos elementos esenciales del diseño de aprendizaje: sus componentes y sus métodos.

Los `components` de un diseño de aprendizaje son tres:

Roles. Los roles de un curso en línea son generalmente tres: profesor o facilitador, administrador del sistema y estudiante. Sin embargo, dado que en un curso hay muchos y muy variados perfiles de estudiantes, en esta especificación es posible representar a todos y cada uno de los perfiles y/o estudiantes por medio de un rol tipo learner. En este caso se representará a todos y cada uno de los estudiantes tal y como se muestra en la figura 8.

```
<imsld:roles identifier="LD-roles">
  <imsld:learner identifier="ROLE-LEARNER-3" create-new="not-allowed">
    <imsld:title>ana </imsld:title>
  </imsld:learner>
</imsld:roles>
```

Figura 8. Representación de roles en IMS-LD.

Propiedades. Las propiedades de un diseño de aprendizaje permiten representar todas aquellas variables dinámicas que pueden formar parte del curso y cambiar a lo largo del mismo. En este caso se representan las calificaciones de una evaluación intermedia a través de un código similar al de la figura 9.

```
<imsld:properties>
  <imsld:locpers-property identifier="QTI-Test1">
    <imsld:title>QTI-Tema1</imsld:title>
    <imsld:datatype datatype="real"/>
    <imsld:initial-value>70</imsld:initial-value>
  </imsld:locpers-property>
</imsld:properties>
```

Figura 9. Representación de propiedades de tipo QTI en IMS-LD.

Actividades. Las actividades, de forma similar a las tareas y acciones de un dominio jerárquico de planificación, se dividen en dos tipos: `learning-activities` y `activity-structures`.

- Las `learning-activities` describen las acciones del dominio, es decir, las actividades u objetos de aprendizaje específicos que el estudiante deberá llevar a cabo para comprender uno o mas conceptos relacionados con un objetivo de aprendiza-

je. Están directamente relacionadas con uno de los recursos de la sección principal de *resources* del IMS-LD, como se puede ver en la figura 10.

```

<imsld:learning-activity identifier="ITEM-action2">
  <imsld:title>LS-Problem-St</imsld:title>
  <imsld:activity-description>
    <imsld:item identifierref="RES-seis"/>
  </imsld:activity-description>
  <imsld:complete-activity>
    <imsld:user-choice/>
  </imsld:complete-activity>
</imsld:learning-activity>

```

Figura 10. Representación de actividades de aprendizaje en IMS-LD.

- Las *activity-structures* por su parte, pretenden describir el conjunto de actividades que habrá de llevar a cabo un estudiante específico que ha obtenido una calificación determinada en la evaluación señalada. Tal y como se muestra en el título de la primera estructura de actividades de la figura 11 *ACTIVITY-STRUCTURE-QTI-Test1-3.0-90.0*, en el que se señala que es una estructura de actividades del QTI Test1 para el estudiante 3.0 que ha obtenido una calificación mayor o igual a 90.0.

O bien, describe las distintas estructuras de aprendizaje que puede llevar a cabo un estudiante para un curso determinado dependiendo de su calificación. Como se puede ver en la segunda estructura de aprendizaje de la figura 11.

Para definir el uso y dinamismo de los componentes de un diseño de aprendizaje descritos en los puntos anteriores, se encuentran los llamados *methods*, que son dos: *play* y *conditions*.

- **Player.** La etiqueta *play* permite asignar una estructura de aprendizaje a un rol específico, en este caso a un estudiante particular. También, como se puede apreciar en la figura 12, se especifica el ¿cómo? y el ¿quién? debe ejecutar una estructura de actividades para que se considere completada. En este caso, cuando la última actividad se ha completado, también la estructura se considera terminada.
- **Condiciones.** La etiqueta *conditions*, junto con sus etiquetas anidadas *if* y *then*, permite definir el valor de una propiedad en base al cual se podrán visualizar, o no, un

```

<imsld:activity-structure
  identifier="ACTIVITY-STRUCTURE-QTI-Test1-3.0-90.02" structure-type="selection">
  <imsld:title>Discrete-Maths</imsld:title>
  <imsld:learning-activity-ref ref="ITEM-action6"/>
  <imsld:learning-activity-ref ref="ITEM-action11"/>
  <imsld:learning-activity-ref ref="ITEM-action12"/>
  <imsld:learning-activity-ref ref="ITEM-action13"/>
  <imsld:learning-activity-ref ref="ITEM-action2"/>
  <imsld:learning-activity-ref ref="ITEM-action5"/>
  <imsld:learning-activity-ref ref="ITEM-action4"/>
  <imsld:learning-activity-ref ref="ITEM-action9"/>
  <imsld:learning-activity-ref ref="ITEM-action15"/>
  <imsld:learning-activity-ref ref="ITEM-action16"/>
  <imsld:learning-activity-ref ref="ITEM-action18"/>
  <imsld:learning-activity-ref ref="ITEM-action7"/>
  <imsld:learning-activity-ref ref="ITEM-action10"/>
</imsld:activity-structure>

<imsld:activity-structure
  identifier="ACTIVITY-STRUCTURE-Course-3" structure-type="selection">
  <imsld:title>Course </imsld:title>
  <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-Discrete-Maths-3.0-90.0"/>
  <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-QTI-Test1-3.0-90.0"/>
  <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-QTI-Test1-3.0-70.0"/>
  <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-QTI-Test1-3.0-30.0"/>
</imsld:activity-structure>

```

Figura 11. Representación de estructuras de aprendizaje en IMS-LD.

conjunto de actividades y/o estructuras de aprendizaje. Un ejemplo de su aplicación se puede observar en la figura 13.

Utilizando los métodos anteriores, será posible definir la estructura interactiva y personalizada básica que se requiere para ejecutar adecuadamente un diseño instruccional.

Una vez etiquetados todos los objetos de aprendizaje y rellenos los campos del perfil de usuario que han sido descritos en esta sección, será posible - a través de diversos procesos de ingeniería de conocimiento - que los expertos en planificación y scheduling inteligentes generen los archivos de dominio y problema de planificación que posteriormente serán utilizados por un planificador para generar un diseño de aprendizaje adaptado a cada estudiante.

Sin embargo, el proceso de generación de ficheros para la planificación sigue resultando bastante lento y costoso en lo que a recursos y capacidades humanas se refiere. Es por eso que en la siguiente sección se proponen un conjunto algoritmos que permitirán generar de forma completamente automatizada los ficheros de dominio y problema de planificación, simplificando así el uso de las técnicas de AIP&S para la posterior generación de diseños de aprendizaje.

```

<imsld:play identifier="COURSE-FOR-STUDENT-3">
  <imsld:title>Course for Student ana</imsld:title>
  <imsld:act identifier="Discrete-Maths-3">
    <imsld:title>Discrete-Maths-3</imsld:title>
    <imsld:role-part identifier="ROLE-PART-3">
      <imsld:title>Participants</imsld:title>
      <imsld:role-ref ref="ROLE-LEARNER-3"/>
      <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-Course-3"/>
    </imsld:role-part>
    <imsld:complete-act>
      <imsld:when-role-part-completed ref="ROLE-PART-3"/>
    </imsld:complete-act>
  </imsld:act>
  <imsld:complete-play>
    <imsld:when-last-act-completed/>
  </imsld:complete-play>
</imsld:play>

```

Figura 12. Código XML que representa el método `play` de un IMS-LD.

```

<imsld:conditions>
  <imsld:if>
    <imsld:greater-than>
      <imsld:property-ref ref="QTI-s2ctest2"/>
      <imsld:property-value>79</imsld:property-value>
    </imsld:greater-than>
  </imsld:if>
  <imsld:then>
    <imsld:show>
      <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-QTI-Test1-3.0-90.0"/>
      <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-QTI-Test1-4.0-90.0"/>
    </imsld:show>
    <imsld:hide>
      <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-QTI-Test1-3.0-70.0"/>
      <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-QTI-Test1-3.0-30.0"/>
      <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-QTI-Test1-4.0-70.0"/>
      <imsld:activity-structure-ref ref="ACTIVITY-STRUCTURE-QTI-Test1-4.0-30.0"/>
    </imsld:hide>
  </imsld:then>
</imsld:conditions>

```

Figura 13. Código XML que representa el método `conditions` de un IMS-LD.

La siguiente sección, por tanto, introduce en el proceso de ingeniería de conocimiento que llevan a cabo los algoritmos propuestos. Se han creado dos algoritmos de generación de dominios de planificación y dos de problemas, que se usan dependiendo del paradigma que deseemos utilizar para construir el diseño de aprendizaje.

CAPÍTULO 4

Generación Automática de los Ficheros de Dominio y Problema de Planificación

Como se ha clarificado en el capítulo anterior, un diseño instruccional se puede obtener a partir de un proceso de planificación y scheduling inteligentes; y estas técnicas han demostrado hacer mas eficiente el proceso de la generación de estos diseños, tal como se expresa en [Kontopoulos et al., 2008, Peachy and McCalla, 1986, Ullrich and Melis, 2009], entre otros.

Para poder utilizar las técnicas de AIP&S es preciso generar un dominio y problema de planificación en lenguaje PDDL, lo cual requiere un arduo proceso de ingeniería de conocimiento para los expertos en AIP&S sobre el dominio a codificar[Castillo et al., 2005, Wilkins and Desimone, 1992]. Dicho proceso, así como la escritura de los dominios y problemas en el lenguaje PDDL, suele llevarse a cabo manualmente, siguiendo la arquitectura presentada en la figura 14, y ello toma un tiempo que en ocasiones es muy poco razonable. También, sobre todo en dominios dinámicos, este proceso requiere un tiempo y esfuerzo que no siempre es viable, si se compara con el que normalmente se utiliza en la generación

manual (sin ayuda de las técnicas de AIP&S) de un plan.

El caso de los dominios e-learning, y mas específicamente de la generación de diseños instruccionales adaptados a las necesidades de un conjunto de estudiantes, no es la excepción. Por ejemplo, en el entorno de una universidad a distancia con cientos de miles de alumnos y cientos de cursos en línea, es una tarea titánica, aún para un experto en AIP&S, obtener toda la información relativa a los perfiles de los alumnos, generar su correspondiente problema de planificación y hacer las modificaciones pertinentes en el cada vez que los alumnos lleven a cabo alguna actividad trascendente como un examen, la matrícula en un nuevo curso o modifiquen alguno de sus datos del perfil. Al experto también le resultaría casi imposible diseñar y mantener a mano el dominio de planificación para cada uno de los cursos que la universidad ofrezca, dada la complejidad y dinamismo imprevisible de dichos cursos, y sus materiales correspondientes, a lo largo del ciclo escolar.

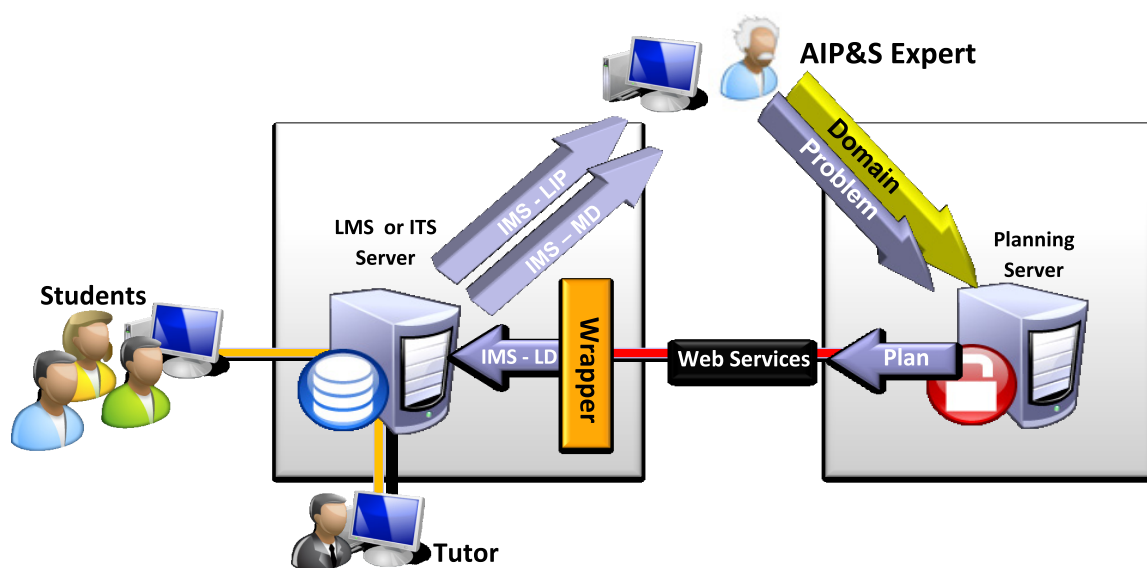


Figura 14. La vista de la arquitectura general utilizada por las pocas propuestas que integran la adaptación de cursos en LMS's por medio del AIP&S. Veamos que el trabajo del experto en AIP&S es crucial dentro de la misma.

Resumiendo la problemática anterior, se puede decir que aunque las técnicas de planificación y scheduling han demostrado ser sumamente eficientes durante el proceso de generación de diseños instruccionales, su uso sigue teniendo los siguientes inconvenientes:

1. Los dominios y problemas de planificación son codificados manualmente por personal altamente cualificado que no es fácil encontrar y requiere de una gran cantidad de tiempo, dinero y esfuerzo, no solo para ser entrenado, si no también para llevar a cabo el proceso de ingeniería de conocimiento que permita hacer una versión inicial de estos ficheros de planificación.
2. Cada curso dentro de un LMS requiere que se codifiquen para él un dominio y problema específico. Por lo tanto, cada vez que surja un cambio en la estructura de un curso o en el perfil de un estudiante, será necesario recurrir de nuevo al o los expertos en AIP&S, considerando que pueden existir cientos de cursos y cientos de miles de alumnos dentro de cada LMS.

Dada la problemática anterior, en este capítulo se presentará un método para abordar el inconveniente (1). Aunque existen muchas soluciones, la propuesta planteada a continuación se basa en encontrar la forma de generar *automáticamente* los ficheros de dominio y problema de planificación que serán utilizados por el planificador para la posterior generación del diseño instruccional de un curso e-learning, tal y como se describe en la arquitectura de la figura 4.

En el capítulo 5 se abordará el inconveniente (2) haciendo uso de algunas de las técnicas de planificación para entornos dinámicos descritas en el capítulo 2.

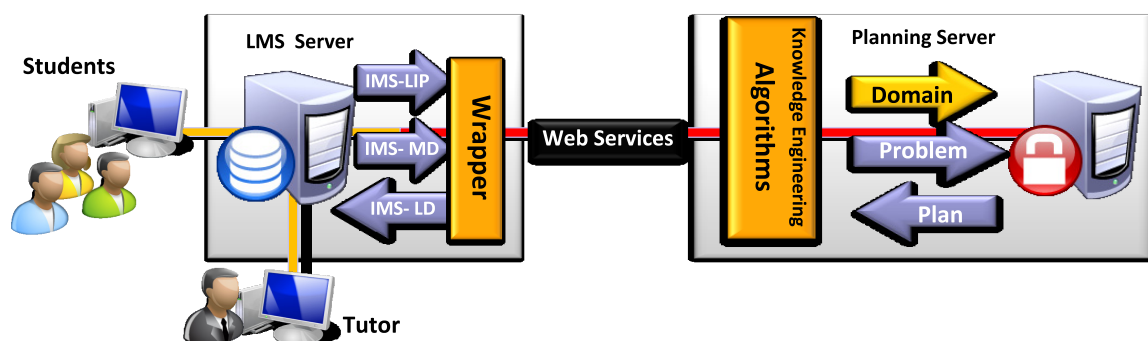


Figura 15. La vista de la arquitectura general utilizada en nuestra propuesta que puede ser construída dado cualquier curso bien etiquetado en un LMS que pueda proporcionar la información requerida por los algoritmos presentados en las siguientes secciones.

Para poder generar automáticamente los ficheros de un problema y dominio de e-learning, ha sido necesario pasar por varias etapas(fig. 16).

La primera es la definición de los datos sobre los cuales se llevaría a cabo el proceso de ingeniería de conocimiento. Esta labor puede ejecutarse en forma simultánea por dos actores principales, los tutores y/o generadores de contenido y los estudiantes y ya ha sido descrita en las primeras dos secciones del capítulo 3.

La segunda etapa del proceso de generación automática, implica la recuperación de la información proporcionada en la primera etapa y su procesamiento a través de un algoritmo específico. Para ello han sido diseñados dos conjuntos de algoritmos de generación diferentes, uno para planificadores de tipo jerárquico y otro para planificadores basados en estados. A través de estos algoritmos se lleva a cabo el proceso de ingeniería de conocimiento necesario para obtener el dominio y problema de planificación en PDDL, con el detalle suficiente como para que el planificador elegido pueda generar un diseño instruccional adaptado.

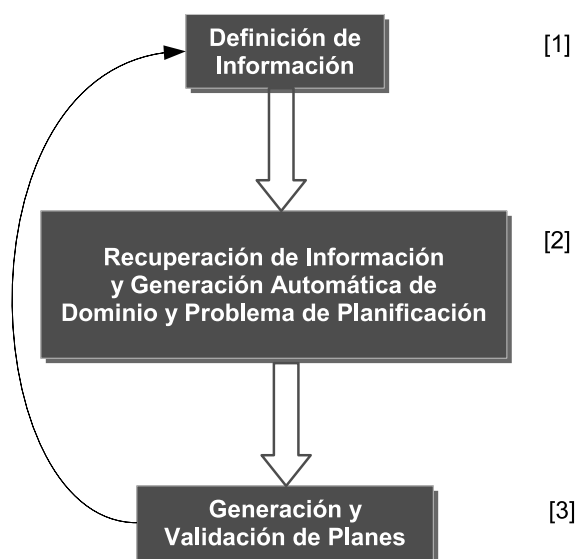


Figura 16. Proceso que se ha seguido para soportar, ejecutar y validar la propuesta teórica de generación automática de dominios y problemas de planificación de esta memoria.

Finalmente en la tercera etapa, que se describirá detalladamente en la sección 4, se muestra una idea general de los dos tipos de diseños instruccionales que es posible obtener como resultado de los dos procesos de planificación. Dichos procesos utilizan la información

proporcionada por los ficheros de dominio y problema de planificación generados en la etapa anterior, y habrán de compararse los resultados de los mismos en base a sus capacidades, necesidades y diferencias que servirán para decidir cual proceso es mas adecuado dependiendo del tipo de dominio de e-learning que se plantee.

Antes de empezar con la explicación formal de los contenidos de esta capítulo, será preciso recordar algunos de los conceptos que se utilizarán a lo largo del mismo y que ya se han mencionado en capítulos anteriores. Después, se irán introduciendo nuevos conceptos de acuerdo a las necesidades de cada algoritmo.

En total, se describirán cuatro algoritmos. Dos para la generación de dominios y dos para la de problemas de planificación, ya que se trabajará sobre dos paradigmas de planificación distintos, jerárquica(HTN) y basada en estados.

1. Preprocesamiento y Preliminares

Para propósitos de este capítulo a los objetos de aprendizaje se les llamará *acciones primitivas* u objetos de aprendizaje primitivos, los temas y subtemas tendrán el nombre de *tareas compuestas* u objetos de aprendizaje compuestos y tanto a los títulos del curso, que describen el objetivo principal del mismo, como a los temas o cursos externos que sean requeridos por alguno de sus temas o subtemas se les llamará *tareas compuestas principales*.

Los *dominios de planificación* describen todas y cada una de las acciones que se tienen que llevar a cabo para alcanzar una, o varias, metas expuestas en el archivo del problema.

Cada una de las acciones primitivas, que posteriormente formarán parte del diseño instruccional de los alumnos cuenta con cinco bloques principales.

1. *La definición de tipo y el nombre de la acción.* Existen dos tipos de acciones, las *durative-action* que tienen un tiempo definido de duración y ejecución dentro del plan y las *action* que no permiten la asignación de una duración establecida.
2. *Los parámetros o variables que utiliza.* Que generalmente se refieren a la variable del estudiante que se esté planificando en ese momento para poder asociar las sentencias

de precondiciones y efectos que formen parte de la acción a dicho estudiante.

3. *Las condiciones o precondiciones de la acción.* En el caso de las acciones con duraciones establecidas se hablará de condiciones y en el caso de las acciones normales de precondiciones. Suelen estar asociadas a las sentencias del estado inicial del estudiante que sean requeridas por esa acción en particular para poder llevarse a cabo. También se pueden agregar otro tipo de precondiciones dentro de las tareas de un dominio HTN.
4. *Los efectos de la acción.* En el paradigma jerárquico la acción ya equivale a un efecto por si misma, aunque es posible añadir otros efectos dependiendo de la influencia que tenga dicha acción sobre otras variables del entorno. Mientras que en el paradigma basado en estados se habrán de especificar los efectos de cada acción o, de lo contrario, será como si no se hubiera llevado a cabo.

Por su parte, en un *problema de planificación* se describen el estado inicial de todas las variables que intervienen en el problema y las metas que queremos conseguir después del proceso de planificación.

En particular, los problemas que habrán de describirse para la futura adaptación del diseño instruccional de un curso comprenden:

1. La descripción del estado inicial de cada uno de los estudiantes del curso que se esté adaptando en ese momento. Esta descripción se hace mediante sentencias declarativas en el lenguaje PDDL que pueden implicar el uso de operadores de comparación. Por ejemplo, si el estilo de aprendizaje de la estudiante Ana es pragmático, entonces la sentencia quedaría de la siguiente manera: `(learning-style ana pragmatic)`
2. La definición de las metas u objetivos principales del curso. Esta definición se hace dependiendo del paradigma que se vaya a codificar y puede incluir, no solo los objetivos principales, si no también restricciones temporales definidas por el estudiante o el profesor, entre otras “métricas” mas.

Los objetos de aprendizaje generalmente suelen crearse y etiquetarse dentro de un contexto determinado para posteriormente ser incluidos dentro de un paquete SCORM y utilizados

dentro de un curso específico. Sin embargo, dentro de un repositorio de objetos de aprendizaje en el que pueden existir cientos de paquetes SCORM, es posible encontrar objetos de aprendizaje que no estén dentro del mismo paquete y que se encuentren relacionados entre sí.

Antes de describir un dominio de planificación es preciso identificar, dentro del o los repositorios de objetos de aprendizaje \mathcal{LOR} , el conjunto de objetos de aprendizaje \mathcal{O} que formarán parte del curso que se desea adaptar y que se deben representar en el dominio, ya sea como acciones primitivas \mathcal{P}_o , o como tareas compuestas \mathcal{C}_o en dominios jerárquicos. Para ello se recomienda utilizar el algoritmo descrito en la figura 17.

Dado un conjunto de objetos de aprendizaje en el repositorio \mathcal{LOR} y el conjunto de metas para el curso que se desea adaptar $\mathcal{G} \subseteq \mathcal{LOR}$ como el conjunto de objetos de aprendizaje de mas alto nivel que se deben llevar a cabo, EXTRAER el conjunto de todos los objetos de aprendizaje $\mathcal{O} \subseteq \mathcal{LOR}$ que son candidatos a aparecer en dicho curso.

(a) $\mathcal{O} = \mathcal{G}$

(b) **Mientras** existan $o \in \mathcal{O}$ no explorados

(i) **Seleccionar** $o \in \mathcal{O}$ no explorado

(ii) **Obtener** conjunto $Next(o) = \{o' | o \in \text{IsPartOf}(o') \vee o' \in \text{Requires}(o)\}$

(iii) **Extender** \mathcal{O} , $\mathcal{O} = \mathcal{O} \cup Next(o)$

Esto va a formar el conjunto cerrado \mathcal{O} de todos los objetos de aprendizaje y sub-objetos de los mismos que pudieran ser asignados a cualquier estudiante dado un conjunto de metas de aprendizaje, por lo que $\mathcal{O} \subseteq \mathcal{LOR}$. Este conjunto \mathcal{O} estará compuesto por objetos de aprendizaje primitivos \mathcal{P}_o o compuestos \mathcal{C}_o , en cuyo caso $\mathcal{O} = \mathcal{P}_o \cup \mathcal{C}_o$.

Figura 17. Algoritmo de extracción desde el repositorio de objetos de aprendizaje de los objetos primitivos o compuestos que formarán parte de un curso.

El conjunto de metas \mathcal{G} necesarias para cumplir los objetivos del un curso, como se ha dicho anteriormente, pertenece al conjunto de objetos de aprendizaje compuestos \mathcal{C}_o y, al representarlas dentro del problema de planificación será preferible que se describan no solo como objetivos literales, si no también que se detalle el tiempo disponible para cumplirlas. Un curso tiene, por defecto, un tiempo mínimo requerido para llevarse a cabo y un tiempo máximo que incluye el tiempo que se requiere para completar aquellos objetos de aprendizaje de mas alto nivel u opcionales. Por lo tanto, el alumno deberá concretar dentro de su meta si tiene poco o mucho tiempo para cumplirla, en cuyo caso se asignará el mínimo o el máximo de cada meta respectivamente.

Por otro lado, dentro del proceso de generación de cualquier dominio de planificación, además de las reglas de ordenación explícitas de objetos dadas por los metadatos (en particular el metadatos `IsBasedOn`, es preciso conocer determinadas reglas de ordenación implícitas

cuyos valores deberán considerarse como segundo o tercer criterio para definir condiciones de las acciones primitivas o de los métodos de ordenación, así como el orden en el que se desplegarán las tareas que formen parte de dichos métodos.

En el caso del dominio de e-learning, se considera que existe una sola regla de ordenación \mathcal{R} basada en el estilo de aprendizaje del estudiante. Para propósitos de esta investigación se toman en cuenta los valores simplificados del estilo de aprendizaje Honey-Alonso: teórico y pragmático (según lo descrito en el apéndice B); en base a la relación de dichos estilos con el *Learning Resource Type* de los objetos de aprendizaje el orden en que deben desplegarse los objetos debe ser el que se muestra en la tabla 3. Por lo tanto, cada uno de los “valores” del estilo de aprendizaje Honey-Alonso es en si mismo un *valor de orden v* que describe el orden en que deben secuenciarse los objetos de aprendizaje de un curso.

	Estilos de Aprendizaje	
	<i>Teórico</i>	<i>Pragmático</i>
Objetos de Aprendizaje - Learning Resource Type	Narrative Text	Simulation
	Table	Example
	Slide	Experiment
	Diagram	Exercise
	Problem Statement	Problem Statement
	Example	Diagram
	Simulation	Slide
	Experiment	Table
	Exercise	Narrative Text

Tabla 3. Orden de despliegue de los objetos de aprendizaje, según su *Learning Resource Type*, para un estudiante con estilo de aprendizaje teórico o uno pragmático.

Además de las acciones y las reglas que permiten ordenarlas, existen otros elementos que forman parte de cualquier dominio de planificación. Estos son la cabecera, los tipos de datos, las constantes, las funciones y los predicados. En los siguientes puntos se describe cada uno de ellos, así como la forma en que se generarán automáticamente dentro de cualquier dominio de planificación.

- La *cabecera* define el conjunto de requerimientos del dominio, es decir, si se utilizarán acciones durativas o simples, si se llevará a cabo una expansión jerárquica de tareas, si se permitirá el uso de valores numéricos asignados a predicados, de metatags, entre otras. Por lo tanto, la cabecera para un dominio de e-learning suele ser generalmente la

misma para cualquier curso y considerarse como una constante dentro de la generación automática del mismo.

- Los *tipos de datos* también son los mismos para todo dominio de e-learning, pueden ser numéricos u objetos. El resto de los tipos de datos se derivan de estos dos; por ejemplo, un estudiante es un tipo de dato objeto, la calificación de un examen un tipo de dato numérico, los valores de las reglas de ordenación son tipos de datos objetos normalmente, etc.

- En el caso de las *constantes*, éstas se refieren al conjunto de valores que se utilizarán para darle sentido a los parámetros, predicados y funciones que formen parte de las condiciones o parámetros de una acción, o tarea, en el caso de dominios HTN. Para el caso de un dominio de e-learning las constantes pueden ser los nombres de los valores de las reglas de ordenación, los valores de los idiomas que podrán manipularse dentro del curso, el tipo de facilidades de cómputo que tiene el estudiante, etc.

Como ahora mismo no se cuenta con una interfaz para extraer la información de los valores de las reglas de ordenación ni de los idiomas, entre otras tantas variables consideradas durante la personalización de un curso, éstas se definen en base a arreglos de datos almacenados en un fichero que se pueden modificar manualmente dependiendo del tipo de constante al que se quiera hacer referencia.

- Los *predicados*, por su parte, son sentencias con variables incluídas que pueden tomar valores de verdad falso o verdadero dependiendo de su existencia o no para un determinado conjunto de variables que pueden ser especificadas dentro del problema de planificación o durante el proceso de planificación. Los predicados descritos para un dominio de planificación e-learning son: (availability ?student - student ?value - object), (action-name ?student - student done) y (equipement ?student - student ?value - object).

El segundo predicado se genera una vez que se conocen los nombres de todas las acciones que se generarán para el dominio, y se genera un predicado por cada acción ya que action-name es igual al nombre de cada acción. También el predicado (learning-style ?student - student ?value - object), que define la regla de ordenación, se genera

automáticamente para los dos tipos de dominio.

Sin embargo, como dicha regla puede cambiar, es decir, no siempre se basará en el estilo de aprendizaje del estudiante, su nombre y valores - que en este caso son `learning-style` con valores `theoretical` y `pragmatic` respectivamente - se obtienen de arreglos de datos almacenados en un fichero plano, como en el caso de las constantes.

- Las *funciones* son predicados que toman valores numéricos, similares a las variables de un lenguaje de planificación estándar. En este caso, son el nivel que se tiene en un idioma determinado, la calificación de una evaluación previa y el rendimiento promedio de un estudiante los que se representan por medio de las funciones (`language-level ?language - language ?student - student`), (`score ?student - student ?value - object`) y (`performance ?student - student`) que se generan automáticamente de manera estática.

Sentencia PDDL	Campos de Moodle
<code>(= (performance ana) 90)</code>	Se refiere al <i>Performance Level</i> de Ana. El valor <code>High</code> equivale a un 90 y el resto a 60.
<code>(learning-style ana pragmatic)</code>	Indica el <i>Learning Style</i> de Ana cuyos valores pueden ser <code>theoretical</code> o <code>pragmatic</code> .
<code>(= (language-level en ana) 90)</code>	Da un valor de 90 al <i>English Level</i> de Ana. Esto quiere decir que seleccionó los valores <code>Excelent</code> o <code>Acceptable</code> para su nivel de inglés. Si hubiese seleccionado otros valores le hubiera dado un 40. Cabe recordar que <code>language-level</code> es una función.
<code>(= (score ana LogicAndSets) 60)</code>	Indica la calificación - <i>level</i> de Ana en el <i>previous course</i> <code>Logic and Sets</code> . <code>score</code> es una función.
<code>(availability ana much)</code>	Se refiere al <i>Available Time</i> de Ana para llevar a cabo la meta que se esté planificando en ese momento.
<code>(equipment ana multimedia)</code>	Indica que el <i>Equipment</i> de Ana tiene valor <code>Multimedia</code> . Si el <i>equipment</i> de Ana tiene valor de <code>None</code> , entonces dicha sentencia no se generará.

Tabla 4. Descripción del estado inicial de la estudiante Ana en el problema de planificación, el cual se ha extraído a partir de su perfil descrito bajo el estándar IMS-LIP y/o a partir de los campos de Moodle descritos en la sección 2 del capítulo 3.

En base a los predicados y funciones descritos con anterioridad, será posible comprender la manera en que se propone generar el estado inicial del problema de planificación para cualquier paradigma HTN o basado en estados; siendo el estado inicial de un problema

de e-learning toda aquella información que describa las características específicas de un estudiante que puedan aprovecharse para la personalización de un curso durante el proceso de planificación.

Por lo tanto, tomando como base la información del perfil de la estudiante Ana que se ha extraído de su descripción correspondiente en el estándar IMS-LIP, es posible generar el estado inicial de dicha estudiante formando las sentencias declarativas correspondientes que han sido definidas previamente como predicados o funciones dentro de sus respectivas secciones en el dominio de planificación.

Así, como se puede apreciar en la tabla 4, existen 6 posibles sentencias declarativas que hacen referencia a su nivel de inglés, estilo de aprendizaje, nivel de rendimiento o desempeño académico, disponibilidad horaria y equipamiento técnico disponible.

Los elementos de un dominio y problema de planificación cuya generación automática no se ha descrito en esta sección serán descritos en las secciones siguientes. Estos elementos son las acciones primitivas - y tareas en caso de un dominio de planificación HTN - así como algunos predicados específicos de los dominios basados en estados; además de la generación de metas de un problema de planificación para cada tipo de representación.

2. Dominios y Problemas para Planificadores Jerárquicos

Como se ha mencionado en reiteradas ocasiones, los dominios de planificación jerárquicos permiten representar de una forma más intuitiva la estructura de un curso. Sin embargo, el proceso de ingeniería de conocimiento que se debe llevar a cabo para definirlos resulta sumamente complejo debido, principalmente, a la enorme cantidad de detalles que se deben obtener del dominio real para poder representarlo adecuadamente en el dominio HTN.

Por otro lado, los planificadores jerárquicos que se han estudiado no permiten definir, dentro de la meta del problema, métricas multiobjetivo para optimizar diversas variables; por lo cual, resulta necesario definir un mayor número de precondiciones asociadas a dichas variables dentro del fichero del dominio, aumentando aún más la complejidad de su generación.

Aunque existen muchas propuestas para generar automáticamente dominios y problemas de planificación para el contexto de e-learning utilizando planificadores Jerárquicos [Sicilia et al., 2006, Ullrich and Melis, 2009], en ninguna de ellas ha generado un dominio o problema basándose única y exclusivamente en la información proporcionada por los metadatos de los estándares de e-learning.

En las siguientes secciones se presentan dos algoritmos de ingeniería de conocimiento. Uno para extraer automáticamente, de la información que proporcionan los metadatos de los objetos de aprendizaje de un curso, todos los predicados, funciones, constantes, tareas, acciones y precondiciones que requiere la definición de un dominio de planificación HTN. Y otro para extraer, de los metadatos del perfil del estudiante, todos los predicados y funciones que definan el estado inicial del problema de planificación; así como la meta del mismo a partir de los metadatos del objeto de aprendizaje del curso que represente la tarea compuesta principal del mismo.

Una primera aproximación a la propuesta de esta memoria se puede ver en [Castillo et al., 2007], mientras que la primera versión de los algoritmos formales presentados en las siguientes secciones se puede revisar en [Castillo et al., 2010].

2.1. Dominio Jerárquico

Un dominio Jerárquico de planificación está compuesto fundamentalmente por una cabecera o conjunto de requerimientos, los tipos de datos que se utilizarán en el dominio, las constantes, predicados, funciones, predicados derivados, tareas y acciones.

La generación automática de la cabecera, tipos de datos, constantes, predicados y funciones se ha descrito en la sección 1 de Preprocesamiento, ya que es la misma tanto para dominios HTN como basados en estados.

Además de las *acciones primitivas* descritas en la primera parte del capítulo, y que equivalen a los objetos de aprendizaje que van a ser ejecutados por el estudiante como parte de las actividades del curso, un dominio de planificación jerárquico tiene como componentes principales para su descripción a las tareas compuestas.

Las *tareas compuestas* son mas conocidas en un ambiente educativo como temas, subtemas o título del curso. En el lenguaje PDDL los componentes de definición general de una tarea son el título, los parámetros y los métodos. Los métodos de descomposición están formados por diferentes secuencias de acciones y/o tareas por medio de las cuales se puede llegar a cumplir el objetivo de una tarea y están formados por su nombre, las precondiciones específicas del método y la secuencia de acciones a realizar según el orden de aparición.

Siendo $\mathcal{P}_o \subset \mathcal{O}$ un conjunto de acciones primitivas, es decir, aquellos objetos de aprendizaje que no están compuestos por otros objetos de aprendizaje.

Para cada objeto de aprendizaje $o \in \mathcal{P}_o$.

- (a) **[Generar Acción Primitiva Simple]**. Generar una nueva acción durativa a con un solo parámetro $?student$ y $nombre = Identifier(o)$.
 - (i) Asignar el valor de a el valor del metadato $typical-learning-time(o)$ de su respectivo objeto de aprendizaje.
 - (ii) Si $language(o) \neq es$ entonces $preconditions(a) = preconditions(a) \cup (>(language-level language(o) ?student) th)$ donde th es el umbral requerido para entender correctamente el objeto de aprendizaje.
 - (iii) Si $other-platform-requirements(o) = Multimedia$ entonces $preconditions(a) = preconditions(a) \cup (equipment ?student multimedia)$.
 - (iv) Introducir en la sección de metadatos, específica del planificador seleccionado, una tag con la siguiente información: “<item><userid>?student</userid><identifiaer>Identifier(o) </identifiaer></item>”.
 - (v) Introducir un efecto para indicar que la acción ha sido completada con éxito por el estudiante (Figura 19).
- (b) **[Revisar los objetos múltiples]**. Si existen uno o mas objetos $o' \in \mathcal{P}_o$ con el mismo $Title(o)$ que o pero con diferentes valores en uno o mas del resto de sus metadatos, entonces:
 - (i) Generar una acción primitiva para cada o' con el mismo nombre de o pero sus respectivos tiempo y precondiciones, siguiendo los lineamientos de (a) .
 - (ii) Eliminar los metadatos de relación de todos los o' y marcarlos como revisados.
- (c) **[Revisar la cobertura]**. Si el metadato $coverage(o) = Optional$, entonces crear un nuevo objeto compuesto auxiliar o'' cuyos metadatos serán los mismos que los de o y su $Identifier(o'')$ igual al de o pero con el prefijo $Optional_$.
- (d) **[Revisar la dificultad]**. Si el metadato $difficulty(o) = Difficult$ o $difficulty(o) = Very Difficult$ y $coverage(o) \neq Optional$, entonces crear un nuevo objeto compuesto auxiliar o'' cuyos metadatos serán los mismos que los de o y su $Identifier(o'')$ igual al de o pero con el prefijo $Difficult_$.
- (e) Si se ha creado un nuevo objeto o'' en (c) o (d), entonces
 - (i) Extender el conjunto \mathcal{C}_o con o'' de manera que $\mathcal{C}_o = \mathcal{C}_o \cup o''$.
 - (ii) Eliminar los metadatos de relación de o y asignarle el metadato $IsPartOf(o'')$.

Figura 18. Algoritmo para la generación las acciones primitivas durativas de un dominio de planificación HTN.

Por lo tanto, para completar la definición de un dominio de planificación jerárquico para e-learning será necesario generar automáticamente tareas compuestas y acciones primitivas del mismo.

2.1.1. Definición de Acciones Primitivas

En la figura 18 se puede observar a detalle el algoritmo publicado en [Castillo et al., 2010] en el cual se describe el proceso de generación automática de acciones primitivas a a partir de los objetos de aprendizaje que forman parte del subconjunto \mathcal{P}_o .

En este algoritmo se genera una acción primitiva simple para cada elemento o que forma parte de \mathcal{P}_o . Dicha acción tiene como nombre el identificador de o , ya que si tuviera el nombre de o podrían surgir problemas de codificación e interpretación por parte del planificador debido a espacios o caracteres extraños que el nombre del objeto pudiera tener.

La acción a también tiene un solo parámetro denominado *?student* y que en la definición de tipos del dominio es un objeto tipo *student* que a su vez es un tipo de datos objeto. Además, tiene una duración que se toma a partir del valor del metadato *Typical Learning Time* del objeto o para el cual se está creando la acción, así como unas precondiciones que dependen de los valores de sus metadatos *Language* y *Other Platform Requirements* respectivamente y que, en caso de que las dos existan, serán concatenadas por una disyunción.

También se genera, dentro de la acción, una sección especial de anotaciones o metainformación parte de un subconjunto del lenguaje PDDL descrito detalladamente en [Pérez, 2007] y que permite definir la información específica (y aumentada) que contendrá el fichero de salida del planificador sobre cada una de las acciones que formen parte del plan resultante.

Finalmente, se genera un efecto que especifica que dicha acción ya se ha llevado a cabo y con ello se da por terminada la generación automática de la acción a para un objeto de aprendizaje primitivo o cuyo ejemplo se puede ver reflejado en la figura 19.

Sin embargo, este algoritmo no se da por terminado con el solo hecho de generar una acción para cada $o \in \mathcal{P}_o$. También, en el inciso (b), se revisa el objeto o de manera que, si existen otros objetos de aprendizaje o' con el mismo nombre de o , no tengan que ser revisados posteriormente si no que se genere para cada uno de ellos una acción a' igual a a pero con su respectivo tiempo y precondiciones relacionadas a los metadatos *Language* y *Other Platform Requirements*.¹

¹En este caso, se considera que si hay dos o mas o con el mismo nombre y uno de ellos es difícil y/u opcional entonces todos son difíciles y/u opcionales.

```

(:durative-action ComplexityExperiment
 :parameters(?student - student)
 :meta(
  (:tag prettyprint
   "<<item>
    <userid?id</userid>
    <identifier>ITEM-action24</identifier>
    </item>"))
 :duration(= ?duration 5)
 :condition(and
  (>(language-level en ?student) 50)
  (equipment ?student multimedia))
 :effect(ComplexityExperiment ?student done))

```

Figura 19. Acción *Complexity Experiment* creada de acuerdo al algoritmo de la figura 18.

Por otro lado, el algoritmo de la figura 18 también permite generar un nuevo elemento o'' del conjunto \mathcal{C}_o , es decir, un nuevo objeto de aprendizaje compuesto en caso de que o sea opcional o tenga una dificultad alta, y que este nuevo objeto ocupe el lugar de o dentro de la estructura relacional de los $o \in \mathcal{O}$, para finalmente asignar a o como parte del mismo cambiando así sus metadatos relacionales iniciales.

La generación de los objetos compuestos o'' tiene un sentido meramente auxiliar que permite extender las capacidades del dominio de planificación HTN a las necesidades de un entorno e e-learning, como se explicará mas a detalle en la siguiente sección.

2.1.2. Definición de Tareas Compuestas

Siendo $\mathcal{C}_o \subset \mathcal{O}$ un conjunto de objetos de aprendizaje compuestos, es decir, aquellos objetos de aprendizaje que están compuestos por otros objetos de aprendizaje y que corresponderían a tareas dentro del dominio de planificación, se proponen tres métodos distintos para generar automáticamente tareas compuestas para el dominio de un curso de e-learning, métodos que se describen en los algoritmos de las figuras 20, 24 y 26. Estos métodos, aunque se encuentran en figuras distintas, se pueden ejecutar de manera paralela ya que en los tres se revisan uno a uno todos los elementos de \mathcal{C}_o y las tareas que se generen a partir de los mismos son interdependientes y necesarias para generar un dominio HTN consistente.

El algoritmo de la figura 20 es el que se encarga de generar todas aquellas tareas que se corresponden con un objeto de aprendizaje compuesto que se ha creado y etiquetado durante

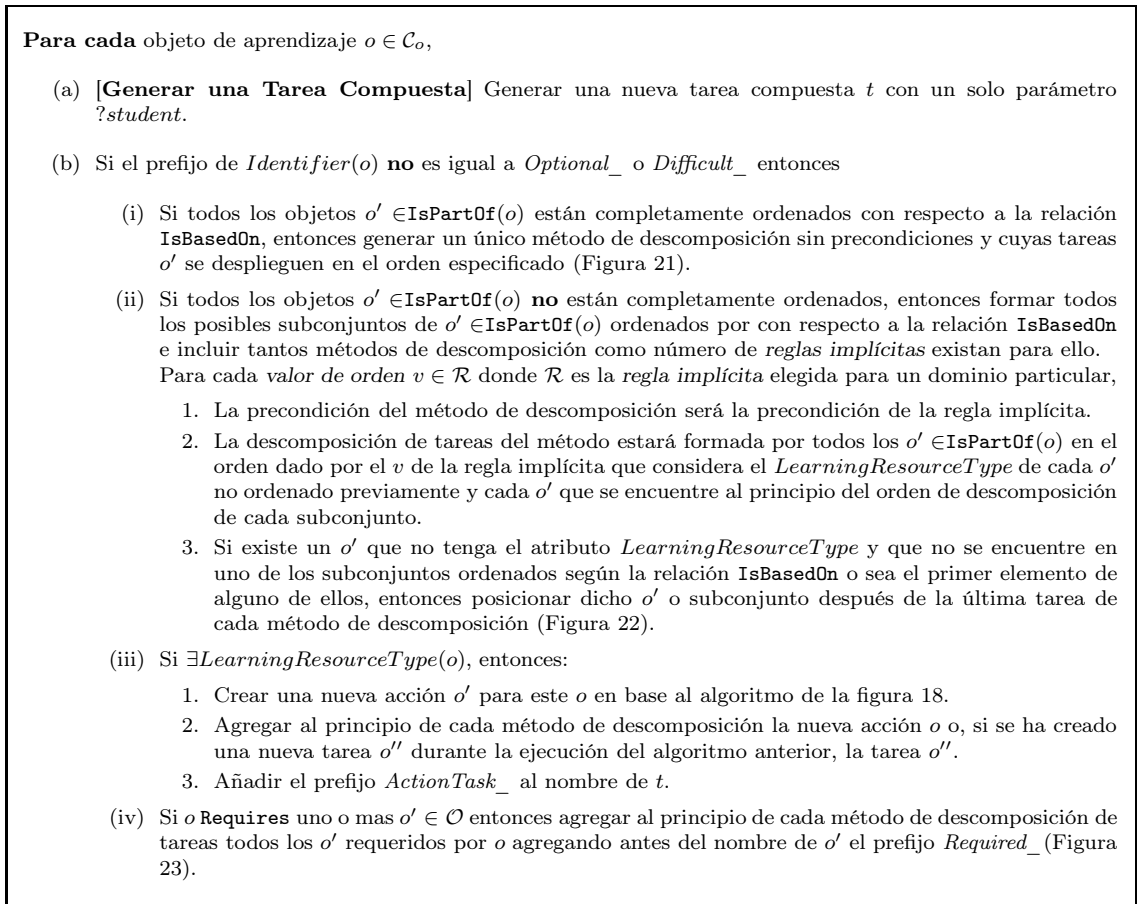


Figura 20. Algoritmo para la generación las tareas compuestas “simples” de un dominio de planificación HTN.

la generación del curso y que, por lo tanto, pueden ser “palpables” dentro del despliegue del mismo al estudiante. Por esta razón, el identificador del objeto o para el cual se este generando la tarea no puede iniciar con el prefijo *Optional_* o *Difficult_*, ya que esto indica que es un objeto que ha sido generado durante el análisis de objetos $o \in \mathcal{P}_o$ del algoritmo de la figura 18.

Las tareas generadas por medio de este algoritmo tienen el nombre del identificador del o explorado, así como un solo parámetro, y pueden contener uno o mas métodos de descomposición de tareas dependiendo de la existencia o no de los suficientes metadatos **IsBasedOn** como para que se relacionen por medio de él todos los objetos $o' \in \mathcal{O}$ que formen parte de o . Es así como, si todos los $o' \in \text{IsPartOf}(o)$ están completamente ordenados por la relación **IsBasedOn**, entonces la tarea generada solo contendrá un método sin precondiciones

que incluya a todos los o' en el orden especificado, tal y como se puede apreciar en la figura 21.

Si la condición anterior no se cumple, entonces se tendrá que recurrir al uso de los valores de las reglas implícitas de ordenación descritas en la sección 1 de preprocesamiento y preliminares para ordenar las $o' \in \text{IsPartOf}(o)$. Y por cada valor de la regla implícita de ordenación, se generará un nuevo método de ordenación de tareas cuya precondition será aquella asociada a la regla y su valor, por ejemplo, la precondition (`learning-style ?student theoretical`) corresponde a la regla `learning-style` cuyo valor de ordenación sea `theoretical`.

```
(:task Complexity
 :parameters(?student - student)
 (:method unique
 :precondition()
 :tasks(
 (ComplexityProblemSt ?student)
 (ComplexityExercise ?student)
 (ComplexityExperiment ?student))))
```

Figura 21. Tarea *Complexity* con un solo método de descomposición y las tres acciones que forman parte de la misma en el orden indicado por la relación `IsBasedOn`.

Un ejemplo de una tarea cuyos métodos de ordenación son generados en base al uso de reglas implícitas de ordenación se puede observar en la figura 22, en la que al encontrarse dos acciones parcialmente ordenadas la acción sobre la que se aplica la regla implícita es la primera de la ordenación explícita, en este caso sobre *Logic Gates 1 Exercise*.

Como las reglas de ordenación se basan en los valores del metadato *Learning Resource Type* de cada o' , si ese metadato no llegara a estar definido para uno o mas objetos o' , entonces dichos objetos se colocarían al final de la ordenación de tareas de cada uno de los métodos de ordenación de la tarea.

Por otro lado, si el objeto o tiene un valor asociado a su metadato *Learning Resource Type* se agregará el prefijo `ActionTask_` al nombre de la tarea t generada previamente, se generará una nueva acción para o en base a los pasos del algoritmo de la figura 18 y se agregará al principio de cada método de descomposición t la llamada a la nueva acción generada, o la llamada al nuevo objeto o'' creado durante la ejecución del algoritmo de generación de

acciones, si ese fuera el caso.

Finalmente, este primer algoritmo indica la introducción de todo $o' \in \text{Requires}(o)$ al principio de la descomposición de tareas de cada método de ordenación de t , precediendo al identificador de cada o' con el prefijo `Required_`.

```

(:task LogicGates
 :parameters(?student - student)
 (:method Theoretical
 :precondition(learning-style ?student theoretical)
 :tasks(
 (LogicGates1NarrativeText ?student)
 (LogicGates1Exercise ?student)
 (LogicGates2NarrativeText ?student)
 (Optional_LogicGates2Exercise ?student)))
 (:method Pragmatic
 :precondition(learning-style ?student pragmatic)
 :tasks(
 (LogicGates1Exercise ?student)
 (LogicGates2NarrativeText ?student)
 (Optional_LogicGates2Exercise ?student)
 (LogicGates1NarrativeText ?student))))

```

Figura 22. Tarea *Logic Gates* con dos métodos de descomposición ordenados por los dos valores de ordenación v que forman parte de la *regla implícita* \mathcal{R} que hace referencia al estilo de aprendizaje Honey-Alonso.

El segundo algoritmo de generación automática de tareas compuestas se ocupa de generar tareas auxiliares para aquellos $o \in \mathcal{P}_o$ cuyos metadatos permiten establecer precondiciones de opcionalidad o dificultad. Los objetos $o \in \mathcal{C}_o$ para los cuales se generan tareas en base a la estructura dictada por el algoritmo de la figura 24 son aquellos que han sido creados durante el algoritmo de la figura 18 y cuyo identificador, por lo tanto, tiene el prefijo *Optional_* o *Difficult_*.

Cuando un objeto es opcional, puede ser o no utilizado por el estudiante dependiendo de la cantidad de tiempo disponible que éste tenga para llevar a cabo las actividades del curso. En caso de que el estudiante tenga suficiente tiempo, entonces el objeto aparecería como una actividad dentro de su plan personalizado. Por otro lado, si tiene suficiente tiempo, pero ya se han seleccionado para su plan otras actividades opcionales que consumen el tiempo específico (en caso de que se pudiera especificar un tiempo intermedio entre el máximo y el mínimo establecido para el curso) que el ha determinado, o si no tiene tiempo para cursar ninguna actividad opcional, no debería incluirse este objeto dentro del plan.

Si las $o \in \mathcal{P}_o$ que son opcionales se incluyeran dentro de los métodos de ordenación de las tareas tal cual, y si se les agregara una condición de opcionalidad respecto al tiempo para representar dicha propiedad; entonces, cuando un estudiante no tuviera tiempo suficiente para llevar a cabo dicha actividad no se generaría ningún plan, por que la no ejecución de dicha acción impediría la inclusión de su tarea padre (que si es obligatoria dentro del plan².

```
(:task ActionTask_BooleanAlgebraDiagram
 :parameters(?student - student)
 (:method Theoretical
 :precondition(learning-style ?student theoretical)
 :tasks(
 (Required_LogicAndSets ?student)
 (BooleanAlgebraDiagram ?student)
 (MinimizationOfCircuitsSlide ?student)
 (Difficult_BooleanAlgebraSimulation ?student)
 (LogicGates ?student)))
 (:method Pragmatic
 :precondition(learning-style ?student pragmatic)
 :tasks(
 (LogicAndSets ?student)
 (BooleanAlgebraDiagram ?student)
 (Difficult_BooleanAlgebraSimulation ?student)
 (MinimizationOfCircuitsSlide ?student)
 (LogicGates ?student))))
```

Figura 23. Tarea *Boolean Algebra Diagram* con un solo método de descomposición y las tres acciones que forman parte de la misma en el orden indicado por la relación *IsBasedOn*.

Lo mismo ocurriría, si una acción cuyo metadato *!Difficulty* tiene los valores de difícil o muy difícil representara los requerimientos de performance del estudiante asociados a dicha dificultad dentro de sus condiciones. No sería posible generar un plan alternativo en caso de que estudiante no cumpliera la condición de rendimiento (\geq (*performance ?student*) *th*)³ ya que, como se supone que la dificultad va asociada al tema del objeto, cualquier objeto que se llamara de la misma forma tendría que ser difícil también.

Por lo tanto, es posible identificar dos tipos de condiciones:

- Condiciones *Fuertes* que están asociadas a requisitos que deben existir obligatoriamente para que una acción pueda ejecutarse, como por ejemplo el nivel de inglés del estudiante o las capacidades de hardware o software necesario para que la acción sea

²Los objetos de aprendizaje compuestos no tienen ningún valor en su metadato *Coverage* y, por lo tanto, su inclusión en el plan es obligatoria

³*th* es el umbral requerido para entender correctamente la acción que forma parte de dicha tarea

desplegada correctamente en el ordenador.

- Condiciones *Débiles* que no forcen la inclusión del objeto de aprendizaje dentro plan. Por ejemplo, la opcionalidad respecto a tiempo o la dificultad de un objeto de aprendizaje.

El problema que surge con las condiciones fuertes es el mismo que se manifiesta cuando se plantea introducir las condiciones de opcionalidad y dificultad dentro de la acción, esto es, si no se cumplen no se genera plan para ese estudiante. Pero éste tiene fácil arreglo, ya que para estos casos se sugiere crear otro objeto de aprendizaje con el mismo nombre del que tiene la precondición fuerte, pero con características que no impliquen el cumplimiento de dichas condiciones.

Continuando con el análisis del objeto o de la figura 20, para cada ciclo:

- (c) Si el prefijo de $Identifia(o)$ es igual a *Optional_* o *Difficult_* entonces generar dos métodos de descomposición:
 - (i) El primero deberá contener la acción $o' = IsPartOf(o)$ y como precondiciones (**availability ?student much**) si $Coverage(o) = Optional$, (**>= (performance ?student) th**) si $Difficulty(o) = Difficult$ o $Difficulty(o) = Very Difficult$ o, en caso de que se cumplan los dos valores anteriores de o , las dos precondiciones unidas por una conjunción.
 - (ii) El segundo no tendrá ninguna tarea, ni condición (Figura 25).

Figura 24. Algoritmo para la generación las tareas compuestas auxiliares para representar condiciones *débiles* en un dominio de planificación HTN.

Sin embargo, para poder abordar el problema que representan las condiciones débiles y ayudar con ello a generar planes que pudieran o no incluir objetos de aprendizaje que las tuvieran manifestadas en sus metadatos, es que se sugiere utilizar el algoritmo de la figura 24.

En este algoritmo se crea una tarea t con un parámetro (*?student*) y dos métodos de descomposición. El primero contiene tantas precondiciones como condiciones débiles se expresen en los metadatos del o explorado y solo la acción que dió origen a o como parte de sus tareas. El segundo método no tiene precondiciones y tampoco acciones que ejecutar. Por lo tanto, cuando un estudiante no cumpla con las condiciones débiles de una acción, al pasar previamente por “el filtro” de la tarea auxiliar generada por este algoritmo, seguirá

generándose un plan para el pero sin dicha acción como parte del mismo.

Un ejemplo de tareas generadas a partir del algoritmo de la figura 24, lo podemos ver en la tarea de la figura 25.

```
(:task Optional_LSNarrativeText
  :parameters(?student - student)
  (:method yes
    :precondition(and
      (>= (performance ?student) 50)
      (availability ?student much))
    :tasks(
      (LSNarrativeText ?student)))
  (:method no
    :precondition( )
    :tasks( )))
```

Figura 25. Tarea *LS Narrative Text* con el prefijo *Optional_* que al ser, su acción concreta tanto difícil como opcional, esas son las precondiciones requeridas para llevarla a cabo.

Finalmente, el tercer camino para generar tareas dentro de un dominio HTN para cursos de e-learning implica la generación de una tarea compuesta auxiliar t' para cada $o' \in Requires(o)$. Dicha tarea deberá tener dos métodos de descomposición, el primero con una precondición del tipo $(\langle \langle score \ ?student \ Identifier(o) \ \text{th1} \rangle \rangle)^4$ y una descomposición de tareas que incluya solo la tarea que corresponda al identificador de o' y el segundo método no deberá tener ninguna precondición ni ninguna tarea como parte de su descomposición.

Para cada objeto de aprendizaje $o \in C_o$,

Si existe un $o' | o \in Requires(o')$ donde o' es cualquier otro objeto parte de \mathcal{O} , entonces generar una nueva tarea compuesta t' cuyo nombre sea $Required_Identifier(o)$, el parámetro $?student$ y dos métodos de ordenación, el primero con la precondición $(\langle \langle score \ ?student \ Identifier(o) \ \text{th1} \rangle \rangle)$ y que llame a la tarea t y el segundo sin precondición ni tarea que lo compongan (Figura 27).

Figura 26. Algoritmo para la generación las tareas compuestas auxiliares que permiten decidir si un objetivo de aprendizaje asociado a un objeto o' se ha cumplido o no previamente en un dominio de planificación HTN.

Este tipo de tareas sirven para representar otro tipo de condiciones débiles que están asociadas a tareas (no a acciones), en este caso las relacionadas con el cumplimiento previo de un objetivo concreto, y podemos ver un ejemplo de ellas en el código de la figura 27.

Siguiendo los pasos descritos por los algoritmos de las dos secciones anteriores, generación

⁴th1 es el umbral requerido para que se considere que dicha acción (objetivo) no ha sido cumplido aún.

de acciones y generación de tareas, podremos generar un dominio de planificación jerárquico suficientemente consistente y completo.

```
(:task Required_LogicAndSets
  :parameters(?student - student)
  (:method yes
   :precondition(
     (<= (score ?student LogicAndSets) 50)
   :tasks(
     (LogicAndSets ?student)))
  (:method no
   :precondition( )
   :tasks( )))
```

Figura 27. Tarea *Logic and Sets* que al ser requerida por *Boolean Algebra Diagram* tiene el prefijo *Required_* en su nombre y, su acción (o tarea) concreta solo puede ser ejecutada si no se ha alcanzado la calificación mínima necesaria como para considerarse bien comprendida previamente.

Pero un dominio no sirve de mucho si no le proporciona un estado inicial y una meta específica que cumplir. Estos dos elementos se describen en el fichero de problema de planificación cuya generación automática se detalla en la siguiente sección.

2.2. Problema Jerárquico

Un problema de planificación en PDDL de tipo jerárquico se puede generar automáticamente según lo señalado en el el algoritmo de la figura 28.

Primero se definen las sentencias para describir el estado inicial de cada uno de los alumnos del curso. Estas sentencias representan precondiciones dentro del dominio y se obtienen de la información proporcionada acerca del perfil del estudiante.

Las sentencias que se producirán automáticamente para cada estudiante han sido descritas anteriormente usando como base la información de la tabla 4 que describe el estado inicial de la estudiante Ana para el curso de matemáticas discretas de la figura 3. Así, se generarán para cada estudiante dichas sentencias dependiendo de sus características específicas, tal y como se describe en la primera parte del algoritmo de la figura 28.

Una vez generadas las sentencias del estado inicial para cada uno de los estudiantes del curso es preciso generar las metas que se pretenden conseguir una vez ejecutado el

planificador. El planificador SIADEX permite introducir n número de tareas como metas a seguir.

Las metas de un planificador HTN para dominios de e-learning generalmente hacen referencia a l objetivo (o conjunto de objetivos) principal de un curso y se generan tantas metas como objetivos para ese curso tenga el estudiante.

Generar un estado inicial

- Para cada estudiante $\langle s \rangle$ y cada atributo $\langle a \rangle$ de dicho estudiante
 1. Si el atributo $\langle a \rangle$ es numérico entonces traducir dicho atributo a una función
(= ($\langle a \rangle \langle s \rangle$) $\langle value \rangle$)
 2. Si $\langle a \rangle =$ *Language Level* del tipo *lang*, entonces:
si su valor es *Excelent* o *Acceptable* traducir a la función
(= ($\langle a \rangle \langle lang \rangle \langle s \rangle$) 90)
o, si tiene los valores *Low* o *Poor* entonces traducir a la función
(= ($\langle a \rangle \langle lang \rangle \langle s \rangle$) 40)
 3. Si $\langle a \rangle =$ *Performance*, entonces: si su valor es *High* traducir a la función
(= ($\langle a \rangle \langle s \rangle$) 90)
o, si tiene los valores *Medium* o *Low* entonces traducir a la función
(= ($\langle a \rangle \langle s \rangle$) 60)
 4. Si el atributo no es igual a *Language Level* o *Performance* y tampoco es numérico, entonces traducir al predicado
($\langle a \rangle \langle s \rangle \langle value \rangle$)

Generar las metas

- Para cada estudiante $\langle s \rangle$ y cada meta $\langle g \rangle$ de dicho estudiante
 1. Si el estudiante tiene el valor *Much* en su atributo *Available Time* traducir la meta como una meta simple en el formalismo HTN-PDDL
($\langle g \rangle \langle s \rangle$).
 2. Si el estudiante tiene restricciones de tiempo, esto es, que la meta $\langle g \rangle$ para $\langle s \rangle$ tenga que terminar en un punto $\langle d \rangle$ dentro del intervalo de tiempo máximo y mínimo establecido para $\langle g \rangle$, entonces traducirla como una meta con deadline en el formalismo HTN-PDDL
($\langle = \ ?end \langle d \rangle \langle g \rangle \langle s \rangle$)

Figura 28. Algoritmo para la generación automática del problema de planificación HTN a partir de la información proporcionada por el estándar IMS-LIP.

Las metas pueden ser de dos tipos dependiendo de la disponibilidad temporal del estudiante.

Cuando el estudiante tiene mucha disponibilidad temporal se genera una meta por medio de un predicado simple del tipo (*objetivo**estudiante*) para cada objetivo del curso.

Sin embargo, cuando el estudiante tiene poca disponibilidad temporal para llevar a cabo todos los objetivos del curso la meta consta de dos partes:

- La sentencia (`<= ?end 12150`) que establece la cantidad de minutos máxima de que dispone el estudiante para terminar el curso, y
- La sentencia (`DiscreteMaths ana`) que se refiere al objetivo principal o *tarea compuesta principal* del curso en cuestión para dicho estudiante, en este caso Ana.

3. Dominios y Problemas de Planificación para Planificadores Basados en Estados

Así como para los planificadores HTN se han descrito en la sección anterior un conjunto de algoritmos de ingeniería de conocimiento que permiten generar automáticamente un dominio y un problema de planificación a partir de la metainformación descrita en el capítulo 3, también para los planificadores *Basados en Estados* es preciso definir otro conjunto de algoritmos ya que la naturaleza, particularmente del dominio, es sustancialmente diferente a la HTN pues solo consta de acciones durativas en su composición.

Por otro lado, estructura y composición de un problema de planificación basado en estados no difiere mucho de la de un HTN, sin embargo, esa mínima diferencia representa un cambio de enfoque sustancial con respecto a la selección de acciones que formarán parte de un plan o diseño instruccional adaptado para cada estudiante.

Aunque existen muchas propuestas para generar automáticamente dominios y problemas de planificación para el contexto de e-learning con planificadores basados en estados [Camacho et al., 2007, Kontopoulos et al., 2008], ninguno de ellos ha generado su dominio o problema basándose única y exclusivamente en la información proporcionada por los metadatos de los estándares de e-learning.

Una primera aproximación a los algoritmos de generación automática de dominios y problemas de e-learning para planificadores basados en estados [Morales et al., 2008a]. En las siguientes secciones se detallarán dichos algoritmos.

3.1. Dominio Basado en Estados

A diferencia de los planificadores jerárquicos, un planificador basado en estados no tiene tareas compuestas que formen parte del dominio de planificación en PDDL. Esta es la razón principal por la cual se han tenido que diseñar algoritmos diferentes para traducir los metadatos de los recursos a un dominio de planificación donde las relaciones de estructura jerárquica entre ellos pudieran ser codificadas de alguna manera dentro de los diferentes tipos de acciones primitivas que permite el planificador elegido, es decir, LPG-td.

Por lo tanto, para poder generar automáticamente el dominio de planificación basado en estados, es necesario pasar por tres etapas:

1. La primera consiste en la generación de una estructura secuencial de objetos de aprendizaje ordenados \mathcal{R}_v para cada valor v de la regla de ordenación \mathcal{R} , proceso que se llevará a cabo a través del algoritmo de ingeniería de conocimiento de la figura 29.
2. La segunda permite la generación de acciones primitivas durativas y sus respectivas condiciones y efectos a partir del análisis de cada uno de los elementos $o \in \mathcal{R}_v$, sus metadatos y su orden secuencia obtenido en la etapa anterior. Dicho algoritmo se detalla en la figura 30.
3. La tercera etapa corresponde a la generación de acciones primitivas **no** durativas que, aunque no aparecerán en el despliegue final del diseño instruccional, servirán de auxiliares para que sea posible generar “siempre” un plan aunque el estudiante no cumpla con las condiciones temporales ni de rendimiento requeridas por su objeto generador.

Las últimas dos etapas deberán ejecutarse de manera paralela.

3.1.1. Estructuras Secuenciales de Objetos de Aprendizaje Ordenados

Cuando un planificador jerárquico genera un plan de acción para un dominio de e-learning, los únicos elementos que aparecen dentro del plan son aquellos que pertenecen al conjunto \mathcal{P}_o o aquellos que $o \in \mathcal{C}_o$ que tienen algún valor en su metadato *Typical Learning Time*, es

decir, aquellos objetos que finalmente terminan siendo representados como acciones durativas dentro del dominio de planificación.

En un dominio de planificación basado en estados solo se dispone de acciones (durativas o no), por lo tanto, las precondiciones y los efectos de las acciones juegan un papel fundamental cuando el planificador intenta decidir cual es la secuencia de objetos de aprendizaje que mejor se adapta al perfil de un estudiante.

Para esto, no solo será preciso definir las condiciones relacionadas con las características del perfil que el estudiante debe tener para llevarla a cabo, si no también la acción o acciones que deberán ejecutarse previamente en base a los distintos perfiles y que serán incluidas como *precondiciones de orden*.

Es conocido que los objetos que forman parte de una tarea compuesta se pueden ordenar en base a dos criterios distintos: las reglas de ordenación explícitas que toman en cuenta la relación *Is Based On* entre los objetos de aprendizaje y las reglas de ordenación implícitas que se basan en los valores de una regla de ordenación para encontrar las distintas configuraciones de objetos dependiendo de alguna característica de los mismos que esté relacionada con otra del perfil del estudiante.

Extrapolando, se puede entonces inferir que los objetos de aprendizaje de todo el plan tendrán tantas secuencias (o configuraciones) distintas de ordenación como valores de la regla implícita que se considere. Y es de ahí de donde surge el algoritmo de la figura 29.

Este algoritmo permite llevar a cabo un “preprocesamiento” de los $o \in \mathcal{O}$ que, dado un número de valores n para la *regla implícita* de ordenación seleccionada para el dominio particular, permite recorrer n veces el árbol formado por todos $o \in \mathcal{O}$ y que tienen como raíz un $o \in \mathcal{C}_o$, que a su vez es $o \in \mathcal{G}$.^{5, 6} Es decir, permite recorrer n veces el árbol de objetos de aprendizaje que forman parte de la subestructura jerárquica de un objeto compuesto meta y extraer una secuencia de objetos primitivos o cuyo metadato *learning – resource – type(o)*

⁵ El recorrido de ese árbol será a profundidad tomando en cuenta la relación *IsPartOf* y de izquierda a derecha de acuerdo a las relaciones *IsBasedOn* y *Requires*.

⁶Sin embargo, se puede decir que esta parte del proceso únicamente es necesaria cuando los objetos de aprendizaje del repositorio están agrupados en capítulos, temas, subtemas y así sucesivamente. Este es el caso de la mayoría de los repositorios, aunque pueden existir algunos que no tengan estructura jerárquica alguna y en este caso solo se ordenarían todos los $o \in \mathcal{O}$ en tantas veces como valores de la regla de ordenación.

Dado, $o \in \mathcal{C}_o$ que a su vez es $o \in \mathcal{G}$, un conjunto de objetos vacío \mathcal{R}_v que contendrá los objetos de aprendizaje $o \in \mathcal{O}$ ordenados en base a uno de los valores de orden v de la Regla Implícita \mathcal{R} y \mathcal{L}_o el conjunto de listas temporales de objetos de aprendizaje ordenadas l que se irán formando durante el algoritmo.

Para cada $v \in \mathcal{R}$,

1. Si $o \in \mathcal{C}_o$ y $\neg \exists LearningResourceType(o)$, entonces
 - a) Si $\exists l \in \mathcal{L}_o | o \in l$, entonces eliminar o como elemento de su l correspondiente.
 - b) Obtener el conjunto l' de objetos de aprendizaje tal que $l' = \{o' | o' \in is - part - of(o) \vee o' \in requires(o)\}$,
 - c) Si existen varios o' con el mismo $title(o')$, dejar solo uno de ellos en la lista l y ordenar los $o' \in l$ restantes en base a la relación de orden $IsBasedOn$ y/o al v actual, además de poner al principio de la ordenación todos los $o' \in requires(o)$ y al final todos los $o' \in \mathcal{C}_o$.
 - d) Asignar $\mathcal{L}_o = \mathcal{L}_o \cup l'$, situando l' como el primer elemento accesible de \mathcal{L}_o .
 - e) Asignar $o = o'$ siendo o' el primer elemento en la ordenación de l' y pasar a 1.
2. Si $o \in \mathcal{C}_o$ y $\exists LearningResourceType(o)$, añadir $\mathcal{R}_v = \mathcal{R}_v \cup o$ como elemento final en la ordenación de los $o \in \mathcal{R}_v$ y pasar a 1-a.
3. Si $o \in \mathcal{P}_o$,
 - a) Añadir $\mathcal{R}_v = \mathcal{R}_v \cup o$ como elemento final en la ordenación de los $o \in \mathcal{R}_v$ y eliminar o como elemento de l .
 - b) Si $\exists o' \in l$ tal que o' sea ahora el primer elemento en la ordenación de l , entonces Asignar $o = o'$ y pasar a 1.
 - c) Si $l = \emptyset$, entonces quitar l de \mathcal{L}_o y
 - 1) Si $\mathcal{L}_o = \emptyset$
Se ha terminado con la ordenación de \mathcal{R}_v , por lo tanto, si existe un $v' \in \mathcal{R}$ sin revisar continuar el algoritmo para ordenar su $\mathcal{R}_{v'}$ correspondiente a partir del $o \in \mathcal{G}$ inicial, de otro modo, terminar el algoritmo.
 - 2) Si $\mathcal{L}_o \neq \emptyset$, revisar el $l \in \mathcal{L}_o$ que se encuentra al principio de la ordenación y pasar a 3-b.

Figura 29. Algoritmo para formar conjuntos ordenados \mathcal{R}_v de los objetos de aprendizaje primitivos que permiten satisfacer los objetivos de un objeto compuesto $o \in \mathcal{G}$.

exista y extraerla de manera ordenada en base al valor v de la regla que se esté analizando.

Así pues, empezando por el objeto $o \in \mathcal{G}$, lo primero que se hace en este algoritmo es revisar si dicho objeto pertenece al conjunto de las tareas compuestas y no tiene el metadato *Learning Resource Type* y, si esto es así, ordenar todos los objetos que formen parte de ese o o sean requeridos por el mismo en base, primero a la regla de ordenación explícita, y después al valor de la implícita que se esté tomando en cuenta en ese recorrido. Una vez ordenados los objetos que componen a o , éstos se añaden como lista l' al principio de la estructura ordenada \mathcal{L}_o y se toma como nueva o al primer elemento de l' .

En caso de que un objeto compuesto sí tenga asociado un metadato *Learning Resource Type*, entonces éste se añadirá al final de la secuencia ordenada \mathcal{R}_v y después se procesará

como objeto compuesto sin tipo de recurso de aprendizaje asociado, pero, si anteriormente dicho objeto se encontraba en una lista l entonces tendrá que ser sustraído de la misma.

Por otro lado, si el objeto que se está analizando forma parte del conjunto \mathcal{P}_o entonces, habrá que añadirlo a \mathcal{R}_v y eliminarlo de su l correspondiente, para después decidir si:

- Se revisará el siguiente $o \in l$ del l actual, o
- Se eliminará el l actual por que esté vacío y revisará el o del siguiente l , o
- Se terminará la integración del \mathcal{R}_v actual y se pasará a conformar el $\mathcal{R}_{v'}$ del siguiente valor v' de la regla \mathcal{R} , o
- Se terminará el algoritmo.

La estructura \mathcal{L}_o permite entonces, recorrer la estructura jerárquica del árbol para la meta elegida, al incluir todos y cada uno de los conjuntos de acciones parte de objeto compuesto que aparezcan en él en el orden y profundidad adecuados.

Al final de la aplicación del algoritmo, por lo tanto, se obtendrán n secuencias ordenadas de \mathcal{R}_v que se utilizarán en la siguiente etapa para crear las acciones primitivas durativas del dominio.

3.1.2. *Acciones Primitivas Durativas*

Una vez que se han creado las secuencias ordenadas \mathcal{R}_v se puede proceder a generar automáticamente todas y cada una de las acciones durativas del dominio de planificación basado en estados; acciones que están directamente relacionadas con el conjunto de objetos de aprendizaje que formarán parte de las actividades a ejecutar por el estudiante y, por lo tanto, tienen una duración.

Se hace hincapié en lo anterior por que, si se espera que durante el proceso de planificación no ocurran errores debido al incumplimiento de condiciones débiles, entonces se propone hacer uso de acciones no durativas auxiliares que se introduzcan dentro del plan en su lugar pero que, al ser fácilmente identificables, no sean incluidas dentro del itinerario formativo final.

Para cada $o \in \mathcal{R}_v$, siendo v cualquiera de los valores de la regla implícita \mathcal{R} para este dominio tal que $v \in \mathcal{R}$:

1. Seleccionar $o \in \mathcal{R}_v$ no explorado aún
2. Crear una acción durativa a con un solo parámetro $?student$, duración = $TypicalLearningTime(o)$, nombre = $Identifier(o)$ y efecto = $(at\ end\ (Identifier(o)\ ?student\ done))$.
3. Las condiciones de a se formarán de la siguiente manera: $conditions(a) = conditions(a) \cup (at\ start\ (and$
 - a) Si $Coverage(o) = optional$, entonces $conditions(a) = conditions(a) \cup (availability\ ?student\ much)$.
 - b) Si $Difficulty(o) = difficult$ o $very\ difficult$, entonces $conditions(a) = conditions(a) \cup (>= (performance\ ?student)\ th)$.
 - c) Si $Language(o) \neq es$, entonces $conditions(a) = conditions(a) \cup (>(language-level\ language(o)\ ?student)\ th)$.
 - d) Si $OtherPlatformRequirements(o) = multimedia$, entonces $conditions(a) = conditions(a) \cup (equipment\ ?student\ multimedia)$.
 - e) Si o no es el primer elemento de \mathcal{R}_v para algún v , entonces $conditions(a) = conditions(a) \cup (or$
 - f) Para cada v , si o no es el primer elemento de \mathcal{R}_v , entonces
 - 1) $conditions(a) = conditions(a) \cup (and\ (nombre(\mathcal{R})\ ?student\ v)$.
 - 2) Siendo $o' \in \mathcal{R}_v$ el elemento anterior a o para la ordenación del conjunto \mathcal{R}_v ,
 - i) Si $Coverage(o') = optional$, entonces $conditions(a) = conditions(a) \cup (or\ (noopt-identifier(o')\ ?student\ done)\ (identifier(o')\ ?student\ done))$.
 - ii) Si $Coverage(o') \neq optional$, pero $Difficulty(o') = difficult$ o $very\ difficult$, entonces $conditions(a) = conditions(a) \cup (or\ (nodif-identifier(o')\ ?student\ done)\ (identifier(o')\ ?student\ done))$
 - iii) En cualquier otro caso $conditions(a) = conditions(a) \cup (identifier(o')\ ?student\ done)$
 - 3) Cerrar las condiciones que abarcan a todas los v , de manera que $conditions(a) = conditions(a) \cup)$
 - g) Si o no es el primer elemento de \mathcal{R}_v para algún v , entonces cerrar las condiciones que involucran a todos los v $conditions(a) = conditions(a) \cup)$

Figura 30. Algoritmo para crear las acciones primitivas a de un dominio de planificación **no** HTN.

El algoritmo en esencia es muy sencillo, y pretende recorrer una de las secuencias ordenadas \mathcal{R}_v y explorar cada uno de los o que la componen. Permitirá crear una acción con el nombre del identificador de o , su duración a partir del tiempo típico de aprendizaje de o y su efecto como $(at\ end\ (Identifier(o)\ ?student\ done))$, que significa que al final de la acción está será considerada como hecha.⁷

La tercera parte del proceso es la clave para que las acciones puedan ser ordenadas correctamente por el planificador y que elimina las barreras que la estructura jerárquica de un curso representa para el mismo. Esta parte del algoritmo se encarga de introducir en la acción las condiciones que se deben cumplir al inicio de la misma y que están relacionadas

⁷Cabe mencionar que, cada vez que se genera un efecto, habría de generarse también un nuevo predicado $(Identifier(o)\ ?student\ done)$.

con la opcionalidad, dificultad, idioma y requerimientos técnicos del objeto.

Las cuatro precondiciones anteriores, en caso de existir, se agregan a la condición global concatenadas por una disyunción y posteriormente se revisan uno a uno todos los \mathcal{R}_v , localizando el objeto o dentro de cada una y revisando el o' anterior. De manera que, sabiendo el o' anterior para el valor v de la regla \mathcal{R} , se podrán generar precondiciones del tipo: `(and (nombre(\mathcal{R}) ?student v) (identifier(o') ?student done))` O, del tipo `(and (nombre(\mathcal{R}) ?student v) (or (noopt/nodif-identifier(o') ?student done) (identifier(o') ?student done)))` en caso de que el objeto o' tenga condiciones de opcionalidad o dificultad.

Cualquiera de estos casos se pueden observar en la representación de la acción *Boolean Algebra Simulation* en la figura 31, donde *Minimization Of Circuits Slides* es el o' para el valor *theoretical* de la regla *learning-style* y *LS Narrative Text* es el o' para el valor *pragmatic*. Pero, como *LS Narrative Text* es también opcional, entonces se agrega por medio de una conjunción la condición que incluya al mismo objeto con el prefijo *noopt-*.

```
(:durative-action BooleanAlgebraSimulation
:parameters(?student - student)
:duration(= ?duration 10)
:condition(at start
  (and
    (equipment ?student multimedia)
    (>= (performance ?student) 50)
    (or
      (and
        (learning-style ?student theoretical)
        (MinimizationOfCircuitsSlides ?student done))
      (and
        (learning-style ?student pragmatic)
        (or
          (LSNarrativeText ?student done)
          (noopt-LSNarrativeText ?student done))))))
)
:effect(at end (BooleanAlgebraSimulation ?student done)))
```

Figura 31. Acción durativa *BooleanAlgebraSimulation* creada de acuerdo al algoritmo de la figura 30.

Dentro de la generación de precondiciones de orden, juega un papel clave la inclusión del predicado `action-name ?student done` en los efectos de cada acción, siendo `action-name` el

nombre de la acción para la que se genere el efecto. Este tipo de estrategia para detectar cuando la acción representativa de un objeto de aprendizaje ya se ha llevado a cabo ha sido utilizada anteriormente en otros trabajos como el de [Garrido and Onaindía, 2010].

3.1.3. *Acciones Primitivas NO Durativas*

Por último, cada vez que se genera una acción primitiva por medio del algoritmo de la figura 30, es preciso revisar cuatro propiedades del objeto para el cual se ha generado la acción: su multiplicidad, su opcionalidad, su dificultad y lo que se ha denominado *estado meta*.

En caso de que cualquiera de estas propiedades se cumpliera, será preciso generar una o mas acciones (o condiciones) según lo recomendado por el algoritmo de la figura 32.

Multiplicidad

En el algoritmo de la figura 29 solo se incluye uno de los objetos de aprendizaje que tengan el mismo nombre y el mismo padre en la ordenación ya que la mayor parte de los planificadores basados en estados solo toma en cuenta, durante el proceso de planificación, el último objeto descrito en el dominio y los demás los omite. Además, si se intentara incluirlos, la complejidad de las estructuras y el algoritmo aumentaría sustancialmente.

Sin embargo, y como valor añadido, en este algoritmo (fig. 32) si que se generan acciones primitivas durativas para los objetos de aprendizaje que tengan el mismo que nombre que otro para el cual ya se haya generado una acción, teniendo en cuenta que pueden existir otros planificadores que si soporten búsquedas sobre múltiples acciones con el mismo nombre.

Además, permite aprovechar la multiplicidad de tareas para que “siempre” se puedan generar planes ya que, aunque la tarea que se plasme dentro del plan sea la última generada en el dominio, las condiciones pueden ser (al menos en las pruebas llevadas a cabo hasta ahora) las de cualquiera de las tareas con ese nombre y, por lo tanto se evita el incumplimiento de condiciones.

1. Si existen uno o mas $o' \in \mathcal{O}$ con el mismo $title(o)$, entonces crear una acción a' idéntica a la a creada para ese o en la figura 30 en nombre, parámetro y efecto excepto por la duración que será igual al $typical-learning-time(o')$ y las condiciones agregadas en 3-a,b,c y d de la figura 30 que serán las de o' .
2. Si $Coverage(o) = \text{optional}$, entonces crear una nueva acción primitiva no durativa a' con nombre = $noopt-identifier(o)$, un único parámetro $?student$, efecto = $(noopt-Identifier(o) ?student done)$ y como precondiciones aquellas formadas en base a los puntos 3-e, 3-f y 3-g de la figura 30 para o .
3. Si $Coverage(o) \neq \text{optional}$, pero $Difficulty(o) = \{ \text{difficult} \vee \text{very difficult} \}$, entonces crear una nueva acción primitiva no durativa a' con nombre = $nodif-identifier(o)$, un único parámetro $?student$, efecto = $(nodif-Identifier(o) ?student done)$ y como precondiciones aquellas formadas en base a los puntos 3-e, 3-f y 3-g de la figura 30 para o .
4. Si no existe a' con nombre igual al de $Identifier(o')|o' \in \mathcal{G}$ es la meta para la cual se han creado las secuencias de objetos del algoritmo de la figura 29, entonces crear una acción primitiva no durativa a' con nombre $Identifier(o')$, el parámetro $?student$, el efecto $(identifier(o') ?student done)$ y las precondiciones abiertas a cualquier modificación descrita en 5.
5. Si o no es predecesor de ningún o' que pertenezca a alguno de los \mathcal{R}_v existentes, entonces agregar una precondición en la acción a' creada en 4 que se forme según los puntos 3-e, 3-f y 3-g de la figura 30 pero solo para los \mathcal{R}_v en los que o no sea predecesor de ningún o' y considerando al objeto o actual como el o' de dichos puntos.

Figura 32. Algoritmo para crear las acciones primitivas auxiliares a' , ya sea durativas o no, para un dominio de planificación **no** HTN.

Opcionalidad

Si el objeto o que se esté analizando tiene el metadato $Coverage(o) = \text{Optional}$, además de haber agregado la condición correspondiente en la acción generada en la sección anterior, también se tendrá que generar una acción no durativa alternativa a' que se ejecute en caso de que dicha condición no se cumpla.

La nueva acción no durativa tendrá como nombre el identificador de o precedido por la

```
(:action noopt-BooleanAlgebraSimulation
:parameters(?student - student)
:duration(= ?duration 10)
:precondition(or
  (and
    (learning-style ?student theoretical)
    (MinimizationOfCircuitsSlides ?student done))
  (and
    (learning-style ?student pragmatic)
    (or (LSNarrativeText ?student done)
        (noopt-LSNarrativeText ?student done))))
)
:effect(noopt-BooleanAlgebraSimulation ?student done))
```

Figura 33. Acción *nodif-BooleanAlgebraSimulation* creada de acuerdo a la segunda parte del algoritmo de la figura 30.

cadena `noopt-`, el efecto que corresponde a dicho nombre de acción `done` pero sin la condición `at end` ya que la acción es no durativa y un solo parámetro `?student`.

También se incluirán las precondiciones de orden que se formaron en el inciso *f*) de la figura 30, es decir, aquellas y solo aquellas precondiciones asociadas al valor de las reglas implícitas correspondientes.

Dificultad

Si el objeto *o* que se esté analizando tiene el metadato $Difficulty(o) = \{ \text{difficult} \vee \text{very difficult} \}$ y $Coverage(o) \neq \text{Optional}$, es decir, que sea difícil pero no opcional.

Entonces se generará una nueva acción primitiva no durativa *a'* cuyo nombre será $Identifier(o)$ con el prefijo `nodif-`, un solo parámetro `?student`, el efecto $(\text{nodif-}Identifier(o) \text{?student done})$ y las precondiciones dadas por el inciso *f*) del algoritmo de la figura 30.

Estado Meta

Finalmente, en el punto 4 se crea una acción primitiva no durativa *a'* que tendrá el nombre del objeto compuesto *o'* considerado como meta del curso, es decir, el objeto raíz del árbol de objetos que se esté analizando, un solo parámetro `?student`, el efecto $(\text{identifier}(o') \text{?student done})$.

Y en el punto 5 si no existe el sucesor del *o* analizado en ese momento en alguna de las secuencias \mathcal{R}_v se añadirá a la acción *a'* una precondición de orden por cada \mathcal{R}_v para el cual no exista sucesor de *o*. En dicha precondición se hará referencia a la precondición de los valores de la regla implícita y a *o* como antecesor de la acción meta *a'* para dicho valor, así como a su acción opcional o difícil en caso de que existiera.

Estos dos últimos puntos, por lo tanto, indican que serán precondiciones de la acción meta *a'* todas aquellas acciones derivadas de los objetos *o* que se encuentren al final de una o varias secuencias de objetos \mathcal{R}_v , a lo cual se le ha nombrado estar en estado meta.

Al terminar de ejecutar de forma paralela estos dos algoritmos, será posible obtener

todas las acciones primitivas durativas que ejecutará el estudiante durante la aplicación de su diseño instruccional y las acciones primitivas no durativas que servirán de apoyo al planificador para encontrar, *siempre que el curso esté etiquetado según lo descrito en esta memoria*, planes formativos que se adapten al perfil de cada estudiante del curso.

3.2. Problema Basado en Estados

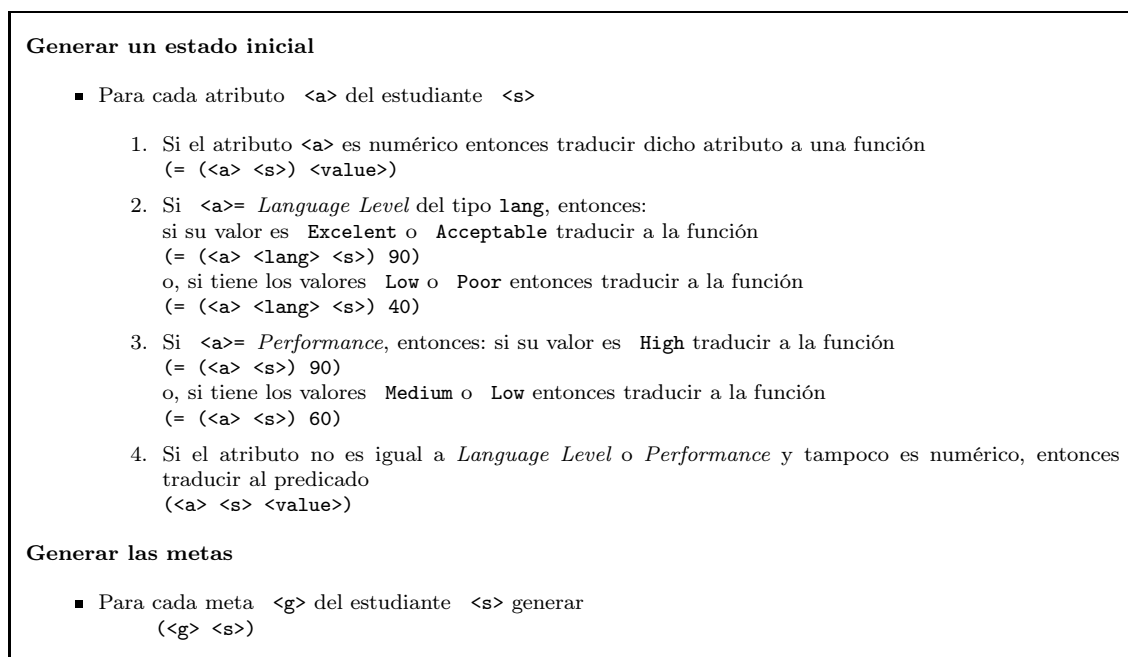


Figura 34. Algoritmo para la generación automática del problema de planificación **no** HTN a partir de la información proporcionada por el estándar IMS-LIP.

Durante la generación automática de un problema de planificación basado en estados, la cual se describe en el algoritmo de la figura 34, es posible observar como el estado inicial de dicho problema se genera exactamente igual que el estado inicial de un problema para un planificador HTN debido a que, para la generación del plan, se contemplan las mismas características del perfil del estudiante solo que se integran y utilizan de formas distintas por el dominio de planificación.

Para los planificadores basados en estados la meta se describe a través de una función. Esta función puede, o bien incluir sentencias del tipo función, o sentencias declarativas; pero no las dos en la misma función.

La generación automática de una meta, en este caso y dado que lo que interesa a los dominios e-learning es alcanzar un objetivo concreto, no contempla la generación de restricciones temporales sobre las metas generadas y mucho menos una función de optimización sobre los objetivos del curso. Las metas se expresan mediante sentencias declarativas que expresan el objetivo principal del curso para cada estudiante y no hacen referencia a una tarea compuesta, como en el caso de las metas para problemas HTN, si no a una acción no durativa que sería la última acción a incluir en el plan para cualquiera de las ordenaciones posibles según el número de valores de la regla implícita de ordenación elegida.

4. Planes Generados (Diseños Instruccionales)

Para comprobar que los algoritmos descritos en las secciones anteriores funcionan correctamente se han diseñado algunos experimentos para comprobar que los planes generados por los planificadores representen diseños instruccionales consistentes.

Para tal propósito se han generado automáticamente los dominios de dos paquetes SCORM:

- Uno para el curso **Discrete Maths** de la figura 3 con 23 objetos de aprendizaje: 12 primitivos y 6 compuestos. Este mini-curso se utilizará como ejemplo corto para esta memoria.
- Uno de los objetivos del seminario a distancia “Educación e Inteligencia Artificial” impartido como parte de las actividades de divulgación de esta memoria y experimentación real de esta memoria. Este objetivo/meta resulta interesante para los propósitos de esta fase de la experimentación ya que cuenta con 95 objetos de aprendizaje: 68 primitivos y 27 compuestos, número promedio de objetos de aprendizaje que un SCORM complejo podría contener.

El estado inicial del problema que se ha utilizado con dichos dominios se ha generado con el perfil de 36 estudiantes reales, de los cuales se han elegido 3 como perfiles representativos. En la tabla 5 se muestran los perfiles de Chris, Thom y María;⁸ y podemos observar como

⁸Los nombres se han cambiado para conservar el anonimato de los estudiantes.

Thom tiene un rendimiento alto al igual que María, un estilo de aprendizaje pragmático y mucha disponibilidad temporal a diferencia de Chris y María; Chris es el único que ha cursado y pasado el objetivo de **Logic and Sets** previamente, pero comparte el resto de sus características con María, incluido su estilo de aprendizaje, su baja disponibilidad horaria, su buen nivel de inglés y su disponibilidad multimedia.

Chris	Thom	María
(= (performance chris) 40)	(= (performance thom) 90)	(= (performance maria) 90)
(learning-style chris theoretical)	(learning-style thom pragmatic)	(learning-style maria theoretical)
(= (language-level en chris) 90)	(= (language-level en thom) 40)	(= (language-level en maria) 90)
(= (score chris LogicAndSets) 90)	(= (score thom LogicAndSets) 40)	(= (score maria LogicAndSets) 40)
(available-time chris little)	(available-time thom much)	(available-time maria little)
(equipment chris multimedia)		(equipment maria multimedia)

Tabla 5. Perfiles de los estudiantes del curso de Matemáticas Discretas que han sido generados automáticamente en el lenguaje PDDL para su descripción dentro del estado inicial del archivo de problema.pddl.

Así, dados los estados iniciales de la tabla 5 y las metas correspondientes a los dos SCORMS anteriores, se han generado con los planificadores SIADEX y LPG-td planes o diseños instruccionales para cada uno de estos estudiantes.

En las siguientes secciones se mostrarán dichos planes para comprobar que el proceso de adaptación ha resultado ser exitoso y mostrar las diferencias entre los planes generados por los diferentes paradigmas para un mismo estudiante, así como entre las capacidades y procesos soportados por cada uno de los planificadores. Sin embargo, antes de mostrar los planes y para tener una visión más clara y resumida de los mismos, es preciso asociar cada objeto del curso de *Discrete Maths* con su respectivo identificador. Esta asociación se puede observar en la tabla 6, donde también se muestra el tipo de recurso de aprendizaje de cada objeto (si procede), así como sus respectivos requisitos ya sea de idiomas, equipamiento, opcionalidad y/o dificultad. Además, los objetos se muestran en diferentes niveles de enumerados, siguiendo la estructura jerárquica del curso descrita por las figuras 3 y 5.

4.1. Jerárquico

Los planes generados para Thom, Chris y María por el planificador jerárquico SIADEx y que se muestran en la tabla 7 se adaptan claramente a todas y cada una de las especificaciones planteadas en el perfil de cada uno de ellos.

* <i>Discrete Maths</i>	ITEM-task1
o <i>Algorithms</i>	ITEM-task2
+ <i>Basic Algorithms Narrative Text</i>	ITEM-action1
+ <i>Complexity</i>	ITEM-task4
# <i>Complexity Problem Statement</i>	ITEM-action5
# <i>Complexity Exercise</i>	ITEM-action6
# <i>Complexity Experiment</i>	ITEM-action7
# <i>Complexity Experiment</i>	ITEM-action8
Equipment Access - Language_English	
o <i>Boolean Algebra</i>	ITEM-task3
+ <i>Logic and Sets</i>	ITEM-task6
# <i>Logic and Sets Experiment</i>	ITEM-action17
# <i>Logic and Sets Exercise</i>	ITEM-action16
# <i>Logic and Sets Problem Statement</i>	ITEM-action14
# <i>Logic and Sets Problem Statement</i>	ITEM-action15
Equipment Access - Language_English	
# <i>Logic and Sets Narrative Text</i>	ITEM-action13
Coverage: Optional	
+ <i>Boolean Algebra Simulation</i>	ITEM-action2
Equipment Access,Difficulty: Very Difficult	
+ <i>Boolean Algebra Simulation</i>	ITEM-action3
Difficulty: Very Difficult	
+ <i>Minimization of Circuits Slide</i>	ITEM-action4
+ <i>Logic Gates</i>	ITEM-task5
# <i>Logic Gates1 Exercise</i>	ITEM-action10
# <i>Logic Gates2 Narrative Text</i>	ITEM-action11
# <i>Logic Gates2 Exercise</i>	ITEM-action12
Coverage: Optional	
# <i>Logic Gates1 Narrative Text</i>	ITEM-action9

Tabla 6. Objetos de Aprendizaje en relación a sus identificadores.

El plan de Chris es el mas corto debido a que ya había cursado previamente el tema de *Logic and Sets-task6*, con lo cual se descartan automáticamente 4 acciones primitivas y una tarea compleja del plan; por otro lado Chris tampoco cuenta con mucha disponibilidad horaria, por lo que los objetos opcionales *Logic Gates 2 Exercise-action12* y *Logic and Sets Narrative Text-action13* tampoco se incluyen dentro de su plan y, finalmente, tampoco se incluye el objeto *Boolean Algebra Simulation-action2/3* debido a que su rendimiento es bastante bajo y estos objetos son de alta dificultad.

Por otro lado, el plan de Thom es el mas largo, ya que ni ha cursado *Logic and Sets-task6*, ni tiene restricciones temporales lo cual permite la inclusión de los dos objetos

opcionales y, como en sus cursos anteriores ha tenido un rendimiento mas bien alto, entonces podrá optar por cursar actividades dificiles como *Boolean Algebra Simulation-action3*, que no la *action2* ya que no tiene suficientes capacidades de equipamiento multimedia.

<i>Chris</i>	<i>Thom</i>	<i>María</i>
<ul style="list-style-type: none"> * ITEM-task1 <ul style="list-style-type: none"> o ITEM-task2 <ul style="list-style-type: none"> + ITEM-action1 + ITEM-task4 <ul style="list-style-type: none"> # ITEM-action5 # ITEM-action6 # ITEM-action8 o ITEM-task3 + ITEM-action4 + ITEM-task5 <ul style="list-style-type: none"> # ITEM-action9 # ITEM-action10 # ITEM-action11 	<ul style="list-style-type: none"> * ITEM-task1 <ul style="list-style-type: none"> o ITEM-task2 <ul style="list-style-type: none"> + ITEM-action1 + ITEM-task4 <ul style="list-style-type: none"> # ITEM-action5 # ITEM-action6 # ITEM-action7 o ITEM-task3 <ul style="list-style-type: none"> +ITEM-task6 <ul style="list-style-type: none"> # ITEM-action17 # ITEM-action16 # ITEM-action14 # ITEM-action13 + ITEM-action3 + ITEM-action4 + ITEM-task5 <ul style="list-style-type: none"> # ITEM-action10 # ITEM-action11 # ITEM-action12 # ITEM-action9 	<ul style="list-style-type: none"> * ITEM-task1 <ul style="list-style-type: none"> o ITEM-task2 <ul style="list-style-type: none"> + ITEM-action1 + ITEM-task4 <ul style="list-style-type: none"> # ITEM-action5 # ITEM-action6 # ITEM-action8 o ITEM-task3 <ul style="list-style-type: none"> +ITEM-task6 <ul style="list-style-type: none"> # ITEM-action15 # ITEM-action17 # ITEM-action16 + ITEM-action4 + ITEM-action2 + ITEM-task5 <ul style="list-style-type: none"> # ITEM-action9 # ITEM-action10 # ITEM-action11

Tabla 7. Planes generados por un planificador HTN para los perfiles de estudiantes representativos Thom, Chris y María utilizando el dominio del curso *Discrete Maths* .

Finalmente podemos ver que el plan de María es mas bien equilibrado. Ella no ha cursado el tema *Logic and Sets-task6* por lo cual ha sido necesario incluirlo en su plan junto a los objetos que forman parte del mismo. Sin embargo, como su disponibilidad horaria es mas bien baja, se ha tenido que omitir la inclusión de dos objetos opcionales pero, por otro lado, se ha podido incluir el objeto difícil *Boolean Algebra Simulation-action2* ya que, a diferencia de Thom, ella si cuenta con equipamiento multimedia.

Respecto a la ordenación de los objetos, se puede observar que Chris y María tienen una ordenación idéntica, excepto por el objeto extra que se ha incluido en el plan de María, y esto es por que los dos tienen estilos de aprendizaje que equivalen a *theoretical*. Pero Thom, al tener un estilo de aprendizaje con valor *pragmatical* tiene una ordenación muy diferente a

las de Chris y María por que a pesar de que incluye los objetos parte de la `task4`, a partir de la misma, estos objetos se encuentran en un orden totalmente diferente. Por ejemplo, las acciones 9, 10 y 11 para Chris y María, se encuentran en el orden 10, 11 y 9 para Thom; y todo esto debido al valor de sus estilos de aprendizaje e la conveniencia de la regla implícita de ordenación que dice que unos tipos de recursos deben de mostrarse primero que otros dependiendo del estilo de aprendizaje de un estudiante.

<i>Chris</i>	<i>Thom</i>	<i>María</i>
<pre> * ITEM-task1 o ITEM-task2 + ITEM-action1 + ITEM-task4 # ITEM-action5 # ITEM-action6 # ITEM-action8 o ITEM-task3 & +ITEM-task6 & # ITEM-action15 & # ITEM-action17 & # ITEM-action16 + ITEM-action4 + ITEM-task5 # ITEM-action9 # ITEM-action10 # ITEM-action11 </pre>	<pre> * ITEM-task1 o ITEM-task2 + ITEM-action1 + ITEM-task4 # ITEM-action5 # ITEM-action6 # ITEM-action8 & # ITEM-action8 o ITEM-task3 +ITEM-task6 # ITEM-action17 # ITEM-action16 & # ITEM-action15 # ITEM-action13 + ITEM-action3 + ITEM-action4 + ITEM-task5 # ITEM-action10 # ITEM-action11 # ITEM-action12 # ITEM-action9 </pre>	<pre> * ITEM-task1 o ITEM-task2 + ITEM-action1 + ITEM-task4 # ITEM-action5 # ITEM-action6 # ITEM-action8 o ITEM-task3 +ITEM-task6 # ITEM-action15 # ITEM-action17 # ITEM-action16 + ITEM-action4 & + ITEM-action3 + ITEM-task5 # ITEM-action9 # ITEM-action10 # ITEM-action11 </pre>

Tabla 8. Planes generados por un planificador LPG para los perfiles de estudiantes representativos Thom, Chris y María utilizando el dominio del curso *Discrete Maths* .

4.2. Basado en Estados

Normalmente los planes generados por un planificador basado en estados, como LPG-td, deberían de ser iguales a los generados por un planificador jerárquico. Sin embargo, si se comparan los planes de la tabla 7 con los de la tabla 8 es posible constatar que los segundos, es decir, los generados por el planificador basado en estados, contienen una serie de errores derivados de la imposibilidad de representar acciones múltiples y tareas requeridas dentro del dominio de planificación generado bajo el paradigma de planificación basada en estados.

La representación de deadlines, la cual también está restringida en la mayoría de los planificadores basados en estados existentes, no se encuentra representada en este ejemplo pero si comentada en las siguientes secciones.

Los errores en la generación de planes para los estudiantes representativos Chris, Thom y María, utilizando el planificador LPG-td, están señalados en la tabla 8 por medio del signo & que precede a cada acción que no debería aparecer en el plan.

En el plan de Chris se puede observar como, a pesar de que el había cursado previamente el tema requerido *Logic and Sets-task6* y todos los objetos que forman parte del mismo, estos se incluyen en su plan. Lo anterior ocurre por que no se ha podido diseñar una manera sencilla de representar las condiciones de “opcionalidad” sobre un objeto compuesto dentro de un dominio basado en estados y mucho menos de generar automáticamente condiciones tan complejas sin elevar la complejidad y, por lo tanto, reducir la eficiencia, del algoritmo de generación automática del dominio basado en estados.

Algo similar ocurre con los errores en el plan de Thom, sin embargo, en este caso no se debe a la inclusión de objetos compuestos añadidos debido al metadato *Requires* si no a la de objetos múltiples que no coinciden con las características del perfil de Thom.

Como se comentaba durante la descripción del algoritmo de generación automática del dominio para planificadores basados en estados, la mayor parte de estos planificadores no soportan correctamente la inclusión de acciones con el mismo nombre lo cual obligaría a modificar el algoritmo para que generara precondiciones de orden mas complejas en torno a los n objetos que tuvieran el mismo nombre para cada subconjunto l y, por lo tanto reducir la simplicidad, generalización y eficacia del algoritmo. En este caso se ha optado por incluir en el dominio las acciones múltiples con el mismo nombre pero diferentes condiciones, pero el planificador lo que hace es tomar en cuenta las condiciones de las dos acciones para validar su cumplimiento por parte del estudiante y luego incluir solo la última acción que aparece en el dominio y no la que corresponde a las condiciones cumplidas.

Así, en lugar de incluir la acción *Complexity Experiment-action7* en el plan de Thom, ha incluido *Complexity Experiment-action8* cuyas pero Thom no tiene ni el equipamiento

ni el idioma necesarios para llevarla a cabo, y lo mismo ocurre con la acción *Logic and Sets Problem Statement-action15*; pero las ha elegido por que las acciones 8 y 15 se encuentran definidas después que las 7 y 14 dentro del fichero del dominio. La inclusión de la acción *Boolean Algebra Simulation-action3* en el plan de María responde a las mismas causas; aunque en su caso debería haberse incluido la acción 2 por que cumple, además de con la condición de rendimiento, con la condición de equipamiento, se ha incluido la acción 3 por que es la última que se define dentro del fichero del dominio.

5. Comparativa

Después de estudiar el proceso de generación de dominios y problemas, además de los diferentes itinerarios formativos adaptados que se han obtenido como resultado de la ejecución de los planificadores SIADEX y LPG-td utilizando sus respectivos archivos generados, es preciso a resumir las diferencias entre diversos procesos y resultados para poder sacar algunas conclusiones importantes de ello.

5.1. Procesamiento Estructural

La naturaleza jerárquica de un curso es una ventaja muy grande para los planificadores HTN, como SIADEX, debido a que su dominio de planificación, codificado en el lenguaje PDDL, representa de una forma muy clara la estructura del curso que podemos observar, por ejemplo, en la figura 3. Los temas, subtemas, etc.. son representados por tareas compuestas y los objetos de aprendizaje, también llamados actividades concretas, son representados por medio de una acción primitiva con duración y ciertas condiciones asociadas.

En el caso de la extracción de un dominio de planificación para un paradigma basado en estados, el proceso es un poco mas complicado y su resultado en el lenguaje PDDL es mucho menos claro, e incluso engorroso. Este tipo de planificadores solo permiten la declaración de acciones primitivas con precondiciones asociadas y, tal vez, un tiempo que, al no poder incluirse en la meta del problema, no tiene ningún sentido práctico mas que el informativo. Además, cada relación estructural debe ser representada en base a complicadas precondiciones que requieren de un costo de preprocesamiento mucho mayor que el requerido

para el HTN, al tener que recorrer en varias ocasiones los objetos del árbol y duplicar la información en varias estructuras de datos complejas para poder ordenar adecuadamente los objetos.

5.2. Representación de las Restricciones Temporales

La capacidad de representación y manipulación temporal de la que cada planificador es capaz, es determinante para decidir que es lo que se puede representar acerca del dominio de un curso que se impartirá on-line.

Por el momento existen dos tipos de limitaciones temporales que, comparado con el planificador SIADEX, restringen la expresividad temporal en el dominio de planificación que puede ser generado para el planificador LPG-td. Estas limitaciones no son generales de todos los planificadores basados en estados, pero se ha utilizado utilizado LPG-td ya que es uno de los planificadores mas sobresalientes con respecto a la resolución de problemas con restricciones temporales.

Estas diferencias entre los planificadores que se deben a las restricciones en la representación de conocimiento temporal se describen en los puntos siguientes:

Plazos. Generalmente, los cursos e-learning tienen una fecha límite de término, es decir, un plazo establecido en el cual será necesario cumplir con todas las metas propuestas por el tutor a través de las actividades disponibles dado cada uno de los objetos de aprendizaje. En los cursos en línea los estudiantes tienen la oportunidad de establecer sus propios plazos previos al plazo general establecido por el tutor. Esta propiedad es representada por medio de la información del campo *Available Time* descrito por cada uno de los estudiantes en su perfil. Este valor se utiliza para definir una meta dentro del archivo del problema de planificación HTN, como se ha visto anteriormente.

El planificador basado en estados LPG-td no soporta este tipo de metas en su definición del problema de una forma natural. El método utilizado por la mayor parte de los planificadores basados en estados para soportar restricciones temporales es una métrica que minimiza o maximiza el tiempo requerido por las acciones durativas para cumplir con la meta propuesta.

Este método no permite lidiar con plazos globales e individuales a la vez, manteniendo siempre, aunque esté declarada dicha meta, una misma duración del plan para cada grupo de alumnos con perfiles iguales excepto por el tiempo disponible. Además, algunos planificadores tampoco permiten definir metas relacionadas con predicados y metas de maximización o minimización de variables dentro del mismo problema de planificación, por lo que, si se quiere incluir una métrica, no siempre es posible definir una meta basada en el cumplimiento de una acción específica.

Duraciones. Al obtener un plan adaptado por medio de un planificador jerárquico el estudiante puede decidir como de largo puede ser su recorrido de aprendizaje y, según ese valor, decide que acciones cuyo metadato *Coverage* tiene valor `Optional` pueden ser incluidas o no dentro del plan.

Un planificador basado en estados siempre intenta minimizar el tiempo necesario para cumplir con la meta, es decir, el objetivo de aprendizaje del curso; por tal motivo, si se incluye una métrica de minimización temporal dentro de la meta del problema basado en estados, las acciones primitivas de tipo opcional no serían incluidas bajo ninguna circunstancia.

5.3. Análisis de Complejidad

Como se puede ver en la figura 17 los pasos para construir el conjunto \mathcal{O} no exceden el tamaño del repositorio \mathcal{LOR} dado que los objetos de aprendizaje son procesados en una sola ocasión. Por lo tanto, se puede decir que este paso del proceso de generación automática se puede ejecutar en un tiempo lineal dado el tamaño del repositorio $\mathcal{LOR} : \mathcal{O}(|\mathcal{LOR}|)$.

Los dos ciclos principales del algoritmo de transformación a dominios de planificación HTN de las figuras 18, 20 y 24 están dado por $|\mathcal{P}_o|$ y $|\mathcal{C}_o|$ respectivamente, teniendo en cuenta que se han añadido a \mathcal{C}_o un pequeño número objetos compuestos auxiliares para cada acción opcional o difícil, número que puede ser despreciado. Por lo tanto, el número de iteraciones para el algoritmo de generación automática de dominios HTN está dado por $|\mathcal{O}| = |\mathcal{P}_o| + |\mathcal{C}_o|$ tal que $\mathcal{O} \subseteq \mathcal{LOR}$, lo cual indica que depende del número de objetos en el repositorio. Así pues, la complejidad del algoritmo de transformación es lineal con respecto al tamaño del repositorio $\mathcal{O}(|\mathcal{LOR}|)$.

El algoritmo para la generación de dominios para planificadores basados en estados es un tanto mas complejo que el anterior. La primera parte, que comprende la generación de las estructuras de objetos de aprendizaje ordenados (fig. 29), se encarga de recorrer todos los $o \in \mathcal{O}$ que sean necesarios

para cumplir con una meta, tantas veces como valores v de una regla de ordenación implícita \mathcal{R} existan; y como \mathcal{O} no exceden el tamaño del repositorio \mathcal{LOR} , entonces $\mathcal{LOR} : \mathcal{O}(|\mathcal{LOR}|)$. Es decir, que la complejidad de este primer paso del algoritmo depende del producto de $|\mathcal{R}|$, es decir de $v \in \mathcal{R}$, por $|\mathcal{LOR}|$ y sería $\mathcal{O}(n \times |\mathcal{LOR}|)$.

La segunda parte del algoritmo (fig. 30), que se ejecuta en paralelo con la tercera (fig. 32), pudiera considerarse de complejidad $\mathcal{O}(|\mathcal{LOR}|)$ e incluso menor ya del conjunto de objetos de la estructura \mathcal{R}_v han sido eliminados los objetos múltiples, sin embargo, el ciclo interno que revisa cada estructura \mathcal{R}_v indica que habrá que revisar $|\mathcal{LOR}| \times |\mathcal{R}_v|$ y, como $\mathcal{R}_v \subseteq \mathcal{LOR}$, el número total de iteraciones equivaldría a \mathcal{LOR}^2 . Por lo tanto, la complejidad de todo el proceso de generación de dominios para planificadores basados en estados quedaría de la siguiente manera $\mathcal{O}(n \times |\mathcal{LOR}| + |\mathcal{LOR}^2|)$.

Con respecto a la generación del fichero del problema tanto para el algoritmo HTN (fig. 28) como para el basado en estados (fig. 34), se puede observar que la generación de las metas es inmediata y la generación del estado inicial es también muy simple dado que solo depende linealmente del número de estudiantes registrados en el curso y el número de características del perfil de cada uno.

Por lo tanto, *para planificadores HTN el proceso completo de transformación sería lineal dado un número limitado de objetos en el o los repositorios, mientras que para planificadores basados en estados sería cuadrático.*

5.4. Resumen Comparativo

En la tabla 9 se muestran seis aspectos básicos que sirven para comparar los dos enfoques de planificación utilizados para generar el dominio de planificación de un curso y que fueron publicados en [Castillo et al., 2010] y [Morales et al., 2008a]. También se ha contrastado con ellos el proceso de generación de un diseño instruccional, definido a través de un dominio real y construido por expertos en educación.

Primero se evalúan las ventajas y desventajas de la generación de cada uno de los dominios de planificación comparados con la elaboración de los diseño de aprendizaje de forma manual. Después se evalúan las características de los recorridos de aprendizaje que se obtienen dado cada uno de los procesos anteriores; en esta parte se puede notar una brecha amplia entre los planes generados por los planificadores y los correspondientes al diseño de aprendizaje dado que la desventaja derivada de un exhaustivo proceso de generación y seguimiento manual se ve retribuida con muchas mas capacidades de adaptación que terminarán siendo, en las implicaciones teóricas, las pautas a seguir para futuras investigaciones.

Aunque los dos planificadores son ampliamente utilizados en dominios reales con restricciones temporales, es posible observar que el dominio para el planificador SIADEX es menos complejo y rápido de generar automáticamente, además de más expresivo dadas las limitaciones de representación de jerarquías que presenta LPG-td, lo que implica un costo bastante grande al momento de construir las precondiciones que las representen. Esto se puede notar tanto en los costos de generación de dominios y problemas como en el de generación del plan⁹.

Para el planificador LPG los tiempos de generación del plan se reducen considerablemente entre más estudiantes haya, pero esto solo es posible si se utiliza una opción que no está documentada, ni viene por defecto, llamada `-incremental_goal_resolution`. Usando las opciones por defecto, resulta imposible crear un plan para más de 12 estudiantes, y para 12 estudiantes toma en promedio 92.24 segundos, es decir, casi el triple de lo que toma para 36 con la opción anteriormente mencionada.

Por otro lado, los resultados del análisis de complejidad de los dos algoritmos, sitúan al proceso completo de generación automática de ficheros y dominios HTN como los menos complejos a comparación de los diseñados para el planificador basado en estados.

Además, el soporte de plazos y opcionalidad incluido en el planificador HTN le da un punto extra frente al basado en estados, añadiendo en el proceso de adaptación del recorrido de aprendizaje dos aspectos que suelen ser de un gran provecho para alcanzar las metas deseadas por el tutor y dar libertades a los estudiantes en cuestión de tiempo dedicado al curso. Esta última cualidad suele ser muy apreciada, sobre todo, por alumnos que reciben formación continuada por internet y que no disponen de mucho tiempo para el estudio debido a compromisos familiares o de trabajo.

El soporte de multiplicidad en los nombres de acciones, lo cual implica que la misma acción puede ser desplegada al estudiante de formas distintas, es algo muy valorado también y que a la mayor parte de los planificadores basados en estados les cuesta manejar. Sin dejar de lado la condicionalidad en la inclusión de objetos de aprendizaje complejos que en esta memoria están representados como objetos requeridos por otros, que tampoco ha sido posible representar en planificadores basados en estados por medio de dominios generados automáticamente.

⁹Los tiempos mostrados en la tabla son el promedio de 50 ejecuciones de cada algoritmo o del planificador, respectivamente.

	HTN (SIADEX)	No HTN (LPG-td)	Real (e-learning)
Aplicación Práctica	<p>→ El conocimiento necesario para la generación del dominio se extrae directamente del etiquetado de metadatos.</p> <p>→ La expresividad del planificador, debido a la similitud de su representación a la de un dominio real y su “facilidad” de generación.</p>	<p>→ El conocimiento necesario para la generación del dominio se extrae completamente del etiquetado de metadatos.</p> <p>→ El algoritmo de generación del dominio tiene una complejidad bastante grande en comparación con el HTN.</p> <p>→ Demasiadas condiciones para las acciones primitivas.</p>	<p>→ Es demasiado costoso generarlo ya que implica un tiempo enorme, un análisis exhaustivo de las características de cada estudiante, conocimientos pedagógicos y de estándares de e-learning bastante amplios, además de un proceso de seguimiento del curso imposible en ocasiones.</p>
Calidad	<p>Soporta interacción adicional y adaptación:</p> <p>→ Acciones adaptadas.</p> <p>→ Plazos temporales dependiendo de la disponibilidad del estudiante y los requisitos que el tutor ha impuesto al curso.</p> <p>→ Actividades opcionales dependiendo de la capacidad del estudiante y de su tiempo.</p>	<p>Soporta las típicas condiciones de requerimientos</p> <p>→ Acciones adaptadas.</p> <p>→ No hay adaptación respecto al tiempo, simplemente toma la primera actividad que encuentra.</p> <p>→ Si te puedes permitir cursar material extra u opcional, lo da todo de un tirón sin importar el tiempo requerido para ello.</p>	<p>La calidad de los recorridos de aprendizaje se basa en la experiencia del experto y en las características principales de los estudiantes:</p> <p>→ Objetos de aprendizaje adaptados a su situación.</p> <p>→ Soporta restricciones temporales.</p> <p>→ Soporta interacción entre diferentes roles y grupos de estudiantes.</p> <p>→ Soporta material extra para reforzar el conocimiento adquirido.</p>
Generación de Dominio y Problema	→ 8.01 segundos.	→ 8.16 segundos.	→ Tiempo requerido para el etiquetado de objetos de aprendizaje: 5 horas aproximadamente.
Generación de Plan	<p>→ Plan para 1 estudiante: 0.11 segundos ya convertido al formato de Moodle.</p> <p>→ Plan para 36 estudiantes: 37.81 segundos ya convertido el plan al formato de Moodle.</p>	<p>→ Plan para 1 estudiante: 0.41 segundos + 0.002 segundos para convertirlo al formato de Moodle.</p> <p>→ Plan para 36 estudiantes: 33.14 segundos + 0.55 segundos para convertirlo al formato de Moodle.</p>	<p>→ Tiempo requerido para analizar su perfil y asignar un plan a un estudiante: 30 minutos.</p> <p>→ Tiempo requerido para analizar su perfil y asignar un plan a 36 estudiantes: aproximadamente 18 horas.</p>
Complejidad en la Transformación	$\mathcal{O}(LOR)$	$\mathcal{O}(LOR ^2)$	No responde a complejidad algorítmica si no a la necesidad de expertos en planificación inteligente en todo momento durante el proceso de transformación.
Implicaciones Teóricas	El planificador soporta la mayor parte de las capacidades de adaptación requeridas en esta fase de la solución.	Su dominio no es intuitivo, ni permite elegir entre las distintas acciones “múltiples” en base a criterios temporales.	Implica mas tiempo y esfuerzo, pero marca la pauta para la mejora de los actuales procesos de planificación.

Tabla 9. Tabla comparativa entre distintas características y divergencias entre los planificadores utilizados y el dominio real de un curso.

CAPÍTULO 5

Planificación de Diseños Instruccionales en Entornos Dinámicos

Cuando se habla de entornos dinámicos en el marco de un curso a distancia se puede relacionar con diversas características del curso o los estudiantes que pueden variar a lo largo de la realización del mismo. Normalmente dichas variaciones pueden provocar cambios inmediatos o progresivos en la organización del diseño instruccional del curso para cada estudiante y, por lo tanto, resultan sumamente interesantes para los expertos en la aplicación de técnicas de planificación para entornos dinámicos.

En las siguientes secciones se detallará el problema de la planificación sobre entornos educativos dinámicos. Posteriormente se mencionarán las distintas propuestas que han implementado hasta ahora otros autores para resolver este problema utilizando o no técnicas de planificación y scheduling inteligentes. Y finalmente se describirán las dos propuestas que, en base a las herramientas descritas en los preliminares de esta memoria, se pueden aplicar para dar una solución parcial a este problema.

1. El Problema de los Entornos Educativos Dinámicos

Los entornos educativos requieren de la intervención de agentes externos que creen situaciones de cambio y/o incertidumbre durante el desarrollo del curso a distancia. Estos agentes pueden ser humanos o artificiales, pero generalmente los agentes artificiales solo actúan como apoyo a la toma de decisiones o emulando a los agentes humanos.

Por lo tanto, para propósitos de esta memoria se considerarán solo los agentes humanos, ya sea profesores y/o estudiantes, que debido a sus características provocan situaciones de cambio o incertidumbre en las propiedades del curso que son condiciones para generar un diseño instruccional adaptado del mismo.

Algunas de las características de los *estudiantes* que pueden variar a lo largo de la realización del curso y que, por lo tanto, requieren del soporte de dinamismo para readaptar el itinerario formativo del curso a uno o varios estudiantes cuyo perfil haya cambiado, pueden ser las siguientes:

- Su nivel de idiomas.
- Sus resultados en las distintas evaluaciones intermedias programadas previa ejecución del mismo.
- Su disponibilidad temporal.
- Su rendimiento en las actividades realizadas hasta cierto punto temporal del curso o en la actividad previa.

Existen muchas otras características de un estudiante que, al ser modificadas dentro de su perfil o analizadas a través de un seguimiento exhaustivo de su interacción con la plataforma del curso, pueden provocar la necesidad de utilizar técnicas de planificación inteligente para entornos dinámicos.

Por otro lado, algunas de las características del curso que el *profesor* puede modificar a lo largo de la realización del mismo, y que pueden disparar la necesidad de procesos dinámicos de readaptación del curso a los estudiantes, podrían ser:

- Los objetos de aprendizaje que lo componen, ya sea agregando o eliminando manualmente los objetos seleccionados inicialmente, modificando las relaciones de orden y/o jerarquía entre ellos o algún otro de sus metadatos¹.
- La asignación y/o modificación del puntaje a las evaluaciones intermedias realizadas por los estudiantes.
- La regla implícita de ordenación seleccionada para determinar el orden no explícito de las actividades del curso.

Estos son solo algunos ejemplos de los elementos del curso que, al ser modificador por el profesor durante la ejecución del curso, pueden requerir la aplicación inmediata de técnicas de planificación para entornos dinámicos.

En la actualidad, los cursos en línea suponen la generación de una gran cantidad de objetos de aprendizaje y su inclusión en la plataforma de e-learning antes o durante el desarrollo del curso, sin ningún tipo de adaptación a las nuevas necesidades de cada estudiante o grupo de estudiantes. Lo más normal es que, si el profesor detecta alguna necesidad en el grupo durante la impartición del curso, modifique el contenido de algunos objetos para después volverlos a integrar en la plataforma de e-learning.

Por otro lado, los Sistemas Tutoriales Inteligentes si que se han esforzado por integrar técnicas de adaptación sobre todo al dinamismo del estudiante. Así se tiene por ejemplo el trabajo de [Vassileva, 1997] quien introdujo por primera vez los conceptos de planificación, reparación de planes y re-planificación para soportar el dinamismo de un curso a través de planificación. Su trabajo fue pionero en aplicar estos conceptos en un tutor inteligente, pero no consideraba ni los estándares de representación SCORM o IMS-LD que estaban surgiendo en esos momentos, ni la técnica de planificación continua que se adapta más a la forma de impartir tradicionalmente un curso a distancia dentro de un LMS. También se puede observar el trabajo de [Weber and Brusilovsky, 2001] cuyo manejo de la dinamicidad consiste en integrar nuevos hipervínculos con ejemplos del tema que se esté tratando en ese

¹La modificación del contenido de un objeto de aprendizaje, siempre y cuando no se cambien sus características tan drásticamente que esto implique la modificación de alguno de sus metadatos, no implica cambio alguno en el diseño instruccional original de los estudiantes

momento con el fin de reforzar un conocimiento que podría resultar escaso.

Sin embargo estos sistemas, como se ha comentado en la introducción y preliminares de esta memoria, tienen un número restringido de usuarios, una estructura particular para los cursos que pueden soportar y un software sobre el que es difícil hacer modificaciones constantes ya que se encuentra hecho a la medida. Además la mayoría de estos ITS's no soportan los principales estándares de e-learning, tampoco están integrados en plataformas de e-learning con un espectro mas amplio de usuarios, ni utilizan técnicas de planificación para abordar la dinamismo del entorno, pilares fundamentales de esta investigación.

El único intento serio por manejar la dinamismo del entorno en ITS's ha sido el de [Ullrich and Melis, 2009], en ese trabajo se propone utilizar la capacidad del planificador SHOP2 de pausar el proceso de planificación, esperar información que pudiera ser incierta en algún punto del proceso de planificación mientras se despliega al estudiante la porción del diseño instruccional planificada hasta ese momento y, cuando la información incierta sea proporcionada, reactivar el proceso de planificación de SHOP2 hasta el siguiente punto de espera. Este proceso se lleva a cabo una y otra vez hasta que se alcanza la meta definida al principio de la ejecución del plan.

Sin embargo, [Ullrich and Melis, 2009] no utiliza ningún tipo de estándar para recabar la información incierta y enviarla al planificador y, por otro lado, se ve limitado a la definición previa a la ejecución del plan de condiciones inciertas no genéricas.

Por otro lado, en [Santos et al., 2007] se utilizan agentes inteligentes para hacer una monitorización selectiva de la interacción del estudiante con un LMS específico e ir cambiando dinámicamente el estado de su perfil y por lo tanto propiedades de un IMS-LD generado previamente por un planificador inteligente. Este proyecto resulta sumamente interesante pues ha permitido estudiar la interacción del estudiante y ofrecer recomendaciones en tiempo real; sin embargo no queda claro como, existiendo secuencias estáticas de objetos de aprendizaje para cada perfil de estudiante, es posible que los agentes recomienden en tiempo real el uso de otros objetos que no se encuentran dentro de dicha secuencia y que, por lo tanto no son accesibles a dicho estudiante.

Por lo tanto, la arquitectura presentada en esta memoria y la sugerida en [Santos et al., 2007] podrían ser complementarias si, utilizando las técnicas de planificación en entornos dinámicos, se integraran a la llamada unidad de aprendizaje o **activity structure** los objetos recomendados por los agentes y se generara nuevamente el IMS-LD de cada estudiante. Pero como se ha mencionado anteriormente, solo el LMS dotLRN soporta el uso del estándar IMS-LD y su costo de integración en otros LMS's sería a mas largo plazo que el tipo de integración que se propone a continuación.

En las siguientes secciones se analizarán dos propuestas para utilizar las técnicas de planificación en entornos dinámicos en conjunto con las mas innovadoras especificaciones de e-learning y los LMS's mas utilizados en la actualidad.

La primera propuesta se basa en representar automáticamente un diseño instruccional dinámico y genérico dentro de la especificación IMS-LD en su nivel B. Dada la poca madurez en la implementación (mas no en la especificación teórica) de arquitecturas de comunicación entre este y otros estándares, solo se representarán dentro del mismo hilos condicionales dinámicos que hagan referencia a las evaluaciones intermedias de un curso y su posible calificación.

Para lidiar con otras características de incertidumbre y dinamismo de un curso a distancia, además de la referente a las evaluaciones intermedias, se ha diseñado e implementado la propuesta de la sección 3. Las características dinámicas que no se tomarán en cuenta dentro de este trabajo ya que requieren la intervención de otras técnicas de ingeniería de software e inteligencia artificial diferentes del AIP&S, son la evaluación del rendimiento actividad por actividad y la modificación de las reglas de ordenación implícita.

2. Pre-Planificación Condicional para la Generación de un IMS-LD nivel B

En el nivel B de la especificación IMS-LD, denominada comúnmente diseño de aprendizaje, es posible representar valores condicionales que permitan decidir entre las distintas opciones de continuidad del diseño instruccional para un estudiante. Por ejemplo, un estudiante con determinado perfil puede iniciar un curso a distancia siguiendo el conjunto

de actividades previamente adaptado a su perfil pero, en cierto punto predefinido de la secuencia de actividades de aprendizaje del mismo, se le proporciona como actividad una evaluación en la cual puede aprobar o reprobar. Obviamente su secuencia a partir de ese punto de inflexión será distinta si aprueba a que si reprueba y, por ende, habrán de haberse planificado previamente las dos posibles secuencias de actividades que, junto con la información de su perfil, reestablezcan la coherencia y den continuidad a su diseño instruccional original.

Tomando en cuenta esta situación de ejemplo y codificando manualmente una situación similar usando dicha especificación, es posible observar que en cada condición del IMS-LD se pueden representar puntos de inflexión del plan que deberán estar previamente establecidos. Algo muy similar a la propuesta de [Ullrich and Melis, 2009] pero que, en vez de esperar a continuar con el proceso de planificación en tiempo real, necesita de una planificación a priori de las posibles secuencias de objetos de aprendizaje para cada valor o rango de valores que pueda tomar el punto de inflexión, como se describe en [Morales et al., 2009] .

Para efectos prácticos, y dado que las evaluaciones intermedias de un curso son los puntos de inflexión mas comunes de de un curso, se considerará solo la incertidumbre sobre los resultados de los estudiantes en las evaluaciones intermedias del curso que hayan sido definidas previamente por el profesor. Esta decisión se ha tomado no solo para poder ejemplificar mejor y de una forma mas sencilla el proceso de generación automática de generación de un IMS-LD nivel B, si no por que el resto de las variables que pueden ser modificadas por el profesor o el estudiante durante la ejecución del curso requieren de la interacción entre IMS-LD y otras especificaciones como IMS-LIP e IEEE-LOM; interacción que, aunque esté teóricamente definida, aún no está implementada dentro de la versión estable de ningún LMS.

2.1. El manifiesto IMS-LD

Hablando en términos de la especificación IMS-LD descrita en el capítulo 3 sección 4 y cuyos elementos y estructura relacional se muestran también en la figura 35, para desplegar el diseño instruccional de cada estudiante o *Role-Part* existe un *Play* compuesto por un

conjunto de *Acts* que corresponden a cada uno de los subobjetivos del curso que habrán de cumplirse para poder obtener el valor de las propiedades o *Properties* que se analizarán en las condiciones o *Conditions*. Cada Role-Part está asociada a un *Role* o perfil de estudiante específico y cada rol tiene asignada una *activity-structure* o estructura de actividades general para completar un Act, dicha estructura de actividades general se dividirá en un conjunto de *activity structures*.

Cada *activity-structure* del conjunto principal podrá ser ejecutada, sí y solo si, el estudiante ha adquirido el valor específico de la propiedad requerida por la condición que permite visualizarla. Por ejemplo, en la figura 13 las condiciones indican que solo será posible visualizar las estructuras de aprendizaje *ACTIVITY-STRUCTURE-QTI-Test1-3.0-90.0* para el estudiante cuyo role es 3 y *ACTIVITY-STRUCTURE-QTI-Test1-4.0-90.0* para el estudiante cuyo role es 4, si el valor de la propiedad *QTI-s2ctest2* es mayor o igual a 8.

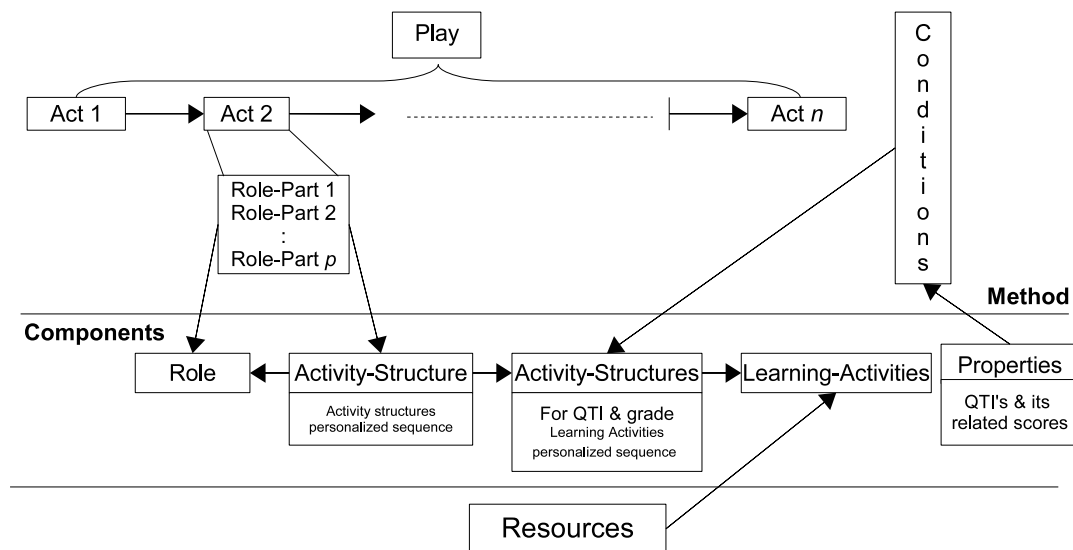


Figura 35. Estructura del IMS-LD nivel B en base a la propuesta de esta sección.

Así, deberá existir al menos una condición para ejecutar cada Act a partir del Act2, y el valor de verdad de dicha condición deberá estar asociado al valor que el estudiante, cuyo perfil cumpla las características de ese Role, haya generado durante el desarrollo de los Act previos para alguna de las propiedades predefinidas.

Cabe mencionar que cada activity-structure, excepto la general, puede estar formada por otras activity-structures o por *learning-activities* o actividades de aprendizaje que están asociadas a uno o mas recursos o *Resources*.

2.2. Modificación a los Algoritmos de Generación Automática de Dominios y Problemas de Planificación

Para utilizar los algoritmos de generación de dominios y problemas de planificación descritos en el capítulo anterior aplicando este nuevo enfoque, será necesarias cuatro modificaciones:

1. Que a cada predicado, acción y tarea se agregue el parámetro `?score` y que en la metainformación de las acciones se impriman los contenidos de una *learning-activity*.
2. Que se generen reglas de asociación entre la calificación o score obtenida por un estudiante y la modificación de las características de su perfil en base a un valor o rango de valores de la misma.
3. Que para cada estudiante del curso se generen tantas descripciones de su estado inicial como reglas de asociación se definan en el punto anterior.
4. Que para cada estudiante del curso se generen metas como reglas de asociación se definan en el punto 2.

Las reglas de asociación que se han definido para propósitos de esta memoria son las siguientes:

- Si `?score <= 5`, entonces el estudiante se considera reprobado y, por lo tanto, para dicha instancia de su perfil

<i>Meta</i>	(DiscreteMaths maria 3)	(DiscreteMaths maria 7)	(DiscreteMaths maria 9)
<i>Estado Inicial</i>	(= (performance maria 3) 40) (learning-style maria 3 theoretical) (= (language-level en maria 3) 90) (= (score maria LogicAndSets 3) 40) (available-time maria 3 much) (equipment maria 3 multimedia)	(= (performance maria 7) 40) (learning-style maria 7 theoretical) (= (language-level en maria 7) 90) (= (score maria LogicAndSets 7) 40) (available-time maria 7 little) (equipment maria 7 multimedia)	(= (performance maria 9) 90) (learning-style maria 9 theoretical) (= (language-level en maria 9) 90) (= (score maria LogicAndSets 9) 40) (available-time maria 9 little) (equipment maria 9 multimedia)

Tabla 10. Múltiples perfiles y metas en el fichero de planificación del problema para la estudiante María; todo ello con el fin de obtener distintas secuencias que posteriormente se puedan asociar a condiciones en el IMS-LD.

- La disponibilidad habrá de ser alta, ya que el estudiante tendrá que dedicarle mas horas al estudio de los objetivos del curso, y
- El rendimiento del estudiante será bajo, aunque inicialmente haya sido alto.
- El resto de sus características permanecerán iguales a las del perfil original.
- Si $?score >5 \text{ and } \leq 8$, entonces el estudiante ha aprobado pero con un rendimiento pobre y, por lo tanto, para dicha instancia de su perfil
 - Al rendimiento del estudiante se le otorgará el valor de *Medium* , y
 - El resto de sus características permanecerán iguales a las del perfil original.
- Si $?score >8 \text{ and } \leq 10$, entonces el estudiante ha aprobado con buena nota y. por lo tanto, para dicha instancia de su perfil
 - Al rendimiento del estudiante se le otorgará el valor de *High* , y
 - El resto de sus características permanecerán iguales a las del perfil original.

Estas reglas habrían de ser definidas o elegidas por el profesor y, por lo tanto, como trabajo futuro se espera permitir al profesor modificar a su antojo las reglas de asociación a través de una interfaz de what-ifing diseñada especialmente para este entorno.

Entonces, el estado inicial y las metas para cada estudiante quedarían según lo ejemplificado para la estudiante María en la tabla 10; en donde para cada regla se define una calificación dentro del rango de la condición de la misma [3,7,9] y esta es la que se incluye en los predicados de su meta y perfil. Los predicados de disponibilidad y rendimiento han

tenido que ser modificados en algunos casos para que coincidan con la calificación asociada a esa representación del perfil de María.

2.3. El Proceso de Pre-Planificación Condicional

A diferencia de la arquitectura propuesta en el primer acercamiento no dinámico a la planificación automática de diseños instruccionales (fig. 4), podemos ver como en este caso (fig. 36) el proceso consta de un ciclo que genera, por cada meta intermedia del curso, todas las posibles secuencias de actividades adaptadas a todos los perfiles de estudiantes existentes.

En cada ciclo (excepto en el primero) se van agregando al manifiesto IMS-LD las m secuencias de actividades **activity-structures** adaptadas para cada perfil, las actividades que las componen y las condiciones para poder desplegar alguna de ellas al estudiante que cumpla con uno u otro perfil y calificación asociada.

La primera secuencia es diferente ya que, al no tener una evaluación intermedia que la preceda, solo se genera una estructura ordenada de actividades o plan para cada rol o perfil de estudiante.

Una vez generadas todas las secuencias que cumplan con la meta principal del curso g_n y agregadas en el manifiesto, este será enviado a un visualizador o Player de IMS-LD's (en este caso RELOAD) para que pueda ser utilizado por los estudiantes.

Entonces, para cada rol $r \in \mathcal{R}$ que pertenece al conjunto de roles posibles dependiendo del p número de perfiles de estudiantes distintos en un curso, existirán n subconjuntos de objetos de aprendizaje \mathcal{S} cuya secuenciación y posterior unión generará el diseño instruccional completo de un curso.

También habrá $n - 1$ puntos de inflexión (en este caso evaluaciones) o metas intermedias g por medio de los cuales se podrá obtener la información necesaria para generar los distintos perfiles del problema de planificación para cada regla de asociación.

Con lo anterior se podrán generar las m secuencias adaptadas s_i de objetos de aprendizaje que usarán las m reglas de asociación existentes y su asociación con el valor del punto de inflexión g_{i-1} para definir la submeta que habrá de alcanzar el planificador para elaborar

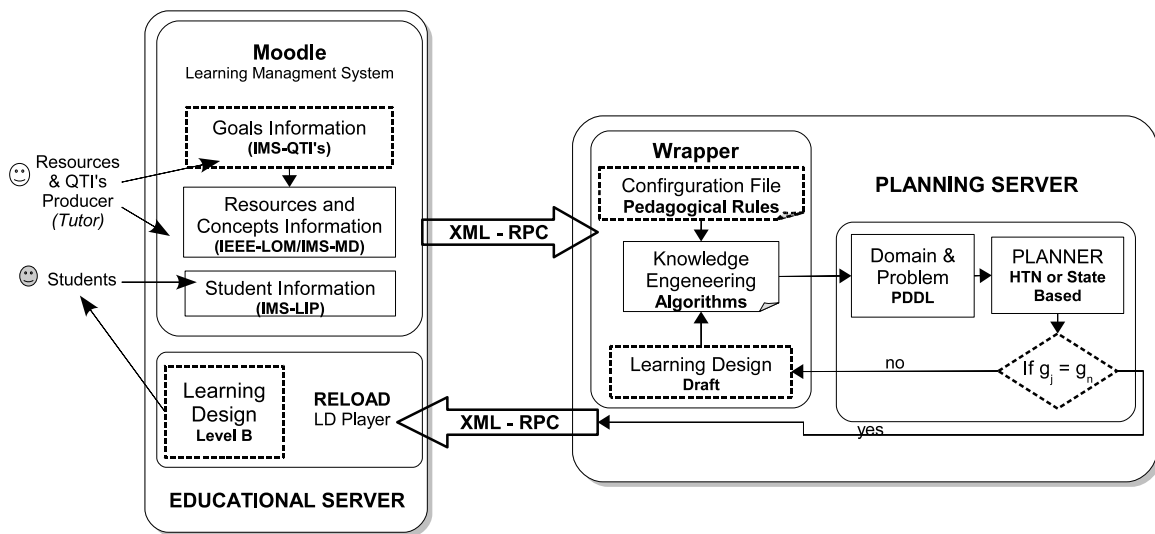


Figura 36. Generación automática y despliegue del IMS-LD nivel B en base a la propuesta de Pre-Planificación Condicional, los rectángulos con líneas punteadas son los elementos que se agregan a la propuesta original de la figura 4.

cada secuencia. Cada secuencia de objetos de aprendizaje será una **activity-structure** $s_{i,j}$ que el estudiante con rol r podrá seguir en caso de que obtenga la calificación j asociada a la misma en su evaluación intermedia g_{i-1} .

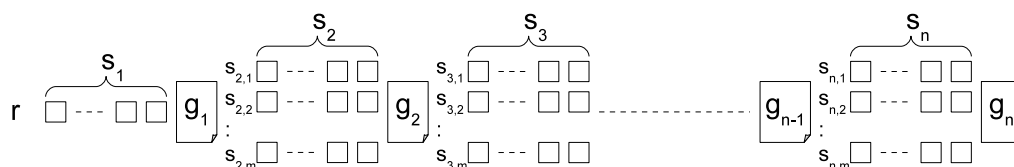


Figura 37. Visualización formal de la generación de secuencias condicionales.

3. Planificación Continua y Re-planificación para la Interacción con un LMS

Aunque muchos de los sistemas gestores de aprendizaje que se han estudiado en los preliminares soportan estándares, se ha encontrado que la especificación IMS-LD de diseños de aprendizaje, que es el estándar que tiene mas capacidades para representar las características de un diseño instruccional original, no cuenta con especificaciones lo suficientemente maduras como para interactuar con la información y funciones de los LMS's si no solamente

mostrar los diseños instruccionales previamente generados previamente, como se ha podido comprobar en la sección anterior.

Es por esta razón que en las siguientes secciones se describen dos procesos por medio de los cuales se puede intensificar el soporte de dinamismo en entornos de e-learning a través de otras dos técnicas de planificación, pero utilizando como salida para desplegar al estudiante su diseño instruccional adaptado la especificación SCORM en lugar del estándar IMS-LD.

3.1. Planificación Continua en Entornos e-Learning Dinámicos

La propuesta de *planificación continua*, como se puede ver también en [Morales and García, 2010], consiste a grandes rasgos en dividir el curso a distancia en un conjunto de paquetes SCORM. Cada paquete representa un sub-objetivo o meta intermedia y el profesor será el encargado de solicitar la planificación de cada paquete de manera que el SCORM que se muestra a cada estudiante dentro del LMS pueda ser modificado en base a su perfil.

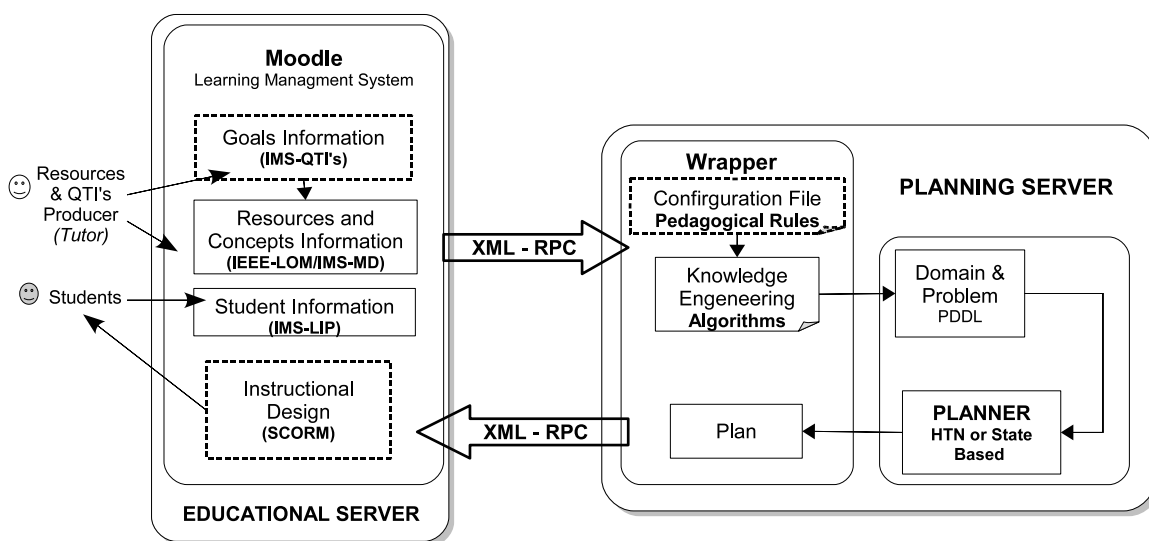


Figura 38. Arquitectura cíclica de planificación para entornos dinámicos.

Como el primer paquete no tiene normalmente ninguna condición previa asociada a evaluaciones intermedias, el profesor puede solicitar la generación de su SCORM adaptado desde el principio del curso.

El profesor podrá solicitar la adaptación del segundo SCORM, si y solo si, ya se ha aplicado y calificado la evaluación intermedia requerida para llevarlo a cabo.

Por lo tanto, a partir del segundo SCORM hasta el último, antes de generar los ficheros de dominio y problema de planificación utilizando los algoritmos del capítulo 4, se modificará el perfil de cada uno de los estudiantes del curso en base a una serie de reglas pedagógicas de asociación que toman en cuenta la calificación obtenida en la evaluación intermedia previa y necesaria para acceder a dicho sub-objetivo.

Estas reglas serán las mismas que se han descrito en la sección 2.2 solo que, en este caso, la generación del problema solo generará un solo estado inicial para cada estudiante utilizando el perfil modificado previamente en base a las reglas de asociación.

3.2. Re-planificación en Entornos e-Learning Dinámicos

Por su parte, la propuesta de *re-planificación* consiste en planificar nuevamente el SCORM con el cual esté interactuando el estudiante cada vez que ocurran los siguientes eventos:

- Que alguna de las características del perfil del estudiante cambie.
- Que se modifique la calificación de la evaluación previa al SCORM que se esté ejecutando.
- Que se añada o se modifiquen los metadatos de al menos un objeto de aprendizaje del SCORM actual.

La evaluación del rendimiento actividad por actividad y la modificación de las reglas de ordenación implícita son temas de dinamismo sobre cursos a distancia que se han omitido de esta propuesta, una por que de implementarse requeriría el uso de otras técnicas inteligentes como por ejemplo el uso de agentes inteligentes para monitorizar constantemente la interacción del estudiante con la plataforma de e-learning y/o la definición de una cantidad aún mas exhaustiva de reglas para evaluar el rendimiento del estudiante dentro de cada objeto de aprendizaje, así como para cambiar el enfoque pedagógico que se utilizará para decidir el orden implícito de los objetos.

Hasta el momento los eventos anteriormente mencionados son “detectados” por el estudiante o el profesor y es el profesor el encargado de solicitar que se planifique nuevamente el SCORM que se esté ejecutando en ese momento.

Por lo tanto, si el estudiante modifica su perfil tiene la obligación de pedir al profesor que vuelva a adaptar su diseño instruccional y si el profesor hace alguna modificación a las calificaciones o SCORM's, deberá solicitar el mismo la re-planificación del SCORM correspondiente.

Lo mas interesante e innovador de este proceso de re-planificación es que se elimina el tiempo de generación de, al menos uno, de los ficheros de planificación debido al proceso de razonamiento sobre la información de los ficheros almacenados en el historial de recepción del servidor de planificación que:

- Si detecta que ha cambiado el fichero de perfiles de estudiantes, ya sea por que ha cambiado el perfil de alguno de ellos o una calificación previa, solo genera el fichero del problema.
- En cambio, si lo que detecta es que ha cambiado el manifiesto previamente enviado de ese SCORM, lo que se infiere como la modificación en alguno de los objetos o sus metadatos, entonces genera solo el fichero de dominio de planificación.

Si bien el tiempo ahorrado por medio de este proceso pudiera resultar mínimo, traducido a horas hombre resulta significativo.

Por otro lado, puede resultar bastante engorroso el hecho de que el profesor tenga que detectar los cambios y mandar re-planificar un SCORM cada vez que estos ocurren. Es por eso que como trabajo futuro de esta memoria se propone la generación de interfaces de automatización para estos procesos. Dichas interfaces con comportamiento de agentes inteligentes no se han desarrollado dentro de la presente memoria debido a que involucran altos conocimientos de agentes inteligentes y una modificación profunda de algunos procesos del LMS específico que quitarían generalización a la propuesta actual.

CAPÍTULO 6

Integración con el LMS

Para que la información proporcionada por los metadatos del curso y la referente al perfil del estudiante pueda ser utilizada por los algoritmos de generación de los dominios y problemas de planificación propuestos en esta memoria, y posteriormente por los planificadores para que se generen los planes adaptados, o diseños instruccionales, es preciso definir un protocolo de comunicación entre las diferentes instancias estudiadas en el capítulo anterior.

Además, el uso de todas las técnicas de planificación mencionadas en capítulos anteriores y las grandes ventajas que estas suponen no tendría sentido si, una vez generados los diseños instruccionales, éstos no pudieran ser mostrados al alumno dentro de la misma plataforma de donde se obtuvo la información.

Es conocido que existen visores de la especificación IMS-LD [Nederland, 2009, Liber, 2007] que una vez instalados en el ordenador del estudiante pueden desplegar el diseño de aprendizaje específico proporcionado previamente por el profesor. Sin embargo, estas herramientas no resultan de interés para la aplicación real de la propuesta si no pueden mostrar el LD de forma transparente dentro de una plataforma que integre, sin necesidad

de llamar aplicaciones externas o preinstaladas. Y, como se ha dejado claro en capítulos anteriores, las plataformas mas utilizadas para la impartición de cursos en línea y que, además, integran todas las características de un curso, profesores y estudiantes, son los Sistemas Gestores de Aprendizaje.

El único LMS que soporta actualmente el despliegue del estándar IMS-LD y tiene integrado el IMS-QTI es dotLRN[dotLRN Consortium, 2009], como se describe en [Morales et al., 2008b], pero hasta ahora no se ha encontrado documentación que soporte la integración del estándar IMS-LIP y tal parece que el despliegue de el learning design solo soporta un manifiesto por estudiante, cosa que restringe las capacidades del diseño de aprendizaje en su representación de nivel B.

Por otro lado, es preciso insistir en que es muy importante el uso generalizado del LMS a utilizar (el mas utilizado es Moodle) y en que, como se ha mencionado en el capítulo anterior, la especificación IMS-LD (a menos que esté altamente integrada con el LMS y el resto de los estándares IMS) no permite tanta flexibilidad para el manejo de entornos dinámicos como la propuesta en la sección 3 del capítulo 5.

Dadas las consideraciones anteriores, es de esperarse que habrían de hacerse algunas modificaciones mínimas al LMS que se ha elegido de manera que soportara, tanto la transmisión de datos hacia la herramienta de planificación automática que se ha propuesto, como la recepción y presentación de los planes generados de manera personalizada a cada estudiante.

Con el propósito de que sea posible soportar las propuestas y necesidades mencionadas en los párrafos anteriores, es que en la siguiente sección se establecen las características detalladas que debiera tener cualquier LMS que pretendiera utilizar la tecnología generada en esta memoria. Posteriormente se describirán de forma detallada la arquitectura propuesta para integrar el LMS y los procesos de planificación inteligente. Y se cierra el capítulo con un ejemplo de prueba de esta arquitectura dentro del LMS Moodle utilizando el curso de prueba de Matemáticas Discretas.

1. Características Deseables en los LMS's

Para que un sistema gestor de aprendizaje sea compatible con la arquitectura descrita por la figura 38 debe de cumplir con ciertas características básicas.

1. *Soporte de estándares sobre objetos de aprendizaje.*

Existen diversos grupos de sistemas gestores de aprendizaje dependiendo del nivel de uso que le dan a los objetos de aprendizaje y los estándares relacionados con ellos. A continuación se enlista una descripción de cada uno de estos grupos.

- a) Aquellos que solo manejan sus actividades internas para soportar el proceso de aprendizaje de un curso, es decir, no soportan objetos de aprendizaje.
- b) Algunos otros que posibilitan el uso de objetos de aprendizaje externos, mas no el despliegue de los mismos basándose en los estándares mas básicos, por ejemplo SCORM o IMS-CP.
- c) Los sistemas gestores de aprendizaje que se precian de seguir al pie de la letra la mayoría de los estándares educativos definidos hasta el momento.
Estos permiten el despliegue de los objetos de aprendizaje basándose en los estándares IMS-CP o SCORM, e incluso que se inserten objetos con etiquetado; aunque el proceso de etiquetado requiera de aplicaciones externas para llevarse a cabo y los metadatos no puedan ser usados para ninguna causa específica.
- d) Solo algunos de ellos permiten el despliegue de dichos objetos de forma adaptativa basándose en el estándar IMS-LD, pero no como parte del mismo sistema, si no como una aplicación externa que se manda llamar cuando es necesario.
Esto último quiere decir que el alumno en realidad accede a las actividades del curso a través de esta aplicación externa y no dentro del mismo sistema gestor de aprendizaje.

Lo que la arquitectura propuesta requiere por el momento son LMS's que cumplan los requisitos de c).

2. Soporte del estándar IMS-LIP para los perfiles de usuarios ó capacidad de incluir,

dentro de su sección de administración, los metadatos o campos con la información indispensable para iniciar el proceso de adaptación de cursos.

3. Soporte del estándar IMS-QTI o similares que puedan comunicarse con el IMS-LIP a fin de proporcionarle información sobre calificaciones de los exámenes de cada estudiante.
4. Posibilidad de comunicación con servidores externos a través del protocolo XML-RPC, SOAP o similares.
5. Flexibilidad para la integración de nuevos módulos dentro del código base del LMS.

Como se puede observar en los puntos anteriores, a pesar de que la propuesta de esta investigación está basada en estándares, se da cierto grado de flexibilidad a los sistemas gestores de aprendizaje (sobre todo respecto al soporte de los estándares IMS-LIP e IMS-QTI) de manera que puedan utilizar las oportunidades de adaptación de diseños instruccionales en su beneficio sin tener que hacer modificaciones excesivas en su núcleo funcional.

2. Arquitectura del Sistema

Como se ha mencionado en ocasiones anteriores, cuando surgieron las primeras propuestas de soporte a la adaptación en e-learning por medio del AIP&S y hasta ahora, el papel del experto en temas de análisis de dominios reales y planificación inteligente era indispensable a la vez que poco viable si la intención era generar cursos adaptados a gran escala. Solo era posible hacer pruebas de laboratorio con grupos pequeños de estudiantes, codificando en PDDL los dominios “reales” de cursos de prueba implementados sobre un sistema tutorial inteligente.

Aunque también existieron propuestas innovadoras (en su momento) en las que se llegó a implementar una semi-automatización del proceso o se llegaron a proponer plantillas generales muy poco prácticas dada la gran cantidad de datos especializados que se pueden considerar cuando se está modelando un dominio e-learning.

La arquitectura original de integración del AIP&S se ha representado en la figura 14 en la cual se ve claramente el papel crucial que juega el experto dentro del proceso de generación

de dominios y problemas, sobre todo si se trata de integrarlos con sistemas gestores de aprendizaje.

Sin embargo, en la propuesta de esta memoria se ha ideado una arquitectura donde la participación del experto se vuelve prácticamente nula. Y con el valor añadido de que su implementación es posible dentro de cualquier sistema gestor de aprendizaje que cumpla con unas restricciones mínimas de codificación y diseño de cursos, además de la posibilidad de comunicación con servidores externos, entre otras mas que se mencionan en la sección 1.

La arquitectura que se muestra en la figura 38 es la que finalmente se utilizará para proporcionar a un Sistema Gestor de Aprendizaje la máxima capacidad de adaptación posible que se ha llegado a concretar en los capítulos anteriores. En ella se ve claramente como el profesor o tutor del curso proporcionará la información de los objetos de aprendizaje del curso, además de las calificaciones de los QTI's o evaluaciones intermedias cuando sea necesario, a través de la interfaz del Sistema Gestor de Aprendizaje. En este caso se trata de Moodle pero podrían ser fácilmente dotLRN o ILIAS, entre otros.

También se puede apreciar como el estudiante puede cumplimentar los datos obligatorios de su perfil a través de la interfaz del IMS-LIP de la que dispone el LMS.

Posteriormente, por medio de un servidor web que utiliza la tecnología XML-RPC, el Educational Server enviará toda la información recabada anteriormente al servidor de planificación o *Planning Server*, todo ello a través de la interfaz de Cliente de Planificación o *Planner Client* que se detalla en la sección 2.1 y que se ha integrado de forma transparente en el LMS de manera que pueda ser utilizada por el profesor en cualquier momento.

Una vez en el servidor de planificación, se utilizarán las reglas pedagógicas y de planificación necesarias para crear correctamente los ficheros de dominio y problema de planificación para utilizar alguno de los planificadores estudiados y obtener con ello un plan que incluya los diseños instruccionales de todos los estudiantes del curso, diseños instruccionales que corresponderán solo al SCORM seleccionado por el profesor.

Después, el servidor de planificación contestará a la llamada del cliente enviándole el plan que será integrado posteriormente a las bases de datos que almacenan la información

referente al SCORM adaptado de cada estudiante.

Como puede apreciarse en la figura, este es un proceso cíclico que solo termina cuando se ha planificado la información del SCORM que corresponda al último objetivo del curso.

En las siguientes secciones se detallan los componentes y requerimientos del cliente y servidor de planificación, así como el proceso de comunicación entre ellos.

Se concluye con la descripción de los resultados de la integración de todos estos elementos dentro del LMS Moodle, a través de sus interfaces correspondientes.

2.1. El Cliente de Planificación

El *Planner Client* o cliente de planificación se encarga de transmitir la información procesada de los objetos de aprendizaje y los perfiles de los estudiantes al servidor de planificación y recibir de este el plan en formato XML para después procesarlo y separar los diseños instruccionales de cada estudiante que serán remitidos a sus respectivas instancias de SCORM.

El cliente de planificación es un módulo que deberá instalarse o integrarse en el Sistema Gestor de Aprendizaje. Normalmente, con solo pegar el código del mismo en el lugar correspondiente dentro de la estructura de ficheros del LMS y llevar a cabo unas mínimas modificaciones a la presentación de los cursos en el sistema este módulo estará listo para utilizarse.

En particular habrá que agregar, según los pasos detallados en el apéndice A, los campos del perfil del estudiante necesarios para el proceso de adaptación y marcarlos como de carácter obligatorio, de manera que éste pueda acceder a las máximas capacidades de adaptación de sus cursos de las que el sistema disponga. Estos campos se muestran al profesor por medio de la interfaz de Moodle de la figura 39.

En el LMS Moodle en particular, las instancias de SCORM se almacenan por medio del fichero manifiesto y, además, cuando el paquete es cargado en el LMS este fichero es procesado para que su información estructural se almacene dentro de las bases de datos del sistema.

Este comportamiento no es exclusivo de Moodle, también se han encontrado estas ca-

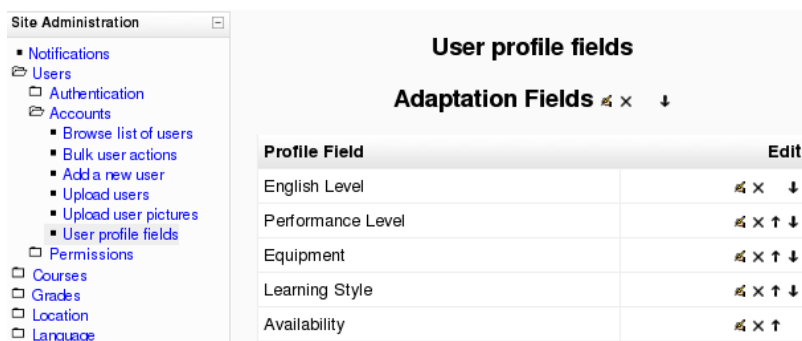


Figura 39. Nuevos campos del perfil del usuario para la adaptación de los SCORM's.

racterísticas en ILIAS y Caroline que sustentan su interfaz en los lenguajes PHP, HTML y javascript, ya que resulta necesario almacenar los objetos de aprendizaje y su estructura en medios mas accesibles que un fichero de texto XML.

El cliente de planificación, con el fin de integrar la funcionalidad de adaptación de diseños instruccionales en el núcleo del procesamiento de un objeto SCORM dentro del LMS o de revisar el correcto funcionamiento e instalación del cliente de planificación, se apoya en cuatro utilerías para la configuración del sistema que deben ser ejecutadas por el administrador del sistema.

Las funciones de estas tres utilerías, con las cuales se debe tener especial cuidado y que pueden ser instaladas con modificaciones mínimas en la mayor parte de los LMS's que soporten SCORM, son las siguientes:

- Update Database. Esta utilería debe ejecutarse una vez instalado el cliente del planificación dentro del LMS y antes de solicitar la adaptación de cualquier SCORM.

Se encarga de agregar el campo referente al identificador del usuario en la tabla de la base de datos del LMS a la que le corresponda almacenar los items, metadatos y secuencia de los objetos de aprendizaje. Una vez agregado el campo, todos los objetos del SCORM se multiplicarán según el número de usuarios para los que vaya a ser necesaria la adaptación. Una réplica mas será también necesaria para conocer la estructura original y completa del SCORM.

En el caso de Moodle ésta tabla se llama "scoes" y dado que en un principio solo

soportaba la visualización de los objetos del SCORM en el mismo orden para todos los integrantes del curso ha sido necesario agregar el campo de `userid` para diferenciar esta forma de visualizarlo.

- To Check Server Connection. Permite revisar la comunicación con el servidor de planificación ya que en ocasiones puede ser que el servidor de planificación no esté disponible o no sea accesible dadas las restricciones de seguridad que la empresa que maneja el servidor LMS tenga en su red.

Esta opción, por lo tanto, permite comprobar la comunicación y, en caso de que existiera algún problema, indicar la posible razón para que un encargado de soporte técnico pueda resolver la incidencia.

- Profile User Fields. Permite visualizar los campos de adaptación del usuario mostrando su nombre, tanto el que se muestra al estudiante como el de identificación dentro del sistema, de manera que sea posible detectar fácilmente un error en el fichero de configuración principal del cliente de planificación, en caso de que se produjera un error en la generación de planes.

Generalmente, ya sea por no saturar al estudiante de requerimientos de información o por que no se desea tener una adaptación tan refinada, los administradores de cursos deciden no agregar todos los campos de adaptación. Si alguno de los campos de adaptación que se le solicitan al estudiante no se encuentra disponible dentro del sistema, es imprescindible que se le haga saber al planner por medio del fichero de configuración, de lo contrario éste intentara solicitarlos todos al momento de la comunicación con el servidor de planificación y esto provocará un error.

- Pretty Print Plans. Esta utilidad permite al profesor y/o administrador revisar la lista de SCORMs que se han planificado para cada curso hasta ese momento.

También permite seleccionar un estudiante para visualizar el plan de un SCORM específico generado para este, de manera que sea posible comprobar la correcta generación y adaptación de los diseños instruccionales de cada SCORM y/o cada curso.

Las utilerías descritas en los puntos anteriores forman parte del cliente de planificación y pueden ser ejecutadas antes, durante o después de la llamada al servidor de planificación.

Por medio de la información almacenada en los campos del perfil del usuario y las bases de datos con los objetos de aprendizaje que forman parte de de los paquetes SCORM de un curso, es posible que el *Planner Client* obtenga de una forma mas eficiente toda la información necesaria acerca de los estudiantes (fig. 40) y los objetos de aprendizaje(fig. 41) que permiten alcanzar el objetivo de un curso; todo ello por medio de una función de preprocesamiento que se ejecuta para extraer de estas fuentes solo aquella información que es relevante para la generación de dominios y problemas en el servidor de planificación.

```
<rows>
  <row>
    <id>3</id>
    <name>ana</name>
    <englishLevel>Acceptable</englishLevel>
    <learningStyle>Active</learningStyle>
    <performanceLevel>High</performanceLevel>
    <equipment>Multimedia</equipment>
    <previousCoursesLevel>60</previousCoursesLevel>
    <availableTime>Much</availableTime>
    <score>Much</score>
  </row>
  <row>
    ...
  </row></rows>
```

Figura 40. Extracto del fichero XML de estudiantes generado por el cliente de planificación que integra solo la información necesaria para que el proceso de adaptación.

Sin embargo, cabe recordar que toda esta información también está almacenada dentro de los estándares IMS-LIP, IMS-QTI y la especificación SCORM de manera puede compartirse con otras plataformas de aprendizaje y/o utilizarse directamente para extraer la información que se enviará al *Planning Server*.

La interfaz que se describirá a detalle en la sección 2.3, es la pieza clave del cliente planificación ya que es la que permite seleccionar el curso y el SCORM que se desea adaptar, así como la evaluación intermedia previa a ese SCORM, para después enviar los ficheros de objetos y estudiantes correspondientes al servidor de planificación y posteriormente recibir e integrar los SCORM adaptados de cada uno.

```

<rows>
  <row>
    <Identifier>ITEM-action3 </Identifier>
    <Title>Ejercicios </Title>
    <language>es </language>
    <coverage>mandatory </coverage>
    <otherPlatformRequirements>none </otherPlatformRequirements>
    <learningResourceType>Exercise </learningResourceType>
    <difficulty>medium </difficulty>
    <typicalLearningTime>30 </typicalLearningTime>
    <resource>resources/images/A1.1.jpg </resource>
    <resourceId>RES-3 </resourceId>
    <isPartOf>
      <IP00>ITEM-task1 </IP00>
    </isPartOf>
    <isBasedOn>
      <IB00>ITEM-action1 </IB00>
      <IB01>ITEM-action2 </IB01>
    </isBasedOn>
    <requires/>
  </row>
  <row>
    ...
  </row>
</rows>

```

Figura 41. Extracto del fichero XML de objetos de aprendizaje generado por el cliente de planificación que integra solo la información necesaria para que el proceso de adaptación.

2.2. El Servidor de Planificación

El servidor de planificación o *Planning Server* es una parte fundamental dentro del proceso de adaptación del curso. Aunque no contenga ninguna parte visual dada la naturaleza del mismo, consta de diversos componentes realmente complejos, los cuales se enumeran a continuación.

Primero que nada se tiene el módulo del *servidor XML-RPC*. Este módulo se encarga no solo de recibir los archivos XML del perfil del usuario y el manifiesto con los metadatos de los objetos de aprendizaje, si no también de comunicar, secuenciar y controlar los procesos llevados a cabo por el resto de los módulos y, finalmente regresar el plan global con todas las secuencias de objetos de aprendizaje adaptadas a cada uno de los alumnos del curso en proceso.

Dentro de la secuencia de procesos que este módulo se encarga de controlar están los que se mencionan en los párrafos siguientes según su orden de ejecución.

El primer proceso que entra en ejecución es el de *decisión*. Como se ha mencionado

anteriormente, el servidor de planificación almacena los ficheros de dominio y problema de planificación de los SCORMS para los cuales se ha solicitado su adaptación previamente, así como los de estudiantes y objetos. Así, dependiendo de ciertas condiciones sobre el análisis de esos ficheros, se llevará el proceso de generación automática de dominios y problemas de planificación para utilizar alguna de las técnicas listadas a continuación:

- **PLANIFICACIÓN.** Las condiciones para que se ejecute este proceso se cumplen cuando no existe ningún fichero dentro del historial que haga referencia al SCORM para el cual se solicita la planificación y, además, cuando no se ha considerado ninguna evaluación intermedia, o <score> dentro del fichero de estudiantes, como condición para la ejecución de este SCORM.

Cuando se cumple con estas condiciones, el proceso de generación automática de dominios se ejecuta de la forma señalada en el capítulo 4.

- **PLANIFICACIÓN CONTINUA.** Cuando no se encuentran ficheros en el historial del servidor de planificación para el SCORM que se ha solicitado planificar y dentro del fichero del problema se ha considerado la calificación de la evaluación intermedia previa a dicho SCORM, es cuando se cumplen las condiciones para que se lleve a cabo el proceso de generación automática de dominios siguiendo las pautas de lo señalado en el capítulo 5 sección 3.1.

Es decir, modificando el fichero de estudiantes recibido del cliente de planificación en base a un conjunto de reglas de asociación.

- **RE-PLANIFICACIÓN.** Finalmente, si el fichero de los estudiantes o el fichero de los objetos ya existían previamente y uno de ellos ha cambiado, entonces el proceso de generación automática se lleva a cabo para el fichero que cambió. Si el que ha cambiado es el de estudiantes se regenera el problema de planificación y si es el de objetos el dominio. Y la generación se lleva a cabo dependiendo de las pautas marcadas por alguno de los dos puntos anteriores, dependiendo del contenido o no de la calificación de la evaluación previa en el fichero de los estudiantes.

El siguiente proceso que es llamado por el módulo XML-RPC del servidor es

el de *planificación*, que por medio de la sentencia `siadex -d HTNDomain.pddl -p HTNProblem.pddl -o plans.xml` ejecuta el planificador *SIADEx* o utilizando la sentencia `lpg -o LPGDomain.pddl -f LPGProblem.pddl -n 1 -incremental_goal_resolution -search_steps 500 -out plan` ejecuta el planificador *LPG-td* proporcionándole como entradas los archivos de dominio y problema generados en el proceso anterior y pidiéndole que el plan de salida lo plasme en un fichero XML llamado “plans”. Cabe recordar que sobre el plan generado por el planificador *LPG-td* es preciso ejecutar una función de post-procesamiento para obtener el fichero de planes con una estructura similar a la del extracto de la figura 42.

Finalmente, el módulo XML-RPC regresa el fichero “plans.xml” al cliente de planificación.

```

<root>
  <item>
    <userid>3</userid>
    <identifier>ITEM-action3</identifier>
  </item>
  <item>
    <userid>3</userid>
    <identifier>ITEM-action5</identifier>
  </item>
  <item>...
</root>

```

Figura 42. Extracto del fichero XML de planes generado por los planificadores *SIADEx* o *LPG-td*.

La estructura de este fichero corresponde a una secuencia ordenada de identificadores de objetos de aprendizaje, y cada objeto tiene asociado el identificador de uno de los usuarios estudiantes del curso. Así, un objeto puede estar referenciado dentro de este fichero tantas veces como el planificador lo haya incluido en el diseño instruccional de alguno de los estudiantes.

2.3. El Proceso de Comunicación

Como ya se ha mencionado anteriormente, una de las características principales con las que debe cumplir un LMS para integrarse adecuadamente a la arquitectura planteada, es que permita la inclusión de nuevos módulos dentro de su código fuente.

Se ha elegido Moodle como el sistema gestor de aprendizaje para llevar a cabo las pruebas de implementación ya que cuenta con una documentación amplia y una estructura de código muy accesible. Esto ha facilitado la integración del módulo del cliente de planificación que

está conformado por un conjunto de scripts en el lenguaje PHP que implementan diferentes protocolos de servicios web, bases de datos y manejo de ficheros XML para llevar a cabo todas las funciones que se requieren.

Las opciones de configuración del *Planner Client* ya se explicaron en la sección 2.1. Sin embargo, la opción principal de este elemento de la arquitectura merece una mención especial. Este es el módulo de *comunicación con el servidor de planificación*.

Utility to Generate Plans

Course: Educación e IA
Scorm: Introducción General
Quiz: Encuesta Inicial

Goals:
 Introducción

Plans created for next students:

[victoryanez](#)
[alexmatico](#)
[joralmocha](#)
[thelma](#)
[ulises](#)
[cariander](#)
[samuel](#)
[angelo](#)
[arturoe](#)
[alerios](#)
[diego](#)
[rubensn](#)
[anaflor](#)
[amairani,bq](#)
[eusebio](#)
[adalmiranda](#)
[yara](#)
[edgararvizu](#)
[fernando,adriel](#)
[jesusp](#)
[alumno](#)
[zhuzho](#)
[ferminivan](#)
[garlant](#)
[pakoruelas](#)

Figura 43. Comunicación con el servidor de planificación a través del módulo de planificación integrado en Moodle. Se puede elegir el curso, el SCORM a planificar, la evaluación previa, los objetivos específicos del curso y los objetos disponibles que dicho curso requiere (si es que estos existen).

Este módulo en concreto se muestra en la figura 43 y funciona de la siguiente forma.

Primero que nada, permite seleccionar el curso sobre el cual se encuentra el paquete

SCORM que se desea adaptar. Después, es posible seleccionar el referido SCORM y entonces habrá que indicar cuales son las metas u objetivos principales de aprendizaje de que se quieren adaptar para dicho paquete, además de los objetos “externos” que estos objetivos requieren. Finalmente se pincha el botón “Get Plan” y, si todo sale bien, la aplicación muestra una lista con los nombres de los alumnos del curso y un enlace a cada uno de sus planes.

Cuando se pincha el botón “Get Plan” se inicia el proceso de generación de los ficheros de estudiantes y objetos correspondientes al curso, quiz y SCORM seleccionados. Posteriormente se se hace una llamada al servidor de planificación por medio del protocolo XML-RPC; en dicha llamada se envían los ficheros anteriormente mencionado y se espera recibir un fichero XML con los diseños de aprendizaje para todos los estudiantes de ese curso. El fichero fichero del plan será finalmente procesado para integrar su información dentro de las bases de datos de Moodle, es decir, modificar las réplicas del SCORM para cada estudiante generadas por la utilidad *Update Database* en base a eliminando los objetos de aprendizaje que no se incluirán en su diseño instruccional o cambiando las relaciones de orden entre ellos.

Puede ocurrir que el servidor de planificación no regrese al cliente un fichero XML con el plan si no un mensaje de error indicando dos problemas principales: problemas de configuración del sistema o problemas en el proceso de creación o procesamiento de los ficheros de estudiantes y objetos.

El primer error ocurre cuando la instalación y configuración del cliente de planificación no se ha llevado a cabo adecuadamente. Es por ello que, como parte de la instalación del mismo, se incluye un paquete SCORM de prueba que no solamente sirva de ejemplo para la generación de posteriores paquetes, si no también para probar el sistema antes de su uso en cursos reales.

El segundo error ocurre cuando no se han etiquetado correctamente los metadatos de los objetos de aprendizaje, lo cual puede generar incoherencias en las relaciones entre los mismos que no permitan generar un plan, como por ejemplo un objeto *A* que esté basado en un objeto *B* y éste a su vez en el mismo *A*, pero obviamente a mayor escala. También puede causar precondiciones que generen un plan incoherente o precondiciones que no permitan generar un plan; por ejemplo, cuando no se define un objeto con una condición fuerte como

el idioma, y no se define también como opcional o se genera otro con el mismo nombre pero que no tenga dicha precondition.

Para evitar ese tipo de problemas, se sugiere que, dentro de un curso de prueba, se introduzca cualquier SCORM a utilizar en otro curso y se pruebe previamente con estudiantes ficticios; así podrá detectarse cualquier error de este tipo y solicitar la ayuda de un experto en el etiquetado de objetos de aprendizaje o en planificación para corregirlo.

Para aplicar el proceso de planificación y generar automáticamente los diseños instruccionales de 36 estudiantes para un SCORM con 95 objetos de aprendizaje, 68 primitivos y 27 compuestos, se han requerido en promedio 52 segundos utilizando el planificador jerárquico. Esto sucede por que, además del tiempo de planificación y generación automática de ficheros de planificación, todos los procesos que lleva a cabo el cliente de planificación (preprocesamiento de información, envío a través de servicios web e integración de planes en las bases de datos del sistema) toman aproximadamente 10 segundos para esta cantidad de estudiantes y objetos que en definitiva resulta bastante grande en comparación con el promedio de objetos que contiene un paquete SCORM tradicional.

Sin embargo, aunque el número de estudiantes sea el mismo, si la cantidad de objetos del SCORM y complejidad de sus relaciones es menor, el proceso completo (procesos del cliente y el servidor) puede tomar en promedio 8 segundos, como es el caso de otro SCORM con 38 objetos de aprendizaje, 27 primitivos y 11 compuestos para el cual se generaron 36 diseños instruccionales en ese tiempo (8segundos).

Con esta sencilla comparación es posible comprobar como la cantidad de estudiantes no tiene un gran impacto en el tiempo de generación que el usuario final percibe desde el cliente de planificación. Y, en base a lo observado en los experimentos llevados a cabo hasta el momento y que se describirán en el siguiente capítulo, generalmente el tiempo máximo para generar los diseños instruccionales de un SCORM de tamaño promedio (35 objetos de aprendizaje) y un número promedio de estudiantes (50) no supera el minuto.

3. La Vista Final del LD en Moodle

Cuando el profesor y el diseñador de contenidos han terminado su trabajo de diseño del curso y etiquetado de los objetos, habrán de empaquetarlo en uno o varios paquetes SCORM almacenados en un fichero zip. A partir de ese punto ya será posible introducir el SCORM en un sistema gestor de aprendizaje, según el método definido por cada uno de ellos.

The screenshot displays the Moodle course interface for 'Minicurso de Matemáticas Discretas'. At the top, a message states: 'SCORM creado específicamente para utilizar como ejemplo en el proyecto tutelado.' Below this, the course content is organized into two main sections: 'Vistas del Administrador del Curso' and 'Contents Organizations'. The 'Vistas del Administrador del Curso' section shows a hierarchical tree of course objects, including 'Discrete Maths', 'Algorithms', 'Complexity', 'Boolean Algebra', and 'Logic Gates'. The 'Contents Organizations' section shows a list of objects under the 'Logic and Sets' organization, including 'LS NarrativeText', 'LS Problem St', 'LS Exercise', and 'LS Experiment'.

Figura 44. Curso de ejemplo de Matemáticas Discretas Visto desde el LMS Moodle. Esta vista corresponde a la forma en que el profesor, administrador o generador de contenidos del curso lo verá.

En el caso de Moodle, solo es necesario introducir una nueva actividad tipo SCORM dentro del curso, ponerle nombre, seleccionar el fichero zip que contenga el paquete deseado, seleccionar el resto de las opciones disponibles y pedir que se despliegue el paquete.

Cuando se visualice un SCORM dentro de un curso como profesor, se verá algo como lo que se muestra en la figura 44 con todos los objetos que el curso contiene sin eliminar u ordenar de forma distinta ninguno de ellos ya que el perfil del profesor no se toma en cuenta para mostrarlo. En Moodle es posible que un profesor de cierto curso pueda fungir también como estudiante y, dado que a los estudiantes se les genera un curso adaptado en el que no es posible ver la estructura original del SCORM, se recomienda a los profesores que se creen

una cuenta paralela a la suya para contar siempre con una visión completa del curso.

Una vez que se ejecuta todo el proceso descrito por la arquitectura presentada en las secciones anteriores, el despliegue de la actividad SCORM se muestra de forma diferente para cada estudiante, por esta razón es que se ha dado la recomendación del párrafo anterior.

Minicurso de Matemáticas Discretas

SCORM creado específicamente para utilizar como ejemplo en el proyecto tutelado.

Contents
Organizations
Discrete Maths ▾

Vista del Estudiante
Thom

- Discrete Maths
 - Algorithms
 - Basic Algorithms Lecture
 - Complexity
 - Complexity Problem St
 - Complexity Exercise
 - Complexity Experiment
 - Boolean Algebra
 - Logic Gates
 - Logic Gates1 NarrativeText
 - Logic Gates1 Exercise
 - Logic Gates2 NarrativeText
 - Logic Gates2 Exercise
 - Boolean Algebra Simulation
 - Minimization of Circuits Exercise

Contents
Organizations
Logic and Sets ▾

- Logic and Sets
 - LS NarrativeText
 - LS Problem St
 - LS Experiment
 - LS Exercise

Figura 45. Curso de ejemplo de Matemáticas Discretas Visto desde el LMS Moodle. Esta vista corresponde a la forma en que el estudiante Thom vería el curso.

Algunos estudiantes pueden tener cursos idénticos si tienen perfiles idénticos. Incluso hay algunos estudiantes que tendrán cursos realmente cortos dado que no se les mostrarán los objetos requeridos. En el caso del curso adaptado que se ha generado para Thom y que se muestra en la figura 45 es posible observar que en su diseño instruccional se añade un contenido extra del tipo Organization, es decir, el objeto requerido *Logic and Sets*.

No está por demás mencionar que la forma de desplegar un paquete SCORM en Moodle permite marcar como vistas las actividades que ya se han llevado a cabo, pero esta característica puede variar dependiendo del sistema gestor de aprendizaje que se utilice.

Una vez integrada toda la arquitectura de generación automática de diseños de aprendi-

zaje soportando dinamismo en los cursos en un LMS, y probada esta integración a través de un curso y estudiantes de prueba, se puede dar el siguiente paso que es probarla en entornos reales, tema del que se hablará en el siguiente capítulo.

CAPÍTULO 7

Experimentación en Dominios Reales

En el capítulo anterior se ha comprobado que es factible que los algoritmos de ingeniería de conocimiento y técnicas de planificación en entornos dinámicos propuestos en los capítulos 4 y 5 puedan integrarse de manera relativamente sencilla en la arquitectura y uso actual que de los Sistemas Gestores de Aprendizaje hacen de la especificación SCORM; sistemas que se utilizan mas comúnmente para impartir cursos a distancia.

Pero la comprobación técnica de esa integración no implica que los diseños de aprendizaje generados por un planificador, los parámetros de adaptación que se han elegido y las técnicas de respuesta a cambios dinámicos puedan ser representadas en cursos reales. Tampoco implica que se pueda sacar suficiente provecho al esfuerzo previo de diseño objetos de aprendizaje y reglas pedagógicas. Ni que finalmente los estudiantes terminarán aceptando tanto o mas los diseños instruccionales generados a través de este medio.

Todo lo anterior solo puede ser comprobado si se lleva a cabo una ardua labor de implementación de estas tecnologías en cursos reales y la posterior experimentación con las respectivas muestras de estudiantes. Esto es lo que se pretende con los experimentos inicia-

les que se han llevado a cabo por medio de los casos de uso y sus respectivas validaciones descritas en la siguientes secciones.

1. Casos de Uso

Los siguientes casos de uso se desarrollaron en diferentes etapas del proceso de investigación y, por lo tanto, algunos de ellos no implementan todas las técnicas y/o capacidades de adaptación que se han mencionado en los capítulos anteriores.

El primer caso describe solo las técnicas de planificación iniciales, para las que ya se utiliza un servidor de planificación y los ficheros de planificación generados por los algoritmos de ingeniería de conocimiento descritos en el capítulo 4, así como el paquete SCORM y el LMS Moodle como medios de información.

En el segundo caso se muestra como se ha implementado la arquitectura completa para entornos dinámicos con una muestra mucho mas variada, utilizando mas información de los perfiles de los estudiantes, en un curso mucho mas grande (con mas objetivos y evaluaciones intermedias).

El servidor de planificación¹ es un procesador Pentium IV con 2 GB de memoria RAM que utiliza el sistema operativo Madriva Linux 2007.

Los algoritmos de ingeniería de conocimiento para la traducción de los objetos de aprendizaje y perfil del estudiante a dominio y problema de planificación respectivamente, fueron implementados en el lenguaje interpretado python que, a pesar de que no se caracteriza por su velocidad de procesamiento, permite un fácil descripción y depuración de errores, así como una rápida depuración y conectividad con otros lenguajes y/o servicios como PHP, XML-RPC, etc.

Los servicios web, algoritmos de traducción, llamadas a planificadores y demás procesos soportados por el servidor de planificación son controlados por scripts en el lenguaje PHP. Se ha elegido este lenguaje pues es fácil y compatible con la línea de programación seguida

¹Servidor de Planificación: <http://sepia.ugr.es/server>

desde el desarrollo del cliente de planificación que también se implementó en el. Sin embargo, esto no influye en el lenguaje en el cual deba estar programado el LMS a utilizar.

1.1. Grupo Tadel: Planificación

A través de un contrato de investigación con la empresa española de e-learning Grupo Tadel², como se reportó en [Figuroa-Martínez et al., 2008] y [Toro et al., 2008], un grupo de 17 estudiantes siguió un curso de 20 horas de *Microsoft Power Point 2003*.

Los resultados fueron muy prometedores y tuvo un impacto significativo en medios de comunicación locales y portales de investigación oficiales de España[Alguacil-Martin, 2008].

La tabla 11 indica la distribución del estilo de aprendizaje Honey-Alonso de la muestra de 17 estudiantes, así como el número de estudiantes que tenían restricciones temporales y los que no, es decir mucho o poco tiempo para llevar a cabo las actividades del curso.

Estilo <i>Activo</i>	1
Estilo <i>Reflexivo</i>	12
Estilo <i>Teórico</i>	2
Estilo <i>Pragmático</i>	2
Estudiantes <i>con</i> restricciones temporales	11
Estudiantes <i>sin</i> restricciones temporales	6

Tabla 11. Muestra de 17 estudiantes de un curso real de MS Power Point 2003, sus estilos de aprendizaje y sus restricciones temporales.

El repositorio de objetos de aprendizaje, almacenado en el mismo servidor de los cursos y donde también se instaló el cliente de planificación, para este curso estaba compuesto por 50 objetos de aprendizaje, los cuales fueron traducidos a los ficheros de dominio y problema de planificación soportados por el planificador HTN SIADEX.

El dominio de planificación incluía todos los objetos del curso transformados a 50 acciones durativas, 19 tareas compuestas y 27 métodos de descomposición de tareas, sin mecanismos de inferencia o procedimientos de control de búsqueda adicionales.

El proceso de traducción de los objetos de aprendizaje al dominio de planificación tomó

²Ver: <http://www.grupotadel.com>

2.1 segundos, mientras que los perfiles de los estudiantes fueron traducidos a un problema de planificación en menos de 1 milisegundo.

Todos los planes o diseños instruccionales del curso completo para los 17 estudiantes fueron encontrados por el planificador en 0.76 segundos. Debe considerarse que las principales restricciones que se consideraron dentro del dominio fueron las de disponibilidad del estudiante y ordenación de acciones en base al estilo de aprendizaje del mismo. Por lo tanto, clasificando los planes en base a estas dos características del perfil del estudiante, se obtuvieron diferentes longitudes del plan tal y como se muestra en la tabla 12, en la cual se puede observar como la mayor diferencia entre la longitud de unos planes u otros la marca la disponibilidad temporal del estudiante.

<i>Id del Estudiante</i>	<i>Activo</i>	<i>Reflexivo</i>	<i>Teórico</i>	<i>Pragmático</i>	<i>Sí Deadlines</i>	<i>No Deadlines</i>	<i>Extensión del Plan</i>
1, 12			X		X		28
2, 6, 8, 15		X				X	35
3	X					X	34
4				X	X		29
5, 7, 9, 10, 13, 14, 16, 17		X			X		29
11				X		X	36

Tabla 12. Número de acciones en el plan para el curso de MS Power Point 2003 por cada perfil de estudiante existente clasificado en base a su estilo de aprendizaje y restricciones temporales.

1.2. Seminario de Inteligencia Artificial y Educación: Planificación Continua y Re-planificación

Este experimento se llevó a cabo por medio de una colaboración entre la Universidad de Granada en España y las Universidades de la Sierra y de Sonora en México y sus resultados se ha difundido a nivel internacional a través de su publicación en [Morales and García-Gorrosieta, 2010].

El cliente de planificación se encontraba en el servidor de cursos Moodle de la Universidad de la Sierra y contenía un repositorio de objetos de aprendizaje dividido en cuatro paquetes SCORM que contienen 20 objetos de aprendizaje el que menos y 95 objetos de aprendizaje el que mas en las distribuciones de objetos primitivos y compuestos que se muestran en la tabla 13, así como también 4 evaluaciones intermedias. El etiquetado de estos objetos se

efectuó en un máximo de 4 horas por cada SCORM y el tiempo de generación de los diseños instruccionales de los mismos para todos los estudiantes del curso, tomado desde el cliente de planificación, tuvo una variación de entre 3 y 52 segundos en promedio para los SCORMS con menos y mas mas objetos de aprendizaje respectivamente.

SCORM	OBJETOS		
	<i>Primitivos</i>	<i>Compuestos</i>	<i>Total</i>
I	68	27	95
II	35	9	38
III	17	6	20
IV	23	8	29

Tabla 13. Cantidad de objetos de aprendizaje para cada SCORM del seminario de IA y Educación. Se puede ver como en ocasiones la suma de objetos de aprendizaje es mayor que el total de los mismo debido a que algunos de ellos se encuentran en los dos conjuntos.

El curso aplicó a una muestra de 20 estudiantes de entre 18 y 43 años de edad, con escolaridad mínima de bachillerato y máxima de doctorado, quienes se matricularon en un seminario a distancia titulado *Educación e inteligencia Artificial* el cual tuvo una duración de 4 semanas.

Antes de iniciar el curso se aplicó el test de estilos de aprendizaje CHAEA y con sus resultados, además del resto de su información personal, los estudiantes cumplimentaron los campos correspondientes a la adaptación dentro de la sección especificada para ello en Moodle. Cabe resaltar que algunos de ellos (4) no rellenaron esa información si no hasta pasados algunos días o después de la primera semana del curso.

El sumario de la información recabada por medio de la información de su perfil, que es de interés para la generación de diseños de aprendizaje se puede observar en la tabla 14. En esta tabla se puede observar como la distribución de estilos de aprendizaje entre todos los estudiantes está muy bien distribuida pues, aunque ninguno obtuvo el estilo de aprendizaje Teórico, debido al paralelismo descrito en el apéndice B, se puede considerar que un 40% de ellos resultó ser Teórico mientras de un 60% obtuvo Pragmático como estilo de aprendizaje al unir con el porcentaje de los que obtuvieron un estilo Activo.

También se puede ver como solo es 35% de ellos tenía restricciones temporales y, aunque

ESTILO DE APRENDIZAJE	Cantidad	Porcentaje
Activo	7	35 %
Reflexivo	8	40 %
Teórico	0	0 %
Pragmático	5	25 %

NIVEL DE INGLÉS	Cantidad	Porcentaje
Excelente	0	0 %
Aceptable	13	65 %
Bajo	7	35 %
Muy Bajo	0	0 %

DISPONIBILIDAD	Cantidad	Porcentaje
Mucha	13	65 %
Poca	7	35 %

RENDIMIENTO	Cantidad	Porcentaje
Alto	11	55 %
Normal	8	40 %
Bajo	1	5 %

Tabla 14. Resumen de los Perfiles de Estudiantes para el Curso de IA y Educación.

la mayoría contaba con un rendimiento alto o medio, ninguno tenía un excelente nivel de inglés y solo el 65 % consideraba aceptable esta última característica. Todos estos precedentes explican por que se han podido obtener planes tan variados para los distintos estudiantes.

Cabe resaltar que como soporte y seguimiento al desarrollo del curso, se llevaron a cabo sesiones de videoconferencia semanales para aclarar dudas respecto al uso de la plataforma utilizando el software Elluminate³. También se utilizaron las herramientas de foro y mensajes privados para tener una comunicación directa con el tutor y promover la participación y colaboración entre los estudiantes.

Estas variaciones no aumentaron conforme se aplicaban los métodos de planificación continua o re-planificación, tampoco se mostró una disconformidad en los estudiantes cuando se aplicaba la replanificación a sus SCORMs debido a que se optó por aplicar este método solo antes de que se hubieran cursado como máximo la mitad de las actividades de un SCORM. En el SCORM que si fue notorio este método fue en el número 2 que cuenta con una gran cantidad de objetos de aprendizaje, por lo que se se ha concluido que se recomendará la cantidad máxima de 50 objetos de aprendizaje por paquete SCORM.

2. Validación a Posteriori de los Diseños Instruccionales

La validación que se ha llevado a cabo de los diseños instruccionales generados solo toma en cuenta la opinión del estudiante acerca de sus diseños instruccionales personalizados, en

³Disponible en: <http://www.illuminate.com/>

especial con respecto a los factores de adaptación, para lo cual se han aplicado una serie de encuestas antes y después de impartir el curso. Lo que no se pretende en esta sección es generar documentos para la evaluación exhaustiva y comparativa de estos diseños, ni comparar las opiniones expresadas en las encuestas con las de otras muestras de estudiantes ya que esa es una labor que se está llevando a cabo actualmente como trabajo futuro de la tesis.

En el caso de la generación de diseños instruccionales para el curso del primer caso de uso, las calificaciones obtenidas en la evaluación general del mismo fueron relativamente buenas, lo cual indica que son y similares al promedio de cursos anteriores en los que nos se han utilizado estas tecnologías de adaptación.

Por otro lado, aunque la motivación de los estudiantes al inicio del curso no era muy elevada - el 60% tenía una motivación media - al final del curso se pueden resaltar los siguientes aspectos:

- Solo el 33% seguía pensando que el contenido del curso no era suficientemente interesante.
- Solo el 20% de los estudiantes no aprendió suficiente o mucho sobre los conceptos impartidos.

Según los resultados de la encuesta de la tabla 15 se mostró un cambio de opinión generalizado en el grupo de estudiantes con respecto a lo que el curso podría aportarles, lo cual es sumamente notorio dadas las respuestas obtenidas en las últimas tres preguntas sobre el grado de satisfacción en diferentes aspectos del curso.

El segundo caso de uso, los estudiantes de este curso obtuvieron un promedio general de 8.27 sobre 10 de calificación en las 4 evaluaciones que se aplicaron durante el desarrollo del mismo, lo cual indica un rendimiento un tanto superior a la media con respecto al rendimiento del curso anterior, pero aún no se ha aplicado dicho curso en particular de manera no adaptada como para poder confirmar esta suposición.

En la tabla 16 se puede apreciar como la final del seminario la mayor parte de los estu-

Motivación de los estudiantes antes del curso.	
<i>Media</i>	60 %
<i>Alta</i>	27 %
<i>Muy Alta</i>	13 %
Calificaciones obtenidas por los estudiantes al final del curso (0-10).	
<i>menos de 5</i>	2 de 17
<i>Entre 5 y 7</i>	7 de 17
<i>Entre 7 y 9</i>	4 de 17
<i>Menos 9</i>	4 de 17
¿Encontraste interesantes los contenidos?	
<i>Poco</i>	33 %
<i>Suficiente</i>	53 %
<i>Mucho</i>	14 %
¿Podrías decir que has aprendido?	
<i>Poco</i>	20 %
<i>Suficiente</i>	60 %
<i>Mucho</i>	20 %
¿Cómo ha estado el desarrollo temporal del curso?(1-10)	
<i>Promedio</i>	8.4

Tabla 15. Resultados de la encuesta aplicada a los estudiantes al final del curso de Microsoft Power Point 2003.

diantes encontraron sus contenidos muy interesantes lo cual puede indicar, no solo que el tema ya en sí resultaba interesante, si no también que los contenidos se ajustaron mas a su estilo de aprendizaje.

Pregunta	¿Piensas que el contenido de los temas fue interesante?		¿Piensas que el tiempo asignado para llevar a cabo las actividades fue suficiente y se adaptó a tus necesidades?				¿Cuánto piensas que has aprendido?	
	Cantidad	Porcentaje	POCO TIEMPO		MUCHO TIEMPO		Cantidad	Porcentaje
Valor			Cantidad	Porcentaje	Cantidad	Porcentaje		
<i>Mucho</i>	13	65 %	5	71 %	7	53 %	4	20 %
<i>Bastante</i>	5	25 %	1	14 %	5	38 %	9	45 %
<i>Normal</i>	2	10 %	1	14 %	1	7 %	6	30 %
<i>Poco</i>							1	5 %

Tabla 16. Resumen de la Encuesta Final para el curso de IA y Educación.

Por otro lado, el ajuste temporal resultó ser satisfactorio o muy satisfactorio independientemente de si la disponibilidad horaria de los estudiantes era mucha o poca, aunque se continúa con la tendencia del curso anterior de que solo un porcentaje mínimo de los estudiantes tiene poca disponibilidad horaria.

Con respecto a la curva de aprendizaje se puede resaltar como un 65 % de los estudiantes aprendió mucho o bastante de los contenidos del curso, mientras que el resto aprendió lo normal o poco contando entre estos últimos a la mayor parte de las personas que tenían poca disponibilidad horaria o que tenían un alto grado académico, por lo que se supone que muchos de los conceptos del curso podrían conocerlos previamente.

Finalmente, y aunque no se encuentra dentro de la tabla, es importante mencionar que un 75 % de los estudiantes estarían interesados en utilizar de nuevo la plataforma para cursar otros temas.

De este último caso de uso se está llevando a cabo en estos momentos un experimento en el que se imparta dicho seminario por medio del curso a distancia pero sin utilizar la arquitectura de generación de diseños instruccionales de esta memoria, lo cual permitirá contrastar la información actual y verificar si esta técnica funciona igual, mejor o peor para la impartición del mismo curso.

Lo que si es preciso mencionar es que durante el desarrollo de ese seminario se están reutilizando los objetos de aprendizaje definidos para el seminario anterior, con unas mínimas modificaciones. Lo cual hace constar que la reusabilidad de estos contenidos retribuirá a la larga el esfuerzo de producción de los mismos.

CAPÍTULO 8

Conclusiones y Trabajo Futuro

El presente capítulo está dedicado a discutir los resultados que se han obtenido a raíz de la investigación llevada a cabo en los años que se estuvo desarrollando el proyecto que se expone en esta memoria. Se intentará resaltar la relevancia de dichos resultados y su impacto, no solo dentro de la comunidad educativa, si no también en la de inteligencia artificial.

Para ello se ha decidido dividir el capítulo en dos secciones. La primera muestra las principales contribuciones de esta investigación, ciertas carencias con las cuales es necesario lidiar y la difusión del mismo a través de congresos, simposios y revistas. La segunda sección proporciona una visión general de las principales líneas de investigación que se podrían abordar en un futuro, e incluso algunas propuestas generales para atacar estos problemas que son clave para la viabilidad del futuro desarrollo e implementación masiva en entornos reales de las técnicas de AIP&S dentro de las plataformas de educación a distancia que se apoyan en el internet.

1. Conclusiones

A lo largo de esta memoria se ha dejado claro el echo de que los métodos actuales de generación de dominios y problemas de planificación para resolver problemas de adaptación a estudiantes de cursos en línea pueden llegar a ser tan costosos como la adaptación manual de los mismos, ya que este procedimiento manual es equiparable al tiempo y esfuerzo que requiere un experto en AIP&S para realizar todo el proceso de ingeniería de conocimiento necesario para la generación de los archivos en el lenguaje PDDL.

Por esto mismo es mas que justificable, como se propone por primera vez en [Vassileva and Wasson, 1996] y posteriormente en [Mohan et al., 2003], el esfuerzo adicional que harían los tutores y diseñadores de contenidos al recabar una mayor información sobre los objetos de aprendizaje, en este caso a través del etiquetado de los mismos por medio de los metadatos definidos en el estándar IEEE-LOM, en particular los mencionados en el capítulo 3 sección 1 como mínimo. De estos metadatos se obtendría la información suficiente para generar un dominio de planificación consistente y, dado que los objetos de aprendizaje pueden ser reutilizables, e incluso comercializables, esto daría un valor añadido al dicho esfuerzo.

Tampoco se puede dejar de lado que esta es la primera investigación en la que se adquieren, tanto la información del perfil del estudiante, como de las actividades del curso directamente de la información proporcionada por los estándares integrados en un LMS y, que se ha logrado que este los metadatos que describen el perfil del estudiante sean suficientes para generar el archivo de problema de planificación en PDDL compatible con el dominio obtenido a partir de la información de los objetos de aprendizaje.

Para poder generar estos dominios y problemas se han diseñado diversos algoritmos que realizan un proceso de ingeniería de conocimiento sobre los metadatos de los objetos y los perfiles de los alumnos (capítulo 4). Gracias a estos algoritmos es posible generar automáticamente los ficheros de dominio y problema de planificación en PDDL para dos paradigmas de planificación inteligente distintos, jerárquico y basado en estados, que al ser utilizados por los planificadores SIADEX y LPG-td respectivamente han dado como resultado diseños

instruccionales adaptados aceptables.

Estos paradigmas de planificación son los que se han aplicado anteriormente en este tipo de problemas y resultaba relevante saber cual de ellos es el mas completo dados los requerimientos del problema de adaptación de cursos. A través de esta investigación, se podido comprobar empíricamente que el paradigma que mas necesidades soporta, hasta el momento, es el *jerárquico*. En particular, si se utiliza el planificador SIADEX, dada su naturaleza jerárquica, su manejo de acciones con el mismo nombre y sus enormes capacidades al lidiar con restricciones temporales, resulta aún más sencillo y completo el proceso de adaptación de contenido de un curso.

Por otro lado, una vez que se ha comprobado que es posible generar automáticamente planes de cursos adaptados a partir de objetos de aprendizaje y perfiles de estudiantes bien etiquetados, ha resultado relativamente sencillo adaptar dicho proceso al soporte de entornos e-learning dinámicos. En este tipo de entornos existen cambios constantes en los perfiles de los estudiantes o en las decisiones que toma el profesor respecto a los contenidos del curso, así como también cierto grado de incertidumbre respecto a variables que influyen en la generación de diversos tramos del diseño instruccional de un estudiante, como por ejemplo, la calificación que se obtendrá en una evaluación intermedia.

Para lidiar con este dinamismo del entorno se han propuesto dos estrategias:

- La primera se basa totalmente en estándares pero tiene una adaptación restringida a las variables y valores de las mismas que se determinen previa ejecución del curso. Sin embargo, resulta un primer gran acercamiento a la generación automática de diseños de aprendizaje en su nivel B, el cual puede irse refinando con el paso del tiempo.
- La segunda resulta ser mas flexible ya que permite utilizar tanto estándares como información propia de un LMS para generar los diseños instruccionales y lidiar con cambios, tanto en el curso como en el perfil de estudiante, que se vayan efectuando y procesando en tiempo real.

Para cada una de estas dos estrategias, que en su núcleo principal integran los algoritmos de ingeniería de conocimiento definidos en el capítulo 4, se ha definido e implementado

una arquitectura de comunicación entre el sistema gestor de aprendizaje y el servidor de planificación. Dicha arquitectura permite soportar el proceso de extracción de conocimiento a raíz de los metadatos de los objetos de aprendizaje y perfiles de los estudiantes definidos en el LMS transmitiéndolo al servidor de planificación en el cual se generarían posteriormente los diseños de aprendizaje o diseños instruccionales adaptados, para después ser devueltos al player de LD's o a Sistema Gestor de Aprendizaje donde serían mostrados cada uno al alumno que le corresponde.

El despliegue de los objetos de aprendizaje dentro del sistema gestor de aprendizaje que se ha utilizado, en este caso Moodle, requiere del soporte del estándar IMS de paquetes de contenidos y/o de paquetes SCORM. Sin embargo, la plataforma también debe adaptarse al proceso de despliegue personalizado de las secuencias generadas y para éste propósito se ha diseñado dentro del llamado cliente de planificación una utilidad que prepara el sistema gestor de aprendizaje para recibir y mostrar las secuencias de objetos.

Se ha seleccionado la arquitectura de la segunda estrategia de planificación para entornos dinámicos para ser evaluada por medio su implementación en un curso real, con alumnos reales. Esto se debe a que es la que integra la mayor parte de las estrategias de adaptación descritas en la memoria y la que permite mayor flexibilidad para su integración a corto plazo en diversos LMS's.

Los resultados obtenidos de la experimentación con esta integración, según las opiniones de los alumnos una vez finalizado el curso, han sido bastante satisfactorios y han marcado la pauta para saber cuáles son las líneas de trabajo sobre las cuales se debe trabajar con más premura para mejorar esta propuesta.

Las líneas de trabajo futuras están descritas en la sección 2 y se han definido no solo gracias a la retroalimentación obtenida de los estudiantes que han participado de la experiencia de los cursos piloto, si no también a raíz de los consejos de revisores expertos que han evaluado las conclusiones de este trabajo para su publicación en revistas y memorias de los congresos que mencionamos en la siguiente sección.

1.1. Publicaciones Obtenidas

Los resultados obtenidos a raíz de este proyecto de investigación han sido presentados en congresos de gran relevancia y publicadas por editoriales muy prestigiosas en el mundo de la investigación en ciencias de la computación e inteligencia artificial, así como en un par de revistas incluídas en el Journal Citation Reports de Thomson Reuters. Las referencias son las siguientes:

1. Luis Castillo, Lluvia Morales, Arturo González-Ferrer, Juan Fdez-Olivares, Óscar García-Pérez. *Knowledge Engineering and Planning for the Automated Synthesis of Customized Learning Designs*. Current topics in Artificial Intelligence (CAEPIA 2007). Springer LNAI 4788, 2007.
2. Lluvia Morales, Luis Castillo, Juan Fdez-Olivares, Arturo González-Ferrer. *Automatic Generation of User Adapted Learning Designs: An AI-Planning Proposal*. Adaptive Hypermedia and Adaptive Web-Based Systems(AH2008). Springer LNCS 5149, 2008.
3. Lluvia Morales, Luis Castillo, Juan Fdez-Olivares, Arturo González-Ferrer. *Building Learning Designs by Using an Automatic Planning Domain Generation: A State-Based Approach*. Frontiers in Artificial Intelligence and Applications (STAIRS2008). IOS Press 179 Ed. A. Cesta and N. Fakotakis.
4. José Figueroa-Martínez, Lluvia Morales, Luis Castillo. Modificación del Módulo SCORM para el Soporte Básico del IMS-LD. Moodle Moot 2008 (Barcelona, SPAIN).
5. Torres Toro, Sebastián; Castillo Vidal, Luis; Morales, Lluvia; Jiménez-Galera, M^a Dolores; Llorca-Díez, María Ángeles; Ortega Carrillo, José Antonio. Personalización de itinerarios formativos en Moodle adaptados a los Estilos de Aprendizaje de Honey-Alonso. Moodle Moot 2008 (Barcelona, SPAIN).
6. Morales, L., Castillo, L. and Fdez-Olivares. Planning for Conditional Learning Routes. Advances in Artificial Intelligence (MICAI 2009). Springer LNCS.
7. Lluvia Morales, José Figueroa-Martínez, Luis Castillo: Adapted SCORM/AICC Objects. iMoot 2010. At: <http://2010.imoot.org/>

8. Luis A. Castillo, Lluvia Morales, Arturo González-Ferrer, Juan Fernández-Olivares, Daniel Borrajo, Eva Onaindia: *Automatic Generation of Temporal Planning Domains for E-Learning Problems*. J. Scheduling 13(4): 347-362 (2010).
9. Lluvia Morales, Jesús Miguel García-Gorrosieta: *LMS Plan: An Architecture to Integrate Adapted Learning Sequences with LMS's*. Workshop on Intelligent Learning Environments, MICAI 2010. Hidalgo, México.
10. Lluvia Morales, Jesús Miguel García: *LMS Plan: An architecture to integrate Adapted Learning Sequences with LMS's*. Research in Computing Science: Special Issue on Advances in Artificial Intelligence. Vol. 51. pp 189-196.

También se espera la publicación en la revista Knowledge Engineering Review (el trabajo ya está aceptado) del artículo *On the automatic compilation of e-learning models to planning* redactado en colaboración con las universidades Carlos III de Madrid y Politécnica de Valencia en el cual se presenta la comparativa de los resultados del proceso de ingeniería de conocimiento de esta memoria para generar dominios y problemas de planificación a partir de un dominio de e-learning, con los de otros dos métodos diseñados en las mencionadas instituciones.

Durante el desarrollo de la tesis también se han redactado publicaciones paralelas que, si bien no consideran la aplicación de las técnicas de AIP&S en dominios de e-learning, si sirvieron de refuerzo para entender a profundidad el funcionamiento del planificador HTN SIADEX, técnicas de ingeniería de conocimiento sobre otro tipo de información que pueden ser utilizadas para generar dominios y problemas de planificación para otro tipo de dominios reales, el flujo de trabajo de diferentes organizaciones (particularmente las de e-learning), así como temas tan diferentes como la biotecnología y la aplicación de algoritmos genéticos y lógica difusa para resolver algunos de los problemas de la misma.

Las siguientes son algunas de las publicaciones que se han obtenido a lo largo de estos años y que abordaron los temas descritos en el párrafo anterior:

- González-Ferrer, A., Castillo, L., Fdez-Olivares, J., Morales, L. Workflow Planning

for E-learning Center Management. 8th IEEE International Conference on Advanced Learning Technologies(ICALT 2008), IEEE Computer Society Press.

- Morales, L., González, J.R., Pelta, D. Fuzzy Contact Maps Overlap for Protein Comparison: the Roles of Fitness and Normalization. XIV Congreso Español Sobre Tecnologías y Lógica Fuzzy(ESTYLF08).
- González-Ferrer, A., Castillo, L., Fdez-Olivares, J., Morales, L. Towards the Use of XPDL as Planning and Scheduling Modeling Tool: the Workflow Patterns Approach. Advances in Artificial Intelligence (IBERAMIA 2008) Springer LNAI.
- González, J.R., Pelta, D.A., and Morales, L. Fuzzy Contact Maps Overlap for Protein Comparison: the Roles of Fitness and Normalization and the relation with Crisp Contact Maps. In *Mathware & Soft Computing*, Vol 16, No 1 pp.87-102 (2009).
- Luis Castillo, Juan Fernandez-Olivares, Antonio González, Gonzalo Milla, David Prior, Lluvia Morales, José Figueroa, Victor Pérez-Villar A Knowledge Engineering Methodology for Rapid Prototyping of Planning Applications. 23rd International FLAIRS Conference. Daytona Beach, Florida, USA. May 2010.

2. Trabajo Futuro

En esta sección se describirán de forma breve algunas de las líneas de trabajo que se han abierto a partir de esta investigación y que pudieran ser abordadas como nuevas propuestas para proyectos de fin de carrera, proyectos de investigación tutelada o tesis doctorales (ya se existen algunas expresiones de interés al respecto). O bien como desarrollos empresariales inteligentes que pudieran ser comercializados a mediano plazo.

2.1. Estrategias de What-iffing y/o Iniciativa Mixta

What-iffing se refiere al diseño de interfaces que permitan la modificación de las variables pedagógicas de adaptación de manera previa al proceso de ejecución del curso. Esto servirá al profesor para elegir la mejor estrategia de diseño instruccional para cada curso particular, permitiendo una adaptación no solo a las preferencias del estudiante si no también del tutor.

Las interfaces de iniciativa mixta, por el contrario, se ejecutan durante el proceso de ejecución del plan y permiten recomendar al profesor las variables y/o acciones necesarias para continuar con el proceso de planificación continua o re-planificación que se debiera ejecutar y recabar dicha información para tomarla en cuenta para la construcción de las siguientes secciones del diseño instruccional.

2.2. Dominios e-Learning Colaborativos

Se intentaría incluir, dentro de las precondiciones de las acciones que se generan actualmente en el dominio, algunas restricciones temporales y de dependencia entre acciones que pueden ser llevadas a cabo por alumnos diferentes al que estuviésemos planificando en ese momento.

Se está estudiando la manera de ampliar las definiciones de las metas en el archivo del problema y hacer uso de las capacidades temporales que el planificador HTN, SIADEX, nos ofrece.

Se está trabajando en conjunto con un experto en planificación temporal basada en estados para utilizar la técnica de *mezcla de planes* (merge plan) para crear diseños instruccionales paralelos para diferentes estudiantes. De manera que dichos diseños puedan interconectarse automáticamente cuando exista la necesidad de ejecutar acciones colaborativas y se minimice el tiempo que un estudiante tiene que esperar a que otro termine una parte de su diseño para que puedan iniciar la actividad colaborativa.

2.3. Sistema de Identificación de Precondiciones Inconsistentes

El problema de las inconsistencias dados ciertas definiciones de estados iniciales y el conjunto de precondiciones del dominio de planificación no es nuevo. Dada nuestra experiencia tratando de evitar inconsistencias entre dominios y problemas de planificación generados automáticamente hemos decidido que hacerlo de forma automática puede ser, no solo un tema de investigación sumamente relevante para los expertos en AIP&S, si no también una herramienta importantísima para nuestra aplicación.

Pongamos un ejemplo. El método de cierta tarea compuesta con calidad de obligatoria

requiere de una acción múltiple (con el mismo nombre pero diferentes precondiciones). La precondición de una pide un nivel elevado de inglés, la precondición de la otra requiere de disponibilidad multimedia. El estudiante Tom, no tiene ni un nivel elevado de inglés, ni el equipo necesario para satisfacer los requerimientos multimedia. Bajo estas circunstancias es imposible generar un curso adaptado para el estudiante Tom. ¿Qué opciones tenemos? La primera opción puede ser indicarle al profesor que genere otro objeto de aprendizaje que satisfaga el objetivo requerido o que haga una búsqueda de dicho objeto sobre los repositorios existentes, de manera que éste tenga menos restricciones para poder llevarse a cabo por un alumno como Tom. La segunda opción es decirle al alumno que aprenda inglés o que consiga el software o hardware necesario para poder realizar la actividad.

La decisión de elegir a quien se le darán instrucciones para que el plan pueda ser “adaptado” se le puede dejar al administrador del sistema o se puede hacer un estudio para ver cual sería la opción mas conveniente, incluso se podría buscar la manera de que la labor del profesor de generación o búsqueda de nuevos objetos sea reemplazada por una herramienta de automatización. Pero, ¿cómo puede detectar el planificador dicho problema de inconsistencias?. Actualmente existen herramientas para detectarlas, mas no para definir las. He ahí un nuevo tema de investigación.

2.4. Dominios Diferentes de e-Learning (Otras Aplicaciones)

Para dar mayor relevancia a nuestro trabajo, se intentará probar hasta que punto es compatible la generación de un dominio e-learning de planificación jerárquica, con la generación de otro tipo de dominio que por naturaleza puedan codificarse de forma jerárquica. La idea es buscar la equivalencia entre las funciones de generación, para después generalizarlas.

2.5. Generalización las Reglas para los Estilos de Aprendizaje y las Precondiciones Derivadas del Perfil de los Alumnos

Como hemos visto a lo largo de esta memoria, se ha dado un primer acercamiento a la generación automática de dominios y problemas de planificación para cursos e-learning etiquetados bajo el estándar LOM e impartidos sobre un sistema gestor de aprendizaje. Sin embargo, dada la experiencia obtenida, nos hemos dado cuenta de que es posible generalizar

aún mas la generación de éstos dominios y problemas implementando clases que representen los diferentes “tipos” de precondiciones que es posible generar y las reglas para la generación de órdenes en los métodos de una tarea compuesta dados distintos estilos de aprendizaje.

2.6. Búsquedas sobre Repositorios Distribuidos de Objetos de Aprendizaje

Paralelamente a la línea inicial de este trabajo también sería posible resolver los problemas que existen respecto de la búsqueda de objetos de aprendizaje sobre repositorios distribuidos y su posterior revisión, para comprobar que se ha hecho un etiquetado correcto sobre los metadatos básicos de los objetos; dejando fuera los metadatos relacionales debido a que esos son definidos por el profesor para cada curso particular. Ésta línea de trabajo es sumamente relevante ya que el costo de diseño y etiquetado de todos los objetos para un curso particular suele ser realmente grande, para resolver este contratiempo existen una gran variedad de empresas e instituciones académicas que ofrecen el servicio de diseño de estos objetos para su posterior colocación en un repositorio libre o de paga de manera que puedan ser reutilizados en un futuro.

El problema de búsqueda sobre éstos repositorio ya es bien conocido y sus deficiencias bastante sorteables si se cuenta con una idea innovadora. El problema de comprobación de la validez de su etiquetado (y posterior corrección, si fuera posible) es un tema totalmente innovador debido, no solo al propósito para el cual se lleva a cabo dado el interés de esta memoria, si no también por que el tema de los estándares es realmente nuevo y nunca está de mas una herramienta tipo revisor que sirva de apoyo a los tutores y diseñadores de contenidos en esta especializada labor.

APÉNDICE A

Manual del Usuario para Integrar y Utilizar el Cliente de Planificación en Moodle

Documentación del Sistema de Adaptación de Objetos SCORM en Moodle

1. Primeros Pasos

1.1. La generación y etiquetado de un objeto SCORM

Se ha decidido utilizar el software Reusable eLearning Object Authoring and Delivery¹ (RELOAD) para generar un objeto Sharable Content Object Reference Model (SCORM)[SCORM, 2004]. Este software se puede descargar de su página oficial² y para aprender a utilizarlo solo es preciso consultar la documentación³ del mismo, sin embargo en este documento se presenta una guía rápida de los aspectos del software que interesan para

¹Available at <http://www.adlnet.gov/Technologies/scorm/default.aspx>

²<http://www.reload.ac.uk/new/editor.html>

³<http://www.reload.ac.uk/guide.html>

los propósitos de este documento.

Dado que SCORM es un estándar, su estructura e información dada por el estándar IMS Metadata[IMS-GLC, 2003] se guarda en un fichero de tipo manifest XML schema. Éste manifiesto define al tema principal del objeto SCORM como una *organization* y a los subtemas como *items* que no tienen asociada ninguna referencia a un recurso.

Por su parte, las actividades de aprendizaje son items que tienen asociada una referencia a un recurso que puede ser una página web ya existente referenciada por medio de su dirección web, o una página web creada para ese fin específico que se encuentre en un directorio local, además de documentos de power point, pdf, word, flash, video, audio, etc creados por el profesor, administrador y/o diseñador de contenidos del curso.

Se recomienda guardar todo el material creado especialmente para este curso en una subcarpeta llamada *resources*, *items*, *learning_activities* o *learning_objects* (se ha elegido *resources* como nombre para esta subcarpeta y ese será el nombre que se le dará durante el resto del documento) dentro de la carpeta principal del SCORM donde se guarde el manifiesto. Y las imágenes que utilicen las páginas web o los documentos en power point, entre otros, en una subcarpeta de la carpeta *resources* llamada *images* o *sources* (aquí se le llamará *images*).

Cada *item* y *organization* habrá de tener asociados obligatoriamente unos metadatos básicos que forman parte del estándar IMS Metadata. Éstos metadatos son los siguientes:

- **Identifier** y **name** del item que se pueden rellenar en la ventana principal RELOAD que corresponde al paquete SCORM. Una vez que se haya seleccionado el item a etiquetar⁴⁶ aparecerá un recuadro en la parte inferior central (El panel de atributos). El primer campo a rellenar es el **name** que hace referencia al nombre de la actividad y, luego, dentro de la tabla con columnas **attribute** y **value** tenemos el **Identifier** que deberá ser único y de preferencia fácil de reproducir, por lo que se sugiere cambiarlo y nombrarlo en base a los siguientes criterios:
 - Para los temas o subtemas, llamarle ITEM-task1.
 - Para las actividades con un recurso asociado, llamarle ITEM-action1.

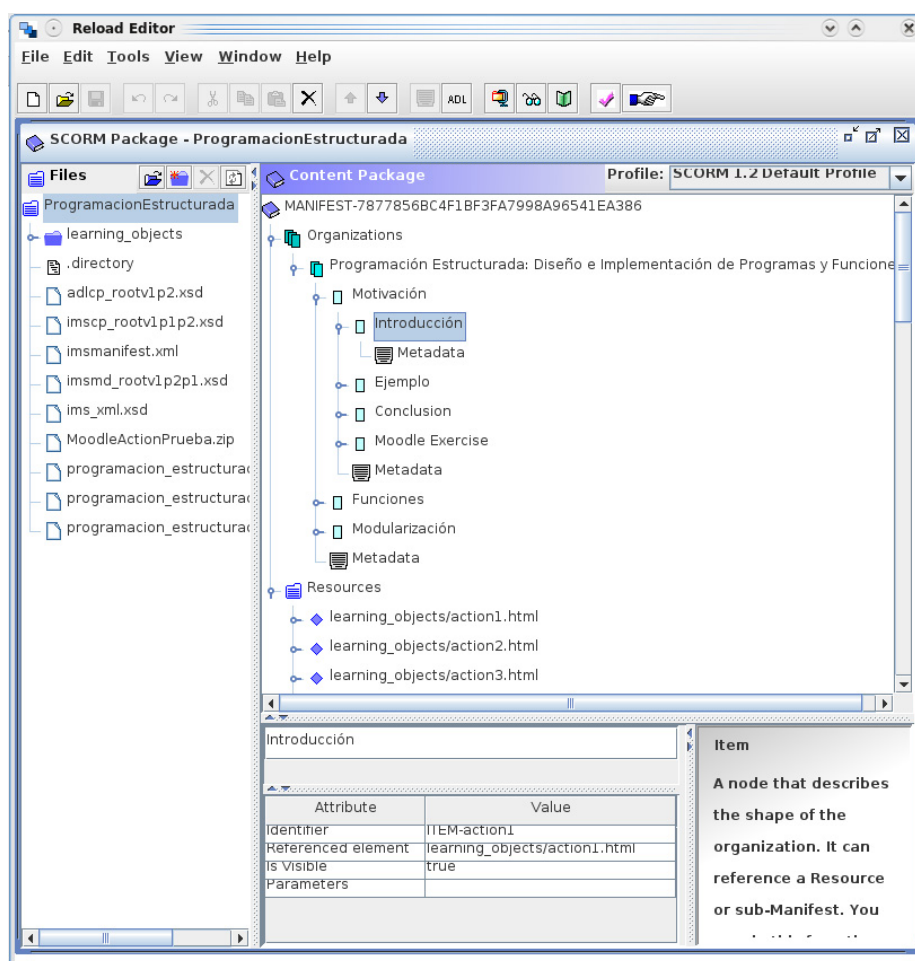


Figura 46. Ventana principal de RELOAD

Siendo el número diferente (de preferencia consecutivo) para cada uno de los dos grupos de nomenclaturas.

- Los items que tengan un recurso asociado, es decir las actividades, deberán tener adicionalmente un Recurso asociado al atributo **Referenced element**.
- El metadato **General** → **Coverage** deberá tener asociados los valores *Mandatory* u *Optional*, según sea el caso.
- El metadato **General** → **Language** tendrá los valores *en*(English) o *es*(Español).
Nota: en un futuro podrán especificarse mas idiomas.
- Para el metadato **Technical** → **Other Platform Requirements** es necesario espe-

cificar si la actividad requiere de aplicaciones específicas de software o hardware que no son tan comunes al ponerle el valor *Multimedia*, o si no las tiene, al ponerle el valor *NoOne*.

- **Educational** → **Learning Resource Type** sirve para indicar el tipo de recurso de aprendizaje. Por el momento se aceptan para una mejor adaptación: *Exercise* (para ejercicios que deben llevarse a cabo dentro de la misma actividad), *Narrative Text* (para lecturas variadas), *Example* (para lecturas que contengan ejemplos del tema a tratar), *Problem Statement* (para ejercicios que deben realizarse de manera independiente al recurso, por ejemplo, prácticas de laboratorio, investigaciones, participaciones en foros, exposiciones) y *Simulation* (para videos o transparencias que nos indiquen como interpretar o hacer algo).
- **Educational** → **Difficulty** que indica la dificultad relativa de la actividad, de manera que si el estudiante tiene un performance alto y la dificultad de la actividad es alta ésta actividad pueda mostrarse al estudiante, pero si el estudiante tiene un performance bajo no será incluida dentro de la secuencia de actividades de su SCORM personalizado. Los valores que podrá tomar este metadato son: *very easy*, *easy*, *medium*, *difficult* y *very difficult*.
- **Educational** → **Typical Learning Time** → **Date/Time** que determina el número de minutos promedio que un alumno necesita para llevar a cabo la actividad. Una vez etiquetadas todas las actividades con este valor, será necesario sumar los valores de todas ellas para asignar un typical learning time global con formato *[máximo de minutos,mínimo de minutos]* a la organization a la que pertenezcan.
- Las organizations no suelen tener relaciones asociadas, pero el resto de los items deben contener por lo menos la relación IsPartOf y, de ser necesario, la relación IsBasedOn.
 - *IsPartOf* describe una relación jerárquica entre dos items, por ejemplo, el item de subtema suma podría contener a los items de actividad definición, ejemplos, ejercicios y minitest por lo que cada uno de ellos tendría una relación IsPartOf con suma.

- La relación *IsBasedOn* describe una relación de orden entre dos items, por ejemplo, para hacer el mini-test del subtema suma primero debemos haber leído la definición, visto los ejemplos y hecho los ejercicios y por lo tanto la actividad mini-test tendrá una relación *IsBasedOn* que apunte al identificador de cada uno de ellos.

Para definir el tipo de relación rellenaremos el metadato **Relation** \rightarrow **Kind** \rightarrow **value**.

Para indicar a que actividad o tema apunta esa relación rellenaremos el metadato **Relation** \rightarrow **Resource** \rightarrow **Identifier** con el identificador de la actividad a la que apunta.

Tendremos que agregar un metadato value e identifier por cada actividad o tema al que apunta la actividad o tema que estemos etiquetando en ese momento. Por ejemplo, para el *IsBasedOn* que definíamos anteriormente, la actividad mini-test tendrá tres metadatos de tipo relation con sus respectivos valores asociados.

PARA EVITAR ERRORES

Al etiquetar las relaciones debemos tener especial cuidado de no generar ciclos entre las actividades poniendo relaciones de orden que impliquen que una actividad sea la base de otra que a su vez la tiene como base o de jerarquía que contengan actividades o subtemas que estén en un nivel mas alto, por ejemplo, que el subtema aritmética sea parte del subtema suma que a su vez es parte de aritmética.

PARA DAR ROBUSTEZ AL TEMA

Es importante mencionar que podemos generar actividades con el mismo nombre y utilidad dentro de un mismo subtema pero que difieran en su duración, idioma o requerimientos de hardware. Al hacer esto daríamos mas riqueza adaptativa a nuestro curso ya que nuestro planificador podría decidir que a un estudiante con poca disponibilidad horaria se le mostrará la actividad con menos duración o a un estudiante que no tiene buen nivel de inglés se le mostrará la actividad en español.

Finalmente, una vez que hemos terminado de formar toda la estructura de temas, subtemas y actividades, además de etiquetar cada una de ellas, tendremos que utilizar la opción

The screenshot shows a window titled "Metadata - ProgramacionEstructurada" with an "Edit" menu. The "Profile" is set to "IMS LRM Profile". The "Full Form View" tab is active. The form contains the following fields:

- General**
 - Language: es
 - Coverage: mandatory
- Technical**
- Educational**
 - Learning resource type**
 - Source: LOMv1.0
 - Value: Narrative Text
 - Difficulty**
 - Source: LOMv1.0
 - Value: easy
 - Typical learning time**
 - Date/Time: 5
 - Relation**
 - Kind**
 - Source: LOMv1.0
 - Value: IsPartOf
 - Resource**
 - Identifier: ITEM-task1

At the bottom, there are buttons for "Import...", "Export...", "OK", and "Cancel".

Figura 47. Ventana de Metadatos de RELOAD.

de RELOAD de *zip Content Package* para que pueda ser utilizado en la siguiente sección.

1.2. La Instalación y Configuración del add-on dentro del Módulo SCORM de Moodle

¿CÓMO COLOCO EL OBJETO SCORM EN EL CURSO DE MOODLE DEL QUE FORMA PARTE?

Una vez generado el objeto SCORM y una vez activado el modo de edición del curso de moodle con permisos de administrador, el zip del mismo deberá ser agregado al curso en Moodle[Dougiamas,] por medio de la opción *Add an Activity...* en donde habrá que

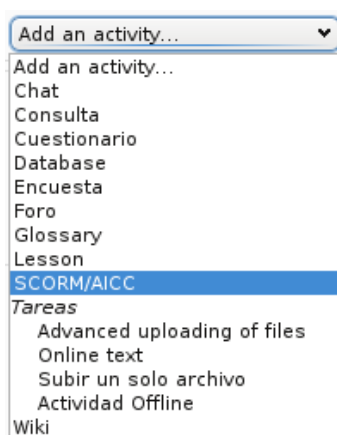


Figura 48. ¿Cómo agregar una actividad SCORM/AICC en Moodle?.

seleccionar la actividad *SCORM/AICC*., tal y como se muestra en la figura 48.

Para crear dicha actividad tendrá que rellenar atributos obligatorios *Name* y *Summary* (figura 49) para después cargar por medio de la opción *Choose or Upload a file . . .* el *Package file* que contenga el SCORM deseado.

Finalmente se presionará el botón inferior llamado *Save and Display* que deberá mostrarnos el objeto, sus jerarquías y contenidos correctamente, de manera que comprobemos que se ha cargado bien.

EL PERFIL DEL USUARIO DEBE TENER *Campos Especiales*: ¿CÓMO LOS AGREGO A MI PLATAFORMA DE MOODLE?

El segundo paso después de subir el SCORM a Moodle será, pedir al administrador del sistema que agregue los campos del perfil del usuario que sean necesarios para la adaptación.

Esto se hace en Site Administration → Users → Accounts → User Profile Fields en donde habrá que crear una nueva *profile category* y agregar varios fields del tipo *menu of choices* con los atributos name, short name, options y default value siguientes:

- short name: EnglishLevel name: **English Level** options: *Excelent, Acceptable, Low, Poor* y default value: Acceptable
- short name: Performance, name: **Performance Level**, options: *High, Medium, Low*

Name* Programación Estructurada

Summary* ?

Trebuchet 1 (8 pt) Lang B I U # x x

Tema de Programación Estructurada

Path:

Package file* ? programacion_estructurada021109.zip Choose or upload a file ...

Save and return to course Save and display Cancel

Figura 49. Atributos de la Actividad SCORM/AICC.

Site Administration

- Notifications
- Users
 - Authentication
 - Accounts
 - Browse list of users
 - Bulk user actions
 - Add a new user
 - Upload users
 - Upload user pictures
 - User profile fields
 - Permissions
- Courses
- Grades
- Location
- Language

User profile fields

Adaptation Fields x ↓

Profile Field	Edit
English Level	x ↓
Performance Level	x ↑ ↓
Equipment	x ↑ ↓
Learning Style	x ↑ ↓
Availability	x ↑

Figura 50. Nuevos campos del perfil del usuario para la adaptación de los SCORM's.

y default vlaue: Low.

- short name: Equipment, name: **Equipment**, options: *Multimedia* or *NoOne* y default value: Multimedia. (aquí, en un futuro, cada administrador deberá de especificar que software o hardware son necesarios en los diferentes cursos de la plataforma y mencionarlos al usuario en la descripción del campo para que le sea posible elegir).
- short name: HALS (Honey-Alonso Learning Style), name: **Learning Style**, options: *Theoretical*, *Reflexive*, *Pragmatic*, *Active* y default value: Active.
- Y finalmente short name: disponibilidad, name: **Availability**, options: *Much*, *Little* y

default value: Much.

¿CÓMO AGREGO EL ADD-ON DE PLANIFICACIÓN Y ADAPTACIÓN DE CURSOS A LA INSTALACIÓN DE MOODLE?

El tercer paso será descomprimir el zip que contenga el add-on dentro de la carpeta moodle/mod/scorm/ quedando de esta manera una nueva carpeta, dentro de ésta última llamada planner.

Lo siguiente que hay que hacer es ajustar el archivo de configuración. Especialmente la definición de la GLOBAL 'fields_group', en la parte derecha del arreglo se especificarán los short name de los campos de adaptación. Si se han seguido los pasos descritos hasta el momento no debería ser necesario cambiar esta sección, pero si hemos decidido poner otro short name diferente del sugerido, si que será necesario.

Por otro lado, si nuestra categoría de campos de adaptación no ha sido la primera categoría generada en los campos del perfil del usuario entonces tendremos que buscar el identificador de dicha categoría en la base de datos por medio de la consulta: `select * from pre_user_info_category` y cambiarla en la constante `PLANNER_PROFILE_ADAPTATION`.

Utilities to Create Adapted Scorm Packages

Options

- [Planner](#). Utility to Create Plans
- [To Check Server Connection](#). Checks communication with planners server **sepia.ugr.es**
- [LDAEMUL utilities](#).

Authors:

Lluvia Carolina Morales Reynaga
José Figueroa Martínez

Figura 51. Utilerías para la instalación y uso del módulo de adaptación de un paquete SCORM.

Otra cosa muy importante que habrá que cambiar dentro de este archivo de configuración será la clave para acceder al servicio web cuya llamada deberá proceder del rango de ip's asignado para la misma pues de otro modo el servicio no podrá ser utilizado. (Éste servicio se encuentra en proceso de desarrollo)

Finalmente, dentro de la página <http://moodleserver/moodle/mod/scorm/planner/>, don-

de *moodleserver* es la dirección web donde está alojado su servidor moodle, encontraremos dos opciones de configuración. La primera es la llamada *Check Planning Server Communication* en donde comprobaremos que nuestro sistema se conecta correctamente con el servidor de planificación que no regresa los scorms adaptados. La segunda es la de *LDAEMUL utilities*, en la que deberemos seleccionar la opción *Update Database*, la cual agrega un campo extra llamado `userid` a la tabla `pre_scrom_scoes` de la base de datos de moodle, este campo nos servirá para almacenar los planes adaptados generados hasta el momento.

Si tuviésemos un error con los campos del perfil del usuario de nuestro fichero de configuración, la opción *profile user fields* nos permitirá revisarlo vía web para luego modificarlo en el fichero de configuración si fuese necesario. Y la opción *Pretty Print Fields* se utiliza para visualizar de una manera orientativa todos los planes generados para cada alumno de cada curso y scorm dentro de él, para los cuales hayamos generado planes previamente.

1.3. Uso del add-on de Adaptación

Para poder utilizar el add-on de adaptación de los recorridos de aprendizaje de un scorm entraremos a la web <http://moodleserver/moodle/mod/scorm/planner/> y seleccionaremos la opción Planner.

LAS VARIABLES COURSE, SCORM, QUIZ Y GOAL.

Al acceder a esta opción nos permitirá elegir el curso donde se encuentra el scorm que queremos modificar (solo si hemos accedido previamente y no hecho logout a moodle como profesores o administradores de dicho curso).

Después de eso podremos elegir por su nombre uno de los scorms que contiene dicho curso.

Y el quiz que se haya realizado y evaluado previo al tema de ese scorm y que nos ayudará para mejorar la adaptación dadas las siguientes reglas:

- Si el estudiante tiene una nota entre 80 y 100, el nivel de performance de su perfil actual se incrementará al máximo.

Utility to Generate Plans

Course: Metodología de la Programación I ▾

Scorm: Vectores y MAarices de Prueba ▾

Quiz: QTI Programación Estructurada ▾

Goals:

Vectores y Matrices

Planes creados para:

[luis](#)
[lluvia](#)
[alumno](#)
[luisete](#)
[75166797](#)
[esteby](#)
[elmo](#)
[henry](#)
[juanmmm](#)
[juliorm](#)
[oskyar](#)
[royinf](#)
[linlin](#)
[josefb](#)
[jvillena](#)
[chemaraisin](#)
[guadix](#)
[wiber](#)
[angeljcc](#)
[juangv](#)
[irene90](#)
[estibaliz](#)
[pesadilla91](#)
[joseog](#)
[jaroalbier](#)
[mhm2009](#)
[pmmr](#)
[javia](#)
[marcofg](#)

Figura 52. Utilería para la generación de planes.

- Si el estudiante tiene una nota entre 50 y 80 (para el caso de los residentes en España y entre 60 y 80 para los residentes en México) su nivel de performance se modificará a un nivel medio.
- Y si es menos de 50 o 60, su nivel de performance se modificará temporalmente como bajo y pero se incrementará su tiempo disponible al máximo para que refuerce sus conocimientos con actividades optativas.

Recordemos que la elección de un quiz no es obligatoria, pero si quieren utilizar esta

opción y no saben como evaluar los quizez que generen, sobre todo las preguntas que no tengan una evaluación automática, en el anexo A ponemos a su disposición un mini-tutorial para hacerlo.

Finalmente deberá seleccionarse la Goal del tema, que inicialmente, al tener solo una organización será solo una, pero si decidiéramos crear un scorm con varias organizations podrían ser dos o mas.

LA LLAMADA AL SERVICIO WEB DE PLANIFICACIÓN

Una vez seleccionados el curso, el scorm, el quiz previo y la eta podremos pulsar el botón *Get Plan*. Este botón hará una llamada al servidor de adaptación (que pude tardar varios minutos dependiendo del número de alumnos matriculados en el curso y/o de la complejidad del scorm) que, si todo ha salido bien, nos regresará las secuencias de scorms adaptadas para cada uno de los estudiantes del curso y nos permitirá visualizar las mismas de manera orientativa. Pero, por el contrario, si ha existido un error ya sea con la comunicación con el servidor o con la generación de las secuencias de actividades adaptadas nos lo hará saber.

Los errores pueden ser de dos tipos:

1. Que en los metadatos de relaciones se referencie a identificadores de actividades o temas no existentes o que haya un problema en la generación de las secuencias, ya sea por que un alumno no cumple con los requisitos mínimos necesarios para hacer una actividad obligatoria (esto pasa cuando, por ejemplo, ponemos dentro de un tema una actividad con nivel de inglés elevado, pero no damos la opción de hacer la misma actividad en español), o
2. Por que dentro de nuestras relaciones de orden o jerárquicas hemos definido un ciclo infinito por error, por ejemplo, cuando además de poner las relaciones *IsBasedOn* del ejemplo de etiquetado agregamos una mas que diga que la definición de suma está basada en el mini-test, por lo tanto, como se requieren entre ellas, no sería posible hacer ninguna y, dado que son obligatorias, no se podría generar una secuencia válida de acuerdo a las condiciones impuestas en los metadatos del scorm.

2. Preguntas Frecuentes

En resumen, ¿Que variables de adaptación voy a utilizar? English Level, Performance Level, Equipment, Learning Style y Availability

¿Cómo puedo reciclar mis materiales de un curso en Moodle a un objeto SCORM? Asociando su dirección web a un resource del scorm y luego llamando a dicho resource por medio del referenced element del item.

¿Cómo puedo saber el id de los campos del perfil del usuario que han sido creados para la adaptación? Consulta en la base de datos que identificador corresponde a la categoría de los campos de adaptación por medio de la consulta: `select * from pre_user_info_category` Según ese el id de la categoría que corresponda a los campos de adaptación hacer la siguiente consulta: `select id, shortname, name from pre_user_info_field where categoryid = "id de la categoría de campos de adaptación"`.

¿Puedo agregar nuevos criterios y/o reglas de adaptación al sistema? Estamos trabajando en eso, en 6 meses aproximadamente estará listo. Solo que este servicio de adaptación solo se activará si se compra una licencia de uso del servicio web de planificación.

3. Notas para el Desarrollo

Tener en cuenta que hasta ahora hemos pensado que la categoría de los campos de adaptación tiene el identificador 1, pero en el archivo de configuración esto no está especificado.

Por otro lado, si nuestra categoría de campos de adaptación no ha sido la primera categoría generada en los campos del perfil del usuario entonces tendremos que buscar el identificador de dicha categoría en la base de datos por medio de la consulta: `select * from pre_user_info_category` y cambiarla en la constante `PLANNER_PROFILE_ADAPTATION`.

Otra cosa muy importante que habrá que cambiar dentro de este archivo de configuración será la clave para acceder al servicio web cuya llamada deberá proceder del rango de ip's asignado para la misma pues de otro modo el servicio no podrá ser utilizado. (Éste servicio se encuentra en proceso de desarrollo).

APÉNDICE B

Aplicación y Evaluación del cuestionario CHAEA

CUESTIONARIO HONEY-ALONSO DE ESTILOS DE APRENDIZAJE
CHAEA C.M.ALONSO,D.J.GALLEGO Y P.HONEY

Instrucciones: Marque con un signo (+) las sentencias de la siguiente lista que sean afirmativas para usted:

1. Tengo fama de decir lo que pienso claramente y sin rodeos.
2. Estoy segur@ de lo que es bueno y lo que es malo, lo que esta bien y lo que esta mal.
3. Muchas veces actúo sin mirar las consecuencias.
4. Normalmente trato de resolver los problemas metódicamente y paso a paso.
5. Creo que los formalismos coartan y limitan la actuación libre de las personas.
6. Me interesa saber cuales son los sistemas de valores de los demás y con que criterios actúan.

7. Pienso que el actuar intuitivamente puede ser siempre tan válido como actuar reflexivamente.
8. Creo que lo más importante es que las cosas funcionen.
9. Procuro estar al tanto de lo que ocurre aquí y ahora.
10. Disfruto cuando tengo tiempo para preparar mi trabajo y realizarlo a conciencia.
11. Estoy a gusto siguiendo un orden, en las comidas, en el estudio, haciendo ejercicio regularmente.
12. Cuando escucho una nueva idea enseguida comienzo a pensar como ponerla en práctica.
13. Prefiero las ideas originales y novedosas aunque no sean prácticas.
14. Admito y me ajusto a las normas solo si me sirven para lograr mis objetivos.
15. Normalmente encajo bien con personas reflexivas, y me cuesta sintonizar con personas demasiado espontáneas, imprevisibles.
16. Escucho con más frecuencia que hablo.
17. Prefiero las cosas estructuradas a las desordenadas.
18. Cuando poseo cualquier información, trato de interpretarla bien antes de manifestar alguna conclusión.
19. Antes de hacer algo estudio con cuidado sus ventajas e inconvenientes.
20. Crezco con el reto de hacer algo nuevo y diferente.
21. Casi siempre procuro ser coherente con mis criterios y sistemas de valores. Tengo principios y los sigo.
22. Cuando hay una discusión no me gusta ir con rodeos.
23. Me disgusta implicarme afectivamente en mi ambiente de trabajo. Prefiero mantener relaciones distantes.
24. Me gustan más las personas realistas y concretas que las teóricas.
25. Me gusta ser creativo, romper estructuras.
26. Me siento a gusto con personas espontáneas y divertidas.
27. La mayoría de las veces expreso abiertamente cómo me siento.
28. Me gusta analizar y dar vueltas a las cosas.

29. Me molesta que la gente no se tome en serio las cosas.
30. Me atrae experimentar y practicar las últimas técnicas y novedades.
31. Soy cauteloso a la hora de sacar conclusiones.
32. Prefiero contar con el mayor número de fuentes de información. Cuantos más datos reúna para reflexionar, mejor.
33. Tiendo a ser perfeccionista.
34. Prefiero oír las opiniones de los demás antes de exponer la mía.
35. Me gusta afrontar la vida espontáneamente y no tener que planificar todo previamente.
36. En las discusiones me gusta observar cómo actúan los demás participantes.
37. Me siento incómodo con las personas calladas y demasiado analíticas.
38. Juzgo con frecuencia las ideas de los demás por su valor práctico.
39. Me agobia si me obligan a acelerar mucho el trabajo para cumplir un plazo.
40. En las reuniones apoyo las ideas prácticas y realistas.
41. Es mejor gozar del momento presente que deleitarse pensando en el pasado o en el futuro.
42. Me molestan las personas que siempre desean apresurar las cosas.
43. Aporto ideas nuevas y espontáneas en los grupos de discusión.
44. Pienso que son más conscientes las decisiones fundamentadas en un minucioso análisis que las basadas en la intuición.
45. Detecto frecuentemente la inconsistencia y puntos débiles en las argumentaciones de los demás.
46. Creo que es preciso saltarse las normas muchas más veces que cumplirlas.
47. A menudo caigo en cuenta de otras formas mejores y más prácticas de hacer las cosas.
48. En conjunto hablo más que escucho.
49. Prefiero distanciarme de los hechos y observarlos desde otras perspectivas.
50. Estoy convencido que deber imponerse la lógica y el razonamiento.
51. Me gusta buscar nuevas experiencias.
52. Me gusta experimentar y aplicar las cosas.
53. Pienso que debemos llegar pronto al grano, al meollo de los temas.

54. Siempre trato de conseguir conclusiones e ideas claras.
55. Prefiero discutir cuestiones concretas y no perder el tiempo con charlas vacías.
56. Me impaciento cuando me dan explicaciones irrelevantes e incoherentes.
57. Compruebo antes si las cosas funcionan realmente.
58. Hago varios borradores antes de la redacción definitiva de un trabajo.
59. Soy consciente de que en las discusiones ayudo a mantener a los demás centrados en el tema, evitando divagaciones.
60. Observo que, con frecuencia, soy un@ de l@s más objetiv@s y desapasionados en las discusiones.
61. Cuando algo va mal le quito importancia y trato de hacerlo mejor.
62. Rechazo ideas originales y espontáneas si no las veo prácticas.
63. Me gusta sopesar diversas alternativas antes de tomar una decisión.
64. Con frecuencia miro hacia delante para prever el futuro.
65. En los debates y discusiones prefiero desempeñar un papel secundario antes que ser el/la líder o el/la que más participa.
66. Me molestan las personas que no actúan con lógica.
67. Me resulta incomodo tener que planificar y prever las cosas.
68. Creo que el fin justifica los medios en muchos casos.
69. Suelo reflexionar sobre los asuntos y problemas.
70. El trabajar a conciencia me llena de satisfacción y orgullo.
71. Ante los acontecimientos trato de descubrir los principios y teorías en que se basan.
72. Con tal de conseguir el objetivo que pretendo soy capaz de herir sentimientos ajenos.
73. No me importa hacer todo lo necesario para que sea efectivo mi trabajo.
74. Con frecuencia soy una de las personas que más anima las fiestas.
75. Me aburro enseguida con el trabajo metódico y minucioso.
76. La gente con frecuencia cree que soy poco sensible a sus sentimientos.
77. Suelo dejarme llevar por mis intuiciones.

78. Si trabajo en grupo procuro que se siga un método y un orden.

79. Con frecuencia me interesa averiguar lo que piensa la gente.

80. Esquivo los temas subjetivos, ambiguos y poco claros.

Para calcular su PERFIL DE APRENDIZAJE:

1. En la tabla 17 rodee con una línea cada uno de los números que ha señalado con un signo más (+).
2. Sume el número de círculos que hay en cada columna.
3. Coloque estos totales en la gráfica. Así comprobará cual es su estilo o estilos de aprendizaje preferentes.

	I	II	III	IV
	3	10	2	1
	5	16	4	8
	7	18	6	12
	9	19	11	14
	13	28	15	22
	20	31	17	24
	26	32	21	30
	27	34	23	38
	35	36	25	40
	37	39	29	47
	41	42	33	52
	43	44	45	53
	46	49	50	56
	48	55	54	57
	51	58	60	59
	61	63	64	62
	67	65	66	68
	74	69	71	72
	75	70	78	73
	77	79	80	76
Total				
	Activo	Reflexivo	Teórico	Pragmático

Tabla 17. Tabla para la evaluación del CHAEA y definición del estilo de aprendizaje del estudiante.

CARACTERÍSTICAS DEL ESTILO ACTIVO

Las personas que obtengan un predominio claro del Estilo Activo poseerán algunas de estas características o manifestaciones:

Características principales:

1. Animador
2. Improvisador
3. Descubridor
4. Arriesgado
5. Espontáneo

Otras características:

- Creativo
- Novedoso
- Aventurero
- Renovador
- Inventor
- Vital
- Vividor de la Experiencia
- Generador de ideas
- Lanzado
- Protagonista

- Chocante
- Innovador
- Conversador
- Líder
- Voluntarioso
- Divertido
- Participativo
- Competitivo
- Deseoso de aprender
- Solucionador de problemas
- Cambiante

CARACTERÍSTICAS DEL ESTILO REFLEXIVO

Las personas que obtengan un predominio claro del Estilo Reflexivo poseerán muchas de estas características o manifestaciones:

Características principales:

1. Ponderado
2. Concienzudo
3. Receptivo
4. Analítico
5. Exhaustivo

Otras características:

- Observador
- Recopilador
- Paciente
- Cuidadoso
- Detallista
- Elaborador de argumentos
- Previsor de argumentos
- Estudioso de comportamientos
- Registrador de datos
- Investigador
- Asimilador
- Escritor de informes y/o declaraciones
- Lento
- Prudente
- Distante
- Sondeador

CARACTERÍSTICAS DEL ESTILO TEÓRICO

Las personas que obtengan una mayor puntuación en el Estilo Teórico tendrán características o manifestaciones como estas:

Características principales:

1. Metódico
2. Lógico
3. Objetivo
4. Crítico
5. Estructurado

Otras características:

- Disciplinado
- Planificado
- Sistemático
- Ordenado
- Sintético
- Razonador
- Pensador
- Relacionador
- Perfeccionista
- Generalizador
- Buscador de hipótesis
- Buscador de teorías
- Buscador de modelos
- Buscador de preguntas
- Buscador de supuestos subyacentes

- Buscador de conceptos
- Buscador de racionalidad
- Buscador de "por qué "
- Buscador de sistema de valores, criterios. . .
- Inventor de procedimientos para. . .
- Explorador

CARACTERÍSTICAS DEL ESTILO PRAGMÁTICO

Las personas que obtengan una mayor puntuación en el Estilo Pragmático tendrán características o manifestaciones como éstas:

Características principales:

1. Experimentador
2. Práctico
3. Directo
4. Eficaz
5. Realista

Otras características:

- Técnico
- Útil
- Rápido
- Decidido

- Planificador
- Positivo
- Concreto
- Claro
- Seguro de sí
- Organizador
- Actual
- Solucionador de problemas
- Aplicador de lo aprendido
- Planificador de acciones

Como es posible observar, dadas las características de los distintos estilos de aprendizaje y los valores de perpendicularidad expresados en la figura 53, los estilos Teórico y Reflexivo son muy similares, y lo mismo pasa con los estilos Pragmático y Activo. Por lo cual, si se quisiera simplificar su definición para llevar a cabo cualquier proceso de automatización que implicara un gran nivel de procesamiento sobre los mismos, se podría hacer una equivalencia entre ellos y reducir los estilos a dos principales: Pragmático y Teórico.

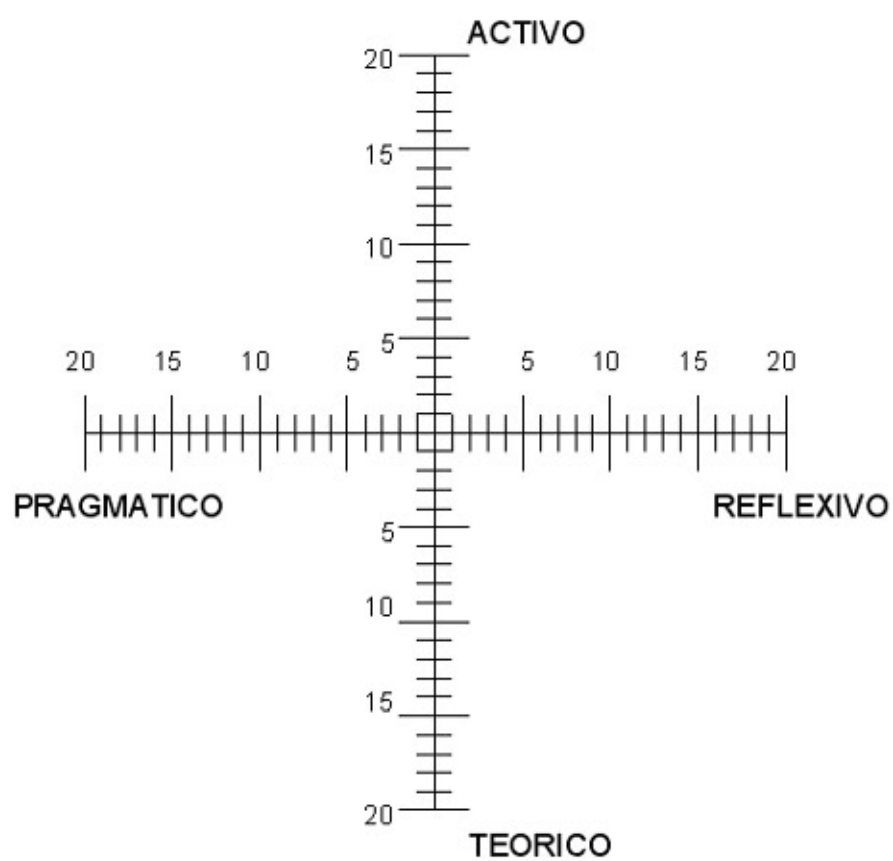


Figura 53. Perpendicularidad de los estilos de aprendizaje Honey-Alonso.

BIBLIOGRAFÍA

- [Alguacil-Martin, 2008] Alguacil-Martin, S. (2008). Tailores Learning Experiences. <http://www.andaluciainvestiga.com/espanol/noticias/9/6915.asp>.
- [Allen et al., 1999] Allen, J. E., Guinn, C. I., and Horvitz, E. (1999). Mixed-initiative interaction. *IEEE Intelligent Systems and Their Applications*, 14(5):14–23.
- [Ambite and Knoblock, 1997] Ambite, J. L. and Knoblock, C. A. (1997). Planning by re-writing: Efficiently generating high-quality plans. In *In Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 706–713.
- [ANSI/IEEE, 2007] ANSI/IEEE (2007). IEEE Standard for Learning Object Metadata. In <http://ltsc.ieee.org/wg12/>.
- [Avgeriou et al., 2003] Avgeriou, P., Papasalouros, A., Retalis, S., and Skordalakis, M. (2003). Towards a Pattern Language for Learning Management Systems. *Educational Technology & Society*, 6:11–24.
- [Bacchus and Kabanza, 2000] Bacchus, F. and Kabanza, F. (2000). Using temporal logics to express search control knowledge for planning. *Artif. Intell.*, 116:123–191.
- [Baldiris et al., 2008] Baldiris, S., Santos, O., Barrera, C., J.G., J. B., Velez, J., and Fabregat, R. (2008). Integration of Educational Specifications and Standards to Support

- Adaptive Learning Scenarios in ADAPTAPlan. *Special Issue on New Trends on AI techniques for Educational Technologies. International Journal of Computer Science and Applications (IJCSA)*, 5(1):88–107.
- [Bienkowski, 1995] Bienkowski, M. (1995). Demonstrating the Operational Feasibility of new technologies: The arpi ifds. *IEEE Intelligent Systems*, 10:27–33.
- [Bitzer et al., 1961] Bitzer, D., Braunfeld, P., and Lichtenberger, W. (1961). PLATO: An Automatic Teaching Device. *IRE Transactions on Education*, 4:157–161.
- [Bonet and Geffner, 1999] Bonet, B. and Geffner, H. (1999). Planning as heuristic search: New results. In *European Conference on Planning*, pages 360–372.
- [Brusilovsky and Vassileva, 2003] Brusilovsky, P. and Vassileva, J. (2003). Course Sequencing Techniques for Large-Scale Web-Based Education. *International Journal Continuing Engineering Education and Lifelong Learning*, 13(1/2):75–94.
- [Camacho et al., 2007] Camacho, D., R-Moreno, M., and U. Obieta, U. (2007). CAMOU: A Simple Integrated e-Learning and Planning Techniques Tool. In *4th International Workshop on Constraints and Language Processing*.
- [Caroline, 2010] Caroline, C. (2010). Caroline LMS. Available at: <http://www.caroline.net/>.
- [Castillo, 1998] Castillo, L. (1998). *Desarrollo y Aplicación de Técnicas de Planificación no Lineal para la Programación del Control de Plantas Industriales*. PhD thesis, University of Granada, Granada, España.
- [Castillo et al., 2005] Castillo, L., Fdez.-Olivares, J., García-Perez, O., and Palao, F. (2005). SIADEX. un Entorno Integral de Planificación para el Diseño de Planes de Actuación en Situaciones de Crisis. In *VI Jornadas de Transferencia Tecnológica de Inteligencia Artificial, (TTIA)*.
- [Castillo et al., 2006] Castillo, L., Fernández-Olivares, J., García-Perez, O., and Palao, F. (2006). Bringing Users and Planning Technology Together. experiences in SIADEX. In *16th International Conference on Automated Planning and Scheduling (ICAPS 2006)*, pages 11–20. AAAI Press.

- [Castillo et al., 2007] Castillo, L., Morales, L., Fernández-Olivares, J., González-Ferrer, A., and Palao, F. (2007). Knowledge Engineering and Planning for the Automated Synthesis of Customized Learning Designs. In *LNAI 4788*, pages 40–49.
- [Castillo et al., 2010] Castillo, L., Morales, L., González-Ferrer, A., Fdez-Olivares, J., Borraro, D., and Onaindía, E. (2010). Automatic Generation of Temporal Planning Domains for E-learning Problems. *Journal of Scheduling*, 13(4):347–362.
- [Cazau, 2001] Cazau, P. (2001). Estilos de Aprendizaje: Generalidades. Available at: [http://www.gestionescolar.cl/UserFiles/P0001/Image/gestion_portada/documentos/CD-48%20Doc.%20estilos%20de%20aprendizaje%20\(ficha%2055\).pdf](http://www.gestionescolar.cl/UserFiles/P0001/Image/gestion_portada/documentos/CD-48%20Doc.%20estilos%20de%20aprendizaje%20(ficha%2055).pdf).
- [Cobos et al., 2007] Cobos, C., Niño, M., Mendoza, M., Fabregat, R., and Gomez, L. (2007). Learning Management System based on SCORM, Agents and Mining. In *WISE'07: Proceedings of the 8th international conference on Web information systems engineering*, pages 298–309, Berlin, Heidelberg. Springer-Verlag.
- [desJardins et al., 1999] desJardins, M. E., Durfee, E. H., Ortiz, C. L., Jr., and Wolverton, M. J. (1999). A survey of research in distributed, continual planning. 20:13 – 22.
- [dotLRN Consortium, 2009] dotLRN Consortium, T. (2009). Dot LeARN. Available at <http://dotlrn.org>.
- [Dougiamas,] Dougiamas, M. Modular Object-Oriented Dynamic Learning Environment. Moodle. In <http://moodle.org/>.
- [eXe eXeLearning, 2010] eXe eXeLearning (2010). eXe. Available at: <http://exelearning.org>.
- [Ferguson et al., 1996] Ferguson, G., Allen, J. F., and Miller, B. W. (1996). Trains-95: Towards a mixed-initiative planning assistant. In *AIPS*, pages 70–77.
- [Figuroa-Martínez et al., 2008] Figuroa-Martínez, J., Morales, L., and Castillo, L. (2008). Extending Moodle SCORM module to Support a Basic IMS-LD. In *Moodle Moot, Spain*.
- [Fikes et al., 1998] Fikes, R. E., Nillson, N. J., and Cocosco, C. A. (1998). A review of "strips: A new approach to the application of theorem proving to problem solving by r.e. fikes, n.j. nillson, 1971".

- [Foulser et al., 1992] Foulser, D., Li, M., and Yang, Q. (1992). Theory and Algorithms for Plan Merging. *Artificial Intelligence*, 57:143–181.
- [Fox et al., 2006] Fox, M., Gerevini, A., Long, D., and Serina, I. (2006). Plan Stability: Replanning versus Plan Repair. In *ICAPS*, pages 212–221.
- [Garrido and Onaindía, 2010] Garrido, A. and Onaindía (2010). On the Application of Planning and Scheduling Techniques to E-learning. In *Proc. 23rd Int. Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE 2010) - LNCS*, pages 244–253. Springer.
- [Georgeff and Lansky, 1987] Georgeff, M. P. and Lansky, A. L. (1987). Reactive reasoning and planning. In *AAAI*, pages 677–682.
- [Gerevini and Serina, 2002] Gerevini, A. and Serina, I. (2002). LPG: A Planner Based on Local Search for Planning Graphs with Action Costs. In *International Conference on AI Planning and Scheduling (AIPS)*, pages 13–22.
- [Gravot and Alami, 2001] Gravot, F. and Alami, R. (2001). An Extension for the Plan-Merging Paradigm for multi-robot Coordiantion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2929– 2934.
- [Herve, 1994] Herve, H. L. M. (1994). Representation and Control in IxTeT, a Temporal Planner. In *Proceedings of the International Conference on AI Planning Systems*.
- [Hoffmann, 2003] Hoffmann, J. (2003). The Metric-FF Planning System: Translating "ignoring delete lists" to numerical state variables. *Journal of Artificial Intelligence Research. Special Issue on the 3rd International Planning Competition*, 20(1):291–341.
- [Hoffmann and Nebel, 2001] Hoffmann, J. and Nebel, B. (2001). The FF planning system: Fast plan generation through heuristic search.
- [IMS-GLC, 2003] IMS-GLC (2003). IMS Specifications. In <http://www.imsglobal.org/>, Open University of the Netherlands.
- [Inc., 2010a] Inc., B. (1997-2010a). Blackboard Learn. Available at: <http://www.blackboard.com>.

- [Inc., 2010b] Inc., D. (199-2010b). Desire2Learn. Available at: <http://www.desire2learn.com>.
- [Knoblock, 1995] Knoblock, C. A. (1995). Planning, Executing, Sensing, and Replanning for Information Gathering. In *IJCAI*, pages 1686–1693.
- [Kontopoulos et al., 2008] Kontopoulos, E., Vrakas, D., Kokkoras, F., Bassiliades, N., and Vlahavas, I. (2008). An Ontology-Based Planning System for e-course Generation. *Expert Systems with Applications*, 35(1/2):398–406.
- [Lewis, 1992] Lewis, M. W. (1992). *Emerging Uses of Computers for Education: An Overview of Tools and Issues for Vocational Educators*. National Center for Research in Vocational Education.
- [Liber, 2007] Liber, O. (2007). REusable eLearning Object Authoring and Delivery. In <http://www.reload.ac.uk/>, University of Bolton.
- [Long and Fox, 2003] Long, D. and Fox, M. (2003). Pddl 2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research*, 20(1):149–154.
- [Luckin et al., 2006] Luckin, R., Underwood, J., du Boulay, B., Holmberg, J., Kerawalla, L., O’Connor, J., Smith, H., and Tunley, H. (2006). Designing Educational Systems Fit for Use: A Case Study in the Application of Human Centred Design for AIED. *International Journal of Artificial Intelligence in Education*, 16(4):353–380.
- [Miksch and Seyfang, 2000] Miksch, S. and Seyfang, A. (2000). Continual planning with time-oriented, skeletal plans. In *ECAI*, pages 511–515.
- [Mohan et al., 2003] Mohan, P., Greer, J., and McCalla, G. (2003). Instructional Planning with Learning Objects. In *IJCAI-03 Workshop Knowledge Representation and Automated Reasoning for E-Learning Systems*, pages 52–58.
- [Morales et al., 2008a] Morales, L., Castillo, L., Fernández-Olivares, J., and González-Ferrer, A. (2008a). Automatic Generation of User Adapted Learning Designs: An AI-

- Planning Proposal. In *Lecture Notes in Computer Science, Adaptive Hypermedia 2008*, pages 324–328.
- [Morales et al., 2009] Morales, L., Castillo, L. A., and Fernández-Olivares, J. (2009). Planning for Conditional Learning Routes. In *MICAI '09: Proceedings of the 8th Mexican International Conference on Artificial Intelligence*, pages 384–396. Springer-Verlag.
- [Morales and García, 2010] Morales, L. and García, J. M. (2010). LMS Plan: An architecture to integrate Adapted Learning Sequences with LMS's. *Research in Computing Science: Special Issue on Advances in Artificial Intelligence*, 51:189–196.
- [Morales and García-Gorrosieta, 2010] Morales, L. and García-Gorrosieta, J. M. (2010). LMS Plan: An Architecture to Integrate Adapted Learning Sequences with LMS's. In *Workshop on Intelligent Learning Environments, MICAI*.
- [Morales et al., 2008b] Morales, R., Santos, O., and Boticario, J. (2008b). Adaptation support in design time through IMS-QTI and IMS-LD specifications in dotLRN web-based learning environment. Valencia (Spain).
- [Mussettola et al., 1998] Mussettola, N., Nayak, P. P., Pell, B., and Williams, B. C. (1998). Remote Agent: To Boldly go Where no AI System has gone Before. *Artif. Intell.*, 103(1-2):5–47.
- [Myers, 1996] Myers, K. L. (1996). Advisable Planning Systems. In *AAAI ARPI*, pages 206 – 209.
- [Myers, 1998] Myers, K. L. (1998). Towards a Framework for Continuous Planning and Execution. In *AAAI Fall Symposium on Distributed Continual Planning*.
- [Nau et al., 2003] Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., and Yaman, F. (2003). SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research*, 20(1):379–404.
- [Nau et al., 2004] Nau, D., Ghallab, M., and Traverso, P. (2004). *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

- [Nau et al., 1999] Nau, D. S., Cao, Y., Lotem, A., and Hector Mu, n.-A. (1999). SHOP: Simple Hierarchical Ordered Planner. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 968–975, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Nederland, 2009] Nederland, O. U. (2009). Copper Core: The IMS Learning Design Engine. Available at: <http://coppercore.sourceforge.net/>.
- [Nilsson, 1982] Nilsson, N. J. (1982). *Principles of Artificial Intelligence*. Springer.
- [of Cologne, 2000] of Cologne, U. (2000). ILIAS Learning Management System. Available at <http://www.ilias.de/ios/index-e.html>.
- [Onder and Pollack, 1999] Onder, N. and Pollack, M. E. (1999). Conditional, Probabilistic Planning: A Unifying Algorithm and Effective Search Control Mechanisms. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 577–584.
- [Peachy and McCalla, 1986] Peachy, D. and McCalla, G. (1986). Using Planning Techniques in Intelligent Tutoring Systems. *International Journal of Man-Machine Studies*, 24(1):77–98.
- [Pérez, 2007] Pérez, O. J. G. (2007). *Planificación en Dominios Temporales usando Técnicas HTN*. PhD thesis, University of Granada, Granada, España.
- [Rintanen, 1999] Rintanen, J. (1999). Constructing conditional plans by a theorem-prover. *Journal of Artificial Intelligence Research*, 10:323–352.
- [Rojas and Defude, 2008] Rojas, J. and Defude, B. (2008). Improving Learning Objects Quality with learning Styles. *Advanced Learning Technologies, IEEE International Conference on*, 0:496–497.
- [Rosenberg, 2001] Rosenberg, M. J. (2001). *E-learning: Estrategias para Transmitir Conocimiento en la Era Digital*. Mc Graw Hill.
- [Ruiz et al., 2008] Ruiz, M. d. P. P., Díaz, M. J. F., Soler, F. O., and Pérez, J. R. P. (2008). Adaptation in Current e-learning Systems. *Computer Standards & Interfaces*, 30(1-2):62–70.

- [Russell and Norvig, 2003] Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education.
- [Sanchez and Lama, 2007] Sanchez, E. and Lama, M. (2007). Introducción a la monografía Técnicas de IA aplicadas a la Educación. *Inteligencia Artificial, Revista Iberoamericana de IA*, 11(33):7–12.
- [Santos et al., 2007] Santos, O. C., Baldiris, S., Velez, J., Boticario, J. G., and Fabregat, R. (2007). *Dynamic Support in ADAPTAPlan: ADA+*, volume 2, pages 131–140. Universidad de Salamanca, Salamanca, Spain.
- [Schoppers, 1987] Schoppers, M. (1987). Universal plans for reactive robots in unpredictable environments. In *IJCAI*, pages 1039–1046.
- [Schwalb and Dechter, 1991] Schwalb, E. and Dechter, R. (1991). Temporal Constraint Networks. *Artificial Intelligence*, 49:61–95.
- [SCORM, 2004] SCORM (2004). Sharable Content Object Reference Model. Available at <http://www.adlnet.gov/Technologies/scorm/default.aspx>.
- [Sicilia et al., 2006] Sicilia, M., Sánchez-Alonso, S., and García-Barriocanal, E. (2006). On Supporting the Process of Learning Design through Planners. *CEUR Workshop Proceedings: Virtual Campus 2006 Post-Proceedings.*, 186(1):81–89.
- [S.L., 2010] S.L., I. I. S. (2010). IActive. Available at <http://www.iactive.es/>.
- [Smith et al., 1996] Smith, S., Lassila, O., and Becker, M. (1996). Configurable, mixed-initiative systems for planning and scheduling. In *Advanced Planning*, pages 235–241. AAAI Press.
- [Toro et al., 2008] Toro, S., Castillo, L., Morales, L., Jiménez-Galera, M., Llorca-Díez, M., and Carrillo, J. (2008). Customizing Learning routes in Moodle based on Honey-Alonso Learning Styles. In *Moodle Moot*.
- [Ullrich, 2005] Ullrich, C. (2005). Course Generation Based on HTN Planning. In *13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems*, pages 74–79. XI.

- [Ullrich and Melis, 2009] Ullrich, C. and Melis, E. (2009). Pedagogically Founded Courseware Generation Based on HTN-Planning. *Expert Systems with Applications*, 36(5):9319–9332.
- [Urretavizcaya, 2001] Urretavizcaya, M. (2001). Sistemas Inteligentes en el ambito de la Educación. *Revista Iberoamericana de Inteligencia Artificial*, 12:5–12.
- [Vassileva, 1997] Vassileva, J. (1997). Dynamic Course Generation. *Journal of Computing and Information Technology*, 5:87–102.
- [Vassileva and Wasson, 1996] Vassileva, J. and Wasson, B. (1996). Instructional Planning Approaches: from Tutoring Towards Free Learning. In *Proc. EuroAIED'96*, pages 1–8.
- [Veloso, 1996] Veloso, M. M. (1996). Towards mixed-initiative rationale-supported planning. In *Advanced Planning Technology, 277-282. Menlo Park, CA*, pages 277–282. AAAI Press.
- [Vrakas et al., 2007] Vrakas, D., Tsoumakas, G., Kokkoras, F., Bassiliades, N., Vlahavas, I., and Anagnostopoulos, D. (2007). PASER: a Curricula Synthesis System based on Automated Problem Solving. *Int. Journal on Teaching and Case Studies, Special Issue on Information Systems: the New Research Agenda, the Emerging Curriculum and the New Teaching Paradigm*, 1(1/2):159–170.
- [Weber and Brusilovsky, 2001] Weber, G. and Brusilovsky, P. (2001). ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education*, 12:351–384.
- [Wiley, 2000] Wiley, D. A. (2000). Connecting Learning Objects to Instructional Design Theory: A Definition, a Metaphor, and a Taxonomy.
- [Wilkins and Desimone, 1992] Wilkins, D. and Desimone, R. V. (1992). Applying an AI Planner to Military Operations Planning. In *Intelligent Scheduling*, pages 685–709. Morgan Kaufmann.
- [Yang et al., 1992] Yang, Q., Nau, D., and Hendler, J. (1992). Merging Separately Generated Plans with Restricted Interactions. *Computational Intelligence*, 8(2):648–676.