

UNIVERSIDAD NACIONAL DE ROSARIO
Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Licenciatura en Ciencias de la Computación

Tesina

**Un Sistema Inteligente para Asistir la Búsqueda
Personalizada de Objetos de Aprendizaje**

Valeria Bibiana Gerling

Directoras de Tesina:
Ana Casali Claudia Deco

Diciembre 2009

Resumen

En este trabajo se describe el desarrollo de un sistema inteligente que ayuda a un usuario a encontrar los recursos educativos electrónicos que le sean más apropiados, de acuerdo con su perfil. Como hipótesis de trabajo se considera que se tienen diferentes repositorios de objetos de aprendizaje, donde cada objeto tiene metadatos descriptivos. Se propone utilizar estos metadatos para recuperar aquellos objetos que satisfagan no sólo el tema de la consulta, sino también el perfil de usuario, teniendo en cuenta sus características y preferencias. El sistema de recomendación se diseña con una arquitectura multiagente para la recuperación de los recursos educativos en la Web. Esta plataforma incluye varios tipos de agentes con diferentes funcionalidades. En particular, en este trabajo se modela el Agente Recomendador (Agente-R), como un agente BDI graduado. Este agente se encarga de realizar una recuperación flexible y presentar una lista ordenada con los mejores recursos, de acuerdo con el perfil de usuario. Se especifica este modelo de agente mediante un sistema multi-contexto y se diseña el Agente-R siguiendo una metodología adecuada. Se implementa un prototipo de este agente utilizando el lenguaje SWI-Prolog y finalmente, se presenta un caso de estudio que permite obtener algunas conclusiones.

Palabras claves: sistemas multiagente, sistemas recomendadores, objetos de aprendizaje, metadatos, perfil de usuario.

Abstract

This paper describes the development of an intelligent system that assists a user to find electronic educational resources that are more qualified according to his/her needs and preferences. Different learning objects repositories, where each object has a descriptive metadata, compose our work scenario. Using this metadata, we propose to assist the retrieval of objects that satisfy not only the subject of the query, but also the user profile, considering his/her characteristics and preferences. We design a recommender system with a multiagent architecture for Web educational resources retrieval. This multiagent platform includes several kinds of agents with different functionalities. We particularly model the Recommender Agent (Agente-R) as a graded BDI agent, which is in charge of performing a flexible retrieval and ranking in an ordered list the best resources according to the user profile. The specification of this model of agent is by using multi-context systems and we design the Agente-R following a suitable methodology. We implement a prototype for this agent using SWI-prolog, and finally through a case of study we obtain some conclusions.

Keywords: multiagent systems, recommender systems, learning objects, metadata, user profile.

Agradecimientos

Quiero agradecer el apoyo de muchas personas. A mis padres, gracias a ellos pude cursar y estudiar esta carrera. Gracias por su apoyo e incentivo constante y por la contención brindada, como el té de tilo que mi mamá me preparaba con mucho amor, el día antes de rendir. A mis tres hermanos, por su apoyo y paciencia, por la cantidad de horas que me apoderé de la computadora.

Me gustaría expresar mi total gratitud a mis directoras Ana y Claudia, ellas me dieron la oportunidad de realizar este trabajo, me guiaron y ayudaron en esta investigación. Gracias por su gran dedicación y apoyo. También deseo agradecerle a Cristina Bender su gran colaboración en este trabajo, su ayuda fue muy importante.

Mi especial gratitud también a todos mis profesores, por su enseñanza y sabiduría, la cual fue muy importante para mí.

A todos mis compañeros de la carrera, con muchos de ellos creció una amistad muy linda. Gracias por el aguante, por los consejos, por las horas de estudio juntos, de ellos aprendí mucho. Y también a los amigos que no conocí en esta carrera, gracias por apoyarme siempre, por las llamadas antes de un examen y después. Recorrí este camino junto a ustedes, gracias.

A mi amor Seba, mi compañero, amigo y novio, agradecerle por su paciencia, por su enorme apoyo y colaboración, por todo su amor, él es un pilar para mí. A su familia, que es parte de mi familia, gracias por acompañarme en esta etapa.

Gracias a todos por estar!!!!.

Valeria, Diciembre 2009.

Índice general

1. Introducción	1
2. Conceptos Preliminares	3
2.1. Objetos de Aprendizaje	3
2.2. Repositorio de Objetos de Aprendizaje	3
2.3. Metadatos	4
2.3.1. Estándar LOM	4
2.4. Agentes Recomendadores	5
2.5. Agentes BDI	7
2.5.1. Agentes g-BDI	7
3. Análisis de Repositorios de Objetos de Aprendizaje	11
3.1. Repositorio FLOR	11
3.2. Repositorio OER Commons	12
3.3. Repositorio ARIADNE	12
3.4. Conclusiones	14
4. Arquitectura del Sistema Recomendador	15
4.1. Arquitectura del Sistema Recomendador	15
4.1.1. El Agente Interfaz (Agente-I)	15
4.1.2. El Agente Refinador Semántico (Agente-RS)	16
4.1.3. El Agente Perfil de Usuario (Agente-PU)	16
4.1.4. El Agente Buscador (Agente-B)	17
4.1.5. El Agente Mediador (Agente-M)	17
4.1.6. El Agente Recomendador (Agente-R)	17
4.2. Diseño del Agente Recomendador	17
5. Implementación del Prototipo del Agente Recomendador	19
5.1. Lenguaje de Programación	19
5.2. Agente Interfaz	19
5.2.1. Feedback del usuario	21
5.3. Agente Recomendador	22
5.3.1. Contexto Desire	22
5.3.2. Contexto Belief	23
5.3.3. Reglas	27
5.3.4. Contexto Intention	37

5.4. Ejemplo	37
6. Caso de estudio	41
6.1. Repositorio Local	41
6.2. Análisis de resultados	42
7. Discusiones	49
7.1. Repositorios	49
7.2. Prueba del Prototipo	49
7.3. Proyectos y Publicaciones	50
7.4. Trabajos Futuros	50

Índice de figuras

3.1. Interfaz de búsqueda FLOR	12
3.2. Interfaz de búsqueda OER Commons	13
3.3. Interfaz de búsqueda ARIADNE	14
4.1. Arquitectura del Sistema Multiagente	16
5.1. Interfaz	20
5.2. Resultado	22

Índice de cuadros

5.1. Datos que puede ingresar el usuario.	21
5.2. Metadatos Seleccionados.	26
5.3. Valores de creencia para Interacción.	28
5.4. Valores de creencia para Rol.	29
5.5. Definición de las funciones f y g	30
5.6. Extracto del perfil del usuario correspondiente a José.	38
5.7. Metadatos relevantes de los Objetos de Aprendizaje recuperados.	38
6.1. Objetivos de los usuarios.	42
6.2. Perfiles de los usuarios.	43
6.3. Lista de Objetos Recomendados.	44
6.4. Feedback del usuario.	44
6.5. Metadatos relevantes de los Objetos de Aprendizaje del repositorio local.	45
6.6. Distancias del Agente-R sobre los casos satisfactorios.	46

Capítulo 1

Introducción

En el dominio de la educación existe gran cantidad y diversidad de material que puede contribuir al proceso enseñanza-aprendizaje. En particular, con el desarrollo de la Web y su utilización masiva, se tiene una amplia gama de posibilidades de acceso a material útil e interesante para ser empleado tanto por un alumno que desea aprender un tema, como por un docente que quiere preparar material didáctico. La Web se ha convertido en una herramienta fundamental para la recuperación de este tipo de material, por lo general a través de buscadores. Pero no siempre el resultado es el esperado por el usuario si la búsqueda se realiza sólo considerando palabras claves, porque un material recuperado puede no ser el adecuado para todos. Esto se debe a que los usuarios poseen distintas características y preferencias personales, que deberían también ser consideradas en el momento de la búsqueda. Como ejemplo de buscador Web tenemos el reconocido buscador Google, donde la búsqueda no se basa en datos referentes al entorno educativo del objeto, y esto puede llevar al usuario a probar varios objetos antes de encontrar el que satisface su deseo. En los últimos años, los sistemas recomendadores surgen para ayudar a resolver este tipo de problema, puesto que son capaces de seleccionar, de forma automática y personalizada, el material que mejor se adapte a las preferencias o necesidades de un usuario.

El enfoque en este trabajo es lograr la personalización de los resultados en la recuperación, utilizando datos del usuario modelados en perfiles personales, y metadatos con las descripciones semánticas de cada recurso. El perfil del usuario se utiliza en la recomendación a partir de sus preferencias y la importancia relativa de cada una al momento de elegir un material dado. Esto, en conjunto con los metadatos de cada recurso educativo, constituye la base para el razonamiento del sistema recomendador.

En este trabajo se presenta una arquitectura de un sistema recomendador cuyo objetivo es apoyar a los usuarios a encontrar recursos educativos de acuerdo con sus características y preferencias, y les brinda los resultados ordenados de acuerdo con su perfil. También se analizan diferentes repositorios de objetos de aprendizaje, así como también los diferentes modelos de datos que especifican los metadatos. Por último, se implementa un prototipo del Agente Recomendador, uno de los agentes de la arquitectura del sistema, el cual se diseña como un agente BDI graduado para poder razonar sobre preferencias graduadas de los usuarios.

El resto del trabajo se organiza de la forma siguiente: en el Capítulo 2 se describen conceptos preliminares como objetos de aprendizaje, sistemas recomendadores y sistemas

multiagentes. En el Capítulo 3 se presenta un breve análisis de algunos repositorios Web. En el Capítulo 4 se describe la arquitectura del sistema multiagente propuesto, en el cual se modela como un agente BDI graduado a uno de los agentes principales del sistema. En el Capítulo 5 se presenta la implementación del prototipo del agente recomendador propuesto. En el Capítulo 6 se describe un caso de estudio. Y finalmente, se plantean algunas discusiones.

Capítulo 2

Conceptos Preliminares

2.1. Objetos de Aprendizaje

Según Wiley [23] un **Objeto de Aprendizaje** (OA) es todo recurso digital que apoya a la educación y que puede ser reutilizado. Por otro lado el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE Institute of Electrical and Electronics Engineers) define un **Objeto de Aprendizaje** como una entidad, digital o no digital, que puede ser usada para aprendizaje, educación o entrenamiento [12]. El concepto de Objeto de Aprendizaje (Learning Object) abarca principalmente a un conjunto de materiales digitales los que como unidad o agrupación permiten o facilitan alcanzar un objetivo educacional. Ejemplos de los recursos digitales más pequeños incluyen a imágenes o fotos, cortos de video o audio, pequeñas porciones de texto, ecuaciones, definiciones, animaciones, pequeñas aplicaciones web, entre otros. Recursos digitales de mayor tamaño son páginas web completas que combinen texto, imágenes y otros medios de comunicación. Es decir, los objetos pueden adquirir formas muy diversas y presentarse en diferentes formatos y soportes.

Dado que estos objetos han de contar con la posibilidad de ser reutilizados, actualizados, combinados, separados, referenciados y sistematizados, se necesitan potentes repositorios que los contengan.

2.2. Repositorio de Objetos de Aprendizaje

Un **Repositorio de Objetos de Aprendizaje** es una gran colección de OA, estructurada como una base de datos con metadatos asociados y que generalmente se puede encontrar en la Web. Los objetos y el repositorio son complementarios. Un objeto que no guarde las características necesarias para poder integrarse en un repositorio pierde todas sus virtudes y, a su vez, un repositorio que no cuente con una buena base de objetos deja de ser operativo.

Algunos ejemplos de repositorios son: MERLOT (Multimedia Educational Resource for Learning and Online Teaching)¹, CAREO (Campus Alberta Repository of Educational Objects)², FLOR (Federación Latinoamericana de Repositorios)³, OER Commons (Open

¹<http://www.merlot.org/>

²<http://www.careo.org/>

³<http://ariadne.cti.espol.edu.ec/FederatedClient>

Educational Resources)⁴, SMETE (Science, Mathematics, Engineering and Technology Education)⁵, CeLeBraTe (Context eLearning with Broadband Technologies)⁶, ARIADNE⁷, entre otros.

Para este trabajo, se han analizado en particular los repositorios FLOR, OER Commons y ARIADNE. FLOR es una federación de repositorios de objetos de aprendizaje de diferentes instituciones educativas de latinoamérica. Para poder comunicarse con este repositorio se utiliza el protocolo SQI (Simple Query Interface), que es una interfaz de programación de aplicaciones. OER Commons es la primera red abierta de aprendizaje donde los docentes pueden acceder a los materiales de sus colegas, compartir los propios y colaborar en la puesta en práctica en el aula. Utiliza para la transmisión de metadatos el protocolo OAI-PMH (Open Archives Initiative - Protocol for Metadata Harvesting). Por último, ARIADNE es una asociación Europea que permite compartir conocimiento y fomenta la cooperación internacional en enseñanza. Al igual que FLOR, utiliza como protocolo de comunicación el protocolo SQI. En el Capítulo 3 se analizan más ampliamente estos tres repositorios.

2.3. Metadatos

Los **metadatos** son un conjunto de atributos necesarios para describir las principales características de un recurso. A través de los metadatos se tiene un primer acercamiento con el objeto, conociendo rápidamente sus principales características. Se utilizan con el propósito de poder encontrar, gestionar, reusar y almacenar objetos de aprendizaje de manera efectiva. Son especialmente útiles en los recursos que no son textuales, por ejemplo los multimedia, y en los que su contenido no puede ser indexado por sistemas automáticos. Los metadatos pueden clasificarse en descriptivos, administrativos y de estructura [3].

Los metadatos descriptivos sirven para encontrar, identificar y seleccionar un recurso, formar colecciones de recursos similares, así como relacionarlo con otros recursos. Los metadatos administrativos incluyen información sobre cuándo y cómo fue creado el recurso, quién es el responsable del acceso o de la actualización del contenido, y otra información técnica, como la versión de software o el hardware necesario para ejecutarlo. Los metadatos estructurales permiten identificar cada una de las partes que componen al recurso y definir la estructura que le da forma. Por ejemplo, un libro, que contiene capítulos y páginas, se puede etiquetar con metadatos que identifican cada parte y la relación que guardan entre ellas.

2.3.1. Estándar LOM

Para lograr una buena interconexión entre repositorios y facilitar el desarrollo de sistemas de búsqueda como es el caso del sistema recomendador, es importante que los metadatos estén escritos y organizados según un estándar. Algunos de ellos son el estándar *Dublin Core*, que es un estándar de propósito general; el estándar LOM, el cual veremos

⁴<http://www.oercommons.org/>

⁵<http://www.smete.org/smete/>

⁶http://celebrate.eun.org/eun.org2/eun/en/index_celebrate.cfm/

⁷<http://www.ariadne-eu.org/>

más en detalle a continuación, y especializaciones de este, lo que se conoce como perfil de aplicación LOM. Algunos de estos perfiles son, el perfil de aplicación *CanCore*, el perfil de aplicación LOM-ES, entre otros esquemas de metadatos.

El estándar IEEE 1484.12.1-2002⁸ acredita al modelo de datos LOM (Learning Object Metadata) como el estándar de metadatos para los objetos de aprendizaje. LOM especifica la sintaxis y la semántica de un conjunto mínimo de metadatos necesario para identificar, administrar, localizar y evaluar un objeto de aprendizaje en forma completa y adecuada. Este estándar contempla la diversidad cultural e idiomática de los contextos en los que se puedan utilizar los objetos y sus metadatos. En LOM, los metadatos se organizan en forma jerárquica, agrupándolos en categorías. En particular, el esquema base LOMv1.0 consiste de nueve categorías:

- General: agrupa información general que describe el objeto de aprendizaje como un todo.
- Ciclo de vida: agrupa las características relacionadas con la historia y el estado presente del objeto.
- Metametadatos: agrupa información sobre los mismos metadatos, no sobre el objeto de aprendizaje que se está describiendo.
- Técnica: contiene requisitos y características técnicas del objeto.
- Educacional: agrupa las características pedagógicas y educacionales del objeto.
- Derechos: agrupa las condiciones de uso para la explotación del recurso.
- Relación: agrupa características que definen la vinculación del recurso descrito con otros objetos de aprendizaje.
- Anotación: provee comentarios sobre el uso educativo del objeto de aprendizaje.
- Clasificación: describe este objeto de aprendizaje en relación a un sistema de clasificación particular.

La personalización de los resultados propuesta en este trabajo se sustenta en los metadatos de los documentos y en el perfil del usuario. Los metadatos siguen el estándar LOM. Esto se debe a que este estándar en la actualidad es el único para el marcado de metadatos educativos y además, la mayoría de los repositorios trabajan con este esquema.

2.4. Agentes Recomendadores

En los últimos años la comunidad de Inteligencia Artificial ha desarrollado un intenso trabajo alrededor de los sistemas recomendadores [15]. Estos sistemas ayudan a las personas a encontrar lo que necesitan especialmente de la Web y han tenido una amplia aceptación entre los usuarios. El objetivo de estos agentes es explorar y filtrar las mejores opciones a partir de un perfil de usuario (preferencias, características, etc.) considerando

⁸IEEE LOM specification. <http://ltsc.ieee.org/wg12>, 2002.

un importante número de posibilidades diferentes, muchas de ellas provenientes de la Web. Esto involucra la construcción de un modelo o perfil de usuario el cual puede ser obtenido de forma implícita o explícita. Una taxonomía detallada de sistemas recomendadores puede apreciarse en [13] y las principales técnicas para su desarrollo pueden agruparse según [22] en:

- Sistemas basados en contenidos: utilizan solamente las preferencias del usuario; intentan recomendar elementos que son similares a aquellos que al usuario le gustaron en el pasado. Se enfocan en algoritmos de aprendizaje de preferencias de usuarios y filtran nuevos elementos, teniendo en cuenta los que más satisfacen las preferencias de estos.
- Sistemas de filtrado colaborativo: requieren que diversos usuarios evalúen elementos según sus preferencias (*ratings*). Luego unen los roles de estos usuarios con el perfil del usuario que busca la recomendación. Es decir, estos sistemas buscan personas que poseen perfiles similares para otorgarle una recomendación al usuario interesado. Este sistema se diferencia del anterior en que su recomendación se basa en las preferencias de otros usuarios, y el basado en contenido, en la similitud con el perfil del usuario.
- Sistemas de *data mining* social: extraen implícitamente preferencias de bases de datos correspondientes a comportamientos sociales, tal como los mensajes Usenet, sistema del historial de uso, citas, hyperlinks. Estos sistemas se basan en características de la interacción entre la computadora y el usuario, sobre la presentación de los resultados. Estas visualizaciones son siempre presentadas de manera de ayudar a la navegación de espacios de información como la Web.
- Sistemas híbridos: combinan características del sistema basado en contenido y del sistema de filtrado colaborativo. De esta manera su complementación aumenta la efectividad del otro, evitando las desventajas particulares de cada uno y permitiendo un sistema integrado para alcanzar confiabilidad.

En los últimos años, se ha incrementado el diseño e implementación de sistemas multiagentes para abordar el desarrollo de sistemas distribuidos complejos. Y en particular se ha utilizado para el desarrollo de sistemas recomendadores. Esta tecnología de agentes es importante a la hora de modelar diferentes características que se esperan de estos sistemas como por ejemplo: generar y considerar el perfil del usuario, inferir y agregar información proveniente de fuentes heterogéneas y distribuidas, obtener sistemas escalables, abiertos y seguros, y realizar la tarea requiriendo la menor intervención de las personas.

Entre las aplicaciones potenciales de los sistemas recomendadores, el dominio de la educación parece ser un buen candidato ya que las ofertas de recursos educativos están en constante crecimiento. Según Zaiane [24], los sistemas recomendadores han surgido con el comercio electrónico pero no se habían aplicado en el dominio de la educación y propone usarlos en este campo del conocimiento. Siguiendo esta propuesta, Romero et al. [19] plantea el uso de técnicas de minería de datos para recomendar la navegación entre links y Soonthornphisaj et al. [21] propone un sistema que integra el material recomendado antes de dárselo al usuario. Zhu et al. [25] propone una arquitectura de un sistema de recomendación personalizado de educación escolar, la cual adopta una tecnología multiagente y

consiste de seis agentes de software, los cuales coordinan el trabajo de forma jerárquica entre ellos para ofrecer una gama de funciones primarias.

El sistema recomendador propuesto en este trabajo está basado en contenidos y utiliza el perfil del usuario, metadatos de OA y conocimiento del dominio para realizar la recomendación.

2.5. Agentes BDI

Una de las arquitecturas de agentes más notorias es el agente BDI (Belief-Desire-Intention) propuesto por Rao y Georgeff [17]. Este modelo está basado en la representación explícita de las creencias (B) del agente, que representan el estado del entorno, sus deseos (D), representando sus motivaciones, y las intenciones (I) del agente, que modelizan sus metas u objetivos. Esta arquitectura ha evolucionado en el tiempo y ha sido utilizada en importantes aplicaciones de sistemas multiagentes. Las distintas arquitecturas de agentes desarrolladas hasta el momento, han sido planteadas para manejar información básicamente bivaluada. Con el propósito de hacer que la arquitectura BDI sea más flexible, Casali et al. [5] han propuesto un modelo general para diseñar agentes BDI graduados (g-BDI). Este modelo permite especificar arquitecturas capaces de tratar con la incertidumbre del entorno y con actitudes mentales graduadas, con el fin de desarrollar agentes que puedan tener una mejor performance en entornos dinámicos e inciertos. En este modelo, los grados en las creencias van a representar en qué medida el agente cree que una fórmula es cierta. Los grados en los deseos, positivos o negativos, permiten al agente establecer respectivamente, diferentes niveles de preferencia o de rechazo. Las graduaciones en las intenciones también estarán dando una medida de preferencia, pero en este caso, modelarán la relación costo-beneficio que le significa al agente alcanzar esa meta. A partir de la representación de estas tres actitudes y según como interactúen unas con otras, se pueden modelar distintos tipos de agentes que tendrán diferentes comportamientos.

Esta arquitectura ha sido utilizada previamente para modelar agentes en el dominio turístico [7].

2.5.1. Agentes g-BDI

Para especificar la arquitectura de un agente BDI graduado, se utiliza la noción de sistema multi-contexto. Este sistema ha sido planteado por Giunchiglia [10] y utilizado por Sabater et al. [20] para especificar distintas arquitecturas de agentes y en particular para la especificación de agentes BDI [16].

Una especificación multi-contexto consta básicamente de unidades o contextos, cada uno con su lógica y reglas puente (bridge rules). De esta forma, un sistema es definido como un grupo de contextos interconectados: $\langle \{C_i\}_{i \in I}, \Delta_{br} \rangle$, donde cada contexto $C_i \in \{C_i\}_{i \in I}$ queda definido por $C_i = \langle L_i, A_i, \Delta_i \rangle$ donde L_i es un lenguaje, A_i axiomas y Δ_i reglas de inferencia, los cuales definen la lógica que utiliza el contexto. También se puede dar una teoría $T_i \subset L_i$ asociada a cada unidad. Δ_{br} son reglas de inferencia donde las premisas y conclusiones pertenecen a distintos contextos, por ejemplo la regla:

$$\frac{C_1 : \psi, C_2 : \varphi}{C_3 : \theta}$$

significa que podemos deducir θ en el contexto C_3 si las fórmulas ψ y φ son deducidas en C_1 y C_2 , respectivamente.

En estos sistemas, la deducción está a cargo de dos tipos de reglas de inferencia, las reglas internas Δ_i de cada contexto y las reglas puente Δ_{br} . Las reglas internas permiten obtener consecuencias dentro de una teoría, mientras que las reglas puente permiten exportar los resultados de una teoría a otra [9].

En el modelo de agente g-BDI se tienen contextos para representar sus creencias (BC), deseos (DC) e intenciones (IC). También se pueden considerar otras unidades funcionales como por ejemplo una unidad dedicada a la comunicación (CC), la cual establece una interfaz única y bien definida con el entorno, representando los sensores y actuadores del agente. Además se incluye una unidad de planificación (PC) que es la encargada de encontrar un plan para cambiar de un estado del mundo a otro nuevo. Luego el modelo de agente es: $A_g = \langle \{BC, DC, IC, CC, PC\}, \Delta_{br} \rangle$.

Cada contexto tiene su lógica asociada, lo que significa que se define un lenguaje, con su semántica y su sistema deductivo.

Para representar y razonar con grados en las creencias, deseos e intenciones, se elige utilizar lógicas modales multivaluadas. En particular se seguirá la propuesta de Godo et al. [11], donde el razonamiento bajo incertidumbre se trata definiendo teorías modales adecuadas sobre lógicas multivaluadas.

Presentamos la idea básica considerando el contexto BC donde los grados de creencia son modelados por ejemplo, como probabilidades. Luego, para cada fórmula clásica (bivaluada), podemos considerar la fórmula modal $B\varphi$ denotando “ φ es probable”. Esta fórmula $B\varphi$ es entonces una fórmula fuzzy (multivaluada), la cuál puede ser verdadera en cierto grado, dependiendo de la probabilidad de φ . En particular, se puede considerar como grado de verdad de $B\varphi$ a la probabilidad de φ .

A continuación se presentan algunas características generales de los distintos componentes (BC, DC, IC, Δ_{br}) de un agente BDI graduado, para más detalles de los mismos ver [6] y [8].

Distintos Contextos

Contexto Belief

Este contexto estará dedicado a modelizar las creencias del agente acerca del entorno. Para representar la incertidumbre de las mismas, se utilizan fórmulas multivaluadas, siguiendo el marco lógico mencionado en la sección anterior y considerando en este caso particular, una medida de probabilidad para modelar la incertidumbre.

Para definir el lenguaje base (crisp) se parte de un lenguaje proposicional L al que hay que incluir los elementos que representan a las acciones, siguiendo la formalización planteada en las lógicas dinámicas. Las acciones, las transformaciones que éstas causan en el entorno, y sus costos asociados serán parte de las creencias del agente. El lenguaje proposicional L es extendido a L_D incluyendo un conjunto de acciones de la forma $[\alpha]$ y una modalidad para indicar las fórmulas que éstas harán cierta, donde α es una acción. Luego, la interpretación de $[\alpha]\varphi$ es “*siempre después de α , φ es cierta*”.

Sobre este lenguaje base L_D se introduce un operador modal fuzzy B, de modo que si $\varphi \in L_D$ la interpretación de $B\varphi$ será “ φ es probable”. Entonces las fórmulas B-modal son

construidas a partir de fórmulas $B\varphi$, donde φ es una fórmula (crisp) de L_D y de constantes racionales en el $[0, 1]$, usando los conectivos de lógica multivaluada de Łukasiewicz (\rightarrow_L , $\&$). En esta lógica, una fórmula modal del tipo $\bar{r} \rightarrow_L B\varphi$ expresa que la probabilidad de φ es al menos r y se nota $(B\varphi, r)$.

La axiomática y reglas de inferencia para completar la parte deductiva del BC pueden verse en [6]. Las creencias de un agente particular se representarán mediante una teoría T_B de fórmulas del lenguaje.

Contexto Desire

En este contexto se representan y se razona sobre los deseos de un agente. Estos manifestarán sus preferencias ideales, independientemente de cuál sea la ubicación del agente y del costo que involucre alcanzarlos. En las preferencias, es interesante poder distinguir entre lo que es positivamente deseado y aquello que es rechazado. Según los trabajos sobre bipolaridad de Benferhat et al. [2], estos tipos de información positiva y negativa pueden modelizarse gradualmente mediante la lógica posibilística, por medio de un esquema bipolar utilizando restricciones expresadas en términos de medidas de la teoría de la posibilidad. Con inspiración en estos trabajos, en este contexto se pueden representar los deseos positivos y negativos de un agente. Los deseos positivos modelizarán lo que el agente desea que ocurra en algún grado, y los deseos negativos corresponderán a lo que rechaza o desea que no ocurra, con cierto peso. Así como para el lenguaje de BC , el lenguaje de DC es definido sobre el lenguaje proposicional L introduciendo dos operadores (fuzzy) modales D^+ y D^- . Donde $D^+\varphi$ se lee “ φ es positivamente deseado” y su grado de verdad representa el nivel de satisfacción del agente cuando φ es cierta. $D^-\varphi$ se lee “ φ es negativamente deseada” y su grado de verdad representa una medida de disgusto del agente cuando φ es cierta.

Los deseos de un agente concreto se representarán mediante una teoría T_D . En particular si $(D^+\phi_i, 1) \in T_D$ representa el caso en que el agente tiene preferencia máxima por ϕ_i y estará totalmente satisfecho ante su cumplimiento. Si $T_D \vdash_{DL} (D^+\phi_j, 0)$ significa que el agente es indiferente a ϕ_j y que su cumplimiento no le trae ningún beneficio. Por otro lado, si $(D^-\psi_i, 1) \in T_D$ representa que el agente tiene total rechazo por ψ_i y que los estados donde ψ_i sea cierta serán totalmente inaceptables para el agente. Si se tiene $(D^-\psi_j, 0)$, significa que ψ_j no es rechazada, al igual que las fórmulas que no estén explícitas en la teoría. En el caso en que $(D^+\phi, \alpha), (D^+\psi, \beta)$, con $\alpha > \beta$, entonces ϕ es más preferido que ψ .

Contexto Intention

En este contexto se representan las intenciones del agente, las cuales son proactivas y representan los objetivos que el agente intentará alcanzar en pro de satisfacer sus deseos. Generalmente la intención deriva en un plan de una o varias acciones, que el agente desarrollará para lograr el objetivo. En este contexto, las intenciones dependerán no sólo del beneficio que le traerá alcanzar al agente el objetivo φ (representado en $D^+\varphi$), sino del estado w en que se encuentre el agente y del costo que le signifique transicionar a un estado w' donde se satisfaga la fórmula φ . Al permitir graduaciones en las intenciones

se puede representar una medida que refleje la relación costo/beneficio que le traerá al agente alcanzar el objetivo.

Reglas Puente

Como dijimos en la Sección 2.5.1 las reglas puente son una parte fundamental en el modelo g-BDI, las cuales nos permiten realizar deducciones donde premisas y conclusiones pertenecen a distintos contextos. A continuación se presenta un ejemplo de regla puente en donde a partir de deseos positivos ($D^+\varphi, d$) y negativos ($D^-\psi, u$) como así también la existencia del plan α : $plan(\varphi, \alpha, P, A, c)$, y conocimiento acerca de cómo la ejecución de dicho plan satisface los deseos ($B[\alpha]\varphi, r$), y que las postcondiciones del mismo no implican un deseo negativo ($B(A \rightarrow \neg\psi)$), se encuentra que α es un *plan posible*: $planPosible(\varphi, \alpha, P, A, c)$.

$$\frac{DC : (D^+\varphi, d), DC : (D^-\psi, u), PC : plan(\varphi, \alpha, P, A, c), \\ BC : (B[\alpha]\varphi, r), BC : B(A \rightarrow \neg\psi)}{PC : planPosible(\varphi, \alpha, P, A, c)} \quad (2.1)$$

Capítulo 3

Análisis de Repositorios de Objetos de Aprendizaje

Como se mencionó en la Sección 2.2, se han analizado los repositorios FLOR, OER Commons y ARIADNE y se investigó sobre los protocolos de comunicación para acceder a estos, sobre los metadatos de los objetos y sobre los modelos de datos utilizados para especificarlos. A continuación se describen estos repositorios.

3.1. Repositorio FLOR

La Comunidad Latinoamericana de Objetos de Aprendizaje (LACLO)¹, es una comunidad abierta, integrada por personas e instituciones interesadas en la investigación, desarrollo y aplicación de las tecnologías relacionadas con objetos de aprendizaje en el sector educativo Latinoamericano. Su principal misión es ayudar a la articulación de los diferentes esfuerzos en la Región para diseminar los avances y beneficios de esta tecnología, a fin de que Latinoamérica pueda hacer frente al gran reto educativo de este siglo: poder ofrecer recursos educativos personalizados y de calidad a cualquier persona, en cualquier momento y en cualquier lugar. Una de las principales herramientas que utiliza LACLO para alcanzar sus objetivos es FLOR, una federación de repositorios de objetos de aprendizaje de diferentes instituciones educativas de Latinoamérica. Cada institución interesada puede compartir sus objetos de aprendizaje en FLOR. Los metadatos de los objetos son especificados utilizando el estándar LOM, y para acceder a recuperar estos recursos se utiliza el estándar SQI (Simple Query Interface), que es una interfaz de programación de aplicaciones (en inglés API, acrónimo de Application Programming Interface), y permite la comunicación sincrónica y asincrónica de búsquedas y resultados entre repositorios.

En su interfaz de búsqueda (Figura 3.1) podemos observar que la búsqueda se realiza especificando sólo palabras claves, es decir, no permite ingresar otras preferencias, como el idioma, o el contexto etc.

¹<http://www.laclo.espol.edu.ec/>



Figura 3.1: Interfaz de búsqueda FLOR

3.2. Repositorio OER Commons

OER Commons es la primera red abierta de aprendizaje donde los docentes pueden acceder a los materiales de sus colegas, compartir los propios y colaborar en la puesta en práctica en el aula. Utiliza para la transmisión de metadatos el protocolo OAI-PMH (Open Archives Initiative - Protocol for Metadata Harvesting). OAI-PMH utiliza transacciones HTTP para emitir preguntas y obtener respuestas entre un servidor o archivo y un cliente o servicio recolector de metadatos. El segundo puede pedir al primero que le envíe metadatos según determinados criterios como, por ejemplo, la fecha de creación de los datos. En respuesta, el primero devuelve un conjunto de registros en formato XML, incluyendo identificadores (URLs por ejemplo) de los objetos descritos en cada registro.

Las peticiones se emiten utilizando los métodos GET o POST del protocolo HTTP y constan de una lista de opciones con la forma de pares del tipo: clave=valor [1].

Se pueden recuperar registros utilizando distintos formatos de metadatos que estén disponibles en el servidor donde se realiza la búsqueda. El repositorio OER Commons cuenta entre otros con el estándar LOM para describir sus objetos. OAI-PMH, a diferencia de SQI, no es un protocolo de búsqueda.

En la Figura 3.2 podemos observar que a diferencia de FLOR cuenta con una búsqueda avanzada, donde además de ingresar palabras claves, se puede especificar criterios adicionales, como idioma, tema al cual pertenecen los objetos que se desean encontrar, tipo de material, formato del recurso, bibliotecas y colecciones, condiciones de uso, etc.

3.3. Repositorio ARIADNE

ARIADNE (nombrada así por la figura de la mitología Griega) es una asociación Europea que permite compartir conocimiento y fomenta la cooperación internacional en enseñanza, abierta a todo el mundo. Una infraestructura ha sido desarrollada en Bélgica y Suiza para la producción de contenido de aprendizaje reutilizable. El núcleo de esta

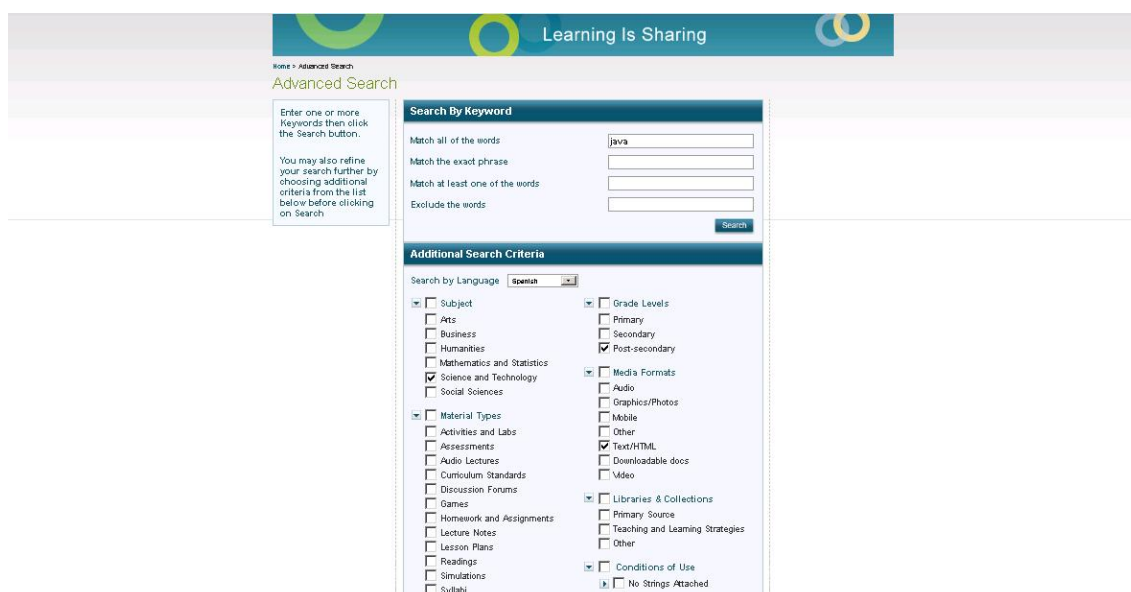


Figura 3.2: Interfaz de búsqueda OER Commons

infraestructura es una biblioteca digital distribuida, que contiene componentes educacionales reusables, es decir es un repositorio de objetos de aprendizaje distribuido, llamado *Knowledge Pool System* (KPS) que actualmente es usado de manera común en contextos académicos y empresariales. Los usuarios finales interactúan con éste a través de clientes. Java y aplicaciones web permiten a los usuarios insertar documentos y sus metadatos asociados en el KPS, buscar documentos relevantes, y descargar estos del KPS. Las aplicaciones Java interactúan con éste a través de los servicios web de ARIADNE.

Desde el punto de vista técnico el KPS consiste de una base de datos relacional que contiene los metadatos, y de un repositorio de documentos que contiene los objetos de aprendizaje. ARIADNE no se enfoca en objetos de aprendizaje de ningún campo particular.

En ARIADNE el repositorio de objetos de aprendizaje consiste en una arquitectura de 3 capas:

- En la capa inferior, el repositorio permite buscar, publicar y recuperar objetos de aprendizaje.
- Una API, que está conectada a servicios web, permite la manipulación de esta capa inferior.
- Aplicaciones tales como Moodle plugin o ALOCom Office plugin, hacen accesible de manera transparente a la información a través de aplicaciones externas.

Un cliente puede navegar ARIADNE a través de SQL.

Por otro lado ARIADNE al igual que FLOR permite búsquedas federadas; algunos de los repositorios de esta federación con los que trabaja son: CGIAR, EducaNext, LACLO-FLOR, Merlot y OER Commons.

ARIADNE entre sus estándares de modelos de datos, utiliza el estándar LOM.

En la Figura 3.3 podemos ver una de las interfaces de búsqueda de ARIADNE. Esta interfaz es un buscador que permite encontrar objetos de aprendizaje en el repositorio local o federado y navegar los resultados por categorías. En la imagen podemos observar que a la izquierda presenta criterios de búsqueda que permiten filtrar por tipo de recurso, formato, contexto, idioma y autor del recurso. El resultado es una lista de objetos, donde para cada uno se puede ver la descripción de estos, es decir sus metadatos. Además si se cuenta con los permisos suficientes se puede descargar el objeto.

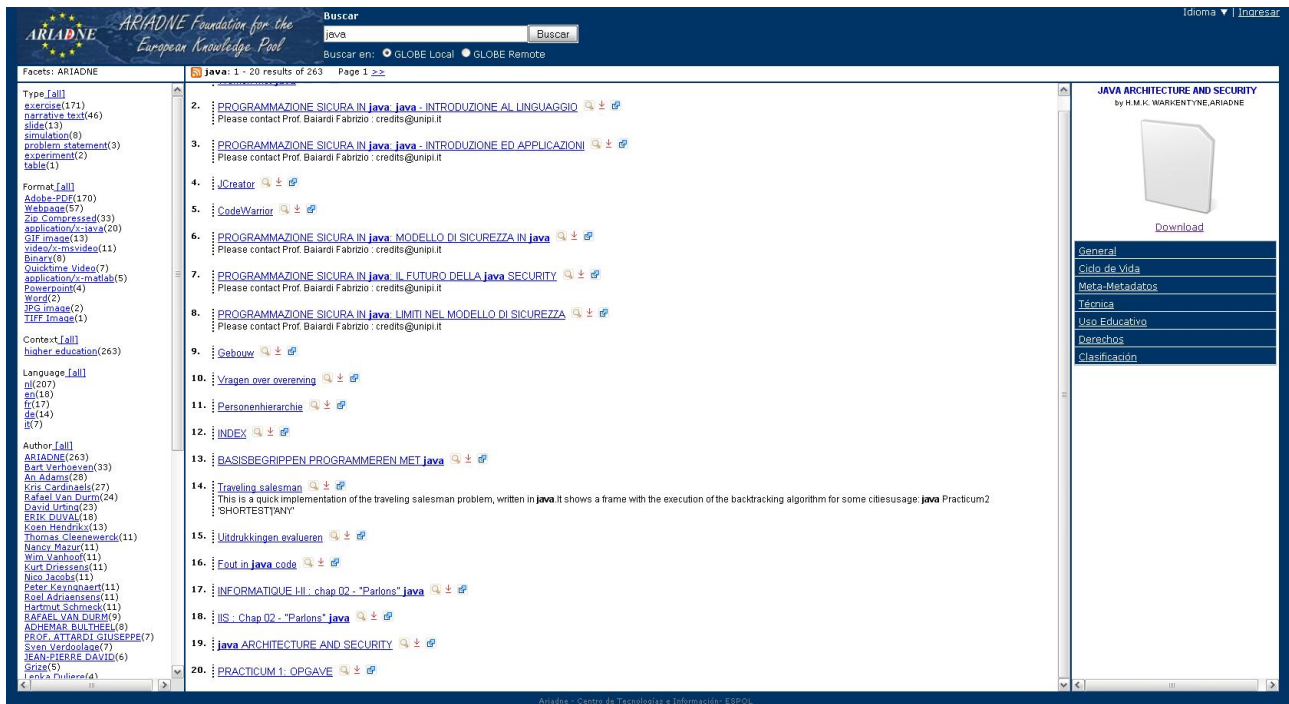


Figura 3.3: Interfaz de búsqueda ARIADNE

3.4. Conclusiones

Al analizar los repositorios, una de las observaciones que resaltamos es la falta de información sobre metadatos educativos. Por ejemplo OER Commons en su mayoría cuenta únicamente con los metadatos que especifican la edad y el idioma del destinatario típico, sin contar con el tiempo aproximado que puede tomar trabajar con el objeto, o el grado de interacción requerido por el mismo, etc. Sin embargo, los registros cuentan con etiquetas (las cuales no son parte del estándar LOM) que permiten especificar los datos educativos como: el tipo de material buscado, y el contexto académico. Por otro lado, FLOR cuenta con algunos metadatos educativos como Learning Resource Type (Tipo de Recurso Educativo), Intended End User Role (Rol del usuario final) y Context (Contexto). Faltando la descripción de los metadatos Interactivity Type, Difficulty, los cuales son necesarios para efectuar una recomendación más personalizada. ARIADNE en cambio, cuenta con la mayoría de los metadatos educativos.

Capítulo 4

Arquitectura del Sistema Recomendador

En este Capítulo se propone una arquitectura multiagente para el Sistema Recomendador y en particular se diseña al Agente Recomendador. Esto se detalla en las secciones siguientes.

4.1. Arquitectura del Sistema Recomendador

La arquitectura multiagente propuesta consta de los siguientes agentes: el *Agente Interfaz (Agente-I)*, que se encarga de capturar los datos ingresados por el usuario; el *Agente Refinador Semántico (Agente-RS)* cuyo objetivo es producir la estrategia de búsqueda asociada al interés del usuario; el *Agente Perfil de Usuario (Agente-PU)* cuya tarea es construir el perfil de usuario; los *Agentes Buscadores (Agente-B_i)* que se encargan de encontrar los objetos educativos (OA) que satisfacen la temática y las restricciones de interés del usuario en los distintos repositorios; el *Agente Mediador (Agente-M)* que integra lo encontrado por cada Agente-B_i y soluciona conflictos, y por último el *Agente Recomendador (Agente-R)* cuyo objetivo es seleccionar los mejores objetos de acuerdo al perfil del usuario.

Esta arquitectura se muestra en la Figura 4.1 y en las secciones siguientes se detallan los agentes que la componen.

4.1.1. El Agente Interfaz (Agente-I)

Interactúa con el usuario a través de una interfaz gráfica. Captura los datos ingresados por el usuario, y le despliega el resultado de su búsqueda. Este agente le provee al Agente-PU las preferencias y restricciones del usuario y al Agente-RS la temática sobre la búsqueda. Luego recibe del Agente-R la recomendación (ranking de objetos de aprendizaje (OA)) que dará al usuario.

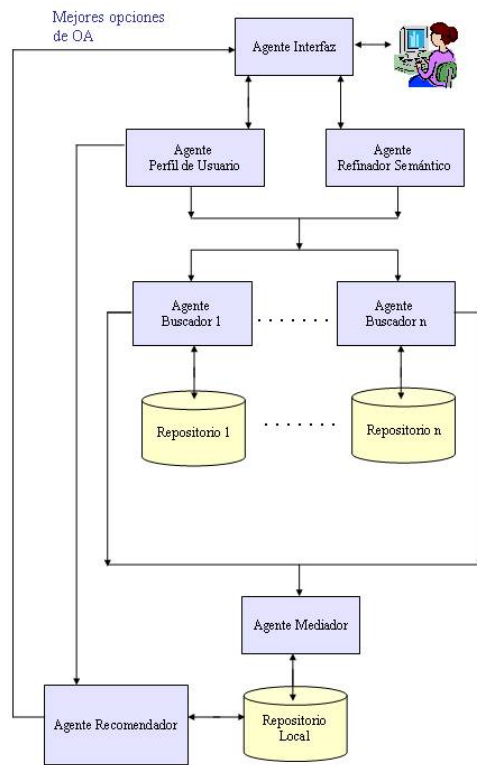


Figura 4.1: Arquitectura del Sistema Multiagente

4.1.2. El Agente Refinador Semántico (Agente-RS)

Produce la estrategia de búsqueda temática. Cuando el usuario hace una consulta a través del Agente Interfaz, ingresa un conjunto de términos que describen el tema de su interés. El Agente-RS desambigua estos términos y los expande semánticamente incorporando sinónimos y conceptos relacionados. La salida de este agente es una estrategia de búsqueda que consiste en la disyunción de las expansiones de cada concepto y luego la conjunción de esas mismas. Esta estrategia es enviada a cada agente buscador.

4.1.3. El Agente Perfil de Usuario (Agente-PU)

En una primera versión, recibe del Agente Interfaz los datos del usuario así como sus preferencias y restricciones, con el objetivo de construir su perfil. Este agente provee a los agentes buscadores (Agente- B_i) algunas restricciones (denominadas primarias) que puedan colaborar en el filtrado de los OA que conformarán el repositorio local. Además, este agente le provee al Agente Recomendador las preferencias y restricciones restantes del usuario, para que los objetos del repositorio local sean adecuadamente ordenados. En versiones más avanzadas se puede extender este agente para que aprenda de la conducta del usuario.

4.1.4. El Agente Buscador (Agente-B)

Busca en un repositorio de objetos de aprendizaje los que satisfagan la estrategia de búsqueda del usuario y, eventualmente, alguna de las restricciones (primarias). Para poder realizar la búsqueda, primero adapta la estrategia de búsqueda a la sintaxis del repositorio al que accede, luego se comunica con el repositorio a través de un protocolo de comunicación (por ejemplo, para comunicarse con el repositorio FLOR utiliza el protocolo SQI) para realizar la consulta correspondiente y de esta manera obtener una respuesta. Cuando recibe esta respuesta, actúa como un wrapper, dado que parseará el resultado de la búsqueda, lo transformará a un formato común (por ejemplo, al lenguaje XML), y se lo proveerá al Agente Mediador el cual lo depositará en el repositorio local. En esta arquitectura se van a tener tantos agentes buscadores como repositorios haya, dado que la manera de consultar varía de acuerdo al protocolo necesario para comunicarse con el repositorio correspondiente.

4.1.5. El Agente Mediador (Agente-M)

Integra la información encontrada por cada uno de los agentes buscadores, y soluciona conflictos de manera que los datos sean consistentes. Por ejemplo, si este agente recibe de dos agentes buscadores información que describen un mismo objeto, complementa los metadatos que lo describen en el repositorio, para obtener una descripción más completa del objeto. Por último, este agente deposita la información que generó en el repositorio local, para que el Agente-R pueda trabajar con estos datos.

4.1.6. El Agente Recomendador (Agente-R)

Este agente es el que finalmente recomienda al usuario los objetos de aprendizaje que satisfacen adecuadamente el perfil del mismo. La salida es una lista ordenada de recursos educacionales, donde el primero es el que más se adecua al deseo del usuario. Este agente busca en el repositorio local y teniendo en cuenta las preferencias y restricciones que fueron provistas por el Agente-PU, utiliza un conjunto de reglas y funciones para determinar cuáles de los OA son los que debe recomendar y en qué orden. Se ha diseñado el Agente-R utilizando el modelo de agente BDI graduado y en la próxima sección se lo describe.

4.2. Diseño del Agente Recomendador

Dado que la arquitectura planteada para este agente ha sido inspirada en el modelo g-BDI (Sección 2.5.1), para su especificación es necesario definir el contenido de los distintos contextos y las reglas puente que los vinculan. Las características particulares de los diferentes contextos en la especificación del Agente Recomendador (Agente-R) son:

- *Contexto Belief (BC)*: Este contexto modela la información del Agente-R sobre el entorno educativo: las características que presentan los OA, descriptas a través de los metadatos (por ejemplo, el idioma, grado de interacción, contexto académico, etc) y cómo estas características pueden ser usadas mediante un conjunto de reglas para estimar la satisfacción de las diferentes preferencias del usuario, es decir el

Agente-R inferirá, utilizando un conjunto de reglas, su grado de creencia b_{ij} en que un objeto O_i satisfaga las distintas preferencias p_j del usuario, representado por la fórmula $B(O_i, p_j, b_{ij})$.

- *Contexto Desire (DC)*: El deseo global del Agente-R, es encontrar el objeto de aprendizaje que satisfaga más al usuario, teniendo en cuenta la temática, las restricciones y las preferencias. En este contexto DC se representan las preferencias que el usuario tiene respecto a los OA (por ejemplo, el idioma o el contexto académico) y las restricciones (por ejemplo, su duración máxima). Tanto las preferencias como las restricciones pueden ser graduadas (tomando valores en el intervalo $[0, 1]$) expresando lo que el usuario desea o rechaza en diferentes grados, de un recurso educativo.
- *Contexto Intention (IC)*: Para esta aplicación, las intenciones serán los objetivos educativos que se intentarán alcanzar a través del mejor objeto (u objetos) seleccionado. En este contexto, las intenciones dependerán de las restricciones del usuario respecto a los OA, sus preferencias (que se espera que se traduzcan en beneficio para el usuario al aprender a través de un determinado OA) y también de la satisfacción esperada de las preferencias a través de un recurso educativo que cuenta con ciertas características representadas en sus metadatos. Por ejemplo, en qué medida un OA cuyo tipo de recurso está catalogado en su metadato como [ejercicio, lectura] satisface a la preferencia de que el recurso sea “práctico”. También se contempla la posibilidad de considerar la confianza en la fuente de la información otorgándole una medida de confianza. Estas variables son combinadas a través de una regla puente adecuada que determina el grado de intención de cada OA (extraído del repositorio local) para satisfacer el conjunto de preferencias del usuario. Este grado de intención será utilizado para ordenar los OA en la recomendación. Distintas funciones para computar este grado de intención asociado a cada OA se pueden utilizar modelando distintos comportamientos del Agente-R. En este trabajo, se plantea este cálculo como el promedio de las satisfacciones esperadas de las distintas preferencias, $P = p_1 \wedge \dots \wedge p_n$, a través de un recurso O_i , considerando distintos factores que provienen de diferentes contextos. Uno de ellos es la prioridad dada por el usuario a cada preferencia (p_j, d_j). Otro factor es el grado de satisfacción de cada preferencia p_j por las características de un recurso, representado por la fórmula $B(O_i, p_j, b_{ij})$. Por lo tanto, el grado de intención $I(O_i, P)$ del objeto O_i se calcula de la siguiente manera:

$$I(O_i, P) = \frac{\sum_{j=1}^n d_j \times b_{ij}}{n} \quad (4.1)$$

En el Capítulo 5 se describe la implementación de este agente, así como del Agente Interfaz, y en el Capítulo 6 se presenta un caso de estudio.

Capítulo 5

Implementación del Prototipo del Agente Recomendador

En este Capítulo se describe la implementación de un primer prototipo del Sistema Recomendador. En particular, en esta primera versión, se implementan el Agente Recomendador y el Agente Interfaz.

5.1. Lenguaje de Programación

El prototipo se implementó en el lenguaje SWI-Prolog,¹ dado que es un lenguaje de programación lógico e interpretado, bastante conocido en el medio de investigación en Inteligencia Artificial. Además, se cuenta con un desarrollo previo de agente recomendador de turismo, modelado también como agente g-BDI, en este mismo lenguaje de programación [4, 7]. SWI-Prolog es una implementación *open source* del lenguaje Prolog. Cuenta con un conjunto rico de características, algunas de estas son:

- ofrece soporte multihilo (*multithread*),
- cuenta con XPCE que es una plataforma independiente, es una *GUI toolkit*,
- es portable para muchas plataformas, incluyendo entre ellas Unix/Linux, Windows (NT/2000/XP/Vista), MacOS X, y otras,
- contiene abundante documentación.

5.2. Agente Interfaz

El Agente Interfaz (Agente-I) se encarga de capturar los datos ingresados por el usuario, como algunos datos personales, sus preferencias y restricciones sobre los objetos que desea buscar. Este agente en la arquitectura presentada en la Sección 4.1, le envía las restricciones y preferencias al Agente Perfil de Usuario y la estrategia de búsqueda a los Agentes Buscadores, pero en la implementación del prototipo en cuestión, se considera que

¹<http://www.swi-prolog.org/>

la búsqueda ya fue realizada. De esta manera los OA que se encuentran en el repositorio local cumplen con la temática y alguna restricción. Por esta razón, en la implementación la comunicación se lleva a cabo directamente entre el Agente Interfaz y el Agente Recomendador, por lo tanto el Agente-I se comunica con este último enviándole los datos ingresados por el usuario.

La interfaz gráfica de la Figura 5.1 se desarrolló utilizando la herramienta XPCE que contiene SWI-Prolog.

Figura 5.1: Interfaz

Los datos que se le preguntan al usuario se pueden ver más en detalle en la Tabla 5.1. En esta tabla se muestran para cada dato, los valores posibles y una descripción del mismo.

Todas las preferencias, exceptuando la Temática, tienen un valor de prioridad, que el usuario puede especificar para darle un valor de importancia con respecto a las demás que fueron elegidas. A la restricción *Duración máxima* también se considera darle un valor de prioridad.

Cuando el Agente-R determina la lista de OA a recomendar, se la envía a este agente, junto con las descripciones, títulos, y enlace de localización, correspondientes a cada uno de los objetos de la lista, siempre y cuando posean esas características. Por lo tanto, a través de la interfaz, el usuario puede ver la lista recomendada resultante y elegir ver para

Datos del Usuario	Valores	Descripción
Nombre	Cualquier cadena de caracteres	Identificación del usuario
Lengua materna	español, inglés, francés	
Preferencias	Valores	Descripción
Temática	Java	Temática que desea que trate el objeto
Idioma	español, inglés, francés	Idiomas que el usuario desea que presenten los documentos.
Rol	estudiante, docente	Si el que realiza la búsqueda lo hace como estudiante o docente.
Interacción	alta, baja, media	Grado de interacción que el usuario desea tener con el objeto
Estilo de aprendizaje	práctico, teórico, mixto	Si prefiere que el objeto satisfaga más algunas de esas características de aprendizaje
Audiencia		
Contexto académico	primario, secundario, terciario/universitario	Nivel educativo en que el usuario desea el material.
Nivel de conocimiento	inicial, medio, avanzado	Nivel de dificultad que el usuario desea sobre el tema.
Restricción	Valores	Descripción
Duración máxima	Número entero	Tiempo máximo que el usuario espera que le tome trabajar con el objeto.
Costo	si, no	Si restringe la búsqueda a los recursos que no sean pagos o permite los que tienen costo.

Tabla 5.1: Datos que puede ingresar el usuario.

cada uno de ellos el detalle, pulsando el botón *Detalles*. En la Figura 5.2, se puede ver, el título, la descripción y el enlace de localización del objeto elegido. El usuario puede entonces dar su opinión de acuerdo a lo que se detalla en la próxima sección, la cual servirá para evaluar la respuesta del sistema.

5.2.1. Feedback del usuario

Como se dijo anteriormente una vez que el sistema le recomienda al usuario una lista de objetos (un ranking), éste puede expresar su opinión a través de la interfaz, eligiendo alguna de las siguientes opciones:

- **Correcto.** Cuando el usuario es satisfecho con la lista de objetos recomendada.
- **Orden diferente.** Cuando el usuario está conforme con los objetos pero el orden no coincide con su preferencia. El usuario puede indicar cuáles son los tres primeros mejores.

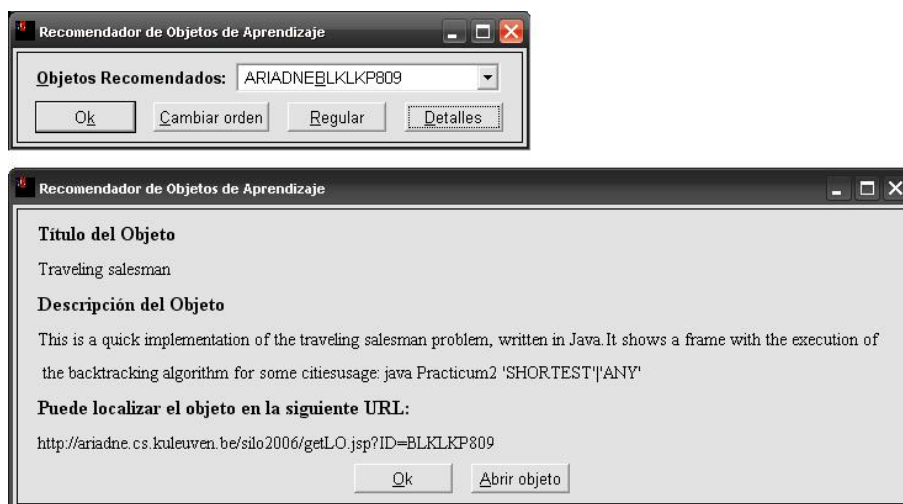


Figura 5.2: Resultado

- ***Incorrecto.*** Cuando el usuario no está satisfecho con la lista de objetos recomendada. En este caso el usuario puede escribir su opinión, es decir un comentario, a través de la interfaz.

5.3. Agente Recomendador

Este agente en base a las restricciones y preferencias (deseos del Contexto Desire) enviadas por el Agente-I, y considerando las reglas pertenecientes al Contexto Belief, calcula a través de una regla puente el grado de intención para cada objeto depositado en el repositorio local, y de esta manera lo ordena decrecientemente según este grado, luego recomienda esta lista de objetos al usuario, enviándola al Agente-I.

5.3.1. Contexto Desire

En este contexto se representan las preferencias y restricciones del usuario. Estos datos, descritos en la Tabla 5.1, como se explicó en la Sección 5.2 se ingresan a través de la interfaz. Las preferencias y restricciones son los deseos positivos y negativos, respectivamente, del agente. Como mencionamos en la Sección 4.2, el usuario puede expresar estas preferencias en diferentes grados, a través de las prioridades. Estos valores pertenecen al conjunto $\{1, \dots, 10\}$, luego se normalizan obteniendo el rango $(0,1]$. Como ejemplo se puede observar en la Figura 5.1 que el usuario, Juan, tiene como preferencia que el idioma del objeto sea español con prioridad 10, inglés con prioridad 8 y con un menor grado de preferencia en idioma francés. Por otro lado ingresa que desea que la interacción con el objeto sea baja, con grado 8. En cuanto a restricciones, Juan indica con prioridad 8, que no desea aquellos objetos cuyo tiempo para trabajar con éstos sea mayor a 60 minutos. Formalmente se representa esto de la siguiente forma: $D^+(\text{idioma, español, 1})$, $D^+(\text{idioma, inglés, 0.8})$, $D^+(\text{idioma, francés, 0.6})$, $D^+(\text{interacción, baja, 0.8})$, $D^-(\text{duración máxima, 60, 0.8})$.

5.3.2. Contexto Belief

El Contexto Belief modela la información del Agente-R sobre el entorno educativo. Las características de los OA que son parte de las creencias del agente, son descritas por los metadatos que consideramos siguen el estándar LOM y están escritos en lenguaje XML, un ejemplo de descripción de un OA en XML sería:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <lom xmlns="http://ltsc.ieee.org/xsd/LOM" xmlns:xsi=...>
  - <general>
    - <identifier>
      <catalog>ARIADNE</catalog>
      <entry>LEA0251</entry>
    </identifier>
    <language>en</language>
    - <aggregationLevel>
      <source>LOMv1.0</source>
      <value>2</value>
    </aggregationLevel>
    - <title>
      <string>JAVA AS A SOLUTION TO IT PLANNING AND CONTROLLING</string>
    </title>
  </general>
  ...
  ...
  ...
- <educational>
  - <interactivityType>
    <source>LOMv1.0</source>
    <value>expositive</value>
  </interactivityType>
  - <learningResourceType>
    <source>LOMv1.0</source>
    <value>slide</value>
  </learningResourceType>
  - <interactivityLevel>
    <source>LOMv1.0</source>
    <value>medium</value>
  </interactivityLevel>
  - <semanticDensity>
    <source>LOMv1.0</source>
    <value>medium</value>
  </semanticDensity>
  - <typicalLearningTime>
    <duration>PT1H</duration>
  </typicalLearningTime>
```

```

- <intendedEndUserRole>
  <source>LOMv1.0</source>
  <value>learner</value>
</intendedEndUserRole>
- <difficulty>
  <source>LOMv1.0</source>
  <value>medium</value>
</difficulty>
...
...

```

Estos metadatos se dividen en nueve categorías, como se explicó en la Sección 2.3.1. Dentro de estos metadatos especificamos en la Tabla 5.2 aquellos que describen las características de los OA que consideramos más relevantes para el recomendador, para más detalle ver [12].

Estos metadatos son parseados con el fin de convertirlos en términos del lenguaje Prolog. Parte del código de esta función la podemos ver a continuación

```

parsearC(R,L,C):-
    chequeaCam(L,C),
    parsear(R,L,C),!.

parsearC(R,_,_):-
    R=[unknown].

parseo(Term,ListIdV,ListIdN):-
    %metadatos necesarios
    parsearC(IdC,Term,[lom,general,identifier,catalog]),
    parsearC(IdE,Term,[lom,general,identifier,entry]),
    parsearC(Lan0,Term,[lom,general,language]),
    parsearC(Desc,Term,[lom,general,description,string]),
    parsearC(Title,Term,[lom,general,title,string]),
    parsearC(Loc,Term,[lom,technical,location]),
    parsearC(Type0,Term,[lom,educational,learningResourceType,value]),
    parsearC(LevelIn,Term,[lom,educational,interactivityLevel,value]),
    parsearC(Role,Term,[lom,educational,intendedEndUserRole,value]),
    parsearC(Con,Term,[lom,educational,context,value]),
    parsearC(Dif,Term,[lom,educational,difficulty,value]),
    parsearC(TimeL,Term,[lom,educational,typicalLearningTime,duration]),
    nth0(0,TimeL,Time),
    append(IdC,IdE,IdL),
    concat_atom(IdL,Id),

    assert(metadato_lan(id(Id),language0(Lan0))),
    assert(metadato_desc(id(Id),des(Desc))),

```

```
assert(metadata_title(id(Id),title(Title))),
assert(metadata_loc(id(Id),loc(Loc))),
assert(metadata_resource(id(Id),resourceType(Type0))),
assert(metadata_inter(id(Id),interLevel(LevelIn))),
assert(metadata_time(id(Id),time(Time))),
assert(metadata_role(id(Id),role(Role))),
assert(metadata_con_dif(id(Id),context(Con),difficulty(Dif))),
...
...
```

N°	Nombre	Explicación	Espacio de valores
1.1.1	General.Identifier.Catalog	El nombre de la identificación o esquema del catálogo para este OA.	Repertorio de ISO/IEC 10646-1:2000
1.1.2	General.Identifier.Entry	El identificador del OA.	Repertorio de ISO/IEC 10646-1:2000
1.2	General.Title	Nombre dado al OA.	-
1.3	General.Language	Lenguajes usados dentro del OA para comunicarse con el usuario.	LanguageID=Langcode (“-”Subcode)* también puede tener el valor “none”
1.4	General.Description	Descripción textual del contenido del OA.	-
1.5	General.Keyword	Palabra clave o frase que describa el tópico del OA.	-
4.3	Technical.Location	Un string que es usado para acceder al OA.	Repertorio de ISO/IEC 10646-1:2000
5.2	Educational.Learning Resource Type	Especifica el tipo de OA. En caso de tener más de uno, el más dominante será el primero.	exercise, simulation, questionnaire, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self assessment, lecture
5.3	Educational.Interactivity Level	El grado de interactividad que caracteriza a este OA. En este contexto se refiere al grado en el que el estudiante puede influir en el aspecto o comportamiento del OA.	very low, low, medium, high, very high
5.5	Educational.Intended End User Role	Usuario principal para el cual está destinado el OA.	teacher, author, learner, manager
5.6	Educational.Context	El principal entorno donde se desea que el aprendizaje y uso de este OA tome lugar.	school, higher education, training, other
5.8	Educational.Difficulty	Cuán dificultoso es trabajar con y a través de este OA, para la audiencia target.	very easy, easy, medium, difficult, very difficult
5.9	Educational.Typical Learning Time	Tiempo típico o estimado que toma trabajar con y a través de este OA, para la audiencia target.	-
6.1	Rights.Cost	Si el uso de este OA requiere pago.	-

Tabla 5.2: Metadatos Seleccionados.

Con estos datos del objeto, el agente aplicará las reglas para inferir su grado de creencia b_{ij} en que un objeto O_i satisfaga las distintas preferencias p_j del usuario, representado por la fórmula $B(O_i, p_j, b_{ij})$. En la siguiente sección detallaremos las reglas, utilizadas en esta implementación.

5.3.3. Reglas

Las reglas utilizadas para hallar los grados de creencia del agente para cada objeto, las determinamos considerando aspectos pedagógicos y en base a esto, utilizamos funciones para determinar estos grados. Para poder estimar en qué medida las características de un OA satisface las preferencias, hemos seleccionado qué metadatos se relacionan con las preferencias, basándonos en el significado de éstos. Según la naturaleza de las distintas preferencias y sus valores posibles se han considerado distintos conjuntos de reglas. Estas podrán ajustarse con la experimentación del sistema.

Preferencia Interacción

Para la preferencia **Interacción**, el metadato asociado es **InteractivityLevel**. Los valores de la preferencia son *baja*, *media*, *alta*. Se contempla el caso en que el usuario no haya ingresado ningún valor (*ignorar*). Los valores del metadato son *very low*, *low*, *medium*, *high*, *very high*. Se contempla el caso en que no exista el valor (*unknown*). En caso de que el usuario elija una preferencia particular y que el metadato correspondiente contenga un valor determinado, se han determinado los valores de creencia considerando lo siguiente:

- En caso de que el valor de la preferencia sea *baja* y la interacción del objeto es *very low* o *low*, el valor de creencia es el valor máximo (1) y a medida que la interacción del objeto es mayor los valores decrecen, hasta el valor mínimo, 0.1.
- En caso de que el valor de la preferencia sea *media* y la interacción del objeto es *medium*, el valor de creencia es el valor máximo (1) y a medida que la interacción del objeto es mayor los valores decrecen y de igual manera si la interacción es menor, decrecen hasta el valor mínimo, 0.1.
- En caso de que el valor de la preferencia sea *alta* y la interacción del objeto es *very high* o *high*, el valor de creencia es el valor máximo (1) y a medida que la interacción del objeto es menor los valores decrecen, hasta el valor mínimo, 0.1.
- En caso de que el valor del metadato no esté cargado (*unknown*), el valor de creencia es 0.5 si el usuario elige como preferencia interacción *baja* o *alta*, y 0.7 si elige interacción *media*. Son valores cercanos a suponer que el valor del metadato es *medium*.

Estos valores se pueden ver en la Tabla 5.3.

A continuación se muestran algunas reglas donde se representan estas relaciones:

(R1) SI $\text{InteractivityLevel}(O_i) = \text{'very low'}$ Entonces $B(O_i, \text{interacción} = \text{baja}, 1)$

Preferencia \ Metadato	<i>very low</i>	<i>low</i>	<i>medium</i>	<i>high</i>	<i>very high</i>	<i>unknown</i>
<i>baja</i>	1	1	0.6	0.2	0.1	0.5
<i>media</i>	0.4	0.6	1	0.6	0.4	0.7
<i>alta</i>	0.1	0.2	0.6	1	1	0.5

Tabla 5.3: Valores de creencia para Interacción.

(R2) SI $\text{InteractivityLevel}(O_i) = \text{'high'}$ Entonces $B(O_i, \text{interacción} = \text{baja}, 0.2)$

(R3) SI $\text{InteractivityLevel}(O_i) = \text{'very low'}$ Entonces $B(O_i, \text{interacción} = \text{alta}, 0.1)$

Por ejemplo, la regla R3 representa: “si el grado de interactividad (representado por el metadato `InteractivityLevel`) del O_i es muy bajo (`'very low'`), entonces la creencia del agente respecto a que ese objeto va a satisfacer la preferencia del usuario $\text{interacción} = \text{alta}$ es muy baja (0.1)”.

Estas reglas están codificadas, respectivamente, en el siguiente fragmento de código:

```
bel_inter(id(Id),interaccion(baja,_),B):-
    metadato_inter(id(Id),interLevel(LevelIn)),
    member('very low',LevelIn), B=1.
```

```
bel_inter(id(Id),interaccion(baja,_),B):-
    metadato_inter(id(Id),interLevel(LevelIn)),
    member(high,LevelIn), B=0.2.
```

```
bel_inter(id(Id),interaccion(alta,_),B):-
    metadato_inter(id(Id),interLevel(LevelIn)),
    member('very low',LevelIn), B=0.1.
```

Preferencia Rol

Para la preferencia **Rol**, el metadato correspondiente es el **IntendedEndUserRole**. Los valores de la preferencia son *estudiante*, *docente*. Se contempla el caso en que el usuario no haya ingresado ningún valor (*ignorar*). Los valores del metadato son *teacher*, *author*, *learner*, *manager*. Se contempla el caso en que no exista el valor (*unknown*). Puede tener más de un valor el metadato, es decir una lista con más de un elemento, y ordenada. Para especificar los valores de creencia, se han tomado en cuenta las posiciones de los valores del metadato en la lista (L) y las siguientes consideraciones:

- Para el valor de la preferencia, *estudiante*, el rol del objeto coincidente es *learner*, luego el más semejante es *teacher*, y luego en menor medida *author* y *manager* de la misma manera.
- Para el valor de la preferencia, *docente*, el rol del objeto coincidente es *teacher*, luego el más semejante es *learner* al igual que *author*, y luego *manager* en menor medida.

- Si el metadato no está cargado y el usuario eligió *learner* como rol, el valor de creencia es 0.8, en cambio si eligió *teacher*, el valor de creencia es 0.2. Estos valores se estimaron teniendo en cuenta la tendencia de los valores del metadato IntendedEndUserRole tomados de los distintos repositorios de OA analizados.

La Tabla 5.4 detalla las relaciones establecidas.

estudiante	[learner,...]	1
	[a,learner,...] and $a \in \{\text{teacher, author, manager}\}$	0.8
	[a,b,learner,...] and $a,b \in \{\text{teacher, author, manager}\}$	0.6
	[a,b,c,learner,...] and $a,b,c \in \{\text{teacher, author, manager}\}$	0.4
	[teacher,...] and $\text{learner} \notin L$	0.3
	([author,...] or [manager,...]) and $\text{learner} \notin L$	0.1
	[unknown]	0.8
docente	[teacher,...]	1
	[a,teacher,...] and $a \in \{\text{learner, author, manager}\}$	0.8
	[a,b,teacher,...] and $a,b \in \{\text{learner, author, manager}\}$	0.6
	[a,b,c,teacher,...] and $a,b,c \in \{\text{learner, author, manager}\}$	0.4
	([learner,...] or [author,...]) and $\text{teacher} \notin L$	0.3
	[manager,...] and $\text{teacher} \notin L$	0.1
	[unknown]	0.2

Tabla 5.4: Valores de creencia para Rol.

Ejemplos de algunas reglas son:

SI $\text{IntendedEndUserRole}(O_i)=[\text{learner},\dots]$ Entonces $B(O_i,\text{rol}=\text{estudiante}, 1)$

SI $\text{IntendedEndUserRole}(O_i)=[a,b,\text{learner},\dots]$
and $a,b \in \{\text{teacher, author, manager}\}$ Entonces $B(O_i,\text{rol}=\text{estudiante}, 0.6)$

SI $\text{IntendedEndUserRole}(O_i)=[a,b,c,\text{teacher},\dots]$
and $a,b,c \in \{\text{learner, author, manager}\}$ Entonces $B(O_i,\text{rol}=\text{docente}, 0.4)$

SI $\text{IntendedEndUserRole}(O_i)=[\text{author},\dots]$ and $\text{teacher} \notin L$
Entonces $B(O_i,\text{rol}=\text{docente}, 0.3)$

El código de estas reglas, es el siguiente:

```
bel_rol(id(Id),rol(estudiante,_),B):-
    metadato_role(id(Id),role(Role)),
    member(learner,Role), nth0(0,Role,learner), B=1.
```

```
bel_rol(id(Id),rol(estudiante,_),B):-
    metadato_role(id(Id),role(Role)),
    member(learner,Role), nth0(2,Role,learner), B=0.6.
```

```
bel_rol(id(Id),rol(docente,_),B):-
```

```

metadato_role(id(Id),role(Role)),
member(teacher,Role), nth0(3,Role,teacher), B=0.4.

```

```

bel_rol(id(Id),rol(docente,_),B):-
  metadato_role(id(Id),role(Role)),
  not(member(teacher,Role)),
  (nth0(0,Role,learner) ; nth0(0,Role,author)), B=0.3.

```

Preferencias Contexto académico y Nivel de conocimiento

Consideramos que las preferencias **Contexto académico** y **Nivel de conocimiento**, especifican el tipo de audiencia para la que se busca el objeto. Cada una es descripta por un metadato, **Context** y **Difficulty**, respectivamente. Los valores de la preferencia **Contexto académico** son *primario*, *secundario*, *terciario/universitario*. Los valores de la preferencia **Nivel de conocimiento** son *inicial*, *medio*, *avanzado*. Los valores del metadato **Context** son *school*, *higher education*, *training*, *other*. Los valores del metadato **Difficulty** son *very easy*, *easy*, *medium*, *difficult*, *very difficult*.

Para determinar el grado de creencia con que un objeto satisface ambas preferencias teniendo en cuenta las características Context y Difficulty, hemos considerado la distancia entre dos funciones, f y g , que especifican respectivamente, el par de preferencias y el par de metadatos. Siendo

$$f : Contexto \times Nivel \longrightarrow \mathbb{N} \text{ y } g : Context \times Difficulty \longrightarrow \mathbb{N}$$

En la Tabla 5.5 podemos ver la definición de las funciones f y g .

f	g
$f(\text{primario}, \text{inicial}) = 1$	$g(\text{school}, \text{very easy}) = 3$
$f(\text{primario}, \text{medio}) = 2$	$g(\text{school}, \text{easy}) = 4$
$f(\text{primario}, \text{avanzado}) = 3$	$g(\text{school}, \text{medium}) = 5$
$f(\text{secundario}, \text{inicial}) = 4$	$g(\text{school}, \text{difficult}) = 6$
$f(\text{secundario}, \text{medio}) = 5$	$g(x) = 7$, $x = (\text{school}, \text{very difficult})$ or $x = (\text{higher education}, \text{very easy})$ or $x = (\text{higher education}, \text{easy})$
$f(\text{secundario}, \text{avanzado}) = 6$	$g(\text{higher education}, \text{medium}) = 8$
$f(\text{terciario/universitario}, \text{inicial}) = 7$	$g(x) = 9$, $x = (\text{higher education}, \text{difficult})$ or $x = (\text{higher education}, \text{very difficult})$ or $x = (\text{training}, \text{very easy})$
$f(\text{terciario/universitario}, \text{medio}) = 8$	$g(\text{training}, \text{easy}) = 10$
$f(\text{terciario/universitario}, \text{avanzado}) = 9$	$g(\text{training}, \text{medium}) = 11$
	$g(\text{training}, \text{difficult}) = 12$
	$g(\text{training}, \text{very difficult}) = 13$

Tabla 5.5: Definición de las funciones f y g

Luego se define la distancia h entre los pares como

$$h((c_1, n_1), (c_2, d_2)) = |g(c_2, d_2) - f(c_1, n_1)|$$

Teniendo esta función h podemos determinar el valor de creencia para las preferencias que haya elegido el usuario, cuyo par valor pertenece al Dominio de f . Considerando que la similitud es inversamente proporcional a la distancia. A continuación podemos ver el grado de creencia (B), dependiendo el valor de h aplicado a un par (c_1, n_1) de preferencias, y al par de metadatos (c_2, d_2) .

$$h((c_1, n_1), (c_2, d_2)) = \begin{cases} 0 & \text{entonces } B = 1 \\ 1 & \text{entonces } B = 0.8 \\ 2 & \text{entonces } B = 0.6 \\ 3 & \text{entonces } B = 0.4 \\ 4 & \text{entonces } B = 0.2 \\ > 4 & \text{entonces } B = 0.1 \end{cases}$$

Si el metadato Context no está cargado, el valor de creencia es 0.5 (el promedio de los valores anteriores). Este mismo valor es considerado si el metadato Context tiene como valor *other*. Por otro lado, si el usuario ingresa un contexto, el valor por defecto del nivel es medio, en caso contrario elige otro. Si el usuario no ingresa un contexto, no se considera el valor de nivel que haya elegido.

Con todo lo anteriormente dicho podemos formular las siguientes reglas:

SI Context(O_i)='higher education' and Difficulty(O_i)='difficult' Entonces
B(O_i , (contexto=terciario/universitario) and (nivel=inicial), 0.6)

SI Context(O_i)='higher education' and Difficulty(O_i)='easy' Entonces
B(O_i , (contexto=terciario/universitario) and (nivel=inicial), 1)

SI Context(O_i)='school' and Difficulty(O_i)='difficult' Entonces
B(O_i , (contexto=terciario/universitario) and (nivel=inicial), 0.8)

SI Context(O_i)='school' and Difficulty(O_i)='medium' Entonces
B(O_i , (contexto=secundario) and (nivel=avanzado), 0.8)

Parte del código que calcula estas reglas podemos verlo debajo:

```
f(X,Y):- ((X=primarioinicial, Y=1);
          (X=primariomedio, Y=2);
          (X=primarioavanzado, Y=3);
          (X=secundarioinicial, Y=4);
          (X=secundariomedio, Y=5);
          (X=secundarioavanzado, Y=6);
          (X='terciario Universitarioinicial', Y=7);
          (X='terciario Universitariomedio', Y=8);
          (X='terciario Universitarioavanzado', Y=9)).
```

```
g(X,Y):- ((X='schoolvery easy', Y=3);
          (X=schooleasy, Y=4);
```

```

(X=schoolmedium, Y=5);
(X=schooldifficult, Y=6);
((X='schoolvery difficult';
  X='higher educationvery easy';
  X='higher educationeasy') , Y=7);
(X='higher educationmedium', Y=8);
((X='higher educationdifficult';
  X='higher educationvery difficult';
  X='trainingvery easy'), Y=9);
(X=trainingeasy, Y=10);
(X=trainingmedium, Y=11);
(X=trainingdifficult, Y=12);
(X='trainingvery difficult', Y=13)).

```

```
h(X1,X2,Y):- f(X1,Y1),g(X2,Y2), Y is abs(Y2-Y1).
```

```

simil(X,GS):-((X=0, GS=1);
              (X=1, GS=0.8);
              (X=2, GS=0.6);
              (X=3, GS=0.4);
              (X=4, GS=0.2);
              (X>4, GS=0.1)).

```

```

calcular_similar(C,N,MC,MN,GS):- atom_concat(C,N,CN),
                                atom_concat(MC,MN,MCMN),
                                h(CN,MCMN,Y), simil(Y,GS),!.

```

```

bel_con_niv(id(Id),contexto(C),nivel(N),B):-
  C\=ignorar,
  metadato_con_dif(id(Id),context(Con),difficulty(Dif)),
  not(member(unknown,Con)),not(member(unknown,Dif)),
  nth0(0,Con,MC), nth0(0,Dif,MD),
  calcular_similar(C,N,MC,MD,B).

```

Cuando el metadato Context está cargado pero no el metadato Difficulty, en tal caso para determinar un valor de creencia se tiene en cuenta el valor del metadato Context y el valor que el usuario ingresa como contexto académico. Es decir:

- Si el contexto académico es *primario* entonces, si el valor de Context es *school*, el grado de creencia será el promedio de los valores 1, 0.8, 0.6, 0.4, 0.2 y 0.1, es decir, 0.5. Estos 6 valores son los grados de creencia posibles en caso de que se conociera el valor de Difficulty. En cambio si el valor de Context es *higher education*, el grado de creencia será 0.2, el promedio de los valores 0.2, y 0.1. Por último, si el valor de Context es *training*, el grado de creencia será 0.1.
- En los casos en que el contexto académico es *secundario* o *terciario/universitario* se realiza un procedimiento similar al anterior descripto.

Preferencia Idioma

Para la Preferencia **Idioma**, el metadato analizado es **Language** (General). Los valores de la preferencia son *español*, *inglés*, *francés*. Los valores del metadato se pueden referir a cualquier lenguaje en un formato específico; si el objeto no tiene ningún lenguaje el valor del metadato es *none*, y si no está cargado es *unknown*. Contamos con el valor de lengua materna (*español*, *inglés* o *francés*). Por otro lado, un objeto puede tener más de un lenguaje.

Para obtener el grado de creencia (B) de que un objeto satisface el o los idiomas elegidos por el usuario se utilizarán las siguientes reglas:

- Si el usuario elige uno o más idiomas y al menos uno de ellos pertenece a Language entonces B tendrá valor 1 y el valor de prioridad utilizado para calcular el grado de intención será igual al máximo de prioridades de cada uno de los idiomas que fue elegido y pertenece a Language.
- Si el usuario elige uno o más idiomas y ninguno pertenece a Language y *none* tampoco pertenece, entonces
 - Si la lengua materna pertenece a Language entonces B tomará el valor 1 y el valor de prioridad utilizado para calcular el grado de intención será el valor 0.6.
 - Si la lengua materna no pertenece a Language entonces ese OA se descarta, no se tiene en cuenta para la recomendación final.
- Si *none* pertenece a Language, entonces B tendrá el valor 1 y el valor de prioridad utilizado para calcular el grado de intención será igual al máximo de prioridades correspondientes a los idiomas que haya elegido el usuario.

El código de estas reglas es el siguiente:

```
bel_materna(Lan0,R):-
    lengua(Lengua), member(X,Lan0) ,
    ((Lengua=español , atom_prefix(X,es));
    (Lengua=inglés , atom_prefix(X,en)) ;
    (Lengua=francés , atom_prefix(X,fr))),
    R is 0.6*1.

bel_materna(Lan0,R):-
    lengua(Lengua), member(X,Lan0) ,
    not((Lengua=español , atom_prefix(X,es));
    (Lengua=inglés , atom_prefix(X,en)) ;
    (Lengua=francés , atom_prefix(X,fr))),
    R=0.

bel_ides(Lan0,Ye,Re):-
    member(X,Lan0), atom_prefix(X,es), Re=Ye.
```

```

bel_ides(Lan0,_,Re):-
    member(X,Lan0),not(atom_prefix(X,es)),
    Re=0.

bel_iden(Lan0,Yi,Ri):-
    member(X,Lan0), atom_prefix(X,en), Ri=Yi.

bel_iden(Lan0,_,Ri):-
    member(X,Lan0),not(atom_prefix(X,en)),
    Ri=0.

bel_idfr(Lan0,Yf,Rf):-
    member(X,Lan0), atom_prefix(X,fr), Rf=Yf.

bel_idfr(Lan0,_,Rf):-
    member(X,Lan0),not(atom_prefix(X,fr)),
    Rf=0.

maximo(X,Y,Z,Max):-
    M1 is max(X,Y),
    Max is max(M1,Z).

bel_idioma(id(Id),idioma(inglés,Yi),idioma(español,Ye),
    idioma(francés,Yf),BD):-
    metadato_lan(id(Id),language0(Lan0)),
    (not(member(none,Lan0))->
    bel_ides(Lan0,Ye,Re),
    bel_iden(Lan0,Yi,Ri),
    bel_idfr(Lan0,Yf,Rf),
    maximo(Re,Ri,Rf,Max),
    (Max\=0 ->
    BD = Max
    ;
    bel_materna(Lan0,Rm),
    BD=Rm)
    ;
    maximo(Ye,Yi,Yf,MY),
    BD=MY).

```

Preferencias Estilo de aprendizaje

Para la preferencia **Estilo de aprendizaje**, el metadato asociado es **LearningResourceType**. Los valores de la preferencia son *teórico, práctico, mixto*.

Los valores del metadato son *exercise, simulation, questionnaire, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self assessment, lecture*.

Dentro del conjunto de valores de metadatos, se consideran dos subconjuntos, uno con valores que se pueden clasificar “prácticos”, denominado P , y otro con valores que se pueden clasificar “teóricos”, llamado T .

$P = \{exercise, simulation, questionnaire, exam, experiment, problem statement, self assessment\}$

$T = \{diagram, figure, graph, index, table, slide, narrative text, lecture\}$

- Si el usuario elige “práctico” realizamos los siguientes cálculos, depende del caso, para determinar el grado de creencia (B).

Dada la lista de metadatos $[a, b, \dots, c]$ y N el tamaño de la misma,

si $N = 1$ ($[a]$) and $a \neq unknown$	entonces $B = i/N$
si $N = 1$ ($[a]$) and $a = unknown$	entonces $B = \#P/(\#P + \#T)$
si $N = 2$ ($[a, b]$)	entonces $B = (3i + 2j)/(N + 3)$
si $N > 2$ ($[a, b, \dots, c]$)	entonces $B = (3i + 2j + \sum_{k=3}^N z)/(N + 3)$

donde $i = 1$ si $a \in P$ sino $i = 0$, $j = 1$ si $b \in P$ sino $j = 0$ y $z = 1$ si el elemento k -ésimo $\in P$ sino $z = 0$.

- Para el estilo “teórico” se realiza el mismo procedimiento pero se chequea si los valores del metadato pertenecen al conjunto T en vez de P , y cuando se tiene el caso

$N = 1$ ($[a]$) and $a = unknown$ entonces $B = \#T/(\#P + \#T)$

- En el caso que el usuario elija “mixto” realizamos el siguiente cálculo:

$$B = (B_p \times B_t) \times 4$$

donde B_p y B_t son las creencias respectivamente de pensar cuán práctico es ese objeto y cuán teórico es ese objeto.

Se utilizan las constantes 3 y 2 como peso para los primeros valores por sus posiciones dentro de la lista de metadatos, dado que tiene importancia el orden. Ya el resto de los valores no tienen peso (sería la constante 1).

Ejemplos de algunas reglas son:

SI LearningResourceType(O_i)=[exercise,lecture] Entonces $B(O_i, \text{estilo}=\text{práctico}, 0.6)$

SI LearningResourceType(O_i)=[exercise, narrative text, slide]
Entonces $B(O_i, \text{estilo}=\text{práctico}, 0.5)$

SI LearningResourceType(O_i)=[exercise, narrative text, slide]
Entonces $B(O_i, \text{estilo}=\text{mixto}, 1)$

SI LearningResourceType(O_i)=[slide] Entonces $B(O_i, \text{estilo}=\text{teórico}, 1)$

Parte del código que calcula estas reglas podemos verlo a continuación:

```

pertenece(práctico, Valor, Res):-
    Cat_Pract = [exercise,simulation,questionnaire,exam,experiment,
                'problem satement','self assessment'],
    member(Valor,Cat_Pract), Res=1.

pertenece(teórico, Valor, Res):-
    Cat_Teor = [diagram,figure,graph,index,table,slide,
                'narrative text',lecture],
    member(Valor,Cat_Teor), Res=1.

calcular(E,L,B,2,Con,Con2):-
    nth0(0,L,V1), pertenece(E,V1,R1),
    nth0(1,L,V2), pertenece(E,V2,R2),
    B is ((Con*R1 + Con2*R2)/(2+Con)).

bel_estilo(id(Id),estilo(E,_),B):-
    E\=mixto,
    metadato_resource(id(Id),resourceType(Type0)),
    length(Type0,T),not(member(unknown,Type0)),
    calcular(E,Type0,B,T,3,2).

```

Restricción Duración máxima

El Agente-R determina qué objetos satisfacen la restricción Duración máxima, es decir si el usuario ingresa una duración como límite máximo, es porque desea trabajar con objetos que le lleven un tiempo menor o igual a este límite. Para aquellos objetos que no satisfacen esta restricción, el agente calcula un valor (un peso negativo) que depende de la prioridad de aquella y de la diferencia entre el tiempo del objeto con el límite ingresado por el usuario. Para la restricción **Duración máxima** el metadato analizado es **TypicalLearningTime**. El valor que ingresa el usuario como restricción es un número entero, que representa el tiempo en minutos. El valor del metadato es del tipo de dato Duration, ver en detalle en [12], este valor se decodifica a un número entero, que representa el mismo tiempo pero en minutos. De esta manera, por cada objeto, se compara el tiempo ingresado por el usuario MU , y el tiempo del objeto MO , de la siguiente manera:

Si $MO \leq MU + P$ la restricción se cumple

Si $MO > MU + P$ la restricción no se cumple

Donde P es un porcentaje de MU . Se consideró el 20% de MU para realizar la prueba de validación.

Sea $MUP = MU + P$, se define la recta (h) que pasa por los puntos $(MUP,0)$ y $(3MUP,1)$

$$h(x) = \frac{1}{2MUP} \times x - 0.5$$

Por lo tanto, el peso negativo, pn , de un objeto O , con tiempo MO el cual no cumple con la restricción, se determina de la siguiente manera:

$$pn = h(MO) \times Pr$$

Donde Pr es la prioridad que el usuario le asigna a la restricción. Finalmente, al grado de intención correspondiente al objeto O se le restará este peso, es decir se “multa” al objeto.

Y por otro lado, el agente directamente no recomienda aquellos objetos con un tiempo mayor a la cota $3MUP$.

5.3.4. Contexto Intention

El Agente-R luego de calcular los grados de creencia para cada una de las preferencias elegidas por el usuario, calcula el grado de intención para aquellos objetos que forman parte de la lista a recomendar; la regla puente implementada es la que se definió en la ecuación 4.1 (Sección 4.2). Finalmente, el Agente-R ordena los objetos según el valor decreciente del grado obtenido, obteniendo así la lista de objetos de aprendizaje que recomendará al usuario, a través del Agente Interfaz.

5.4. Ejemplo

A continuación se presenta un ejemplo simple para poder ilustrar las diferentes funcionalidades y las interacciones de los agentes en el sistema multiagente.

Supongamos que José es un estudiante de Ingeniería que está buscando documentos con información sobre Matrices dado que está cursando un primer curso de Álgebra. Por otra parte, tiene mucho conocimiento sobre el idioma inglés, y también entiende un poco el idioma francés. José prefiere que el estilo de aprendizaje sea práctico. Además, desearía que el tiempo que le lleve el desarrollo del material no sea superior a 60 minutos, y que el documento a buscar sea gratuito.

Cuando realiza la búsqueda, José proporciona como término de entrada la palabra “*matrices*”. El Refinador Semántico interactúa con el usuario y construye la estrategia de búsqueda asociada. Luego el Agente Perfil de Usuario construye el perfil de José mediante un conjunto de preguntas. A partir de esta interacción, su perfil contiene entre otros datos, la información que se muestra en la Tabla 5.6.

Supongamos que los agentes buscadores en base a la estrategia de búsqueda temática resultante y a las restricciones primarias (Costo=“no”), recuperan de los repositorios un conjunto de cuatro objetos de aprendizaje (O_1, O_2, O_3, O_4) con sus metadatos, y son volcados al repositorio local luego de que el Agente Mediador resuelva los posibles conflictos e integre los datos. En la Tabla 5.7 se considera para cada $O_i, i \in \{1 \dots 4\}$, un conjunto reducido de sus metadatos obtenidos por el Agente-M (en el estándar LOM).

Todos estos objetos al estar en el repositorio local ya satisfacen las necesidades temáticas del usuario y la restricción de no tener costo. Ahora el Agente-R a partir de las restantes restricciones y preferencias va a seleccionar los que sean más adecuados para él, aplicando las reglas que definimos en la Sección 5.3.3. En primer lugar aplica la regla sobre la restricción Duración máxima. Como cada $O_i, i \in \{1 \dots 4\}$ satisface esta condición, y a

Preferencia	Prioridad
Lengua Materna=“español”	
Idioma=“español”	1.0
Idioma=“inglés”	1.0
Idioma=“francés”	0.6
Rol=“estudiante”	1.0
Interacción=“baja”	0.7
Estilo de Aprendizaje=“práctico”	1.0
Audiencia	
Contexto Académico=“terciario/universitario”	1.0
Nivel de Conocimiento=“inicial”	
Restricción	
Duración Máxima=“60”	0.7
Costo=“no”	1.0

Tabla 5.6: Extracto del perfil del usuario correspondiente a José.

Metadatos	O_1	O_2	O_3	O_4
Language	en (inglés)	es (español)	fr (francés)	en
Learning Resource Type	lecture	[exercise, lecture]	slide	exercise
Interactivity Level	low	low	very low	high
Intended End User Role	learner	learner	learner	teacher
Context	higher education	higher education	school	higher education
Difficulty	medium	easy	medium	difficult
Typical Learning Time	40	50	20	50
Cost	no	no	no	no

Tabla 5.7: Metadatos relevantes de los Objetos de Aprendizaje recuperados.

su vez el idioma de cada uno de ellos es elegido por el usuario, entonces los cuatro OA son considerados para el ranking.

A partir de las reglas en el Contexto Belief del Agente-R, este agente calcula el grado de satisfacción esperado de cada una de las preferencias del usuario (por ejemplo, interacción=baja, idioma=español) a través de las distintas características de cada O_i . Ejemplos de algunas reglas que aplica el agente son:

(R1) SI InteractivityLevel(O_i)=‘low’ Entonces B(O_i ,interacción=baja, 1)

(R2) SI InteractivityLevel(O_i)=‘high’ Entonces B(O_i ,interacción=baja, 0.2)

(R3) SI Context(O_i)=‘higher education’ and Difficulty(O_i)=‘medium’
Entonces B(O_i , (contexto=terciario/universitario) and (nivel=inicial), 0.8)

(R4) SI Context(O_i)=‘school’ and Difficulty(O_i)=‘medium’
Entonces B(O_i , (contexto=terciario/universitario) and (nivel=inicial), 0.6)

(R5) SI LearningResourceType(O_i)=[exercise] Entonces B(O_i , estilo=práctico, 1)

(R6) SI LearningResourceType(O_i)=[exercise,lecture] Entonces $B(O_i, \text{estilo}=\text{práctico}, 0.6)$

A partir de estas reglas, el Agente-R infiere su grado de creencia b_{ij} en que un objeto O_i puede satisfacer las distintas preferencias p_j del usuario: $B(O_i, p_j, b_{ij})$. Por ejemplo, $B(O_2, \text{estilo}=\text{práctico}, 0.6)$ representa que el Agente-R cree en grado 0.6 que el objeto O_2 satisface la preferencia de que sea un recurso de estilo práctico.

Luego el Agente-R calcula las intenciones de alcanzar sus objetivos (satisfacer sus preferencias de aprendizaje $P = p_1 \wedge \dots \wedge p_n$) a través de un recurso O_i aplicando la regla puente:

$$I(O_i, P) = \frac{\sum_{j=1}^n d_j \times b_{ij}}{n}$$

Computando estos valores para los cuatro objetos se tienen los siguientes grado de intención:

$$I(O_1, P) = 0.72 \quad I(O_2, P) = 0.86 \quad I(O_3, P) = 0.6 \quad I(O_4, P) = 0.608$$

Finalmente, el Agente-R ordena los objetos según el valor decreciente del grado obtenido: O_2, O_1, O_4, O_3 . Luego el sistema recomienda a José el curso O_2 como el más adecuado y además, le da una lista ordenada de las restantes alternativas.

Capítulo 6

Caso de estudio

Se presenta un caso de estudio utilizando el prototipo del Agente Recomendador, considerando un subconjunto de objetos de aprendizaje del repositorio ARIADNE.

6.1. Repositorio Local

Para probar el recomendador se necesita contar con un repositorio que contenga objetos cuya descripción especifique características educacionales, lo cual permite al recomendador basarse en las preferencias del usuario. ARIADNE es un repositorio donde los metadatos que describen a los objetos están cargados en su mayoría, y cuenta con metadatos de la categoría Educational. De este repositorio se seleccionó un conjunto de objetos referentes a un tema particular; para esta prueba seleccionamos objetos que traten sobre el lenguaje Java. ¿Por qué elegimos este tema?. Una de las razones es porque es un tema conocido para los alumnos y docentes de la carrera Licenciatura en Ciencias de la Computación, los cuales podrían contribuir a este proceso de prueba del prototipo; otra de las razones, es porque se necesitaba contar con objetos de posible acceso y que el idioma de estos sea un idioma de un gran alcance para los usuarios. Teniendo en cuenta estas condiciones, se realizó la búsqueda manual de estos objetos a través del buscador de ARIADNE, teniendo la posibilidad de ver sus metadatos, de descargar el archivo XML con estos metadatos y de acceder a los objetos siempre y cuando la descarga no estuviera restringida. Es decir, se realizó la búsqueda a través del buscador de ARIADNE, se ingresó la palabra “Java” y se encontraron 263 objetos (pertenecientes al repositorio local). Aplicando un refinamiento del tema, buscando aquellos objetos cuyo idioma sea inglés, francés o español y que por supuesto se tenga acceso a estos, y luego de actuar como mediador, seleccionamos 27 objetos, los adecuados para el repositorio local. De estos, según el metadato General.Language, 12 objetos están en el idioma inglés, y 15 objetos en francés, pero ninguno en idioma español. En realidad son muy pocos los objetos en este idioma y pertenecen a temas diferentes. La mayoría de los objetos se encuentran en idioma Holandés.

Los títulos de algunos objetos son los siguientes:

- Traveling salesman (Id₁).
- CLASSE JAVA Transaction bancaire (Id₂).

- INFORMATIQUE I-II : SERIE 5 (Id₃).
- INFORMATIQUE I-II : Corrigeserie 5 (Id₄).
- IIS : CORRIGE SERIE 5 (HTML) (Id₅).
- IIS : Chap 01 - Programmer : un exemple (Id₆).
- IIS : Chap 02 - "Parlons" Java (Id₇).
- IIS : Chap 03 - Expressions booleennes (Id₈).
- CLOO : TP2 (Id₉).
- CLOO : CORRIGE TP2 (Id₁₀).
- JAVA AS A SOLUTION TO IT PLANNING AND CONTROLLING (Id₁₁).
- JAVA ARCHITECTURE AND SECURITY (Id₁₂).

En la lista anterior, se indica entre paréntesis la identificación dada a cada objeto para su tratamiento por el agente recomendador.

6.2. Análisis de resultados

Para probar el recomendador se contó con la ayuda de usuarios con diferentes perfiles y se consideró el feedback otorgado por cada uno para establecer si el Agente-R satisface, en algún grado, los diferentes perfiles. En la Tabla 6.1 podemos ver los objetivos que los distintos usuarios desean alcanzar. El contexto académico deseado para todos los usuarios es el universitario.

Usuario	Idiomas	Rol	Objetivo
Usuario1	Inglés	Estudiante	Aprender las herramientas fundamentales con las que cuenta Java
Usuario2	Inglés, Francés	Estudiante	Aprender las herramientas fundamentales con las que cuenta Java
Usuario3	Inglés, Francés	Estudiante	Practicar ejercicios de Java de un nivel medio
Usuario4	Inglés, Francés	Docente	Preparar las primeras clases sobre Java, para un nivel inicial.
Usuario5	Inglés, Francés	Docente	Preparar unas clases teóricas pero para un nivel avanzado.
Usuario6	Inglés, Francés	Docente	Preparar ejercitación Java para alumnos de un nivel inicial-medio. Tiempo aproximado de trabajo, 90 minutos.

Tabla 6.1: Objetivos de los usuarios.

Para cada uno de los usuarios se detalla, en la Tabla 6.2, el perfil de acuerdo al ingreso por Interfaz. En la Tabla 6.3 se muestra el resultado otorgado por el recomendador,

es decir la lista de identificadores de objetos de aprendizaje, y por último en la Tabla 6.4 se muestra el feedback de cada uno de ellos. Para reducir espacios, la notación que utilizaremos es la siguiente: Prioridad (Pr), terciario/universitario (U), inicial (i), medio (m), avanzado (a), estudiante (est), docente (doc).

Perfil	Usuario1	Usuario2	Usuario3	Usuario4	Usuario5	Usuario6
Lengua materna	español	español	español	español	español	español
Temática	java	java	java	java	java	java
Pr del Idioma español	10	10	10	10	10	10
Pr del Idioma inglés	9	10	10	10	10	10
Pr del Idioma francés	0	9	9	9	9	9
(Rol,Pr)	(est,10)	(est,10)	(est,10)	(doc,8)	(doc,8)	(doc,7)
(Interacción,Pr)	(media,5)	(ignorar,-)	(ignorar,-)	(ignorar,-)	(alta,7)	(ignorar,-)
(Estilo,Pr)	(teórico,9)	(teórico,9)	(práctico,9)	(teórico,10)	(teórico,10)	(práctico,10)
(Contexto,Nivel,Pr)	(U,i,9)	(U,i,10)	(U,m,10)	(U,i,9)	(U,a,9)	(U,i,7)
(Duración máxima,Pr)	(0,-)	(0,-)	(0,-)	(0,-)	(0,-)	(90,9)
Costo	no	no	no	no	no	no

Tabla 6.2: Perfiles de los usuarios.

Dentro del repositorio local contamos con objetos que cumplen la condición de no tener costo, es decir es una restricción que no tiene en cuenta el recomendador. En la Tabla 6.5 podemos ver algunos de los metadatos de los 27 objetos que se encuentran en el repositorio local; son los más relevantes junto con Title, Description y Location.

Para evaluar la cercanía entre el ranking recomendado por el Agente-R del ranking propio del usuario, utilizamos la distancia de Manhattan, la cual es adecuada para capturar distancias entre posiciones. La distancia la calculamos entre las posiciones de los tres primeros objetos ordenados por el usuario con las posiciones correspondientes en la lista recomendada, teniendo en cuenta el grado de intención de cada uno de los objetos en la lista.

Sea $LOR = [OR_1, OR_2, OR_3, \dots, OR_n]$ la lista recomendada al usuario (ranking), $GI = [G_1, G_2, G_3, \dots, G_n]$ la lista que contiene los grados de intención de los OA recomendados, es decir G_i es el grado de intención del objeto OR_i y por último se tiene $LOU = [OU_1, OU_2, OU_3]$ los objetos ordenados por el usuario. Si $OU_1 = OR_i, OU_2 = OR_j, OU_3 = OR_k$ entonces la distancia entre el ranking otorgado por el recomendador y el del usuario es:

$$Dist(LOR, LOU) = |1 - x_1| + |2 - x_2| + |3 - x_3|$$

Donde $x_r = \#A_r + 1$, $r \in \{1, 2, 3\}$, $A_1 = \{O_z | O_z \in LOR, G_z > G_i\}$, $A_2 = \{O_z | O_z \in LOR, G_z > G_j\}$, $A_3 = \{O_z | O_z \in LOR, G_z > G_k\}$.

Se considera peor caso para cada usuario cuando se tiene el siguiente caso:
 $OU_1 = OR_n, OU_2 = OR_{n-1}, OU_3 = OR_{n-2}$.

En la Tabla 6.6, se pueden ver las distancias calculadas sobre los casos satisfactorios.

Usuario1	Usuario2	Usuario3	Usuario4	Usuario5	Usuario6
Id ₁₁	Id ₁₂	Id ₁₄	Id ₁	Id ₁	Id ₂₁
Id ₁₃	Id ₁₁	Id ₁₇	Id ₁₂	Id ₂₂	Id ₅
Id ₁₂	Id ₁₃	Id ₁₈	Id ₁₁	Id ₂₄	Id ₁₄
Id ₁₉	Id ₇	Id ₁₉	Id ₁₃	Id ₂₅	Id ₁₇
Id ₂₇	Id ₆	Id ₂₀	Id ₇	Id ₁₂	Id ₁₈
Id ₂₁	Id ₈	Id ₂₇	Id ₆	Id ₁₁	Id ₁₉
Id ₁	Id ₄	Id ₃	Id ₈	Id ₁₃	Id ₂₀
Id ₅	Id ₁₅	Id ₅	Id ₄	Id ₂₆	Id ₂₇
Id ₁₇	Id ₂₆	Id ₂₁	Id ₁₅	Id ₇	Id ₉
Id ₁₄	Id ₂	Id ₉	Id ₂₆	Id ₆	Id ₁₀
Id ₁₈	Id ₂₂	Id ₁₀	Id ₂	Id ₈	Id ₁₆
Id ₂₀	Id ₂₃	Id ₁₆	Id ₂₂	Id ₄	Id ₃
	Id ₂₄	Id ₁₂	Id ₂₃	Id ₁₅	Id ₁
	Id ₂₅	Id ₁₁	Id ₂₄	Id ₂	Id ₁₂
	Id ₂₁	Id ₂₆	Id ₂₅	Id ₂₃	Id ₁₁
	Id ₅	Id ₁₃	Id ₂₁	Id ₁₄	Id ₁₃
	Id ₁₄	Id ₇	Id ₅	Id ₁₈	Id ₆
	Id ₁₇	Id ₂	Id ₁₄	Id ₂₁	Id ₈
	Id ₁₈	Id ₆	Id ₁₇	Id ₁₉	Id ₄
	Id ₁₉	Id ₈	Id ₁₈	Id ₂₇	Id ₂
	Id ₂₀	Id ₄	Id ₁₉	Id ₅	Id ₂₂
	Id ₂₇	Id ₁₅	Id ₂₀	Id ₁₇	Id ₂₃
	Id ₁	Id ₂₂	Id ₂₇	Id ₃	Id ₂₄
	Id ₃	Id ₂₃	Id ₃	Id ₉	Id ₂₅
	Id ₉	Id ₂₄	Id ₉	Id ₁₀	Id ₁₅
	Id ₁₀	Id ₂₅	Id ₁₀	Id ₁₆	Id ₇
	Id ₁₆	Id ₁	Id ₁₆	Id ₂₀	Id ₂₆

Tabla 6.3: Lista de Objetos Recomendados.

Usuario1	Usuario2	Usuario3	Usuario4	Usuario5	Usuario6
Incorrecto	Id ₇	Id ₉	Id ₇	Id ₂₆	Id ₉
	Id ₆	Id ₁₄	Id ₆	Id ₂₁	Id ₁₄
	Id ₈	Id ₃	Id ₈	Id ₁₂	Id ₁₀

Tabla 6.4: Feedback del usuario.

Algunas evaluaciones a tener en cuenta para algunos usuarios:

- En el caso del Usuario1, el recomendador recupera solamente los que satisfacen el idioma elegido (inglés), por lo tanto devuelve como resultado los 12 objetos cuyo idioma es el inglés, según el metadato **Language**. El usuario en su comentario expresa que la respuesta no lo satisface porque el contenido de los documentos no satisface lo que él estaba buscando. Y destaca que el objeto recuperado en primer lugar, no está escrito en idioma inglés, sino en francés, lo cual dificulta su entendimiento. Esto sucede porque el metadato **Language** del objeto Id₁₁ contiene el valor *en*, pero en verdad está escrito en francés. Luego el problema ha sido la mala

Objetos	Language	Learning Resource Type	Interactivity Level	Intended End User Role	Context	Difficulty	Typical Learning Time (min)
Id ₁	en	narrative text	medium	teacher	higher education	difficult	30
Id ₂	fr	narrative text	very low	learner	higher education	difficult	10
Id ₃	fr	exercise	unknown	learner	higher education	medium	120
Id ₄	fr	narrative text	unknown	learner	higher education	unknown	15
Id ₅	en	exercise	unknown	learner	higher education	unknown	15
Id ₆	fr	slide	unknown	learner	higher education	unknown	90
Id ₇	fr	slide	unknown	learner	higher education	unknown	135
Id ₈	fr	slide	unknown	learner	higher education	unknown	90
Id ₉	fr	exercise	unknown	learner	higher education	unknown	45
Id ₁₀	fr	problem statement	unknown	learner	higher education	unknown	30
Id ₁₁	en	slide	medium	learner	higher education	medium	60
Id ₁₂	en	slide	high	learner	higher education	medium	90
Id ₁₃	en	narrative text	unknown	learner	higher education	unknown	3
Id ₁₄	en	simulation	high	learner	higher education	medium	5
Id ₁₅	fr	narrative text	unknown	learner	higher education	unknown	120
Id ₁₆	fr	problem statement	unknown	learner	higher education	unknown	30
Id ₁₇	en	simulation	unknown	learner	higher education	medium	20
Id ₁₈	en	simulation	high	learner	higher education	medium	10
Id ₁₉	en	simulation	medium	learner	higher education	medium	45
Id ₂₀	en	simulation	low	learner	higher education	medium	30
Id ₂₁	en	simulation	high	learner	higher education	easy	10
Id ₂₂	fr	narrative text	high	learner	higher education	difficult	90
Id ₂₃	fr	narrative text	very low	learner	higher education	difficult	10
Id ₂₄	fr	narrative text	high	learner	higher education	difficult	90
Id ₂₅	fr	narrative text	high	learner	higher education	difficult	90
Id ₂₆	fr	narrative text	medium	learner	higher education	medium	200
Id ₂₇	en	simulation	medium	learner	higher education	medium	30

Tabla 6.5: Metadatos relevantes de los Objetos de Aprendizaje del repositorio local.

clasificación del metadato.

- En el caso del Usuario2, analizamos por qué recupera en décima posición al objeto Id₂, el cual es un objeto que consiste en un programa java, pero esto no satisface

Usuario	Distancia	Peor Caso
Usuario2	6	66
Usuario3	14	55
Usuario4	9	66
Usuario5	25	67
Usuario6	14	72
Promedio de distancia	13.6	65.2

Tabla 6.6: Distancias del Agente-R sobre los casos satisfactorios.

lo que el usuario estaba buscando. Observando sus metadatos, en la Tabla 6.5, se puede ver que el metadato **Learning Resource Type** contiene el valor *narrative text*, lo cual influye en un alto grado de creencia de que ese objeto satisface el estilo de aprendizaje elegido (teórico en este caso).

- En el caso del Usuario4, si bien los metadatos que describen el objeto Id_1 son los que mejor satisfacen las preferencias del usuario, el metadato **Learning Resource Type** contiene el valor *narrative text*, y al aplicar las reglas tiene el máximo grado de creencia de que el objeto satisface el estilo teórico. Sin embargo, el usuario que realiza la prueba difiere en esto, considerando que el estilo de este documento es práctico, dado que es una implementación del problema “vendedor ambulante” escrito en Java. Lo mismo sucede con el objeto Id_2 , este objeto consiste en una implementación pero el metadato **Learning Resource Type** contiene el valor *narrative text*.
- En el caso del Usuario5, podemos observar lo mismo que en el caso anterior con respecto al primer objeto recomendado Id_1 , los metadatos que lo describen son los que mejor satisfacen las preferencias del usuario, donde el metadato **Learning Resource Type** contiene el valor *narrative text*, y al aplicar las reglas tiene el máximo grado de creencia de que el objeto satisface el estilo teórico. Sin embargo, el usuario que realiza la prueba difiere en esto, considerando que el estilo de este documento es práctico, dado que es una implementación del problema “vendedor ambulante” escrito en Java. Por otro lado, el usuario ordena posicionando en segundo lugar al objeto Id_{21} el cual es recomendado por el sistema en la posición 18, esto se debe a que el metadato **Learning Resource Type** contiene el valor *simulation*, es decir el grado de creencia de que el objeto satisface el estilo teórico (elegido por el usuario) es muy bajo y además el metadato **Difficulty** contiene el valor *easy*.

De lo anterior podemos concluir que los resultados pueden no satisfacer al usuario dado que un documento aunque cumpla con las preferencias establecidas en la interfaz puede no transmitir lo que el usuario deseaba para aprender, en el caso de un estudiante, o dictar una clase, en el caso de un docente. En la selección de OA como en otro tipo de elementos pueden intervenir factores subjetivos difíciles de captar y modelizar.

Por otro lado, los valores de los metadatos cargados pueden resultar subjetivos a la persona que los carga, como también un error en la carga puede producir que el resultado no sea el adecuado. Aún se puede tener errores de la extracción automática de metadatos.

El desempeño del Agente-R depende de todos estos factores externos. Teniendo en cuenta esto y el promedio de distancias, 13.6, con un promedio 65.2 de peores casos,

podemos afirmar que observamos un buen desempeño del Agente-R, y en la mayoría de los casos, la lista de objetos recomendada no es tan distante a la elegida por el usuario.

También podemos destacar, que una búsqueda recomendada basándose en el perfil del usuario puede resultar ventajosa con respecto a una búsqueda simple, como por ejemplo si se utiliza la interfaz de ARIADNE mencionada en la Sección 3.3. Dado que con esta búsqueda se puede filtrar según valores de algunos metadatos, se obtiene simplemente una lista de objetos, los cuales cumplen con la conjunción de los filtros aplicados. Pero su orden, por lo general, no condice con el perfil del usuario. A su vez, pueden excluirse objetos adecuados al perfil, por no cumplir con algunos de los filtros.

Capítulo 7

Discusiones

En este trabajo se ha presentado la arquitectura de un sistema recomendador de objetos de aprendizaje. Esta arquitectura se basa en un sistema multiagente lo cual permite trabajar de una forma flexible y escalable, con la información heterogénea y distribuida proveniente de los distintos repositorios de objetos de aprendizaje. Se diseñó uno de los agentes de esta arquitectura, el Agente Recomendador, Agente-R, como un agente BDI graduado y se desarrolló la implementación de un prototipo de este agente. En particular, la implementación se desarrolló en el lenguaje SWI Prolog, dado que se cuenta con un desarrollo previo de agente recomendador de turismo, modelado también como agente g-BDI, en este lenguaje de programación. En las siguientes secciones se discutirán algunos puntos.

7.1. Repositorios

Un inconveniente que se presentó para realizar la prueba del prototipo es la falta de información en muchos de los metadatos educacionales de los objetos de aprendizaje en los repositorios evaluados. Por ejemplo en FLOR los metadatos en su mayoría fueron cargados automáticamente y sólo se cargaron algunos de los metadatos pertenecientes a la categoría Educacional, lo que consideramos que dificultaba obtener una recomendación personalizada. Otra dificultad presentada es el acceso limitado a la descripción de los objetos o a ellos mismos. Por otro lado, ARIADNE, el cual presenta metadatos educacionales, no cuenta con un repositorio local de objetos en idioma español.

7.2. Prueba del Prototipo

También fue difícil la evaluación de la recomendación brindada, ya que a través de la selección de los recursos más adecuados, se busca alcanzar mejores resultados del proceso de enseñanza-aprendizaje, lo cual es complejo de evaluar. En una primera etapa se realiza una evaluación por parte del usuario (estudiante o docente) del sistema, respecto a su visión del material recomendado, considerando si los objetos de aprendizaje recomendados le resultan útiles y si el orden es el correcto. El Agente-R determina la lista a recomendar teniendo en cuenta distintos factores pertenecientes a los diferentes contextos mentales,

pero un factor muy importante es el repositorio de objetos, la rica y variada descripción de estos, la subjetividad en la clasificación etc. Sin embargo, a partir del caso de estudio se observa que la propuesta del Agente Recomendador, y de las reglas definidas, es promisorias.

7.3. Proyectos y Publicaciones

El trabajo de esta tesina ha sido desarrollado en el marco de los siguientes Proyectos:

- JARDIN “JUST AN ASSISTANT FOR INSTRUCTIONAL DESIGN”, Proyecto LACCIR (Latin American and Caribbean Collaborative ICT Research), Federación Latinoamericana y del Caribe para la Investigación Colaborativa en Tecnologías de Información y Comunicación. Directora Regina Motz (UdeLAR, Uruguay). Período 2008-2009.
- SISTEMA DE APOYO AL DOCENTE EN LA BÚSQUEDA Y PREPARACIÓN DE MATERIAL DIDÁCTICO PARA LA ENSEÑANZA EN LAS ESCUELAS SANTAFESINAS. Proyecto 219308 de la Secretaría de Estado de Ciencia, Tecnología e Innovación, Provincia de Santa Fe. Directora Ana Casali, año 2009.

Durante el transcurso de la tesina se ha realizado la siguiente Publicación:

- Casali A., Gerling V., Deco C. y Bender C. Un Sistema Inteligente para Asistir la Búsqueda Personalizada de Objetos de Aprendizaje. En Actas del Congreso de Inteligencia Computacional Aplicada (CICA) Esteban Di Tada et al.(eds) 1ª ed. (ISBN 978-987-24967-3-9), 8 pag. Universidad de Palermo, Buenos Aires, 2009.

7.4. Trabajos Futuros

- Difundir la creación de objetos de aprendizaje en repositorios latinoamericanos. En este aspecto, FLOR es un repositorio latinoamericano de objetos de aprendizaje, donde se espera que haya una mayor descripción de los objetos.
- Otra línea a futuro de trabajo sería realizar un sistema de extracción automática/semi-automática de metadatos, y que contemple los metadatos educacionales. Esto ya está en desarrollo, utilizando herramientas del área Inteligencia Artificial y Machine Learning (Aprendizaje Automatizado). De esta manera, se espera obtener la información de una gran cantidad de objetos en cortos plazos.
- A su vez, se necesitaría contar con un control de calidad de metadatos, para asegurar que los valores sean correctos.
- Completar el desarrollo de agentes del sistema multiagente recomendador planteado en este trabajo. En esta línea se llevará a cabo un proyecto que consiste en el desarrollo de cada uno de los restantes agentes del sistema recomendador, es decir, el Agente Perfil de Usuario, los Agentes Buscadores, y el Agente Mediador. En cuanto al Agente Refinador Semántico, se cuenta con una implementación la cual

se adaptará al enfoque de sistema multiagente. Por último, se desea desarrollar una Interfaz Web (PHP) para lograr un alcance masivo del sistema para lograr una efectiva experimentación.

Bibliografía

- [1] Barrueco J M. and Subirats Coll I. OAI-PMH: protocolo para la transmisión de contenidos en Internet. Disponible en [www.uv.es/=barrueco/cardedeu.doc](http://www.uv.es/~barrueco/cardedeu.doc) [consultado septiembre 2009].
- [2] Benferhat S., Dubois D., Kaci S. and Prade, H. Bipolar representation and fusion of preferences in the possibilistic Logic framework. *In Proceedings of the 8th International Conference on Principle of Knowledge Representation and Reasoning (KR '02)*, pages 421-448, 2002.
- [3] Caplan, P. *Metadata Fundamentals for All Librarians* . Chicago : American Library Association, 2003.
- [4] Casali, A., Godo, Ll., Sierra, C. A Tourism Recommender Agent: From theory to practice in *Revista Iberoamericana de Inteligencia Articial, AEPIA* (ISSN 1137-3601) Vol 12:40(2008), pp. 23:38, 2008.
- [5] Casali A., Godo L. and Sierra C. Graded BDI Models For Agent Architectures. Leite J. and Torroni P. (Eds.) *CLIMA V, LNAI 3487*, pp. 126-143, Springer-Verlag, Berling Heidelberg, 2005.
- [6] Casali A., Godo Ll. and Sierra C. Graded BDI Models For Agent Architectures. *Proceedings of the Fifth International Workshop on Computational Logic in Multi-agent Systems, CLIMA V*, Joao Leite and Paolo Torroni eds., 18-33, ISBN:972-9119-37-6. Lisboa, Portugal, setiembre de 2004. Also in *Proceedings of the Second European Workshop on Multi-Agent Systems EUMAS 04*. Barcelona, España, diciembre de 2004.
- [7] Casali A., Godo L. and Sierra C. Modelling Travel Assistant Agents: a graded BDI Approach. *Proceedings of the IFIP-AI, WCC, Volume 217, Artificial Intelligence in Theory and Practice*, Max Bramer (ed.), Springer, pp. 415-424, 2006.
- [8] Casali A., Godo Ll. and Sierra C. Multi-Context Specification for Graded BDI Agents. *Proceedings of the Doctoral Consortium - Fifth International Conference on Modeling and Using Context (CONTEXT-05)*. Research Report LIP 6, Paris, Francia, Julio de 2005.
- [9] Ghidini C. and Giunchiglia F. Local Model Semantics, or Contextual Reasoning = Locality + Compatibility *Artificial Intelligence*,127(2):221-259, 2001.
- [10] Giunchiglia F. and Serafini L. Multilanguage Hierarchical Logics (or: How we can do without modal logics) *Journal of Artificial Intelligence*, vol.65, pages 29-70, 1994.

- [11] Godo, L., Esteva, F. and Hajek, P. Reasoning about probabilities using fuzzy logic. *Neural Network World*, 10:811–824, 2000.
- [12] IEEE Standard 1484.12.1, New York: Institute of Electrical and Electronics Engineers, Learning Technology Standards Committee (2002) (PDF), Draft Standard for Learning Object Metadata. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, [consultado marzo 2009].
- [13] Montaner M., López B., de la Rosa J.L. A Taxonomy of Recommender Agents on the Internet, *Artificial Intelligence Review*, Kluwer Academic Publishers. Volume 19, Issue 4, pp. 285-330. June, 2003.
- [14] Motz R., Guzmán J., Deco C. and Bender C. Applying ontologies to educational resources retrieval driven by cultural aspects. *Journal of Computer Science & Technology*. ISSN 1666-6038. JCS&T Vol 5, N° 4, pp 279-284, December 2005.
- [15] Niinivaara O. Agent-Based Recommender Systems. *Technical Report, University of Helsinki*, Dept. of CS, 2004.
- [16] Parsons, S., Sierra, C. and Jennings N.R. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3): 261-292, 1998.
- [17] Rao A. and Georgeff M. BDI Agents from Theory to Practice, *Technical Note 56, AAIL*, April 1995.
- [18] Rao, A. and Georgeff M. Modeling Rational Agents within a BDI-Architecture. In proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR-92), pages 473-484 (ed R. Fikes and E. Sandewall), Morgan Kaufmann, San Mateo, CA, 1991.
- [19] Romero C., Ventura S., Delgado J. and de Bra P. Personalized Links Recommendation Based On Data Mining in Adaptive Educational Hypermedia Systems. *Second European Conference on Technology Enhanced Learning (EC-TEL 2007)*. Crete, Greece, 2007.
- [20] Sabater, J., Sierra, C., Parsons, S. and Jennings N. R. Engineering executable agents using multi-context systems. *Journal of Logic and Computation*12(3): 413-442 (2002).
- [21] Soonthornphisaj N., Rojsattarat E. and Yimngam S. Smart E-Learning Using Recommender System. *Lecture Notes in Computer Science*. Springer Berlin, Heidelberg, Volume 4114. Computational Intelligence pp 518-523. 2006.
- [22] Terveen L. G. and Hill W. Beyond Recommender Systems: Helping People Help Each Other. In Carroll, J. (Ed.), *HCI in the New Millennium*. Addison Wesley, 2001.
- [23] Wiley, D. Connecting Learning Objects to Instructional Design Theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (ed.) *Instructional Use of Learning Objects*. Editorial Association for Instructional Technology, 2002.

- [24] Zaiane O.R. Building a recommender agent for e-learning systems. Proceedinds of *International Conference on Computers in Education*, pp: 55-59, 2002.
- [25] Zhu F., Ip H. and Cao J. PeRES: A Personalized Recommendation Education System Based on Multi-agents & SCORM, in *Advances in Web Based Learning ICWL 2007*, LNCS 4823/2008, pp 31-42, 2008 .