

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

Ana Casali¹, Valeria Gerling

Universidad Nacional de Rosario,

Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Departamento de Sistemas e Informática,

¹Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas CIFASIS

Rosario, Argentina, 2000

acasali@fceia.unr.edu.ar, valeria.gerling@gmail.com

Claudia Deco, Cristina Bender

Pontificia Universidad Católica Argentina,

Facultad de Química e Ingeniería Rosario, Departamento de Investigación Institucional,

Rosario, Argentina, 2000

cdeco@uca.edu.ar, cbender@uca.edu.ar

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 a sus necesidades y preferencias. 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 multiagente 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. Actualmente se está implementando un prototipo de este sistema de recomendación.

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. The implementation of this recommender system is an ongoing work.

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

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 desea 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 no es el adecuado para todos los usuarios. Esto se debe a que los usuarios poseen distintas características y preferencias personales, que deberían también ser consideradas en el momento de la búsqueda. 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 un sistema recomendador cuyo objetivo es apoyar a los usuarios a encontrar recursos educativos de acuerdo a sus características y preferencias y les brinda los resultados ordenados de acuerdo a su perfil.

El resto del trabajo se organiza de la forma siguiente: en la Sección 2 se describen conceptos preliminares como objetos de aprendizaje, sistemas recomendadores y sistemas multiagentes. En la Sección 3 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. Luego se presenta un ejemplo para ilustrar el funcionamiento de la arquitectura propuesta y finalmente se plantean algunas discusiones.

2. Conceptos Preliminares

2.1. Objetos de Aprendizaje, Repositorios y Metadatos

Según Wiley [10] un **Objeto de Aprendizaje** (OA) es todo recurso digital que apoya a la educación y que puede ser reutilizado. 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. Algunos ejemplos de recursos digitales son imágenes, fotos, cortos de video, pequeñas porciones de texto, ecuaciones, definiciones, aplicaciones Web, páginas Web, documentos en formato pdf, etc. 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. Un **Repositorio de Objetos de Aprendizaje** se entiende como una gran colección de los mismos, 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 la 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 Educational Resources)⁴, entre otros.

Para el diseño del sistema recomendador, se han analizado en particular los repositorios FLOR y OER Commons. 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).

¹<http://www.merlot.org/>

²<http://www.careo.org/>

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

⁴<http://www.oercommons.org/>

Los **metadatos** son un conjunto de atributos necesarios para describir las principales características de un recurso. Son especialmente útiles en los recursos que no son textuales, por ejemplo los multimedia, y en los que su contenido no puede ser indizado por sistemas automáticos. El estándar de la IEEE⁵ 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 nueve categorías: General, que describe el objeto de aprendizaje como un todo; Ciclo de vida, que son características relacionadas con la historia y el estado presente del objeto; Metametadatos, con información sobre los mismos metadatos, no sobre el objeto de aprendizaje que se está describiendo; Técnica, que contiene requisitos y características técnicas del objeto; Educativo, con condiciones del uso educativo del recurso; Derechos, con condiciones de uso para la explotación del recurso; Relación, vinculación del recurso descrito con otros objetos de aprendizaje; Anotación, conteniendo comentarios sobre el uso educativo del objeto de aprendizaje; y Clasificación, con la descripción temática del recurso.

La personalización de los resultados propuesta en este trabajo se sustenta en los metadatos de los documentos. Los metadatos siguen el estándar LOM y una extensión de éste para la consideración de características culturales [4]. Uno de los mayores problemas encontrados en los repositorios analizados es que, aunque se prevé la inclusión de metadatos educacionales en su diseño, éstos generalmente no están cargados.

2.2. Agentes Recomendadores

En los últimos años la comunidad de Inteligencia Artificial ha desarrollado un intenso trabajo alrededor de los sistemas recomendadores [5]. 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 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 verse en [3] y las principales técnicas para su desarrollo pueden agruparse en [9]: sistemas de filtrado colaborativo, filtrado basado en contenidos, filtrado basado en conocimiento y sistemas híbridos.

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 espera 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 [11], 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. [7] plantea el uso de técnicas de minería de datos para recomendar la navegación entre links y Soonthornphisaj et al. [8] propone un sistema que integra el material recomendado antes de dárselo al usuario.

2.3. Agentes BDI

Una de las arquitecturas de agentes más notorias es el agente BDI (Belief-Desire-Intention) propuesto por Rao y Georgeff [6]. 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. Con el propósito de hacer que la arquitectura BDI sea más flexible, Casali et al. [1] 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

⁵IEEE LOM specification. <http://ltsc.ieee.org/wg12>, 2002.

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.

El modelo g-BDI está basado en la noción de sistemas multi-contextos que constituyen una propuesta interesante para el diseño de sistemas lógicos complejos, y que particularmente han sido utilizados para la especificación de agentes. Estos sistemas permiten la definición de diferentes componentes formales y sus interrelaciones. En el modelo g-BDI se utilizan contextos separados para representar cada actitud mental, y cada uno está formalizado con el aparato lógico más apropiado. Las interacciones entre componentes se especifican usando reglas inter-contextos, llamadas reglas puente (bridge rules). Esta propuesta ha sido utilizada previamente para modelar agentes en el dominio turístico [2], donde los diferentes componentes de la arquitectura y sus interacciones pueden ser representados con claridad.

En este trabajo, se presenta una arquitectura multiagente para la recuperación de recursos educativos con el fin de asistir a un estudiante o a un docente a elegir objetos educativos acordes a sus características personales y preferencias. Esta plataforma multiagente incluye varios tipos de agentes en concordancia con las distintas funcionalidades del sistema. En particular, se ha modelado el Agente Recomendador (Agente-R) como un agente BDI graduado, el cual se encarga de la recuperación de los mejores OA acordes al perfil del usuario.

3. Arquitectura del Sistema Recomendador

La arquitectura multiagente propuesta consta de los siguientes agentes: el *Agente Interfaz (Agente-I)*, 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)* 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 1 y a continuación se detallan los agentes que la componen.

- **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 OA) que dará al usuario.
- **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 de la disyunción de las expansiones de cada concepto y luego se considera la conjunción de esas expansiones. Esta estrategia es enviada a cada agente buscador.
- **El Agente Perfil de Usuario (Agente-PU)** 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 (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.
- **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 (denominadas 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 como repositorios haya, dado que la manera de consultar varía de acuerdo al protocolo necesario para comunicarse con el repositorio correspondiente.

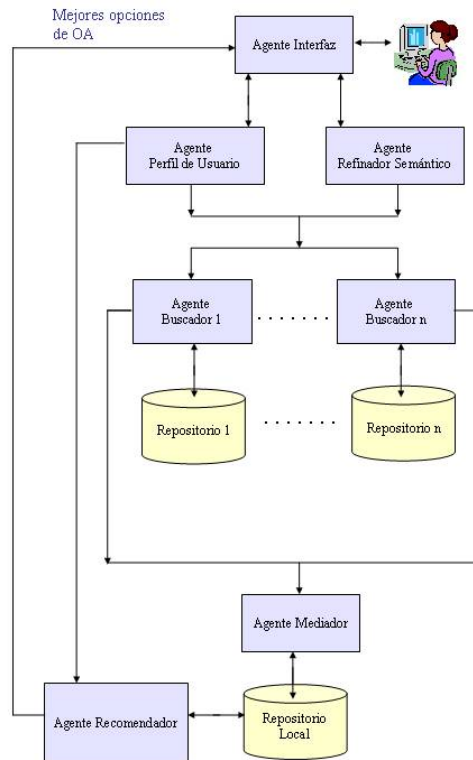


Figura 1: Arquitectura del Sistema Multiagente

- **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.
- **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. Se ha diseñado el Agente-R utilizando el modelo de agente BDI graduado y en la próxima sección se lo describe.

4. Diseño del Agente Recomendador

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, descritas 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.
- **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 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 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.

5. 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, pero no sobre el portugués. 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 1:

Preferencia	Prioridad	Descripción
Idioma=“español”	1.0	Idiomas que el usuario desea que presenten los documentos.
Idioma=“inglés”	0.8	
Rol=“estudiante”	1.0	Si el que realiza la búsqueda lo hace como estudiante o docente.
Interacción=“baja”	0.7	Grado de interacción que el usuario desea tener con el objeto
Contexto Académico=“universitario”	0.7	Nivel educativo en que el usuario desea el material.
Nivel de Conocimiento=“bajo”	1.0	Nivel de dificultad que el usuario desea sobre el tema.
Estilo de Aprendizaje=“práctico”	1.0	
Restricción		Descripción
Duración Máxima=“60”	0.7	Tiempo máximo que el usuario espera que le tome trabajar con el objeto.
Costo=“no”	1.0	Si restringe la búsqueda a los recursos que no sean pagos o permite los que tienen costo.

Tabla 1: Extracto del perfil del usuario correspondiente a José.

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 2 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. En primer lugar busca los que satisfacen la restricción de no

Metadatos	O_1	O_2	O_3	O_4
Language	inglés	español	español	inglés
Format	doc	pdf	doc	pdf
Learning Resource Type	lectura	[ejercicio, lectura]	diapositiva	ejercicio
Interactivity Level	bajo	bajo	bajo	alto
Intended End User Role	estudiante	estudiante	estudiante	docente
Context	universitario	universitario	secundario	universitario
Difficulty	media	baja	media	alta
Typical Learning Time	40	50	20	50
Cost	no	no	no	no

Tabla 2: Metadatos relevantes de los Objetos de Aprendizaje recuperados.

durar más de 60 minutos. Como cada $O_i, i \in \{1 \dots 4\}$ satisface esta condición, los cuatro OA son considerados para el ranking.

A partir de un conjunto de reglas en el Belief Context del Agente-R se calculará el grado de satisfacción esperada de cada una de las preferencias del usuario (por ejemplo, interactividad=baja, idioma=español) a través de las distintas características de cada O_i . Ejemplos de estas reglas son:

- (R1) SI InteractivityLevel(O_i)=bajo Entonces B(O_i ,interactividad=baja, 1)
- (R2) SI InteractivityLevel(O_i)=medio Entonces B(O_i ,interactividad=baja, 0.6)
- (R3) SI Context(O_i)=universitario and Difficulty(O_i)=media Entonces B(O_i , (contexto=universitario) and (nivel=bajo), 0.7)
- (R4) SI Context(O_i)=secundario and Difficulty(O_i)=alta Entonces B(O_i , (contexto=universitario) and (nivel=bajo), 0.8)
- (R5) SI LearningResourceType(O_i)=ejercicio Entonces B(O_i ,estilo=práctico, 1)
- (R6) SI LearningResourceType(O_i)=[ejercicio,lectura] Entonces B(O_i ,estilo=práctico, 0.6)

Por ejemplo, la regla R1 representa: “si el grado de interactividad (representado por el metadato InteractivityLevel) del O_i es bajo, entonces la creencia del agente respecto a que ese objeto va a satisfacer la preferencia del usuario *interactividad=baja* es máxima (1)”.

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_i ,estilo=práctico, 0.6) representa que el Agente-R cree en grado 0.6 que el objeto O_i 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 considerando distintos factores que provienen de distintos contextos. Uno de ellos es la prioridad dada por el usuario a cada preferencia (p_j, d_j). Por ejemplo, en el caso de José tenemos $(p_3, d_3) = (interactividad = baja, 0,7)$. Otro factor es el grado de satisfacción de cada preferencia p_j por las características de un recurso, representado por la fórmula $B(O_i, p_j, b_{ij})$. También se podría incluir el costo y la confianza en la fuente proveedora del recurso (Universidad, Institución, Autor, etc.). Distintas funciones para computar este grado de intención asociado a cada OA se pueden utilizar modelando distintos comportamientos del Agente-R. En este ejemplo, se plantea este cálculo como el promedio de las satisfacciones esperadas de las distintas preferencias:

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

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

$$I(O_1, P) = 0,76 \quad I(O_2, P) = 0,80 \quad I(O_3, P) = 0,71 \quad I(O_4, P) = 0,56$$

Finalmente, el Agente-R ordena los objetos según el valor decreciente del grado obtenido: O_2, O_1, O_3, O_4 . 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.

6. Discusión

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.

Actualmente, se encuentra en desarrollo la implementación de un prototipo del Sistema Recomendador propuesto. En particular, la implementación del Agente Recomendador se está desarrollando 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 [2]. A partir de este prototipo del sistema recomendador se podrá realizar una validación y experimentación de la arquitectura de sistema planteado.

Un inconveniente para la validación del prototipo es la falta de información en muchos de los metadatos educacionales de los objetos de aprendizaje en los repositorios evaluados (por ejemplo, FLOR, OER Commons), lo que hace necesario completar estos metadatos en los recursos sobre algún tema determinado, para poder realizar alguna experimentación. También es 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 prevé realizar una evaluación por parte del usuario (estudiante o docente) del sistema, respecto a su visión del material recomendado, por ejemplo, se puede considerar si los OA recomendados le resultan útiles y si el orden es el correcto.

Agradecimientos

Este trabajo está parcialmente financiado por el Proyecto 219308, Secretaría de Estado y Ciencia, Tecnología e Innovación. Provincia Santa Fe.

Referencias

- [1] 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, Berlin Heidelberg, 2005.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] Niinivaara O. Agent-Based Recommender Systems. *Technical Report, University of Helsinki*, Dept. of CS, 2004.
- [6] Rao A. and Georgeff M. BDI Agents from Theory to Practice, *Technical Note 56, AAIL*, April 1995.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] Zaiane O.R. Building a recommender agent for e-learning systems. Proceedings of *International Conference on Computers in Education*, pp: 55-59, 2002.