

CONCEPTUALIZACIÓN Y ANÁLISIS DE UN SISTEMA MULTI-AGENTE PEDAGÓGICO UTILIZANDO LA METODOLOGÍA MAS-COMMONKADS

CONCEPTUALIZATION AND ANALYSIS OF A PEDAGOGICAL MULTI-AGENT SYSTEM USING MAS-COMMONKADS METHODOLOGY

JOVANI ALBERTO JIMÉNEZ BUILES

Grupo de Investigación y Desarrollo en Inteligencia Artificial, Universidad Nacional de Colombia, jajimen1@unal.edu.co

DEMETRIO ARTURO OVALLE CARRANZA

Grupo de Investigación y Desarrollo en Inteligencia Artificial, Universidad Nacional de Colombia, dovalle@unal.edu.co

JOHN WILLIAM BRANCH BEDOYA

Grupo de Investigación y Desarrollo en Inteligencia Artificial, Universidad Nacional de Colombia, Medellín

Recibido para revisar abril 23 de 2008, aceptado Mayo 29 de 2008, versión final junio 13 de 2008

RESUMEN: En este artículo se presentan la conceptualización y el análisis de un sistema multi-agente (MAS) pedagógico utilizando la metodología MAS-CommonKADS; la cual es una de las más reconocidas para el modelado de MAS. El propósito de la fase de conceptualización es comprender el problema, identificar los actores activos y pasivo, además de elaborar los requisitos que permitan formular unos objetivos para generar la solución. En la fase de análisis se definen los modelos de agente, tareas, coordinación, comunicación, organización y experiencia (conocimientos). Ambas fases fueron validadas mediante la construcción del ambiente multi-agente pedagógico ALLEGRO.

PALABRAS CLAVE: Inteligencia Artificial en Educación, Inteligencia Artificial Distribuida, Sistemas Multi-Agente, Metodología MASCommonKADS.

ABSTRACT: In this paper is presented the conceptualization and analysis of a pedagogical multi-agent system (MAS) using the methodology MAS-CommonKADS; which is one of the most renowned methodologies for MAS modeling. The purpose of the conceptualization phase is understanding the problem, identifying the active and passive actors, and also making the requirements that allow us formulating some objectives to generate the solution. In the analysis phase are defined the models of agents, tasks, coordination, communication, organization and experience (knowledge). Both phases were validated through the construction of the pedagogical multi-agent environment ALLEGRO.

KEYWORDS: Artificial Intelligence in Education, Distributed Artificial Intelligence, Multi-Agent System, MAS-CommonKADS Methodology.

1. INTRODUCCIÓN

El propósito de integrar la inteligencia artificial con la educación radica fundamentalmente en aplicar técnicas al desarrollo de sistemas de enseñanza y de aprendizaje asistida por computador con el objeto de construir sistemas

“más inteligentes”. El término “inteligente” utilizado en estos sistemas queda determinado fundamentalmente por su capacidad de adaptación continua de la instrucción a las características del aprendizaje y del conocimiento de cada uno de los diferentes usuarios [16]. Algunas de estas técnicas y

mecanismo son por ejemplo la Planificación Instruccional (IP), Razonamiento Basado en Casos (CBR), Sistemas Tutoriales Inteligentes (ITS), Ambientes Colaborativos de Aprendizaje (MAS) y Sistemas Multi-Agente (MAS), entre otras [6].

Uno de los más recientes logros adoptados en la ingeniería de software es el paradigma orientado a agentes, el cual se aplica en la representación de sistemas distribuidos dando buenos resultados en dominios tales como: gestión del conocimiento, filtrado y recuperación de información, comercio electrónico, entre otras [10]. Los agentes de software pueden representar conocimiento pedagógico y/o desempeñar tareas de tutoría [3] para soportar y facilitar el aprendizaje humano [7], a este tipo de agentes se les denomina “Agentes Pedagógicos”.

En este artículo se presenta las fases de conceptualización y análisis del Ambiente Multi-Agente de Enseñanza/Aprendizaje ALLEGRO. La metodología utilizada para el desarrollo del ambiente fue MAS-CommonKADS la cual fue propuesta por Iglesias [4]. La metodología ofrece ventajas como es el planteamiento del desarrollo de modelos que reflejan diferentes vistas del proyecto. También es relevante porque ofrece un marco para la especificación del conocimiento independiente de la implementación, combinando un conjunto de modelos de conocimiento reutilizable para unas tareas que se realizan frecuentemente.

ALLEGRO es un ambiente inteligente conformado de un parte por un ITS el cual permite brindar aprendizaje en forma individualizada y de otra por un CSCL que ofrece aprendizaje en modo colaborativo. ALLEGRO fundamenta su paradigma instruccional en cinco modelos pedagógicos: Conductismo, Cognitivismo, Histórico-Social, Cognición Distribuida y Aprendizaje Basado en Problemas. El ambiente fue modelado a través del enfoque de un MAS, debido a que ofrece las siguientes cualidades: Autonomía, flexibilidad y adaptabilidad.

Cuando se habla de autonomía se refiere a la toma de iniciativa (proactividad) de los agentes del MAS para realizar acciones pedagógicas sin

la intervención de los humanos con el propósito de que el aprendiz logre los objetivos instruccionales propuestos. Cuando se habla de flexibilidad se refiere a que el aprendiz utiliza indistintamente cualquiera de las metodologías para su aprendizaje, a saber: Individualizada o colaborativa. Cuando se habla de adaptabilidad se refiere a que el contenido de la instrucción se moldea de acuerdo a las necesidades específicas y preferencias del aprendiz.

La Planificación de la Instrucción (IP) en ALLEGRO utiliza la técnica del Razonamiento Basado en Casos (CBR); es decir, se aprovecha de la experiencia almacenada en la solución exitosa de problemas similares pasados. En este sentido se puede afirmar que el sistema aprende en forma autónoma a partir de la experiencia con los aprendices, convirtiendo a la IP en una herramienta flexible con capacidad de adaptar los conocimientos con determinado grado de abstracción dependiendo del alumno.

En el siguiente capítulo se presenta el marco teórico de referencia haciendo énfasis los MAS y la metodología MAS-CommonKADS. En los capítulos tres y cuatro se exponen las fases de conceptualización y análisis, respectivamente. Finalmente, en el capítulo cinco se presentan las conclusiones.

2. MARCO TEÓRICO DE REFERENCIA

2.1 Sistemas Multi-Agente

Un Sistema Multi-Agente (Multi-Agent Systems, MAS) es una sociedad organizada compuesta por agentes semiautónomos que interactúan entre sí, ya sea para colaborar en la solución de un conjunto de problemas o en la consecución de una serie de objetivos individuales o colectivos [6]. Los principios de los MAS han mostrado un potencial adecuado en el desarrollo de sistemas de enseñanza debido al hecho de que la naturaleza de los problemas de enseñanza y de aprendizaje son más fácilmente resueltos a través de un abordaje cooperativo.

Los agentes de software pueden representar conocimiento pedagógico y/o desempeñar tareas de tutoría para soportar y facilitar el aprendizaje

humano. A este tipo de agentes se les denomina “Agentes Pedagógicos”. Los agentes pedagógicos pueden adaptar sus interacciones instruccionales a las necesidades de los aprendices y al estado actual del ambiente, ayudando a los aprendices en la superación de sus dificultades y en el aprovechamiento de las oportunidades de aprendizaje. Poseen un conjunto de metas de enseñanza, planes instruccionales para la ejecución de esas metas, y recursos asociados en los ambientes de aprendizaje. Ellos colaboran con los aprendices y con otros agentes, proporcionando realimentación continua durante las sesiones de trabajo [3].

Algunos ejemplos de arquitecturas de MAS con propósitos pedagógicos son: MACES: Multiagent Architecture for an Collaborative Educational System [1], AMPLIA: Ambiente Multi-Agente Probabilístico Inteligente de Aprendizagem [2], BAGHERA [15] y JADE: Java Agent framework for Distance learning Environments [14]; entre otros.

2.2 Metodología MAS-CommonKADS

La definición de una metodología de ingeniería del software normalmente no parte de cero, sino que es un proceso de refinamiento, añadiendo los nuevos aspectos y perspectivas de los sistemas y lenguajes e integrando los componentes exitosos de otras metodologías previas. Para modelar el MAS se empleó la metodología MAS-CommonKADS la cual fue propuesta por Iglesias [4] para modelar Sistemas Basados en Conocimientos (KBS).

Rumbaugh [13] define una metodología de ingeniería del software como “un proceso para la producción organizada de software empleando una colección de técnicas predefinidas en las notaciones. Una metodología se presenta normalmente como una serie de pasos, con técnicas y notaciones asociadas a cada paso. Los pasos de la producción del software se organizan normalmente en un ciclo de vida consistente en varias fases de desarrollo”.

La metodología MAS-CommonKADS [4] permite la integración de técnicas de la ingeniería de conocimientos, ingeniería de software orientada a objetos e ingeniería de

software de protocolos. Se desarrolla a través de la construcción de siete modelos: el Modelo de Agente, que describe las características de cada agente; el Modelo de Tarea, que describe las tareas realizadas por los agentes; el Modelo de la Experiencia, que describe el conocimiento que necesitan los agentes para llevar a cabo los objetivos encomendados; el Modelo de Coordinación, que describe las relaciones dinámicas entre los agentes software; el Modelo de Comunicación, que describe las relaciones dinámicas entre los agentes humanos y los agentes software; el Modelo de la Organización, que describe las relaciones estructurales entre los agentes; y el Modelo de Diseño, que refina los modelos anteriores y decide qué arquitectura de agente es más adecuada para cada agente, así como los requisitos de la infraestructura de la red de agentes.

La aplicación de la metodología consiste en el desarrollo de los modelos. Cada modelo está compuesto por constituyentes (entidades modeladas) y relaciones entre los constituyentes. Se define una plantilla textual para describir cada constituyente, y estados de los constituyentes para describir su estado de desarrollo. En este artículo abarcaremos la identificación y comprensión del problema, unido con los actores y requisitos; esta fase se denomina conceptualización. Luego la definición de los modelos de agente, tareas, coordinación, comunicación, organización y experiencia (conocimientos); esta fase se conoce con el nombre de análisis.

3. CONCEPTUALIZACIÓN

3.1 Comprensión Del Problema

El propósito de esta fase es comprender el problema, identificar los actores y elaborar los requisitos que permitan formular unos objetivos para generar la solución. En este contexto se identificó que uno de los mayores problemas en los sistemas tradicionales de aprendizaje asistidos por computador es la dificultad de brindar enseñanza continuamente adaptada a las necesidades y características específicas de cada

aprendiz [14]. El problema crece con la explosión de la Internet y el surgimiento exponencial de estos sistemas los cuales han comenzado a soportar la educación a distancia apoyada en computador, o también conocida como educación virtual.

3.2 Ciclo De Desarrollo De Los Casos De Uso

Los casos de uso permiten describir las posibles interacciones o usos de un usuario con el sistema. Los usuarios se denominan actores y representan a las entidades externas al sistema [5], [11], [4]. A continuación se presentan el conjunto de los casos de uso que ocurren en el desarrollo de la fase de conceptualización.

3.2.1 Identificación y descripción de los actores

En este problema se pueden identificar varios tipos de actores activos que interactúan con el sistema. Así mismo, se pueden considerar en el sistema actores pasivos, tales como las bases de datos.

3.2.2 Identificación y descripción de los casos de uso

Se pueden identificar los siguientes casos de uso para cada actor exceptuando los actores pasivos:

› **Aprendiz.** Tiene cuatro casos de primer nivel: Iniciar sesión en el Sistemas Tutorial Inteligente (ITS), iniciar sesión en el Ambiente Colaborativo de Aprendizaje (CSCL), usar herramientas que le brinda el ambiente y salir (Figura 1).

Para el caso de Iniciar sesión en el ITS, puede realizar los siguientes episodios: Visualizar las Unidades Básicas de Aprendizaje (Basic Unit of Learning, BULs), mirar los Objetivos Instruccionales (Instructional Objectives, IOs) asociados a las BULs, visualizar los conocimientos que desea aprender (mirar simulación, problemas resueltos y realizar evaluación individualizada).

Las BULs son los temas o unidades a tratar en un dominio específico. Los IOs son los logros que debe de alcanzar un aprendiz al finalizar el estudio de una BUL.

Para el caso de uso Iniciar sesión en el CSCL, el aprendiz realiza los siguientes episodios: Visualizar el contenido del foro de debate, modificar, buscar o publicar un artículo en los foros de debate, enviar y recibir archivos y correos electrónicos, compartir pizarra en línea, y conversar con otros utilizando el video, el audio y/o el texto.

En Usar herramientas se encuentra: contemplar bibliografía, glosario, recursos, estadísticas y ayudas.

› **Docente.** Posee ocho casos de nivel superior: Iniciar sesión en el ITS, iniciar sesión en el CSCL; adicionar, borrar, modificar, visualizar y usar herramientas; y salir (Figura 1).

En el caso de uso Iniciar sesión en el ITS, se encuentran los episodios de agregar, borrar, modificar y visualizar: BULs, IOs, conocimientos, evaluaciones, casos, simulaciones, problemas propuestos y recomendaciones.

Para el caso de uso Iniciar sesión en el CSCL, se encuentran los siguientes episodios: Visualizar el contenido de los foros de debate, modificar, buscar o publicar un artículo en el mismo, enviar y recibir archivos y e-mails, compartir pizarra en línea, y conversar con otros utilizando el video, el audio y/o el texto.

› **Asistente de Docencia.** Posee cuatro casos de uso de nivel superior: Iniciar sesión en el CSCL, iniciar sesión en el ITS, usar herramientas y salir (Figura 1).

Para el caso de uso Iniciar sesión en el CSCL se encuentran los siguientes episodios: Agregar, borrar, modificar y visualizar la agenda y el foro, enviar y recibir archivos y e-mails, compartir pizarra en línea, y conversar con otros utilizando el video, el audio y/o el texto.

El monitor es un aprendiz que posee mayor conocimiento que los demás alumnos, por eso, el asistente de docencia realiza los mismos episodios en el caso de uso Iniciar sesión en el ITS. En Usar herramientas se encuentra: contemplar bibliografía, glosario, recursos, estadísticas y ayudas.

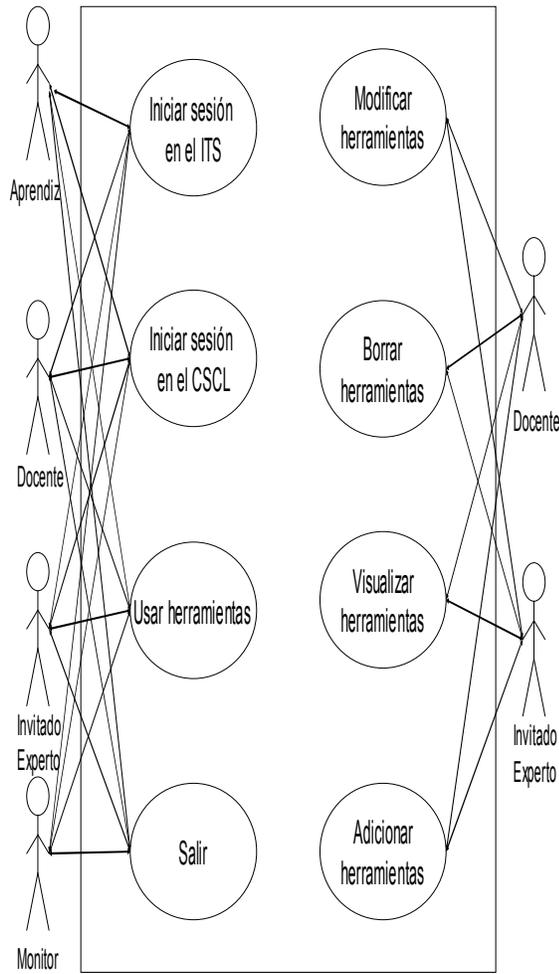


Figura 1. Casos de uso para los actores activos
Figure 1. Use Cases for Active Actors

› **Experto Invitado.** Posee ocho casos de uso de nivel superior: Iniciar sesión en el ITS, iniciar sesión en el CSCL, adicionar, borrar, modificar, visualizar y usar herramientas; y salir (Figura1).

Los episodios asociados al caso de uso Iniciar sesión en el ITS son: Mirar BULs, IOs, agregar, modificar, borrar y/o visualizar conocimientos. En Iniciar sesión en el CSCL se encuentra:

Visualizar el contenido del foro, modificar, buscar o publicar un artículo en el mismo, enviar y recibir archivos y e-mail, compartir pizarra en línea, y conversar con otros utilizando el video, el audio y/o el texto.

3.2.3 Descripción de los casos de uso

A continuación se describe textualmente (Iniciar sesión) los casos de uso (Tabla 1):

Tabla 1. Descripción de los casos de uso:
 Iniciar sesión en el ITS
Table 1. Description of the use cases:
 Session Initiation into the ITS

Caso de Uso	Iniciar sesión en el ITS
Resumen	Un usuario solicita iniciar una sesión de enseñanza y de aprendizaje individualizada y recibe unos conocimientos.
Actores	Aprendiz, Docente, Asistente de Docencia y Experto Invitado.
Precondiciones	Datos de identificación del usuario.
Descripción	El usuario solicita iniciar una sesión de enseñanza y de aprendizaje donde el sistema se encarga de brindarle unos conocimientos relacionados a su perfil cognitivo. Después de ingresar la información de identificación del usuario, el sistema se encarga de clasificarlo en los cuatro tipos de actores activos. El docente y el experto son las personas encargadas de alimentar los conocimientos; el aprendiz y el asistente de docencia reciben la enseñanza que brinda el sistema, lo mismo que el proceso evaluativo.
Excepciones	Datos ilógicos: Para realizar la evaluación de la BUL de estudio, se debió de aprobarse exitosamente la evaluación de la BUL anterior.
Poscondiciones	Tras este caso de uso el aprendiz y el asistente de docencia reciben instrucción.

Para la representación gráfica se utilizo el esquema propuesto por Regnell [12] quien a su vez, utiliza elementos de Jacobson [5] para indicar la relación entre un caso de uso. A

continuación se presenta los casos de uso para el aprendizaje (Figura 2):

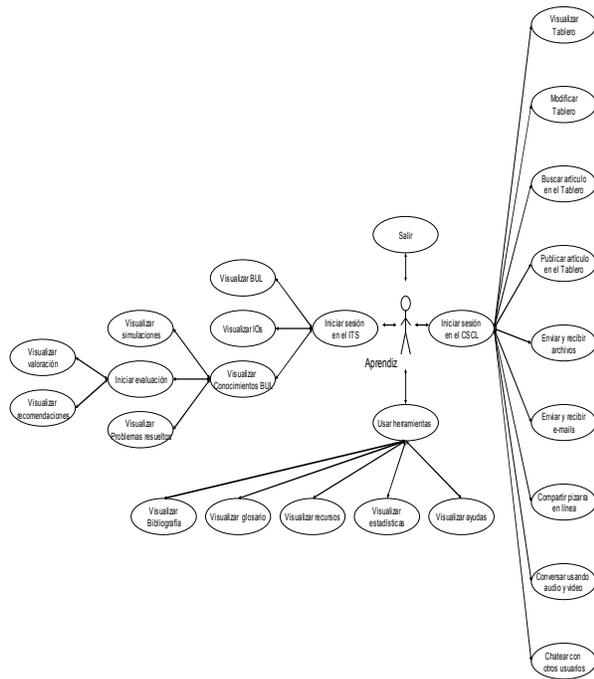


Figura 2. Casos de uso para el aprendiz
Figure 2. Use cases intended for the student

4. ANÁLISIS

En la fase de análisis se utilizaron los diferentes modelos producto de aplicar la metodología MAS-CommonKADS [4], a saber:

4.1 Modelo De Agente

El modelo del agente especifica las características de un agente, tales como: sus capacidades de razonamiento, habilidades, servicios, sensores, efectores, grupos de agentes a los que pertenece y clase de agente. Este modelo después de recoger las características genéricas de los agentes, sirve de puente ente los restos de modelos.

De las características, el Modelo de Agente se orienta esencialmente a los servicios. Los agentes son capaces de realizar tareas, que se especifican en un ejemplar del modelo de tareas. Los agentes pueden ofertar la realización de determinadas tareas (denominadas servicios) a

otros agentes. La especificación de los servicios se realiza en una ontología de servicios, descrita en el modelo de la experiencia. La especificación de los protocolos para ofertar y demandar servicios se describe en los modelos de coordinación y comunicación. Los agentes se caracterizan además por tener objetivos. Un objetivo es una responsabilidad aceptada por el agente [4].

4.1.1 Identificación de los agentes

Para distinguir los agentes del sistema, se tuvo en cuenta las diferentes tareas y roles de los actores abarcados.

› **Agentes Humanos.** Los actores de la fase de conceptualización se pueden identificar como agentes externos, ellos son: Aprendiz, Asistente de Docencia y Docente. El actor Experto Invitado no se tuvo presente debido a su bajo peso dentro del ambiente.

› **Agentes Software.** Tras el análisis de tareas, los agentes determinados son: Tutor, Modelo del Aprendiz, Interfaz, Experto, Diagnóstico y Colaborativo.

Dentro de las plantillas del modelo del agente que se presenta a continuación, solo se tuvieron presente los agentes de software. La tabla 2 presenta la identificación de las características del agente Tutor:

Tabla 2. Identificación del agente Tutor
Table 2. Identification of the Agent Tutor

Agente	Tutor	
Tipo	Básico	
Capacidades-razonamiento	Experiencia	Conocimiento del perfil del estudiante y tareas de los demás agentes de software.
Descripción	Agente de software encargado de guiar los procesos de enseñanza y de aprendizaje, decide las acciones pedagógicas a realizar, cómo y cuándo. Sus funciones son: Planificar y re-planificar continuamente el plan de tutoría, se comunica con los demás agentes solicitando conocimientos, evaluaciones, colaboración, información individualizada del aprendizaje; selecciona la estrategia pedagógica adecuada.	

4.1.2 Identificación y descripción de los objetivos

Los objetivos fueron fijados de acuerdo a la asignación de tareas para cada agente.

4.2 Modelo De Tareas

El modelo de tareas describe las tareas que los agentes pueden realizar: los objetivos de cada tarea, su descomposición, los componentes y los métodos de resolución de problemas para resolver cada objetivo a través de otros aprendices para presentarla al alumno (Figura 3). A continuación se presenta la descomposición de las tareas en subtareas mediante diagramas de Definición Integrada para Modelado de Funciones (Integrated Definition for Function Modeling, IDEF) [8]:

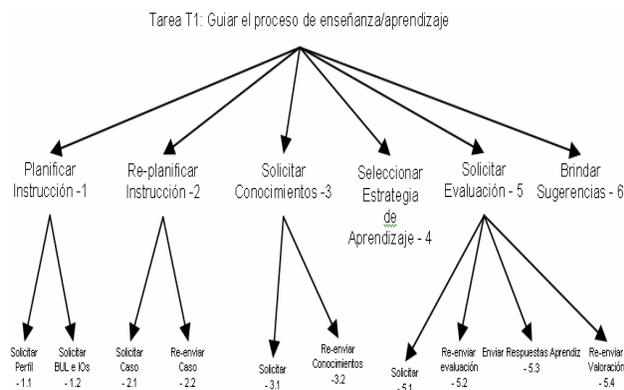


Figura 3. Descomposición de la Tarea T1: Guiar el proceso de enseñanza/aprendizaje

Figure 3. Decomposition of the Task T1: Driving Teaching / Learning Process

4.3 Modelo De Coordinación

El modelo de coordinación describe las interacciones entre los agentes de software. A continuación se describirá uno a uno los componentes de este modelo:

4.3.1 Identificación de las conversaciones

En los elementos anteriores, de manera implícita ya se han determinado algunas conversaciones; sin embargo, se ilustran utilizando la técnica de casos de uso internos.

4.3.2 Descripción de las conversaciones

La descripción de las conversaciones se realiza desde dos puntos de vista:

› **Externo.** Se analizan el objetivo de la conversación, precondiciones, poscondiciones y participantes.

› **Estructural.** Contiene las fases de la conversación y las intervenciones que se dan en cada fase.

La descripción externa de las conversaciones puede ser gráfica, empleando casos de uso interno y textual empleando plantillas del constituyente de la conversación. Las conversaciones identificadas fueron: C1: Determina tutoría, C2: Suministra conocimientos, C3: Establece colaboración, C4: Precisa evaluación, C5: Relaciona perfil / casos, C6: Determina conocimientos y C7: Determina colaboración.

4.3.3 Descripción de los protocolos de comunicación

Una vez identificadas y descritas las conversaciones se describe de forma detallada la descomposición de las intervenciones realizadas en esta, es decir, los mensajes intercambiados. Inicialmente se realizan en lenguaje natural y progresivamente se formalizan. Una conversación se inicia para cumplir un objetivo y puede transcurrir con diferentes protocolos de comunicación (Escenarios): A continuación se presentan el protocolo de comunicación P1:

Protocolo de comunicación: P1: Determina tutoría. Compuesto de:

1. Aprendiz solicita tutoría de acuerdo a la BUL e IOs de estudio.
2. Agente Interfaz recibe la solicitud y la envía al agente Tutor.
3. Agente Tutor contacta a los agentes Experto, Colaborativo, Diagnóstico o Modelo del Aprendiz de acuerdo a la solicitud de Tutoría.

4. De acuerdo a la solicitud de tutoría, los agentes encargados responden brindando lo pedido.
5. Agente Tutor recibe conocimientos, evaluación, colaboración u otros.
6. Agente Tutor envía tutoría al agente Interfaz.
7. Agente Interfaz envía tutoría al aprendiz.

4.3.4 Descripción de las intervenciones

La descripción de las intervenciones de una conversación tiene como objetivo determinar los mensajes intercambiados entre los agentes. A continuación se realiza la descripción de las intervenciones de manera gráfica utilizando el diagrama de secuencias de mensajes (Figura 4):

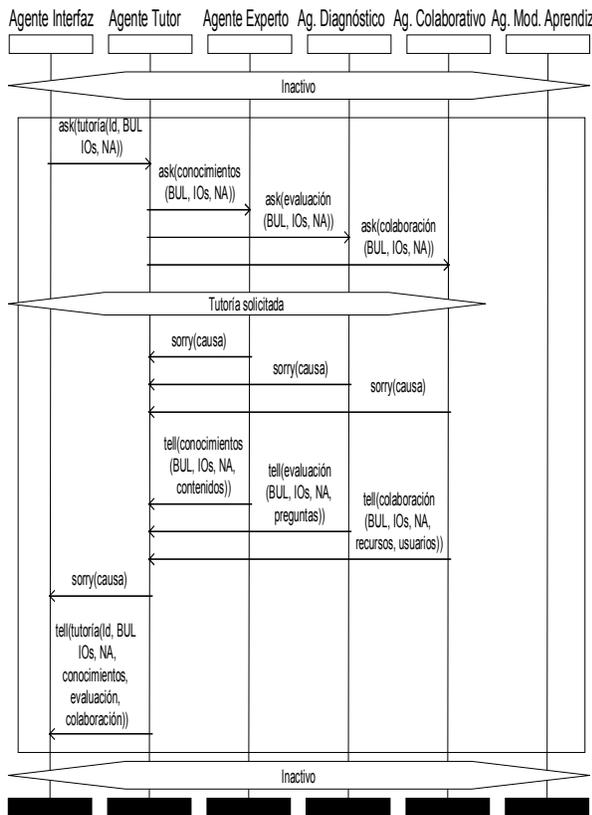


Figura 4. Diagrama de secuencia para las interacciones de la conversación C1: Determina tutoría

Figure 4. Sequence Diagram for interaction of Conversation C1: It determines Tutoring Task

4.3.5 Canales básicos de comunicación

Después de desarrolladas todas las conversaciones se determina los mensajes válidos entre agentes. A continuación se representan por medio de un diagrama (Figura 5):

4.3.6 Descripción de la conducta de la conversación

Después de determinadas las interacciones se describe la manera como se procesan los mensajes recibidos y como se envían (Figuras 5 y 6).

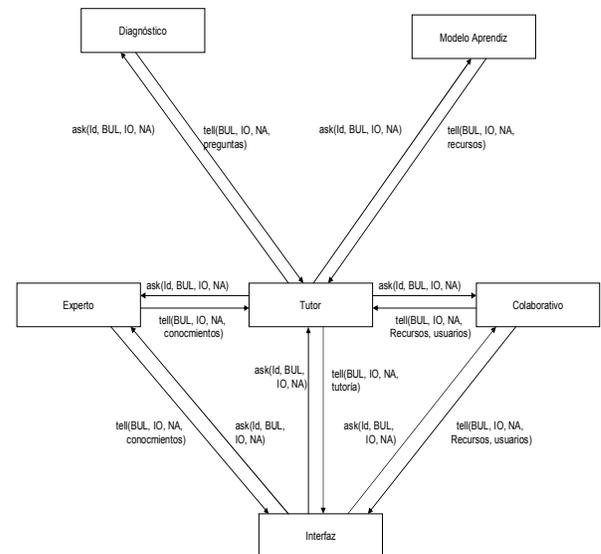


Figura 5. Canales básicos de comunicación y flujo de conversaciones entre agentes de software

Figure 5. Basic Communication Channels and Conversation Mechanism among software agents

4.3.7 Identificación de los servicios

Después de identificar las interacciones entre los agentes, se determinaron los servicios ofrecidos y requeridos por cada agente. Los servicios son facilidades que un agente ofrece a otros agentes y que pueden ser demandadas. El agente acepta la ejecución de un servicio para otros agentes, establece con ellos un compromiso de llevar dicho servicio. Este compromiso significa que el agente tiene como objetivo la ejecución del servicio. Los servicios identificados fueron: S1: Suministro de tutorías, S2: Suministro de conocimientos, S3: Brinda colaboración para

solución de problemas, S4: Brinda evaluación del aprendiz, S5: Suministra casos.

4.4 Modelo de Comunicación

El modelo de comunicación permite describir las interacciones entre un agente humano y un agente de software. Se tiene en consideración los factores humanos en las interacciones. El modelo se describe por medio de las intervenciones humano-máquina de manera gráfica utilizando el diagrama de secuencias de mensajes. A continuación (Figura 6) se presenta las interacciones genéricas de la conversación entre el aprendiz (Humano) y el agente Interfaz (Sistema), en una sesión de aprendizaje individualizado (ITS):

4.5 Modelo de la Organización

El modelo de la organización es una herramienta para analizar la organización humana en que el MAS se encuentra introducido y describe la organización de los agentes de software y su relación con el entorno. A continuación se presenta las relaciones de herencia, los objetos del entorno y la estructura organizativa (Figura 7).

4.5.1 Identificación de las relaciones de herencia

Permiten agrupar clases de agentes que comparten unas capacidades parecidas. En este caso se pueden identificar la siguiente clase de agente:

› **Básico**. Los demás agentes heredan de éste, los atributos esenciales.

4.5.2 Identificación de los objetos del entorno

En este problema no hay un entorno muy definido dado que los agentes no son físicos (p.e.: robots). Esta conclusión también puede haber sido obtenida al comprobar que no se necesitó definir sensores ni actuadores en las plantillas textuales de los agentes. Por lo tanto,

no se pueden identificar objetivos “reactivos” ante determinados eventos del exterior.

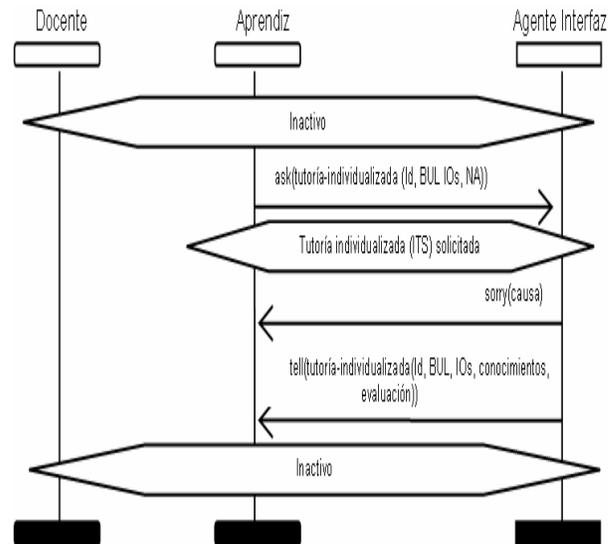


Figura 6. Interacciones genéricas de la conversación entre el aprendiz (Humano) y el agente Interfaz (Sistema), en una sesión de aprendizaje individualizado (ITS)

Figure 6. Generic Interactions of the Conversations between the Student (human actor) and Interface Agent (System), within an Individualized Learning Session (ITS)

4.5.3 Identificación de la estructura organizativa

En la figura 7 se muestra la una representación de la jerarquía de clases de agentes del sistema, donde se observa la plantilla de un agente básico del cual sus características esenciales como: protocolos de comunicación, envío, recepción y procesamiento de mensajes, comportamientos genéricos, entre otros; son heredados por todas las clases de agentes del sistema.

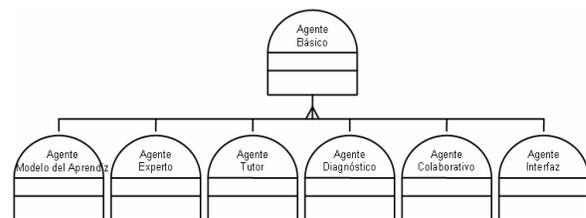


Figura 7. Jerarquía de clases
Figure 7. Classes Hierarchy

4.6 Modelo de la Experiencia

El modelo de la experiencia involucra la identificación, descripción y estructuración del conocimiento que requieren los agentes para realizar sus tareas. La actividad principal es el desarrollo del conocimiento del dominio; la descripción textual de este modelo consiste en una plantilla que describe el diagrama de conceptos.

5. CONCLUSIONES

Este trabajo aprovechó las bondades que ofrecen los Sistemas Multi-Agente (MAS), tales como la autonomía, flexibilidad y adaptabilidad para aplicarlas a un ambiente de enseñanza y de aprendizaje computacional.

La metodología MAS-CommonKADS ofrece ventajas como es el planteamiento del desarrollo de modelos que reflejan diferentes vistas del proyecto. También es relevante porque ofrece un marco para la especificación del conocimiento independiente de la implementación, combinando un conjunto de modelos de conocimiento reutilizable para unas tareas que se realizan frecuentemente.

Con este trabajo se logró enriquecer a los sistemas de enseñanza y de aprendizaje apoyados en computador en la tarea de la IP. Para ello se utilizó un modelo basado en la solución exitosa de problemas similares pasados (CBR). En este sentido, se puede afirmar que el sistema aprende en forma autónoma a partir de la experiencia con los aprendices, convirtiéndose en una herramienta más flexible, que tiene la capacidad de adaptar los conocimientos con determinado grado de abstracción dependiendo del aprendiz. Es decir, utilizando diferentes estrategias pedagógicas para la presentación de objetos de conocimiento en el sentido de visualizar la información de una manera más atractiva.

Para lograr lo anterior, se desarrolló el Ambiente Multi-Agente Pedagógico ALLEGRO a través de un MAS innovador el cual integra las virtudes de un ITS y un CSCL, reafirmando la

importancia de utilizar la inteligencia artificial en la educación para la construcción de nuevas tecnologías educativas.

REFERENCIAS

- [1] BOCCA, E.; Jaques, P.; Vicari, R. (2003) Modelagem e Implementação da Interface para Apresentação de Comportamentos Animados e Emotivos de um Agente Pedagógico Animado. En: Renote - Revista Novas Tecnologias na Educação, Vol. 1 No 2.
- [2] FLORES, C. (2005) Redes Bayesianas Multi-Agentes. Tesis Doctoral. Universidade Federal do Rio Grande do Sul.
- [3] GIRAFFA, L.; Viccari, R. (1998) The Use of Agents Techniques on Intelligent Tutoring Systems. International Conference of Chilean Society of Computer Science, Proceedings: IEEE Computer Society.
- [4] IGLESIAS, C. (1998) Definición de una Metodología para el Desarrollo de Sistemas Multi-Agentes. Tesis Doctoral. Universidad Politécnica de Madrid.
- [5] JACOBSON, I.; Christerson, M.; Jonsson, P.; Övergaard, K. (1992) Object-Oriented Software Engineering. A Use Case Driven Approach. ACM Press.
- [6] JIMÉNEZ, J. (2006) Un Modelo de Planificación Instruccional usando Razonamiento Basado en Casos en Sistemas Multi-Agente para entornos integrados de Sistemas Tutoriales Inteligentes y Ambientes Colaborativos de Aprendizaje. Tesis Doctoral. Universidad Nacional de Colombia.
- [7] JOHNSON, L.; Rickel, J. (1998) Pedagogical Agents. En: International Conference on Autonomous Agents Proceedings of the 3rd Annual Conference on Autonomous Agents.

- [8] NIST, (1993) Secure Hash Standard (SHS). National Institute of Standards and Technology.
- [9] OKAMOTO, T.; Kayama, M.; Cristea, A. (2001) Proposal for Collaborative Learning Standardization. En: Proceedings of the IEEE International Conference on Advanced Learning Techniques -ICALT'01.
- [10] OVALLE, D.; Guzmán, J.; Pastrana, M. (2005) Simulación Basada en Agentes Inteligentes del Proceso de Negociación del Mercado de Energía Eléctrica en Colombia. *Energética: Revista del Instituto de Energía, Facultad de Minas, Universidad Nacional de Colombia*, 33: 25-36.
- [11] PRESSMAN, R. (1996) Software Engineering. En: Software Engineering. IEEE Computer Society, 57-74. USA.
- [12] REGNELL, B.; Andersson, M.; Bergstrand, J. (1996) A Hierarchical Use Case Model with Graphical Representation. In: Proceedings of ECBS'96, IEEE International Symposium and Workshop on Engineering of Computer-Based Systems.
- [13] RUMBAUGH, J. (1996) OMT: The Development Model. *JOOP Journal of Object Oriented Programming*.
- [14] SILVEIRA, R. (2001) Modelagem Orientada a Agentes Aplicada a Ambientes Inteligentes Distribuídos de Ensino: JADE Java Agent framework for Distance learning Environments. Tesis Doctoral. Universidade Federal do Rio Grande do Sul.
- [15] WEBBER, C. (2003) Modélisation Informatique de l'Apprenant: Une Approche basée sur le Modèle CKC (Conception, Knowledge, Concepts) et la Théorie de l'Émergence. Thèse de Docteur. Université Joseph Fourier.
- [16] WENGER, E. (1987) Artificial Intelligence and Tutoring Systems. Los Altos, CA: Morgan Kaufmann.