

Universidad de las Ciencias Informáticas
Facultad 1



Evaluación de la metodología AAll en un caso de estudio.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autora: Leanedey de la Caridad Tur Villazán

Tutores: Dr. Ernesto González Díaz
Ing. Julio Cesar Herrera Soria

Ciudad Habana, Junio 2007

Para que exista una educación válida es necesario que se desarrolle el pensamiento crítico e independiente de los jóvenes, un desarrollo puesto en peligro continuo por el exceso de materias. Este exceso conduce necesariamente a la superficialidad y a la falta de cultura verdadera. La enseñanza debe ser tal que pueda recibirse como el mejor regalo y no como una amarga obligación.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro que yo, Leanedy de la Caridad Tur Villazán, soy el único autor de este trabajo y autorizo al Departamento de Inteligencia Artificial de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 20 días del mes de junio del año 2007.

Leanedy de la Caridad Tur Villazán

Ernesto González Díaz

Julio Cesar Herrera Soria

OPINION DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: Evaluación de la metodología AAll en un caso de estudio.

Autora: Leanedey de la Caridad Tur Villazán.

Los tutores del presente Trabajo de Diploma consideran que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

El trabajo presentado por la diplomante es el primero que se realiza en la Universidad de las Ciencias Informáticas sobre la temática de la tecnología orientada a agentes inteligentes. El mismo reviste una importancia relevante debido a que esta tecnología, según el criterio de algunos expertos, puede en poco tiempo revolucionar toda la tecnología del software. Por otro lado, debido al corte investigativo del mismo, aporta un conocimiento significativo, que en nuestra opinión será muy útil al afianzamiento y establecimiento de una Universidad que ha apostado por líneas de desarrollo novedosas.

El documento presenta un fundamentación teórica muy bien lograda donde se abordan de manera excelente los conceptos y definiciones del tema tratado. La redacción y ortografía son notablemente buenas, destacándose la capacidad léxica de la diplomante. La bibliografía es amplia, novedosa y en perfecta concordancia con los objetivos del trabajo.

Durante el desarrollo del trabajo la diplomante ha demostrado un espíritu favorable y positivo, lo que le permitió enfrentar exitosamente un trabajo sumamente complejo. Toda esta situación valoriza en gran medida el trabajo, y sobre todo dice mucho de la calidad humana de la diplomante, por lo que avizoramos un futuro profesional lleno de éxitos y logros. Es justo destacar que durante la realización de la investigación la diplomante demostró gran independencia, laboriosidad y responsabilidad, siguiendo en todo momento las indicaciones y sugerencias dadas por sus tutores y otros profesores.

Por todo lo anteriormente expresado consideramos que la estudiante esta apta para ejercer como Ingeniera Informática; y proponemos que se le otorgue al Trabajo de Diploma la calificación de 5 puntos.

20 días del mes de junio del 2007.

Dr. Ing. Ernesto González Díaz

Ing. Julio Cesar Herrera Soria

Agradecimientos

Quiero agradecer hoy a todos los que de una forma u otra, han contribuido a que uno de mis mayores sueños se haga realidad, graduarme de ingeniera.

A mi mamá, que siempre me apoya y está a mi lado, yo te quiero mucho.

A mi papá, ya tienes una hija ingeniera.

A mis hermanos, que son parte de mí.

A mi novio Daryll, porque sin él hubiera sido muy difícil llegar hasta aquí.

A mi tutor Julio César por su apoyo desde el principio.

A mi tutor Ernesto, por su dedicación, esfuerzo, paciencia y comprensión. Muchas gracias.

A Dania por su atención desinteresada.

A mis amigas Yuly, Yurina, Yanitza, Indy, Susana, Yanet y Lisset, por acompañarme estos cinco años y hacerme feliz.

A la tía Bertica, gracias por revisar mi trabajo.

A Lorena, por su indispensable ayuda.

A TODOS... MUCHAS GRACIAS.

A mis padres, porque gracias a ellos

Estoy aquí.....

*A Ernesto, por su constante apoyo
y dedicación.....*

Resumen

La tecnología orientada a agentes constituye hoy en día una de las herramientas más prometedoras de la inteligencia artificial, debido a su capacidad para modelar de manera más natural e intuitiva, las soluciones a problemas más complejos. Como toda tecnología incipiente, los sistemas multiagentes aún se encuentran en una etapa de evolución, razón por la cual han surgido una gran variedad de metodologías para el análisis y diseño de los mismos, muchas de las cuales se encuentran solamente en el marco teórico, pues no han sido validadas a través de la elaboración de un sistema multiagentes real.

Al no existir un estándar de metodología en la ingeniería de software orientada a agentes, se hace necesario llevar a cabo investigaciones que permitan establecer cuales de estas metodologías son las más completas para el desarrollo de este tipo de sistemas.

El Instituto Superior Politécnico José Antonio Echeverría (CUJAE), en conjunto con otras universidades del mundo, está inmerso en el proyecto @LIS Technology Net, del cual forma parte un caso de estudio que pretende desarrollar un sistema multiagentes que funcione como un tour operador de viajes real. El objetivo de este trabajo de diploma, es realizar una valoración de la metodología orientada a agentes AAIL (Australian Artificial Intelligence Institute), a partir de su aplicación a ese caso de estudio y de un análisis comparativo entre esta metodología y el proceso de desarrollo de software estandarizado RUP, empleando para ello los modelos y artefactos que propone AAIL en cada una de sus etapas.

La comparación entre la metodología AAIL y RUP, permitirá arribar a importantes conclusiones, ya que RUP representa un proceso estandarizado entre los desarrolladores. Finalmente, se efectuará también una evaluación de la metodología, teniendo en cuenta aspectos reflejados en las plantillas de evaluaciones de metodologías orientadas a agentes, lo que conllevará a definir las ventajas y desventajas de la misma, y a proponer una serie de mejoras que pudieran hacer de AAIL una metodología más completa.

Índice

Introducción	1
Capítulo 1 Ingeniería de Software orientada a Agentes: Metodología AAIL	6
1.1 Introducción.....	6
1.2 Agente.....	6
1.3 Funciones básicas de un Agente	9
1.4 Agentes vs. Objetos	10
1.5 Arquitectura BDI.....	11
1.6 Lenguajes y entornos para agentes.....	12
1.9 Aplicaciones multiagentes reales	16
1.10 Ingeniería de Software Orientada a Agentes	19
1.11 Metodología AAIL.....	20
1.11.1 Ciclo de vida de la metodología AAIL.....	20
1.11.2 Etapa de Análisis.....	23
1.11.3 División de la Organización	23
1.11.4 Modelado del Entorno	24
1.11.5 Modelo Preliminar de Roles	25
1.11.6 Permisos.....	26
1.11.7 Responsabilidades	27
1.11.8 Modelo preliminar de Protocolos de Interacción	28
1.11.9 Definición de la estructura organizacional.....	28
1.11.10 Etapa de Diseño de la Vista Externa.....	30
1.11.11 Definición de los Modelos de Roles y Protocolos de Interacción	30
1.11.12 Modelo de Agentes.....	31
1.11.14 Etapa de Diseño de la Vista Interna	32
1.12 Conclusiones.....	33
Capitulo II La metodología AAIL: desarrollo de un caso de estudio	34

2.1	Introducción.....	34
2.2	Descripción del proyecto @LIS Technology Net.....	34
2.3	Funcionamiento de un tour operador real	35
2.4	Análisis	36
2.4.1	División de la Organización	36
2.4.2	Modelado del entorno	37
2.4.4	Modelo preliminar de Protocolos de Interacción	43
2.4.5	Definición de la estructura organizacional del sistema.....	48
2.5	Diseño	50
2.6	Diseño de la Vista Externa	50
2.6.1	Definición de los Modelos de Roles y Protocolos de Interacción	50
2.6.2	Modelo de Agentes.....	51
2.6.3	Modelo de Clases de Agentes.....	53
2.6.4	Modelo de Instancias de Agente	53
2.6.5	Modelo de Interacción	55
2.7	Diseño de la Vista Interna	58
2.7.1	Modelos de Objetivos	59
2.7.2	Modelos de Creencias.....	62
2.8	Conclusiones.....	70
Capítulo III: Evaluación de la metodología AAIL		71
3.1	Introducción.....	71
3.2	El Proceso Unificado de Desarrollo (RUP)	71
3.4	Análisis comparativo entre la metodología AAIL y el Proceso Unificado de Desarrollo (RUP)	76
3.5	Evaluación de la metodología AAIL	80
3.5.1	Proceso de desarrollo.....	81
3.5.2	Vista de modelos	83
3.5.3	Agentes	86
3.5.4	Modelado de características adicionales.....	88

3.5.5 Documentación.....	88
3.5.6 Otros elementos	89
3.6 Conclusiones.....	91
Conclusiones	92
Recomendaciones	94
Bibliografía Citada	95
Bibliografía Consultada	98
Glosario de Términos.....	100
Anexos.....	101

Índice de Figuras

Figura 1. 1: Visión esquemática de un agente	8
Figura 1. 2 : Funciones de un agente	10
Figura 1. 3 : Sistema Multiagentes	14
Figura 2. 1 : Modelo de Clases de Agentes.....	53
Figura 2. 2 : Modelo de Instancias de Agente	55
Figura 2. 3 : Modelo de creencias del agente Interfaz	62
Figura 2. 4 : Modelo de creencias del agente Personal.....	62
Figura 2. 5: Modelo de creencias del agente Global.....	63
Figura 2. 6 : Modelo de creencias del agente Vuelos	63
Figura 2. 7 : Modelo de creencias del agente Local.....	64
Figura 2. 8 : Modelo de creencias del agente Actividades	64
Figura 2. 9 : Modelo de creencias del agente Alojamiento	65
Figura 2. 10 : Modelo de creencias del agente Transporte	65
Figura 2. 11 : Modelo de planes del agente Interfaz	66
Figura 2. 12 : Modelo de planes del agente Personal.....	66
Figura 2. 13 : Modelo de planes del agente Personal.....	67
Figura 2. 14 : Modelo de planes del agente Vuelos	67
Figura 2. 15 : Modelo de planes del agente Local	68
Figura 2. 16 : Modelo de planes del agente Actividades	68
Figura 2. 17 : Modelo de planes del agente Alojamiento.....	69
Figura 2. 18 : Modelo de planes del agente Transporte	69
Figura 3. 1 : RUP en dos dimensiones	73
Figura 3. 2 : Vista del Modelo de Arquitectura.....	74

Índice de Tablas

Tabla 1: Recursos del entorno.	38
Tabla 2 : Rol Interfaz.....	39
Tabla 3 : Rol Registrar	40
Tabla 4 : Rol Ratificar_ Itinerario	40
Tabla 5 : Rol Itinerario_ Global.....	41
Tabla 6 : Rol Vuelos	42
Tabla 7 : Itinerario_ Local	43
Tabla 8 : Rol Actividades	43
Tabla 9 : Protocolo de Interacción Registrar_ Usuario	44
Tabla 10 : Protocolo de interacción Autenticar_ Usuario	44
Tabla 11 : Protocolo de interacción Solicitar_ Itinerario.....	45
Tabla 12: Protocolo de interacción Solicitar_ Itinerarios_ Destino	46
Tabla 13 : Protocolo de interacción Solicitar_ Vuelos	46
Tabla 14 : Protocolo de interacción Solicitar_ Actividades	47
Tabla 15 : Protocolo de interacción Solicitar_ Alojamiento.....	47
Tabla 16 : Protocolo de interacción Confirmar_ Transporte	48
Tabla 17 : Vista jerárquica de los roles y protocolos de interacción.	51
Tabla 18 : Relación entre Roles, Clases de Agentes e Instancias de Agentes.....	52
Tabla 19 : Artefactos fundamentales de la metodología AAIL.....	84

Introducción

La experiencia ha demostrado la importancia del uso de metodologías en el proceso de desarrollo de software. La Ingeniería de Software (IS) es la utilización de principios de ingeniería sólidos, encaminados a obtener un producto económico fiable y con un funcionamiento eficiente sobre máquinas reales. Esta ha estado en constante evolución, transitando por dos etapas fundamentales: el enfoque Estructurado y el enfoque Orientado a Objetos (OO).

Dentro de la ingeniería de software orientada a objetos existe un proceso que por su flexibilidad y la efectividad de las soluciones que propone, se ha estandarizado entre los desarrolladores, el Proceso Unificado de Desarrollo (RUP)(IVAR JACOBSON, 2000), que utiliza como lenguaje de modelado UML (Lenguaje Unificado de Modelado)(MARTIN FOWLER, 2000). Este cuenta, además, con una herramienta de soporte, Rational Rose, que sustenta todas las etapas que propone RUP y brinda una serie de posibilidades que facilitan el trabajo de los miembros del equipo.

El desarrollo de la Informática ha traído aparejado una mayor complejidad en los problemas a solucionar. Actualmente la Programación Orientada a Objetos, aunque ha demostrado su robustez, no puede dar soluciones óptimas a todos los problemas, como es el caso de los sistemas distribuidos, lo cual ha dado lugar al surgimiento de un nuevo paradigma, el de los Agentes Inteligentes.(VICENTE JULIAN, 2003)

Uno de los conceptos más completo de agente, es en el que se define como una entidad de software con un grupo de propiedades, entre las que se destacan: ser capaz de actuar en un ambiente, comunicarse directamente con otros agentes, estar condicionado por un conjunto de tendencias u objetivos, manejar recursos propios, percibir su ambiente y tomar de él una representación parcial, ser una entidad que posee habilidad, ofrece servicios, y que puede reproducirse.(ESPINO, 2005)

A pesar de que todavía es un tema de profunda investigación, ya se han dado los primeros pasos, desarrollándose algunas aplicaciones exitosas, como son los sistemas para el control de tráfico aéreo, comercio electrónico, gestión de información, aplicaciones médicas y juegos, entre otras ramas.

La creación de sistemas utilizando agentes aumenta la complejidad de los mismos, así como la necesidad de adaptar técnicas ya existentes, o en ocasiones, el desarrollo de técnicas y herramientas nuevas. Ha quedado demostrado que para hacer cualquier tipo de software se necesitan métodos y herramientas que faciliten la obtención de productos finales fiables, y los agentes no son la excepción. En la última década han aparecido diferentes metodologías que proponen procesos para el desarrollo de Sistemas Multiagentes (SMA), lo que ha provocado un auge en la investigación de este paradigma en la rama de la Ingeniería de Software.

En contraste con la historia de las metodologías orientadas a objetos, donde la industria jugó y está jugando un papel fundamental, la mayoría de las metodologías orientadas a agentes, son propuestas de investigaciones académicas que no cuentan con la madurez suficiente. Cada una aborda el problema desde su punto de vista y no hay suficientes aplicaciones reales que prueben su eficacia en la solución de situaciones.

Actualmente en nuestro país, el conocimiento sobre estas metodologías es muy escaso, sin embargo, la CUJAE está trabajando en un proyecto con universidades de varios países para desarrollar un sistema que simule el trabajo de un turoperador de viajes. Este promueve la gestión cultural y turística entre América Latina y Europa, además de que constituye el antecedente principal en el desarrollo de sistemas multiagentes en Cuba. Como parte de este proyecto se han realizado valoraciones de un conglomerado de metodologías de análisis y diseño orientadas a agentes, entre las que se encuentran Prometheus, Gaia, Mase, Tropos e Ingenias. Este estudio no incluyó la metodología AAIL, por lo que se hace imprescindible llevar a cabo esta investigación, a partir de su aplicación a un caso de estudio y la comparación de la misma con RUP, debido a la estandarización de este último.

Esta comparación resultará beneficiosa, ya que se identificarán los aspectos que tienen en común y en los que difieren, con la intención de arribar a conclusiones que sirvan de base para trabajos posteriores, donde se tenga como objetivo mejorar la metodología AAIL o aplicarla a nuevos problemas.

En sentido general se contribuirá a que se conozca con mayor profundidad en nuestro país la existencia de esta metodología, e incentivar su estudio y aplicación en sistemas

que utilicen agentes inteligentes, marcando así un punto de partida en la investigación de este paradigma y las metodologías que lo sustentan, en la UCI.

A partir de la descripción anterior, se arriba a la siguiente **situación problémica**:

Se carece de una valoración de la metodología AAll aplicada a un caso de estudio, que permita determinar las ventajas y desventajas que esta proporciona para el análisis y diseño de sistemas multiagentes. Esto provoca que cuando se pretenda desarrollar un sistema multiagentes en la UCI, no se tenga una valoración adecuada de las metodologías que se pudieran emplear, lo que trae como consecuencia que el proyecto pueda tener atrasos, que no se determinen correctamente los requerimientos funcionales del mismo, o que la tecnología empleada no sea la ideal.

Esta situación conduce a formular como **problema científico**: Necesidad de una evaluación y un estudio comparativo de la metodología de análisis y diseño para sistemas multiagentes AAll.

En afán de solucionar el problema, se plantea como **idea a defender**: Contar con una evaluación de la metodología de análisis y diseño orientada a agentes, AAll, permitirá realizar una adecuada selección de la metodología a emplear en el desarrollo de sistemas multiagentes, teniendo en cuenta que este estudio ya se ha realizado para otras metodologías orientadas a agentes como GAIA, TROPOS, Ingenias, Mase, entre otras.

El **objeto de estudio** de esta investigación lo constituyen las metodologías de análisis y diseño para sistemas multiagente; y como **campo de acción**, la metodología de análisis y diseño orientada a agentes AAll.

Para dar solución al problema planteado se traza como **objetivo general**: Realizar una valoración de la metodología AAll a partir del desarrollo de un caso de estudio y de un análisis comparativo entre esta metodología y el proceso de desarrollo de software estandarizado RUP.

Y los siguientes **objetivos específicos**:

- Analizar aspectos teóricos conceptuales relacionados con la Ingeniería de software orientada a agentes inteligentes.
- Desarrollar el análisis y diseño del sistema multiagentes “Tour operador de viajes”, empleando la metodología orientada a agentes AAIL.
- Comparar esta metodología con otra metodología estandarizada como RUP.

Para dar cumplimiento a los objetivos definidos anteriormente, se llevarán a cabo las siguientes **tareas de investigación**:

- Constatar la existencia del problema propuesto mediante métodos científicos.
- Realizar una búsqueda bibliográfica que permita establecer las bases teóricas de la investigación y los antecedentes de la misma.
- Desarrollar la metodología de análisis y diseño orientada a agentes AAIL, a través de un caso de estudio.
- Llevar a cabo una comparación entre la metodología AAIL, y la metodología estandarizada RUP.

La siguiente tesis de diploma esta estructurada en tres capítulos fundamentales:

Capitulo I: Ingeniería de Software Orientada a Agentes: metodología AAIL.

En este capítulo se definirán los conceptos generales de agente, sistema multiagentes e ingeniería de software orientada a agentes. Luego se detallarán cada una de las etapas y modelos de la metodología AAIL, que servirán de base para el próximo capítulo.

Capítulo II: La metodología AAIL: desarrollo de un caso de estudio.

Sobre la base de lo planteado en el capítulo 1, se seguirán los pasos que establece la metodología AAIL, para aplicarla al caso de estudio “Tour operador de viajes”.

Capítulo III: Evaluación de la metodología AAIL.

En el tercer capítulo se establece una comparación entre las metodologías AAIL, como propuesta del nuevo paradigma de ingeniería de software orientado a agentes, y el Proceso Unificado de Desarrollo (RUP), estándar de referencia del actual paradigma orientado a objetos. Del mismo modo se evalúa la metodología AAIL de acuerdo a determinados aspectos, esenciales en la evaluación de metodologías orientadas a agentes.

Capítulo 1 Ingeniería de Software orientada a Agentes: Metodología AAll.

1.1 Introducción

La inteligencia artificial es la ciencia de la computación que se dedica al estudio de la conducta inteligente en sistemas computacionales, y también al desarrollo de software y hardware que da solución a problemas para los cuales no existen algoritmos, o los que existen son de complejidad no polinomial. Además, tiene como meta construir sistemas que comprendan el lenguaje natural, que piensen de un modo creativo, que entiendan escenas y que usen el sentido común. La Inteligencia Artificial Distribuida (IAD) es la rama de la Inteligencia Artificial (IA) que trata de la resolución de problemas de manera cooperativa en un cierto ambiente, por medio de entidades denominadas agentes.

Hay varias técnicas que permiten el desarrollo de sistemas inteligentes, entre las que se destacan los sistemas de bases de conocimiento, las redes neuronales, los sistemas adaptativos, y los agentes inteligentes, que cuando se integran conforman un sistema multiagentes.(GARCIA-SERRANO, 1998) Para sustentar el desarrollo de las escasas aplicaciones pertenecientes a este nuevo paradigma de la ingeniería de software moderna, se han planteado nuevas metodologías de análisis y diseño orientadas a agentes. El presente capítulo propone un acercamiento a una de ellas, AAll, pasando primero por una detallada descripción de qué son, cómo funcionan y a qué se aplican los agentes inteligentes.

1.2 Agente

Los agentes inteligentes surgen como una alternativa para dar solución a situaciones que la Programación Orientada a Objetos (OO) no puede resolver eficientemente. Este paradigma constituye en la actualidad un área de profunda investigación, lo que trae como consecuencia que existan un gran número de criterios acerca de qué son los agentes y sus características fundamentales.

El tema de los agentes ha despertado un gran interés tanto a nivel académico como industrial. Pero, ¿qué es realmente un agente? Su definición ha resultado ser muy

controvertida, existen numerosas propuestas del concepto de agente, pero ninguna de ellas ha sido aceptada a plenitud por la comunidad científica. Una definición es la de Russell (RUSSELL, 1996), que considera al agente como una entidad que percibe y actúa sobre un entorno. Otra de las definiciones, y que resulta ser una de las más citadas, es la propuesta por Wooldridge y Jennings (M. WOOLDRIDGE, 1995), que plantean que un agente es un sistema informático que está situado en un entorno y es capaz de actuar de forma autónoma y sensible. Según EURESCOM, un agente es una entidad única situada en un ambiente donde interactúa con objetos y con otros agentes. Tiene una identidad que lo distingue de su entorno de desarrollo, y un estado que viene a ser la representación de este. Es responsable para satisfacer ciertas metas y puede planear sus acciones para lograrlas. Es capaz de observar las propiedades (el estado) de ciertos objetos en su ambiente y de percibir algunos eventos que suceden en el mismo, además de realizar ciertas acciones que pueden modificarlo, ya sea de manera directa, cambiando las propiedades de objetos o de manera comunicativa, enviando mensajes con el objetivo de afectar las actitudes mentales de otros agentes.(EURESCOM, 2000)

Un Agente inteligente es un sistema informático capaz de tener un comportamiento autónomo, flexible en algunos ambientes. Cuando se dice flexible se refiere a:

- Reactivo: el agente es capaz de responder a cambios en el entorno (que puede ser el mundo físico, un usuario a través de una interfaz gráfica, una colección de otros agentes, Internet, o todos combinados) en que se encuentra situado.
- Pro-activo: el agente debe ser capaz de tomar la iniciativa para intentar cumplir sus propios planes u objetivos.
- Social: capacidad de un agente para interactuar por decisión propia, en procesos de negociación o cooperación con otros agentes (software/personas), mediante algún tipo de lenguaje de comunicación de agentes.

En la figura 1 se puede ver una visión esquemática de un agente.(V. JULIAN, 2000)

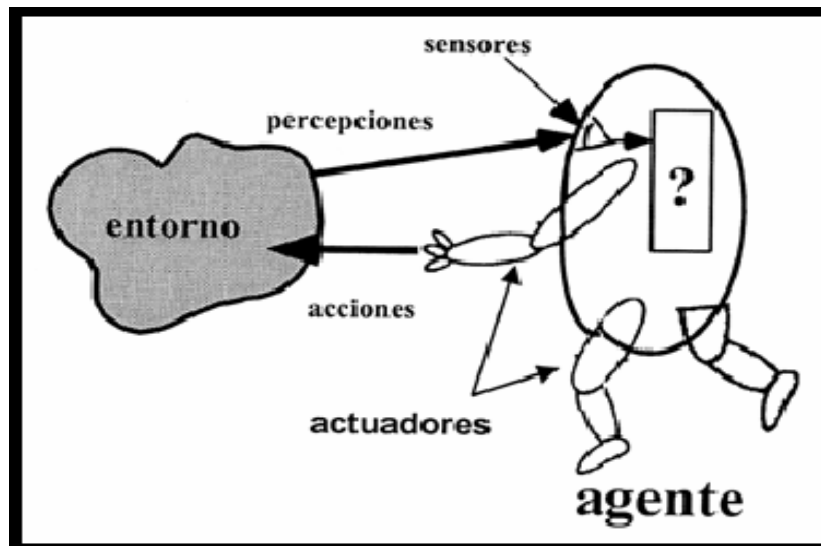


Figura 1. 1: Visión esquemática de un agente

Además de las características antes expuestas, existen en la literatura una serie de características atribuidas a los agentes para resolver problemas particulares, que han sido descritas por los autores Franklin, Graesser (GABRIEL HERNAN TOLOSA, 1999) y Nwana (NWANA, 1996):

- **Autonomía:** un agente es completamente autónomo si es capaz de actuar basándose en su experiencia, y de adaptarse aunque el entorno cambie severamente. Este no está pasivamente sujeto a su entorno, ni a un control externo de sus acciones, es decir, posee su propia información interna para actuar, orientado a la arquitectura de su tarea específica, y puede decidir por el mismo qué acción debe tomar y en qué momento hacerlo.
- **Racionalidad:** un agente actúa de forma racional, intentando cumplir sus objetivos, si estos son viables, y no actuará de manera que impida que se cumplan, al menos en el grado en que se lo permitan los datos que percibe del entorno.

- Adaptatividad: está relacionada con el aprendizaje que un agente es capaz de realizar y si puede cambiar su comportamiento basándose en ese aprendizaje.
- Continuidad Temporal: se considera a un agente como un proceso sin fin, ejecutándose continuamente y desarrollando su función.
- Movilidad: capacidad de un agente de trasladarse a través de una red telemática o mundial (Internet).
- Benevolencia: Se asume que un agente está dispuesto a ayudar a otros agentes si esto no entra en conflicto con sus propios objetivos.
- Veracidad: un agente no comunica información falsa a propósito. (MÁRQUEZ, 1998)

1.3 Funciones básicas de un Agente

Un agente software se caracteriza por llevar a cabo las siguientes funciones:

- Ejecución de Tareas: La capacidad de realizar tareas son las destrezas que el agente posee para lograr sus objetivos.
- Conocimiento de su entorno: El conocimiento del entorno debe de ser introducido o construido por el desarrollador. Algunos agentes, además, adquieren su conocimiento por aprendizaje.

- Capacidad de Comunicación: Se distinguen dos tipos de comunicación, la interacción con el usuario, y la que se refiere a comunicación inter-agente. (NORES, 2001)



Figura 1. 2 : Funciones de un agente

1.4 Agentes vs. Objetos

En ocasiones se puede confundir un agente con un objeto, debido a que estos últimos tienen funcionalidades propias, se definen como entidades computacionales que encapsulan un estado, son capaces de realizar acciones o métodos en ese estado y se comunican por paso de mensajes. Pero los agentes no son objetos, y de la misma forma que existe una relación clara entre ambos conceptos, también se observan diferencias destacables. La primera la encontramos en el grado de autonomía que uno y otro concepto presentan. Un objeto posee autonomía sobre su estado y tiene control sobre él, sin embargo, no tiene ningún control sobre su comportamiento. Si un objeto contiene un método público, cualquier objeto será capaz de invocar dicho método, cuando quiera y cuantas veces quiera. Los agentes tienen una mayor noción de autonomía, ya que deciden por ellos mismos si realizar o no una acción en respuesta a una solicitud de otro agente.

Por otra parte, los agentes son inteligentes, capaces de mostrar un comportamiento flexible (reactivo, pro activo, social), y el modelo de objeto estándar no incorpora este tipo de comportamiento. Otra característica que los distingue de los objetos es que son activos, ya que un sistema multiagentes es inherentemente multihilo, en el cual se asume que cada agente posee al menos un hilo de control activo. Los objetos, en general, permanecen en letargo hasta que otro objeto invoca alguno de los servicios que ofrecen. En fin, se puede afirmar que los objetos están compuestos por datos y funcionalidades, son pasivos expectantes y obedientes, y los agentes están compuestos por datos, funcionalidades y toman decisiones, son más flexibles que meros respondedores de órdenes. (JAQUERO, 2005)

1.5 Arquitectura BDI

La arquitectura BDI (*belief, desire, intention*: creencia, deseo, intención) propuesta por Rao y Goergeff ha sido el modelo más difundido y estudiado para modelar agentes. Este ve a los agentes como sistemas intencionales cuyo comportamiento puede predecirse y explicarse en términos de actitudes, de manera similar a cómo se predice el comportamiento de los sistemas inteligentes complejos (personas). Se entiende por creencia la información disponible para un agente, la cual es tomada del entorno y puede verse como su entrada o lo que conoce del mundo. Para guiar sus acciones el agente tiene deseos, los cuales constituyen actitudes motivadoras capturadas en objetivos, que representan estados que el agente trata de alcanzar a través de intenciones. Estas últimas son actitudes deliberativas que representan los planes con los que el agente trata de conseguir sus objetivos. Además, este debe ser capaz de llevar la cuenta de las acciones realizadas y de los subobjetivos conseguidos.

Para especificar un sistema de agentes, se emplean un conjunto de modelos que operan a dos niveles de abstracción: externo e interno.

- Desde el punto de vista externo, un sistema se modela como una jerarquía de herencia de clases de agentes, de la que los agentes individuales son instancias. Las clases de agentes se caracterizan por su propósito, sus responsabilidades, los servicios que ejecutan, la información acerca del mundo que necesitan y las interacciones externas.

- Desde el punto de vista interno, se emplean un conjunto de modelos (modelos internos) que permiten imponer una estructura sobre el estado de información y motivación de los agentes y las estructuras de control que determinan su comportamiento (objetivos, creencias y planes en este caso).

Otro modelo alternativo al BDI es el de Cohen-Levesque, en el cual las actitudes primitivas son solo dos (creencias y metas) y las otras se derivan de estas. BDI es el modelo teórico más conocido, aunque en la práctica puede no adoptarse un modelo tan teórico para desarrollar un sistema práctico, sino que se asigna a cada agente un conjunto de propiedades que varían su nombre según la metodología que se utilice como son GAIA (permisos, responsabilidades, actividades), AAIL, (creencias, metas y planes, servicios, responsabilidades o roles) y SODA (permisos, actividades, tareas, reglas de interacción), entre otras. Estas decisiones y nombres, en ocasiones tienen que ver con la orientación hacia una plataforma o lenguaje de desarrollo que tenga un mayor o menor acercamiento a BDI.(ZULMA CATALDI, 2006)

1.6 Lenguajes y entornos para agentes

La aparición de cierto número de prototipos de lenguajes de agente es signo de que esta tecnología empieza a ser más ampliamente utilizada, y que en un futuro próximo habrán muchas aplicaciones basadas en agentes. Por lenguaje de agente se entiende: un sistema que permite programar sistemas de hardware o de software en términos de algunos de los conceptos desarrollados por las teorías de agente.

Un agente puede implementarse en cualquier lenguaje, si se asume que son sus propiedades lo fundamental y no los detalles. Sin embargo, es importante destacar que hay lenguajes más puros, orientados a agentes, y que usan BDI como sustento. Tal es el caso de 3APL, mientras que hay otros que se dedican a extender lenguajes de otros paradigmas (fundamentalmente del paradigma OO o el lógico), para que incluyan las potencialidades de la comunicación presentes en los agentes. Ejemplo de esto es el Java Agent Development Framework (JADE), que extiende a Java, Jinni o JavaLog que extienden a Prolog, entre otros. También han surgido extensiones de otros paradigmas

para entornos de agentes, como es DROOLS o JESS que permiten crear Sistemas Expertos en Java, para alimentar de inteligencia a la comunidad de agentes. (ESPINO, 2006)

1.7 Sistemas multiagentes

Los agentes pueden ser usados como entidades independientes, en las que son delegadas funciones o tareas específicas, sin embargo, en la mayoría de los casos, los agentes existen en un entorno donde habitan otros agentes. En estos entornos multiagentes, el comportamiento del sistema deviene como la interacción entre todos los agentes que lo conforman, lo cual brinda un concepto fundamental para los sistemas orientados a agentes, la socialización; entendida como la interacción de un agente (cooperativa, coordinada o negociada) con otros, para lograr un objetivo común, o como una necesidad para alcanzar sus propios objetivos. Un Sistema Multiagentes (SMA) puede definirse, entonces, como grupos de agentes que interaccionan entre sí para conseguir objetivos comunes.

La construcción de SMA integra tecnologías de distintas áreas del conocimiento: técnicas de ingeniería de software para estructurar el proceso de desarrollo; técnicas de inteligencia artificial para dotar a los programas de capacidad para tratar situaciones imprevistas y tomar decisiones, y programación concurrente y distribuida para tratar la coordinación de tareas ejecutadas en diferentes máquinas, bajo diferentes políticas de planificación. Debido a esta combinación de tecnologías, el desarrollo de estos sistemas se complica.(FERNÁNDEZ, 1998)

Cuando se define un Sistema Multiagentes hay que tener en cuenta las características siguientes(J.FERBER, 1999):

- Entorno.
- Conjunto de objetos que se encuentran integrados con el ambiente. Dichos objetos son pasivos, pueden ser creados, modificados y destruidos por agentes.
- Conjunto de agentes que representan las entidades activas del sistema.
- Conjunto de operaciones que hacen posible el trabajo de los agentes sobre los objetos.

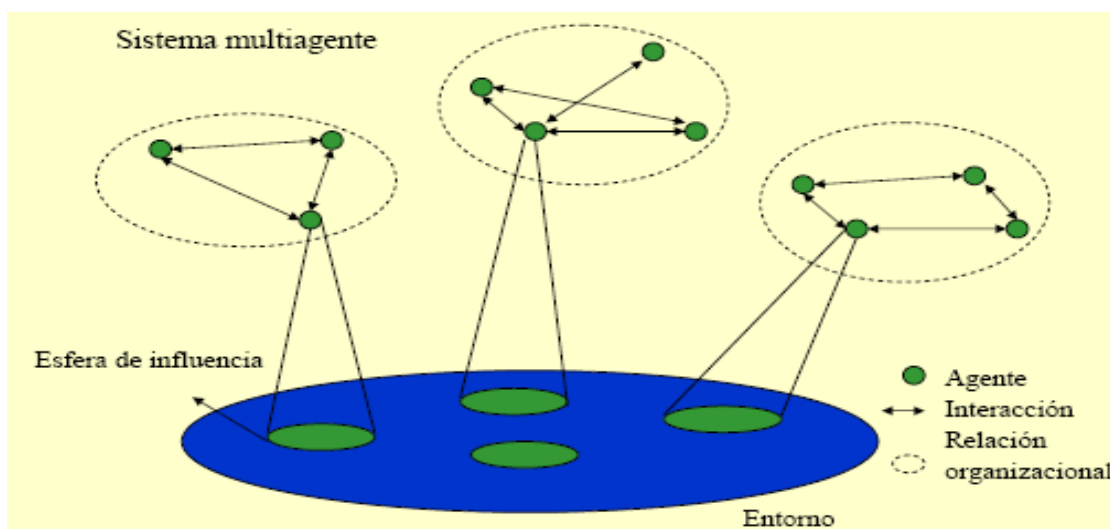


Figura 1. 3 : Sistema Multiagentes

De forma general, es posible distinguir dos clases principales de sistemas multiagentes:

- Los sistemas cerrados: en los cuales cada agente está específicamente designado para cooperar en el logro de un objetivo general.
- Los sistemas abiertos: en los cuales los agentes no se diseñan para compartir un objetivo común, sino que probablemente sean desarrollados de formas diferentes y por tanto, persiguen disímiles objetivos, mientras que por otra parte, la composición de los sistemas está en constante dinamismo, con la salida e incorporación de varios agentes.

La construcción de sistemas más reciente, incluye aplicaciones mucho más distribuidas en un entorno abierto, en el que cada agente tiene que sacar provecho del uso de sus servicios, su conocimiento y su capacidad de encontrar a otros agentes diseminados a través de una red, para alcanzar un flujo de trabajo eficiente.(ZAMBONELLI, 2003)

1.8 Lenguaje para el modelado de Sistemas Multiagentes (AUML)

Agent UML (AUML) surge como extensión de un conjunto de artefactos de UML para ser empleado en el proceso de desarrollo de los SMA, ya que UML es insuficiente para modelar los agentes debido a las características especiales que estos presentan. Esta nueva propuesta intenta unir lo desarrollado en materia de metodologías de desarrollo de software de agentes, con los estándares definidos para el desarrollo de software Orientados a Objetos. Los autores (Bauer, Odell y otros) sugieren una representación en tres capas, denominada Agent Interaction Protocol (AIP):

- 1ra: Representa los protocolos de comunicación y se utilizan los Paquetes y los Templates de UML para la especificación de estos protocolos.
- 2da: Muestra la interacción de los agentes usando Diagramas de Interacción UML.
- 3ra: Muestra el procesamiento interno de los agentes por medio de los Diagramas de Actividad y de Estado.

Los Diagramas de Secuencia son candidatos básicos para modelar la comunicación entre los agentes y han sido los que más se han modificado, ya que no describen de forma satisfactoria estas complejas interacciones. AUML introduce el Diagrama de Protocolo, que extiende el Diagrama de Secuencia de varias formas, específicamente en: roles de agentes, semántica de los mensajes, parametrización anidada de los protocolos, las plantillas de los protocolos y los hilos de ejecución en la línea de vida. Este último es considerado uno de los aportes más significativos y complejos, que provoca transformaciones considerables en la línea de vida de los objetos (agentes), a partir del momento en que esta depende de los mensajes que van arribando.

Son útiles también los diagramas de Estado, pero estos no pueden describir el comportamiento de un grupo de objetos que se encuentran cooperando entre sí, característica básica de los SMA. La notación de los diagramas de estado se combina con los diagramas de Secuencia para la especificación de los protocolos de interacción.(SIMÓN, 2005)

1.9 Aplicaciones multiagentes reales

Según Jennings, los paradigmas de software pasan por tres etapas. En la primera solo hay un grupo de pioneros y visionarios que identifican una tecnología para realizar cosas. En la segunda etapa hay un conjunto de individuos e instituciones que adoptan tempranamente la tecnología, reconocen su potencialidad y la aplican para desarrollar sistemas. Por último en la tercera etapa se produce un reconocimiento amplio de la tecnología y se usa en el desarrollo de distintos software. Para este autor la tecnología de agentes en el año 2000, estaba firmemente en la segunda etapa y comenzando el tránsito a la tercera.

En este tránsito han influido la variedad de aplicaciones exitosas con dicha tecnología, como son los sistemas para:

- Control de procesos: Esta es una aplicación natural para los agentes inteligentes, ya que son autónomos y reactivos, por lo que muchas aplicaciones de control de procesos se basan en agentes. ARCHON es un ejemplo de plataforma de software para la construcción de sistemas multiagentes que se aplica en la gestión del transporte de electricidad. Otros procesos que son controlados por sistemas multiagentes son los de monitorización y diagnóstico de fallos en plantas industriales, como son nucleares o refinerías, además del control de un acelerador de partículas, el proceso de bobinado del acero y la robótica.
- Control de tráfico aéreo: Kinny (1996) describe un sofisticado agente para la realización de un sistema de control aéreo, conocido como OASIS. Es un simulador que recrea las actividades entre aeropuertos reales.
- Comercio electrónico: En este caso los agentes se emplean para proporcionar el entorno virtual donde realizar posibles operaciones comerciales (compra-venta de productos), o también para realizar búsquedas de productos (comparando precios,

consultando disponibilidad) que reúnan las características impuestas por el usuario, todo ello de manera automatizada. En este caso, el agente debe poder comunicarse con las tiendas en línea utilizando protocolos que permitan trabajar con las interfaces de estas. El empleo de esta tecnología aumentará el impacto del comercio electrónico en un futuro cercano, revelando asimismo cómo los agentes basados en la web pueden proporcionar un enorme poder añadido a los consumidores.

- **Juegos y Entornos Virtuales:** En estas aplicaciones, las interacciones entre los agentes permiten construir un entorno virtual donde el usuario se relaciona a través de su agente con los otros que conviven en ese escenario, ya sea para realizar juegos o simuladores, con el objetivo de lograr un alto realismo. Estos agentes se suelen denominar agentes “creíbles” es decir, que proporcionan la ilusión de vida real. En el campo de juegos, podemos mencionar el juego *Creatures* (Grand, 1998), basado en agentes inteligentes que interactúan con el usuario en tiempo real. Son animales domésticos que van desarrollándose en función de la experiencia durante el juego. *Wavis* (1996) también describe muchas aplicaciones para juegos que utilizan la tecnología de los agentes. También existe una versión del popular juego *Tetris*, en el que el agente se pone de parte del jugador cuando ve que éste no puede colocar bien los bloques. Para esto se utiliza un agente reactivo.
- **Gestión de Información:** Los agentes son fundamentales para reducir la sobrecarga de información, ya que los usuarios necesitan información útil y no un montón de datos. Por ejemplo el filtrado inteligente de correo electrónico, de grupos de noticias o la recopilación automática de información disponible en la red. Para realizar estas tareas el agente necesita ser capaz de almacenar, aprender y manipular las preferencias y gustos de cada usuario, así como sus cambios. La imposibilidad en ocasiones de gestionar todo tipo de información suministrada por la red, ha provocado que el agente se especialice en la búsqueda de determinados

tipos de documentos. Otra posible línea sería la planificación de la agenda personal, es decir, disponer de una secretaría virtual o asistente personal.

- **Gestión bibliotecaria:** Mediante la utilización de Agentes Inteligentes estos sistemas tutorizan y monitorizan de alguna manera la acción del usuario. Es posible, de este modo, crear una especie de "historial digital" del usuario, donde queden reflejadas sus necesidades de información. La información proporcionada por este Agente Inteligente le permite al sistema realizar de forma automática las tareas acordes con las necesidades del usuario, así como sugerirle nuevos patrones de búsqueda que refinen la investigación que este está realizando, incluso constituirse como asistente durante el proceso de búsqueda, cuando se detecte que el usuario está utilizando de forma errónea o poco eficiente el módulo de consultas.
- **Agentes personales para viajes:** Análogamente a los agentes de viajes humanos, que nos ayudan a encontrar lo mejor de lo que necesitamos, un usuario le dice a su agente personal para viajes qué es lo que quiere conseguir, qué limitaciones tiene, y cuáles son sus preferencias, y este se encarga de encontrar lo más adecuado. Todo este trabajo requiere conocimiento especializado. Por ejemplo, un buen agente de viajes nos ayuda a encontrar dónde pasar unas vacaciones, cómo llegar, cuándo viajar, actividades que puedo realizar en ese lugar, etc. Localizar información relevante en la World Wide Web está alcanzando la misma complejidad que preparar un viaje o comprar una casa, por tanto se necesita un software capaz de hacer a través de Internet lo que un agente de viajes hace en el mundo real.(MÁRQUEZ, 2005)

En los últimos años se han desarrollado dos grandes proyectos orientados a mostrar la potencialidad de los agentes. Uno de ellos llamado Agencities, lo desarrolla una comunidad de varias decenas de ciudades, y el otro @LIS Technology Net, promueve el uso de esta tecnología para la promoción cultural y turística entre América Latina y Europa, del cual forma parte la Universidad Politécnica de Cataluña en España, en

conjunto con un grupo de trabajo del Instituto Superior Politécnico José Antonio Echeverría (ISPJAE). Actualmente el grupo del ISPJAE como parte de este proyecto, ha realizado una valoración de un conjunto de metodologías orientadas a agentes, a partir del desarrollo de un caso de estudio, abordándose las metodologías GAIA, Tropos, Ingenias, PASSI, Mase, y Message. Este estudio no incluyó la metodología AAIL, por lo que se consideró oportuno desarrollar una investigación de la misma, a partir también del mismo caso de estudio, con vistas a incrementar los conocimientos sobre las metodologías orientadas a agentes y poder hacer una selección más rigurosa cuando se pretenda desarrollar un proyecto basado en esta tecnología. Debido a esto se decide que sea AAIL y no otra, la metodología que se aborde en este trabajo de diploma.

En la UCI, existe una línea de investigación, desarrollada por la Facultad 3, que tiene como objetivo el estudio de las tecnologías basadas en inteligencia artificial. Esta línea tiene varios proyectos de desarrollo, que trabajan con sistemas basados en el conocimiento, redes neuronales, algoritmos metaheurísticos, pero en ninguno trabajan con tecnología orientadas a agentes, por lo que este trabajo constituiría la base para futuros proyectos que se desarrollen en la universidad y que pretendan emplear esta tecnología.

La complejidad que encierra el desarrollo de SMA, sobre todo cuando es muy grande, hace necesario el uso de metodologías que estructuren el proceso de acuerdo con las prácticas de ingeniería de software.

1.10 Ingeniería de Software Orientada a Agentes.

Con el surgimiento de los Sistemas Multiagentes la Ingeniería de Software ha tenido que enfrentar nuevos retos. Las metodologías existentes para el desarrollo de software no cubrían las exigencias para modelar un SMA, ya que las características inherentes a los agentes, tales como la autonomía y la proactividad, aumentan la complejidad en la construcción de los sistemas. Debido a esto surgen diferentes metodologías, que sustentan la modelación de sistemas orientados a agentes, pero no se ha logrado una estandarización, ya que cada una propone una forma diferente de abordar el problema según los criterios de sus desarrolladores.

Estas metodologías provienen de dos vertientes fundamentales: aquellas que son totalmente nuevas y las que surgen como extensiones de las ya existentes. Las primeras se basan en una teoría de agentes, no han alcanzado un nivel de madurez elevado y no están probadas lo suficiente, ya que proponen métodos innovadores para el desarrollo de software, lo cual representa una mayor dificultad para los analistas. Entre estas se encuentran Vowel Engineering, GAIA y Tropos. Las segundas como PASSI, Mase, y MESSAGE/UML intentan aplicar métodos existentes al enfoque de agentes, lo que tiene como ventaja la reutilización de modelos ya conocidos, sin embargo estas propuestas reportan limitaciones, pues los agentes son mucho más que especificaciones de objetos.

Las investigaciones relacionadas con el proceso de desarrollo de los SMA han ido en ascenso y a gran velocidad. Esto se ve reflejado en el volumen apreciable de metodologías que se han definido, ejemplo de algunas de ellas son: PASII, MESSAGE, GAIA, MAS-CommonKADS, Vowel Engineering, TROPOS, Ingenias, SODA, Prometheus, Zeus y AAI (PÉREZ, 2006). En esta última se profundizará a partir del siguiente epígrafe.

1.11 Metodología AAI

La metodología AAI ha sido desarrollada por el Instituto Australiano de Inteligencia Artificial (AAI), basada en la experiencia adquirida en el desarrollo de distintos proyectos de gran envergadura, como el sistema OASIS para el control del tráfico aéreo del aeropuerto de Sydney. Basada en las metodologías orientadas a objetos actuales, las enriquece añadiéndoles algunos de los conceptos de los sistemas basados en agentes. El objetivo de esta metodología es la construcción de un conjunto de modelos, que totalmente completados, definen la especificación del sistema de agentes.

1.11.1 Ciclo de vida de la metodología AAI

Antes de comenzar a introducir los detalles de la metodología AAI, es importante tener una visión general de todas sus etapas y modelos.

El proceso de AAI comienza con la etapa de Análisis, cuyo objetivo es sentar las bases para el diseño de la organización. Esta etapa incluye:

- *Dividir la organización:* Consiste en identificar cómo descomponer fructíferamente la organización en acopladas sub-organizaciones.
- *Modelar el entorno:* Se entiende como una abstracción, es la representación computacional del entorno en el cual el sistema multiagentes será ubicado.
- *Modelar los roles preliminares:* Se identifican las habilidades básicas requeridas por la organización. Este modelo preliminar contiene solo aquellos roles, posiblemente no definidos completamente, que pueden ser identificados independientemente de la imposición de una estructura organizacional.
- *Modelar los protocolos de interacción preliminares:* Se identifican las interacciones básicas requeridas para lograr la funcionalidad de los roles preliminares. Este modelo puede componerse fuera de una estructura organizacional fijada.
- *Definir la estructura organizacional del sistema:* Esta actividad saca provecho de patrones organizacionales que involucran la eficiencia de la organización y la imitación de una organización del mundo real.

Las salidas de la etapa de análisis son aprovechadas por el diseño, donde se definen dos niveles de abstracción, descomponiéndose el sistema desde un punto de vista externo y otro interno.

En el Diseño de la Vista Externa la metodología propone, basada en los roles más significativos de la aplicación, una descomposición del sistema en agentes modelados como objetos complejos, caracterizados por sus propósitos, responsabilidades, servicios que realizan, la información que requieren, y su interacción con el exterior. Los detalles de esta etapa se reflejan a través de las siguientes actividades:

- *Definir los roles y protocolos de interacción:* Esta actividad se basa en la estructura organizacional adoptada y separa, siempre que sea posible, los aspectos independientes de la organización y los dependientes de ella (derivados de la adopción de una estructura organizacional específica).
- *Modelar los agentes:* Esto equivale a identificar las clases de agentes que deben definirse para desempeñar roles específicos, y cuantas instancias de cada una se necesitan emplear en el desarrollo del sistema. Este modelo incluye dos componentes fundamentales:
 - *Modelar las Clases de Agentes:* Consiste en un diagrama de clases que representa gráficamente las relaciones entre los agentes que conforman el sistema, así como los diferentes niveles jerárquicos definidos entre ellos.
 - *Modelar las Instancias de Agentes:* Consiste en un diagrama que representa las instancias de los agentes definidos en el modelo de clases de agentes, los cuales se instancian en tiempo de ejecución.
- *Modelar las Interacciones:* Esta actividad describe las responsabilidades de cada clase de agente, los servicios que provee, las interacciones asociadas y las relaciones de control entre las clases de agentes.

En el Diseño de la Vista Interna la metodología se ajusta al paradigma BDI (Beliefs-Desires-Intentions). Este provee una visión fuerte de la noción de agentes, según la cual, los agentes poseen ciertas actitudes mentales: creencias, deseos e intenciones, que representan, respectivamente, los estados de información, motivación y deliberación del agente. Esto se captura para cada clase de agente en los siguientes modelos:

- *Modelar los objetivos:* Esta actividad describe el conjunto de objetivos que un agente puede adoptar, los eventos a los cuales puede responder y uno o más estados de objetivos utilizados para especificar el estado mental del agente.
- *Modelar las creencias:* Describe la información sobre el entorno y el estado interno de una clase de agente, así como las acciones que puede realizar. Se describen, además, las creencias de los agentes y sus propiedades, las cuales se engloban en conjuntos de creencias. Las instancias de estos conjuntos son denominadas estados de las creencias, que pueden ser utilizados para especificar el estado mental inicial del agente.
- *Modelar los planes:* Describe los planes que un agente podría seguir para lograr sus objetivos. Las propiedades y la estructura de control de planes individuales son agrupadas en conjuntos de planes.

1.11.2 Etapa de Análisis

Toda metodología para el desarrollo de software cuenta con una etapa de análisis donde, de manera general, se divide el problema a resolver en sus partes componentes, con vistas a poder lograr una mejor comprensión del mismo y básicamente obtener una modelación correcta que permita sentar las bases para su futuro diseño e implementación. Si un sistema no se analiza correctamente, su diseño no será el adecuado.

En la metodología AAIL la etapa de análisis consta de dos momentos fundamentales: comprensión del funcionamiento de la organización o entidad, y modelado de los elementos componentes del sistema a partir de la determinación de sus roles e interacciones esenciales.

1.11.3 División de la Organización

El primer paso en el análisis de AAIL es determinar las múltiples organizaciones que coexistirán en la aplicación y sus interacciones de forma autónoma dentro del sistema

multiagentes. Identificar dichas organizaciones es razonablemente fácil si las especificaciones del sistema ya están definidas, o si el sistema imita una estructura del mundo real e involucra varias interacciones entre ellas. De esta manera se pone en evidencia la modularidad del sistema, considerada en términos de sub-organizaciones para subdividir la complejidad global del mismo en un conjunto de pequeños y más manejables componentes. En general, las sub-organizaciones pueden ser establecidas cuando existen porciones en el sistema que:

- Exhiben una conducta especialmente orientada hacia el logro de un determinado objetivo.
- Interactúan holgadamente con otras porciones del sistema.
- Requieren habilidades que no son necesarias en otras partes del sistema.

1.11.4 Modelado del Entorno

Los entornos donde se desarrollan los sistemas multiagentes pueden ser muy diferentes y en ocasiones están relacionados a técnicas específicas. AAll sugiere tratar el entorno en términos de un “conjunto de recursos”, como variables o duplas, accesibles por el agente para pensar (leer sus valores), actuar (cambiar sus valores) o consumir (extraer del entorno). Con tal identificación, el modelo del entorno puede ser visto como una lista de recursos, cada uno asociado a un nombre simbólico, caracterizado por un tipo de acción que el agente puede realizarle, y posiblemente asociado a una descripción.

Existen además, otras especificaciones relacionadas con el modelado de los recursos del entorno que pueden enriquecer o completar el propósito de la notación básica, en particular:

- A menudo, debido a la naturaleza distribuida del entorno o a la lógica relación existente entre los recursos, un gráfico puede ser de gran ayuda para representar tales relaciones e identificar cómo y desde dónde un recurso puede ser accedido.
- El entorno en el cual un sistema multiagentes está inmerso, se caracteriza típicamente por su dinamismo, ya que el contexto de los recursos del sistema

puede cambiar de acuerdo con determinados patrones, y esto requerir de un modelado explícito. Estos casos, generalmente, son tratados mediante anotaciones en el modelo básico del entorno. Por otro lado, algunos de los recursos pueden estar disponibles intermitentemente, necesitando así un enriquecimiento del modelo básico de recursos y ser tomados en cuenta como incertidumbres.

Un importante tema que debe tenerse en cuenta a la hora de modelar el entorno, es que el ambiente operacional de un sistema multiagentes puede incluir componentes activos (servicios y sistemas de bases de datos, por ejemplo) con los que los agentes tienen que interactuar. Esto introduce la problemática de cómo interactuar con las entidades, es decir, modelándolas como recursos del entorno o en términos de agentes adicionales al sistema. Una guía al respecto puede ser:

- Cuando el rol de ese componente activo es simplemente suministrar información (base de datos, servidor web), es recomendable modelarlo como un recurso del entorno. Su fundamento viene dado porque su presencia solo es un mecanismo mediante el cual el agente obtiene los recursos por demandar un servicio, más que por ejecutar una operación.
- Si el entorno contiene componentes y servicios que presentan la capacidad de desempeñar complejas operaciones (bases de datos activas, control de sistemas automáticos), entonces sus efectos pueden identificarse como un simple repositorio de recursos con patrones identificables de cambios dinámicos que son censados por los agentes. En este caso, dichos componentes pueden no ser tratados como parte del entorno, sino que son insertados en el sistema multiagentes como nuevos agentes.

1.11.5 Modelo Preliminar de Roles

El rol de un agente define lo que se espera que realice en la organización, tanto respecto a ella, como en combinación con otros agentes. A menudo, el rol de un agente se define simplemente en términos de una tarea específica que debe cumplir, sin embargo, la

enunciación de un rol es mucho más precisa, es dotar al agente de una muy bien definida posición dentro de la organización, asociada a un conjunto de conductas.

La etapa de análisis de AAll no intenta diseñar por completo la organización de un sistema multiagentes, que es el propósito de la subsiguiente etapa de diseño, sin embargo, aún sin conocer cuál será la estructura de la organización, es posible, en la mayoría de los casos, identificar algunas características del sistema que son independientes de la estructura organizacional que se adopte. En particular, esto equivale a identificar las “habilidades básicas” que requiere la organización para lograr sus objetivos, así como las interacciones para el aprovechamiento de esas actividades. Dada esta identificación, la etapa de análisis puede promover una definición preliminar de los roles y protocolos de la organización, pero esta definición no está completa, pues de hecho, las habilidades básicas (o roles preliminares) solo pueden convertirse en verdaderos roles de la organización cuando se conozca cómo y con cuáles otros roles ellos pueden interactuar, lo que requiere a su vez de una definición general de la estructura de la organización. Análogamente, las necesidades básicas de interacción (o interacciones preliminares) solo pueden ser completamente definidas como protocolos de la organización cuando se adopte una clara estructura organizacional, con los roles involucrados en cada uno y por quién serán ejecutados. La etapa de diseño es la que introduce los roles y protocolos definitivos, que se derivan de la adopción de una estructura organizacional adecuada.

Para representar preliminarmente los roles, AAll adopta una descripción semiformal para expresar las habilidades y la conducta esperada. Esto es representado por dos clases fundamentales de atributos: permisos y responsabilidades.

1.11.6 Permisos

Estos atributos son principalmente encaminados a:

- Identificar los recursos que pueden legítimamente ser usados para llevar a cabo el rol (intuitivamente expresan que puede ser consumido mientras el rol se desempeña).

- Declarar los recursos límites dentro de los cuales el rol puede operar (intuitivamente expresan hasta dónde o qué no puede consumirse mientras el rol es ejecutado).

En general, los permisos relacionan roles de agentes al entorno en el cual están situados. Con el propósito de llevar a cabo un rol, un agente típicamente tiene acceso a los recursos del entorno, y puede cambiar o consumir los mismos; sin embargo, los permisos pueden ser usados para representar conocimientos que el agente puede tener o recibir mediante la comunicación con otros agentes.

1.11.7 Responsabilidades

Estos atributos determinan la conducta esperada de un rol, pudiéndose considerar como cruciales. Las responsabilidades son divididas en dos grupos: propiedades de existencia y propiedades de seguridad. Las propiedades de existencia indican el estado en que “algo bueno ocurre”, esto describe los estados de las cosas que un agente debe efectuar dada determinadas condiciones. Intuitivamente, las propiedades de seguridad indican el estado en que “nada malo ocurre”, lo que describe cuando un estado aceptable de las cosas es mantenido.

Las propiedades de existencia y de seguridad de AAll son especificadas mediante “expresiones de existencia y de seguridad” que definen el ciclo de vida de un rol.

Las expresiones de existencia definen las trayectorias potenciales de ejecución a través de varias actividades e interacciones asociadas con un rol, teniendo como forma general:

- $NOMBROROL = expresión$

Donde NOMBROROL es el nombre del rol de quien las propiedades de existencia son definidas, y *expresión* define las propiedades de existencia del NOMBROROL.

Los requerimientos de seguridad, son especificados por medio de una lista de predicados o expresiones de seguridad, que expresan típicamente la lista de variables de los permisos de un rol. Por convenio, AAll utiliza las expresiones de seguridad en una lista, en la que cada elemento expresa una responsabilidad de seguridad individual o esquema

del rol, dado que su aplicación es a todos los estados durante la ejecución del sistema, así, si el rol es de duración indefinida, las propiedades de seguridad permanecen invariables.

Una muestra de cómo resumir un esquema de rol se ilustra en la plantilla 1 del anexo A.2.

De manera general, un esquema de rol contiene el conjunto de atributos de cada rol, y un modelo de roles comprende el conjunto de todos los esquemas de roles del sistema, que en el caso del modelo preliminar de roles, tales esquemas presentan detalles no bien definidos aún, completándose durante la etapa de diseño.

1.11.8 Modelo preliminar de Protocolos de Interacción

Este modelo captura las dependencias y relaciones entre los roles que intervienen en la organización de un sistema multiagentes, en términos de definir un protocolo para cada tipo de interacción entre roles. Desde que el modelo de roles está en su estado preliminar, el modelo correspondiente de protocolos de interacción lo está también, necesariamente por la misma razón.

En AAll, un protocolo puede ser visto como un patrón de interacción que ha sido formalmente definido y resumido en una secuencia de ejecución de pasos, que enfocan su atención en la naturaleza y el propósito de las interacciones, más bien que en el preciso orden de intercambiar mensajes particulares.

La definición de un protocolo incluye determinados atributos que se representan en la plantilla 2 del anexo A.3.

1.11.9 Definición de la estructura organizacional

La elección de la estructura organizacional es una etapa muy crítica en el desarrollo del sistema multiagentes, pues afecta a las etapas subsiguientes. Desafortunadamente, no es posible identificar un método formal y preciso con el cual obtener el mejor diseño de la arquitectura, no obstante, existen líneas directivas para ayudar a elegir el diseño de la misma.

Es oportuno señalar que fuerzas adicionales pueden influenciar en la elección de la estructura organizacional, y más específicamente de su topología. En muchos casos, un sistema multiagentes tiene la intención de soportar una estructura organizacional del mundo real, por lo que esta última actúa como una fuerza de atracción en la definición de la estructura organizacional del sistema, que tiende a la semejanza con su contrapartida en la realidad.

Notables ejemplos de sistemas basados en agentes son para trabajos cooperativos computacionales, en los cuales los agentes se asocian con todos los miembros del equipo, y en el que cada cual usualmente tiene que cooperar de acuerdo con los patrones de interacción de la organización en el mundo real. De esta manera, los patrones de interacción entre los agentes y sus roles organizacionales son equivalentes a los procedimientos encontrados en las relaciones humanas.

Reflejar la estructura de un sistema como una estructura de una organización en el mundo real, simplificaría la conceptualidad del mismo, aunque en varios casos no siempre es posible que esta fuerza de atracción sea convenientemente acomodada, pues la construcción de una aplicación multiagentes necesariamente implica un análisis en profundidad de la correspondencia con una organización en el mundo real, y a menudo ocurre que estos procesos revelan ineficiencias en dicha organización, lo cual debe ser tomado en cuenta a la hora del diseño de los sistemas, forzando a un reajuste en la selección de la estructura organizacional.

Una vez que la identificación de una adecuada estructura organizacional para el sistema multiagentes está completa, se debe determinar cómo representarla eficientemente. Un control de relaciones, por ejemplo, puede identificar autoridad en la misma de un rol sobre otro: una relación de *paridad* puede expresar el hecho de que dos roles tengan igual estatus, mientras que una de *dependencia*, puede expresar el hecho de que un rol se apoye en algunos recursos o conocimientos de otros roles para poder alcanzar sus objetivos. En cualquier caso, estos prototipos de relación no son mutuamente excluyentes, aunque es bueno señalar que ellos no constituyen ni definen un conjunto completo, pues pueden ser identificados otros tipos de relaciones.

Se debe hacer énfasis en que AAll no propone la adopción de una notación específica. En consideración a la estructura organizacional, además de los métodos simples de

representación descriptos, otras notaciones y representaciones gráficas pueden ser adoptadas para describir y representar los roles y sus interacciones (AUML, por ejemplo).(JORGE GÓMEZ SANZ, 2004)

1.11.10 Etapa de Diseño de la Vista Externa

Las salidas de la etapa de análisis de AAIL documentan todas las características funcionales (y en cierta medida no funcionales) que un sistema multiagentes posee, conjuntamente a las particularidades del entorno en el cual va a ser situado. Estas especificaciones deben ser usadas en el diseño de la vista externa para identificar una eficiente manera de armar la organización y de completar consecuentemente los modelos preliminares de roles y protocolos de interacción.

Es importante señalar que, mientras la etapa de análisis apunta principalmente a entender lo que el sistema multiagentes tendrá que ser, es en la etapa de diseño donde se toman las decisiones sobre las características finales de dicho sistema.

1.11.11 Definición de los Modelos de Roles y Protocolos de Interacción

Una vez definida la estructura organizacional, los modelos preliminares de roles y protocolos de interacción (identificados en la etapa de análisis) pueden ser transformados en completos modelos de roles y protocolos de interacción (describiendo en detalles las características de todos los roles y estados involucrados en el sistema multiagentes que se está desarrollando). De hecho, con la estructura organizacional identificada, se conocen ya cuáles roles tendrán que interactuar con otros y cuáles protocolos tendrán que ser ejecutados.

En consideración a los ya identificados roles y protocolos, la diferencia es que la completa representación en esta etapa incluye todas las características de los mismos (para los roles, la completa identificación de las actividades y servicios, y para los protocolos, la completa identificación de los roles involucrados). Por otro lado, aquellos roles y protocolos cuyas definiciones derivan directamente de la adopción de una estructura

específica, sus representaciones tienen que ser definidas de la explotación de las plantillas de catálogos que constituyen los patrones organizacionales.

1.11.12 Modelo de Agentes

La definición del modelo de agentes equivale a identificar cuáles clases de agentes tienen que ser definidas para desempeñar roles específicos, y cuántas instancias de cada una se necesitan utilizar en el desarrollo del sistema multiagentes. Típicamente, puede existir una correspondencia de uno a uno entre los roles y las clases de agentes: un rol dado puede ser desempeñado por una instancia de un tipo de agente; sin embargo, si la eficiencia de la organización no es afectada y si no emergen problemas de racionalidad, se puede escoger un conjunto de roles fuertemente interrelacionados para ser desempeñados por una misma clase de agente. Existe una relación entre la coherencia de una clase de agente (como su funcionalidad puede ser fácilmente comprendida), las consideraciones de eficiencia que juegan cuando se diseñan clases de agentes, y la necesidad de minimizar incompatibilidades con la organización del mundo real que se soporta.

Este modelo puede ser definido usando un simple diagrama (o tabla), para cada clase de agente, cuyos roles serán delineados en él, así como las instancias de las clases que aparecerán en el sistema multiagentes.

Como parte importante del modelo de Agentes se definen también otros dos modelos:

- El modelo de Clases de Agentes, que consiste en un diagrama de clases que representa gráficamente las relaciones entre los agentes que conforman el sistema, así como los diferentes niveles jerárquicos definidos entre ellos.
- El modelo de Instancias de Agentes, que consiste en un diagrama que representa las instancias de los agentes definidos en el modelo de clases de agentes, los cuales se instancian en tiempo de ejecución.

1.11.13 Modelo de Interacción

Este modelo también forma parte de la etapa de Diseño de la Vista Externa y describe fundamentalmente las responsabilidades de cada clase de agente y los servicios que provee. Permite comprender las funcionalidades de cada agente, así como las relaciones que se establecen entre ellos.

1.11.14 Etapa de Diseño de la Vista Interna

El diseño de la Vista Interna propuesto por la metodología AAll se basa en la arquitectura BDI, que provee una visión fuerte de la noción de agentes, según la cual los agentes poseen ciertas actitudes mentales: creencias, deseos e intenciones, que representan, respectivamente, los estados de información, motivación y deliberación del agente. Esto se define para cada clase agente a través de los siguientes modelos:

- El modelo de objetivos: Describe los objetivos que un agente puede adoptar, y los eventos a los cuales puede responder, así como uno o más estados de objetivos utilizados para especificar el estado mental inicial del agente.
- El modelo de creencias: Describe la información acerca del ambiente y el estado interno que un agente de esa clase puede poseer, así como las acciones que puede ejecutar. Este modelo describe, como su nombre lo indica, las creencias de los agentes y sus propiedades. Todo esto se especifica en un conjunto de diagramas de clases, que denotan el dominio de las creencias de cada clase de agente.
- El modelo de planes: Consiste en un diagrama de transición de estados extendido con la noción de eventos y acciones, para cada clase de agente. Este modelo describe en particular, los planes que un agente puede emplear para lograr sus objetivos, así como las propiedades de cada plan individualmente.(LUCA CERNUZZI, 2003)

1.12 Conclusiones

La IA ha dado un paso más hacia delante con el surgimiento de los sistemas multiagentes. Estos están siendo implementados dentro del ámbito académico, industrial y comercial. Cabe indicar que, como siempre ocurre cuando aparece una nueva tecnología, no se debe creer que cualquier problema es tratable con un sistema basado en agentes, es posible que en ocasiones el empleo de técnicas clásicas resulte más adecuado en función de la naturaleza del problema a resolver.

Para desarrollar cualquier proyecto informático es imprescindible emplear alguna metodología de análisis y diseño que satisfaga las necesidades del modelo de desarrollo que se haya seleccionado. Existen múltiples metodologías de este tipo orientadas a agentes. Cada una de ellas presenta ventajas y limitaciones, por lo que se hace necesario contar con un estudio evaluativo, que permita seleccionar en cada caso cuál es la más adecuada para desarrollar un sistema multiagentes eficiente.

Capítulo II La metodología AAll: desarrollo de un caso de estudio.

2.1 Introducción

En esta sección se desarrollará un caso de estudio utilizando la metodología de análisis y diseño AAll, lo que contribuirá a comprenderla mejor, demostrar su utilidad para la construcción del sistema y evaluar hasta qué punto satisface los requerimientos necesarios para desarrollar sistemas multiagentes.

En el cuerpo de este capítulo sólo se mostrarán los esquemas y diagramas que se consideran más importantes para la comprensión del problema. El resto de ellos que conforman la especificación completa, se encuentran en los anexos de este trabajo de diploma.

2.2 Descripción del proyecto @LIS Technology Net

El proyecto @LIS Technology Net está diseñado para crear un entorno innovador de experimentación y enseñanza en Europa y América Latina. Las funcionalidades de este entorno permitirán la interconexión de todos los socios del consorcio en una red poblada de componentes de software autónomos, capaces de interactuar de forma dinámica entre ellos para proveer servicios a los usuarios.

Este proyecto se está desarrollando entre diferentes universidades: la Universidad Politécnica de Cataluña (España), la Universidad de Bath (Inglaterra), la Universidad de Parma (Italia), la Universidad de Costa Rica (Costa Rica), la Universidad Tecnológica Metropolitana (Chile), el Instituto Politécnico Nacional (México), el Instituto Tecnológico y de Estudios Superiores de Monterrey (México) y el Instituto Superior Politécnico José Antonio Echeverría (Cuba).

El sistema se construye usando como base la plataforma experimental Agentcities con el objetivo de utilizar las más modernas tecnologías de agentes autónomos, Web Internet. Entre los resultados más importantes se espera disponer de:

- Una infraestructura tecnológica de agentes autónomos, que conecte a España, Italia, Inglaterra, Chile, Costa Rica, México y Cuba, así como otras partes del mundo, permitiendo que cualquier tipo de software incluido por cualquier socio, sea capaz de comunicarse.
- Un entorno virtual de enseñanza, extendiendo la infraestructura que permita que estudiantes, profesionales de la educación e investigadores, ganen experiencia al trabajar con tecnologías de punta en las áreas de los agentes autónomos, para crear aplicaciones complejas, dinámicas y disponibles en línea.
- Un demostrador capaz de crear servicios personalizados de carácter cultural o turístico que sean accesibles en línea desde una PC y/o wireless (teléfonos móviles, PDAs con servicio de roaming u otros dispositivos) a través de la composición dinámica de servicios instalados por los distintos socios en los diferentes países del consorcio.

El sistema debe funcionar como un tour operador de viajes, que suministre una posible ruta de itinerario a todo usuario que, conectado desde cualquier dispositivo a una página web, decida que quiere viajar, señalando los destinos (países) a los que desea ir (Chile, Costa Rica, México y/o Cuba, por el momento), el presupuesto con que cuenta para ello y sus preferencias de actividades, alojamiento y transporte. El sistema debe permitir que el usuario modifique el plan de itinerario de viajes propuesto, y una vez aceptado, debe realizar las reservaciones pertinentes y almacenar las preferencias del usuario.

2.3 Funcionamiento de un tour operador real

Un tour operador de viajes real desarrolla un conjunto de procesos que involucra a varios individuos. Como parte de la etapa de “gestión personal”, un individuo (usuario) se comunica con el tour operador y le solicita la confección de un itinerario de viajes, señalándole sus destinos, presupuesto y tiempo del que dispone, así como sus preferencias. Si el tour operador conoce de antemano a su cliente por contactos anteriores, puede retomar algunas de sus preferencias y proponérselas.

Luego en una etapa de “gestión global”, el tour operador se encarga de distribuir adecuadamente entre los destinos seleccionados, el tiempo y presupuesto del usuario, respetando siempre sus preferencias. Posteriormente contacta con los tour operadores propios de cada país, con el objetivo de gestionar el itinerario en cada uno de ellos; negocia con las agencias de vuelos el trayecto entre destinos; y una vez obtenida toda la información (itinerario por cada país y vuelos entre ellos), confecciona un itinerario que satisfaga las necesidades del cliente.

Finalmente durante la etapa de “gestión local”, el tour operador de cada país seleccionado para viajar, tramita las preferencias solicitadas con los respectivos tour operadores de alojamiento y transporte, y una vez coordinado todo, elabora un plan de actividades para satisfacer la estancia del usuario en el destino seleccionado.

El sistema multiagentes “tour operador de viajes” puede ser diseñado sobre la base de una única organización formada por un conjunto de diferentes suborganizaciones, con el objetivo de dividir la complejidad global del mismo en un número de pequeños y manejables componentes. En cada sub-organización los agentes implicados desempeñan los roles de las personas involucradas en el proceso real, lo que puede requerir que estos tengan que interactuar entre sí directa e indirectamente, y con el entorno en el cual están enmarcados.(SARASA, 2006)

2.4 Análisis

A continuación se detallan los modelos identificados para el sistema multiagentes “tour operador de viajes” durante la etapa de Análisis de AAIL.

2.4.1 División de la Organización

Atendiendo a la descripción de los procesos que lleva a cabo un tour operador real, en la organización del sistema “tour operador de viajes” se pueden identificar claramente tres sub-organizaciones, una para cada etapa del proceso.

Primeramente se tiene la sub-organización responsable de la “gestión personal” de los usuarios, la cual se encarga del proceso de identificación, personalización y almacenamiento de las preferencias. Otra sub-organización es la encargada de la “gestión global” de los usuarios, que tiene como función distribuir el tiempo y el presupuesto entre los destinos de viajes, planificar los posibles itinerarios en cada destino, coordinar con las agencias de vuelos el transporte entre destinos, y confeccionar finalmente el itinerario de viaje más adecuado a las preferencias del usuario. Por último se encuentra la sub-organización dedicada a la “gestión local” de los usuarios, responsable de tramitar en cada destino seleccionado, las preferencias de alojamiento, actividades y transporte, además de elaborar un itinerario que satisfaga la estancia del cliente.

La identificación de dichas sub-organizaciones evidencia la modularidad del sistema, ya que cada una se enfoca especialmente hacia el logro de un determinado objetivo e interactúa holgadamente con las otras porciones del sistema.

2.4.2 Modelado del entorno

En el sistema multiagentes “tour operador de viajes”, el conjunto de recursos manejados por los agentes involucrados está formado por: las preferencias de los usuarios, los vuelos entre destinos y los servicios que se brindan en cada país visitado, ya sean actividades, alojamiento o transporte. Se debe señalar que las preferencias de los usuarios, aunque son un recurso que el sistema maneja, no forman parte del entorno, pues constituyen datos propios de la aplicación, por lo que su manipulación se realizará con una base de datos interna de la aplicación.

En la siguiente tabla se relacionan los recursos del entorno del sistema “tour operador de viajes” con un nombre simbólico, la acción que el agente puede efectuar sobre él y una descripción que ilustra lo que representa.

Recurso	Acción	Descripción
Preferencias	Leer Insertar Eliminar Cambiar	El recurso puede estar representado por una base de datos, que incluya información sobre los datos personales y las preferencias del cliente: usuario, contraseña, nombre y apellidos, dirección particular, número de identificación y/o pasaporte, países que ha visitado, actividades que ha realizado, calidad del transporte y el alojamiento que ha recibido durante sus viajes anteriores, rutas de vuelos que ha empleado, entre otros.
Vuelos	Leer Consumir	El recurso puede representarse a través de una base de datos, que incluya información sobre la reservación de vuelos internacionales: numeración del avión, nombre de los países y los aeropuertos de salida y de llegada, clase del vuelo, costo, número de identificación y/o pasaporte en caso de reservación, por ejemplo.
Actividades	Leer Consumir	Este recurso se puede representar por una base de datos, que incluya información sobre las actividades programadas para realizar en cada país: tipo (playa, museo, bosque, etc.), duración, costo, número de identificación y/o pasaporte en caso de reservación, entre otras.
Alojamiento	Leer Consumir	El recurso puede estar representado por una base de datos, que incluya información sobre la calidad del alojamiento en cada país: nombre del hotel, número de la habitación, duración, costo, número de identificación y/o pasaporte en caso de reservación, por ejemplo.
Trasporte	Leer Consumir	El recurso se puede representar por una base de datos, que incluya información sobre la calidad del transporte en cada país: tipo (aéreo, marítimo, terrestre), clase del transporte, costo, número de identificación y/o pasaporte en caso de reservación, duración, entre otros.

Tabla 1: Recursos del entorno.

2.4.3 Modelo Preliminar de Roles

En general, para el sistema “tour operador de viajes” se han identificado un total de 11 roles preliminares, los que se detallarán a continuación. Como parte de la sub-organización encargada de la “gestión personal” de los usuarios, se pueden identificar cinco roles preliminares, independientes de la estructura organizacional que se adopte para representar el sistema “turoperador de viajes”. Estos roles son denominados: Interfaz, Registrar, Autenticar, Buscar_ Itinerario y Ratificar_ Itinerario.

A continuación se reflejan los esquemas de los roles Interfaz, Registrar y Ratificar_ Itinerario. Los restantes esquemas de roles preliminares identificados para esta sub-organización se expresan en el anexo A.4.1.

Nombre del Rol
Interfaz
Descripción
Este rol verifica la información entrada por el usuario. También solicita el registro de un nuevo usuario, la autenticación del mismo si es habitual, un itinerario de viajes según las preferencias del usuario y la confirmación de un itinerario aceptado por el cliente.
Protocolos y Actividades
Registrar_ Usuario Autenticar_ Usuario Solicitar_ Itinerario Confirmar_ Itinerario Chequear_ Información
Permisos
Responsabilidades: Propiedades de existencia
1. Interfaz =Chequear_ Informacion.Registrar_ Usuario 2. Interfaz =Chequear_ Informacion.Autenticar_ Usuario 3. Interfaz =Chequear_ Informacion.Solicitar_ Itinerario 4. Interfaz =Confirmar_ Itinerario
Responsabilidades: Propiedades de seguridad
1. Autenticar_ Usuario→Solicitar_ Itinerario→Confirmar_ Itinerario

Tabla 2 : Rol Interfaz

Nombre del Rol
Registrar
Descripción
Este rol registra a un nuevo usuario en la base de datos del sistema.
Protocolos y Actividades
Registrar_Usuario
Permisos
Insertar preferencias
Responsabilidades: Propiedades de existencia
1.Registrar = Registrar_Usuario
Responsabilidades: Propiedades de seguridad

Tabla 3 : Rol Registrar

Nombre del Rol
Ratificar_Itinerario
Descripción
Este rol solicita la confirmación de un itinerario de viajes aceptado por el usuario, y una vez recibida la misma, almacena las preferencias del usuario en la base de datos del sistema.
Protocolos y Actividades
Confirmar_Itinerario Almacenar_Preferencias
Permisos
Guardar preferencias
Responsabilidades: Propiedades de existencia
1.Ratificar_Itinerario = Confirmar Itinerario.Almacenar Preferencias
Responsabilidades: Propiedades de seguridad
1. Confirmar_Itinerario→Almacenar_Preferencias

Tabla 4 : Rol Ratificar_Itinerario

Dentro de la sub-organización responsable de la “gestión global” de los usuarios, se identifican otros dos roles preliminares, independientes de la estructura organizacional que se adopte para representar el sistema “turoperador de viajes”. Estos se denominan:

Itinerario_Global y Vuelos. A continuación se presentan los esquemas de ambos roles preliminares, respectivamente.

Nombre del Rol
Itinerario_Global
Descripción
Este rol distribuye las preferencias del usuario. Además solicita posibles itinerarios de viajes en cada uno de los destinos, posibles vuelos entre destinos establecidos, conforma también los posibles itinerarios de viajes con toda la información recolectada, y una vez aceptado por el usuario uno de estos, solicita su confirmación a cada destino, así como los vuelos entre destinos.
Protocolos y Actividades
Solicitar_Itinerarios_Destino Solicitar_Vuelos Confirmar_Itinerario_Destino Confirmar_Vuelos Distribuir_Preferencias_Globales Conformar_Itinerarios_Globales
Permisos
Responsabilidades: Propiedades de existencia
1. Itinerario_Global = Distribuir_Preferencias_Globales. Solicitar_Itinerarios_Destino. Solicitar_Vuelos. Conformar_Itinerarios_Globales 2. Itinerario_Global = Confirmar_Itinerario_Destino. Confirmar_Vuelos
Responsabilidades: Propiedades de seguridad
1. Solicitar_Itinerarios_Destino → Solicitar_Vuelos 2. Confirmar_Itinerario_Destino → Confirmar_Vuelos

Tabla 5 : Rol Itinerario_Global

Nombre del Rol
Vuelos
Descripción
Este rol solicita los posibles vuelos entre destinos, y una vez aceptado por el usuario un itinerario propuesto, solicita confirmación de los vuelos.
Protocolos y Actividades
Solicitar_Vuelos Confirmar_Vuelos

Permisos
Leer vuelos Consumir vuelos
Responsabilidades: Propiedades de existencia
1. Vuelos =Solicitar_Vuelos 2. Vuelos =Confirmar_Vuelos
Responsabilidades: Propiedades de seguridad
1. Confirmar_ Itinerario →Almacenar_Preferencias

Tabla 6 : Rol Vuelos

Por último se encuentra la sub-organización que se encarga de la “gestión local” de los usuarios. Aquí se observan cuatro roles preliminares independientes de la estructura organizacional que se adopte para representar el sistema “turoperador de viajes”. Estos roles se identifican como: Itinerario_Local, Actividades, Alojamiento y Transporte.

A continuación se muestran los esquemas correspondientes a los roles Itinerario_Local y Actividades. Los restantes esquemas de roles preliminares identificados para esta sub-organización se expresan en el anexo A.4.2.

Nombre del Rol
Itinerario_Local
Descripción
Este rol distribuye las preferencias del usuario para el destino, solicita además posibles actividades, alojamiento, transporte e itinerarios de viajes en el destino, así como la confirmación de todo esto, una vez que el usuario acepte un itinerario propuesto.
Protocolos y Actividades
Solicitar_Actividades Solicitar_Alojamiento Solicitar_Trasporte Confirmar_Actividades Confirmar_Alojamiento Confirmar_Trasporte Distribuir_Preferencias_Locales Conformar_Itinerarios_Locales
Permisos

Responsabilidades: Propiedades de existencia
1. Itinerario_Local = Distribuir_Preferencias_Locales. Solicitar_Actividades. Solicitar_Alojamiento. Solicitar_Trasporte. Conformar_Itinerarios_Locales
2. Itinerario_Local = Confirmar_Actividades. Confirmar_Alojamiento. Confirmar_Trasporte
Responsabilidades: Propiedades de seguridad
1. Solicitar_Actividades → Solicitar_Alojamiento → Solicitar_Trasporte
2. Confirmar_Actividades → Confirmar_Alojamiento → Confirmar_Trasporte

Tabla 7 : Itinerario_Local

Nombre del Rol
Actividades
Descripción
Este rol solicita posibles actividades en el destino, y una vez aceptado por el usuario un itinerario propuesto, solicita la confirmación de estas actividades.
Protocolos y Actividades
Solicitar_Actividades Confirmar_Actividades
Permisos
Leer actividades Consumir actividades
Responsabilidades: Propiedades de existencia
1. Actividades = Solicitar_Actividades 2. Actividades = Confirmar_Actividades
Responsabilidades: Propiedades de seguridad

Tabla 8 : Rol Actividades

2.4.4 Modelo preliminar de Protocolos de Interacción

Para el sistema “tour operador de viajes” se han identificado un total de 16 protocolos preliminares de interacción, los cuales se detallarán a continuación.

En la sub-organización responsable de la “gestión personal” de los usuarios, se han identificado seis protocolos de interacción preliminares, independientes de la estructura organizacional que se adopte para representar el sistema “turoperador de viajes”. Estos

protocolos son denominados: Registrar_ Usuario, Autenticar_ Usuario, Solicitar_ Itinerario, Confirmar_ Itinerario, Solicitar_ Itinerarios y Guardar_ Preferencias.

A continuación se presentan los esquemas de los protocolos de interacción Registrar_ Usuario, Autenticar_ Usuario y Solicitar_ Itinerario. Los restantes esquemas de protocolos de interacción preliminares identificados para esta suborganización se expresan en el anexo A.5.1.

Nombre del Protocolo
Registrar_ Usuario
Descripción
Este protocolo registra por primera vez las preferencias de un usuario en la base de datos del sistema.
Rol Iniciador
Interfaz Registrar
Rol Asociado
Registrar (base de datos del sistema)
Entradas
Datos personales del usuario
Salidas
Confirmación de que se insertó el usuario en la base de datos del sistema.

Tabla 9 : Protocolo de Interacción Registrar_ Usuario

Nombre del Protocolo
Autenticar_ Usuario
Descripción
Este protocolo busca un usuario habitual en la base de datos del sistema.
Rol Iniciador
Interfaz Autenticar
Rol Asociado
Autenticar (base de datos del sistema)
Entradas
Nombre de usuario y contraseña
Salidas
Datos personales del usuario y preferencias guardadas anteriormente.

Tabla 10 : Protocolo de interacción Autenticar_ Usuario

Nombre del Protocolo
Solicitar_ Itinerario
Descripción
Este protocolo solicita un itinerario de viajes señalando los destinos, el tiempo, el presupuesto y las preferencias del usuario.
Rol Iniciador
Interfaz Autenticar
Rol Asociado
Buscar_ Itinerario
Entradas
Destinos, tiempo, presupuesto y preferencias.
Salidas
Itinerario de viajes completo atendiendo a las características descritas.

Tabla 11 : Protocolo de interacción Solicitar_ Itinerario

Como parte de la sub-organización que se encarga de la “gestión global” de los usuarios, se localizan cuatro protocolos preliminares de interacción, independientes de la estructura organizacional que se adopte para representar el sistema “turoperador de viajes”. Estos protocolos son: Solicitar_ Itinerarios_ Destino, Solicitar_ Vuelos, Confirmar_ Itinerario_ Destino, Confirmar_ Vuelos.

A continuación aparecen los esquemas de los protocolos de interacción Solicitar_ Itinerarios_ Destino y Solicitar_ Vuelos. Los restantes esquemas de protocolos de interacción preliminares identificados para esta sub-organización se expresan en el anexo A.5.2.

Nombre del Protocolo
Solicitar_ Itinerarios_ Destino
Descripción
Este protocolo solicita un conjunto de posibles itinerarios en el destino, señalando el tiempo y el presupuesto asignados para el mismo y las preferencias establecidas por el usuario para su estancia.
Rol Iniciador
Itinerario_ Global
Rol Asociado
Itinerario_ Local
Entradas

Tiempo de estancia en el destino, presupuesto destinado para itinerario en el destino y preferencias en los servicios del destino.
Salidas
Conjunto de posibles itinerarios para el destino atendiendo a las características expresadas.

Tabla 12: Protocolo de interacción Solicitar_ Itinerarios_ Destino

Nombre del Protocolo
Solicitar_ Vuelos
Descripción
Este protocolo solicita un conjunto de posibles vuelos entre los destinos señalados, apuntando los días de vuelo entre destino, el presupuesto asignado para ello y las preferencias de vuelo establecidas por el usuario.
Rol Iniciador
Itinerario_Global Vuelos
Rol Asociado
Vuelos (base de datos del entorno)
Entradas
Días de vuelos entre destinos, presupuesto para vuelos y preferencias de vuelo.
Salidas
Conjunto de posibles vuelos entre destinos atendiendo a las características expresadas.

Tabla 13 : Protocolo de interacción Solicitar_ Vuelos

Finalmente en la sub-organización responsable de la “gestión local” de los usuarios, se identifican seis protocolos de interacción preliminares, independientes de la estructura organizacional que se adopte para representar el sistema “turoperador de viajes”. Estos protocolos son: Solicitar_ Actividades, Solicitar_ Alojamiento, Solicitar_ Transporte, Confirmar_ Actividades, Confirmar_ Alojamiento y Confirmar_ Transporte.

A continuación aparecen los esquemas de los protocolos de interacción Solicitar_ Actividades, Solicitar_ Alojamiento y Confirmar_ Transporte. Los restantes esquemas de protocolos de interacción preliminares identificados para esta suborganización se expresan en el anexo A.5.3.

Nombre del Protocolo
Solicitar_Actividades
Descripción
Este protocolo solicita un conjunto de posibles actividades en el destino señalado, teniendo en cuenta los días asignados para ello, el presupuesto y las preferencias establecidas por el usuario.
Rol Iniciador
Itinerario_Global Actividades
Rol Asociado
Actividades (base de datos del entorno)
Entradas
Días para las actividades, presupuesto y preferencias establecidas por el usuario.
Salidas
Conjunto de posibles actividades en el destino, según las características expresadas.

Tabla 14 : Protocolo de interacción Solicitar_Actividades

Nombre del Protocolo
Solicitar_Alojamiento
Descripción
Este protocolo solicita un conjunto de posibles alojamientos en el destino señalado, atendiendo a los días asignados para ello, el presupuesto y las preferencias establecidas por el usuario.
Rol Iniciador
Itinerario_Global Alojamiento
Rol Asociado
Alojamiento (base de datos del entorno)
Entradas
Días de alojamiento, presupuesto y preferencias establecidas por el usuario.
Salidas
Conjunto de posibles alojamientos en el destino, atendiendo a las características expresadas.

Tabla 15 : Protocolo de interacción Solicitar_Alojamiento

Nombre del Protocolo
Confirmar_Transporte
Descripción
Este protocolo solicita la confirmación de los transportes aceptados por el usuario para el destino.
Rol Iniciador
Itinerario_Local Transporte
Rol Asociado

Transporte (base de datos del entorno)
Entradas
Días para los transportes, presupuesto y preferencias establecidas por el usuario.
Salidas
Confirmación de que se ratificaron los transportes aceptados por el usuario.

Tabla 16 : Protocolo de interacción Confirmar_ Transporte

2.4.5 Definición de la estructura organizacional del sistema.

La elección de la estructura organizacional es un proceso fundamental en el desarrollo de cualquier sistema multi-agentes con la metodología AAll, pues influye en las etapas posteriores. Tomando como referencia los roles identificados hasta el momento en el sistema “tour operador de viajes”, es posible establecer ciertos niveles de relaciones jerárquicas entre ellos, ya sean de paridad (cuando los roles tengan igual estatus), o de dependencia (cuando cierto rol se apoya en otros para lograr sus objetivos).

El rol Interfaz representa la conexión entre el sistema y el usuario, y sus funciones son básicamente recibir y mostrar información. Los roles Registrar y Autenticar se encargan de la parte administrativa del sistema. Tienen como funciones insertar un nuevo usuario en la base de datos del sistema y cargar las preferencias de un usuario habitual perteneciente al mismo, respectivamente. Estos tres roles (Interfaz, Registrar y Autenticar) forman parte del sistema “tour operador de viajes” como complemento del proceso básico que realiza un tour operador, de esta manera, su nivel jerárquico está a la altura de los roles Buscar_ Itinerario y Ratificar_ Itinerario, todos en el nivel superior (1er nivel de la topología) en la estructura organizacional del sistema.

Por otra parte los roles Buscar_ Itinerario y Ratificar_ Itinerario le solicitan al rol Itinerario_ Global, un conjunto de posibles itinerarios de viajes y la confirmación de un itinerario de viajes aceptado por el usuario, respectivamente, estableciéndose entre ellos una relación de dependencia, ya que los dos primeros dependen de la respuesta del último para alcanzar el logro de sus objetivos; por tanto, el rol Itinerario_ Global está un nivel más abajo (2do nivel de la topología) en la estructura organizacional del sistema.

Seguidamente el rol Itinerario_ Global solicita al rol Itinerario_ Local un conjunto de posibles itinerarios de viajes en el destino, y posteriormente, con la información recibida, solicita al rol Vuelos, posibles vuelos entre los destinos seleccionados por el usuario; del mismo modo, una vez aceptado un itinerario de viajes por el usuario, solicita la confirmación del itinerario en el destino al rol Itinerario_ Local , y una vez obtenida la ratificación, solicita también la confirmación al rol Vuelos de los vuelos respectivos. Como se puede apreciar la relación que se establece entre estos tres últimos roles (Itinerario_ Global, Itinerario_ Local y Vuelos), también es de dependencia, pues el primero necesita para su correcto funcionamiento de las respuestas de los dos últimos. Por esto ambos roles Itinerario_ Local y Vuelos se encuentran en un mismo estatus (3er nivel de la topología) en la estructura organizacional del sistema. Es importante aclarar que aunque no existe una relación explícita entre los dos últimos roles (Itinerario_ Local y Vuelos), tiene lugar una cierta dependencia implícita entre ellos, pues para solicitar el rol Itinerario_ Global algún servicio al rol Vuelos, es necesario que ya conozca las posibles respuestas ofrecidas por el rol Itinerario_ Local.

El rol Itinerario_ Local solicita al rol Actividades un conjunto de posibles actividades en el destino, luego con la información recibida, solicita al rol Alojamiento un conjunto de posibles alojamientos, y finalmente, con estos datos, solicita al rol Transporte los posibles transportes en el destino. De este mismo modo, y siguiendo la misma secuencia (actividades, alojamiento y transporte, en ese orden), solicita la confirmación del itinerario, una vez que el usuario lo acepte. De esta manera, la relación existente entre el rol Itinerario_ Local y los roles Actividades, Alojamiento y Transporte es de dependencia, pues el primero necesita de los restantes para lograr sus propósitos, situándose los tres últimos en un mismo estatus (4to nivel de la topología) en la estructura organizacional del sistema. Igualmente se debe señalar que existe una relación implícita de dependencia entre los roles Actividades, Alojamiento y Transporte, pues este último depende de los posibles alojamientos que le suministre el rol Alojamiento al rol Itinerario_ Local, y este a su vez de las posibles actividades que le proporcione el rol Actividades al rol Itinerario_ Local.

Esta estructura organizacional sigue un orden descendente, donde los roles que ocupan los niveles superiores o primeros, dirigen y controlan a los restantes.

2.5 Diseño

Las salidas de la etapa de análisis son aprovechadas en la etapa de diseño del sistema, la cual se descompone en dos partes fundamentales, el diseño de la vista externa y el diseño de la vista interna, según lo propuesto por la metodología AAIL.

2.6 Diseño de la Vista Externa

En esta etapa la metodología propone una descomposición del sistema basada en los roles más significativos en la aplicación. La identificación de los roles y sus relaciones guían la especificación de la jerarquía de clases de los agentes. El análisis de las responsabilidades de cada clase de agente conduce a la identificación de los servicios que ofrecen, así como de aquellos que usan de otros agentes, lo que permite identificar las interacciones externas de los agentes.

Todos estos detalles se capturan en dos modelos principales: el Modelo de Agentes y el Modelo de Interacción. A su vez el Modelo de Agentes incluye dos componentes fundamentales para la descripción detallada del sistema multiagente “tour operador de viajes”, estos son el Modelo de Clases de Agentes y el Modelo de Instancias de Agentes.

2.6.1 Definición de los Modelos de Roles y Protocolos de Interacción

Después de definir la estructura organizacional del sistema multi-agentes “tour operador de viajes”, es posible establecer niveles de jerarquías para roles y protocolos de comunicación. Los modelos preliminares de roles y protocolos de interacción se completan convirtiéndose en los respectivos Modelo de Roles y Modelo de Protocolos de interacción, pues la definición de la estructura organizacional no identificó nuevos roles o protocolos de interacción derivados de la misma.

Una vista jerárquica de los roles y los protocolos de interacción que emplea el sistema “tour operador de viajes”, se muestra a continuación en la siguiente tabla:

Nivel	Roles	Protocolo
1er nivel	1. Interfaz 2. Registrar 3. Autenticar 4. Buscar_ Itinerario 5. Ratificar_ Itinerario	1. Registrar_ Usuario 2. Autenticar_ Usuario 3. Solicitar_ Itinerario 4. Confirmar_ Itinerario 5. Solicitar_ Itinerarios 6. Guardar_ Preferencias
2do nivel	6. Itinerario_ Global	7. Solicitar_ Itinerario_ Destino 8. Solicitar_ Vuelos 9. Confirmar_ Itinerario_ Destino 10. Confirmar_ Vuelos
3er nivel	7. Itinerario_ Local 8. Vuelos	11. Solicitar_ Actividades 12. Solicitar_ Alojamiento 13. Solicitar_ Transporte Solicitar_ Vuelos 14. Confirmar_ Actividades 15. Confirmar_ Alojamiento 16. Confirmar_ Transporte Confirmar_ Vuelos
4to nivel	9. Actividades 10. Alojamiento 11. Transporte	Solicitar_ Actividades Solicitar_ Alojamiento Solicitar_ Transporte Confirmar_ Actividades Confirmar_ Alojamiento Confirmar_ Transporte

Tabla 17 : Vista jerárquica de los roles y protocolos de interacción.

2.6.2 Modelo de Agentes

La definición del modelo de agentes equivale a identificar cuáles clases de agentes tienen que ser definidas para desempeñar roles específicos y cuántas instancias de cada una se necesitan emplear en el desarrollo del sistema multiagentes.

En el sistema “tour operador de viajes” es considerable tener un agente (agente interfaz) que desempeñe el rol Interfaz, así como resulta provechoso tener un mismo agente (agente personal) desempeñando los roles Registrar, Autenticar, Buscar_ Itinerario y

Ratificar_ Itinerario, pues con ello no se afecta la eficiencia de la organización del sistema, ya que dichos roles están estrechamente relacionados y son jugados en diferentes momentos. Es igualmente ventajoso tener un agente (agente global) que desempeñe el rol Itinerario_ Global, uno para el rol Itinerario_ Local (agente local) y otro para el rol Vuelos (agente vuelos). Para los roles Actividades, Alojamiento y Transporte, se designan tres agentes (Agente Actividades, Agente Alojamiento y Agente Transporte), respectivamente.

En la siguiente tabla se detallan las clases de agentes, los roles que desempeñan y las instancias de cada clase de agente que aparecerán en el sistema multi-agentes “tour operador de viajes”:

Clases de Agente	Instancias del Agente	Roles que Desempeña
Agente Interfaz	1	Interfaz
Agente Personal	0 o más	Registrar Autenticar Buscar_ Itinerario Ratificar_ Itinerario
Agente Global	1	Itinerario_ Global
Agente Local	1 o más	Itinerario_ Local
Agente Vuelos	1	Vuelos
Agente Actividades	1	Actividades
Agente Alojamiento	1	Alojamiento
Agente Transporte	1	Transporte

Tabla 18 : Relación entre Roles, Clases de Agentes e Instancias de Agentes

2.6.3 Modelo de Clases de Agentes

El modelo de Clases de Agentes es una representación gráfica de las relaciones entre los agentes que conforman el sistema, así como de los diferentes niveles jerárquicos con los que el mismo opera. En el caso de estudio abordado se pudieron identificar 4 niveles de jerarquía y 8 tipos de agentes involucrados, los cuales se presentan a continuación.

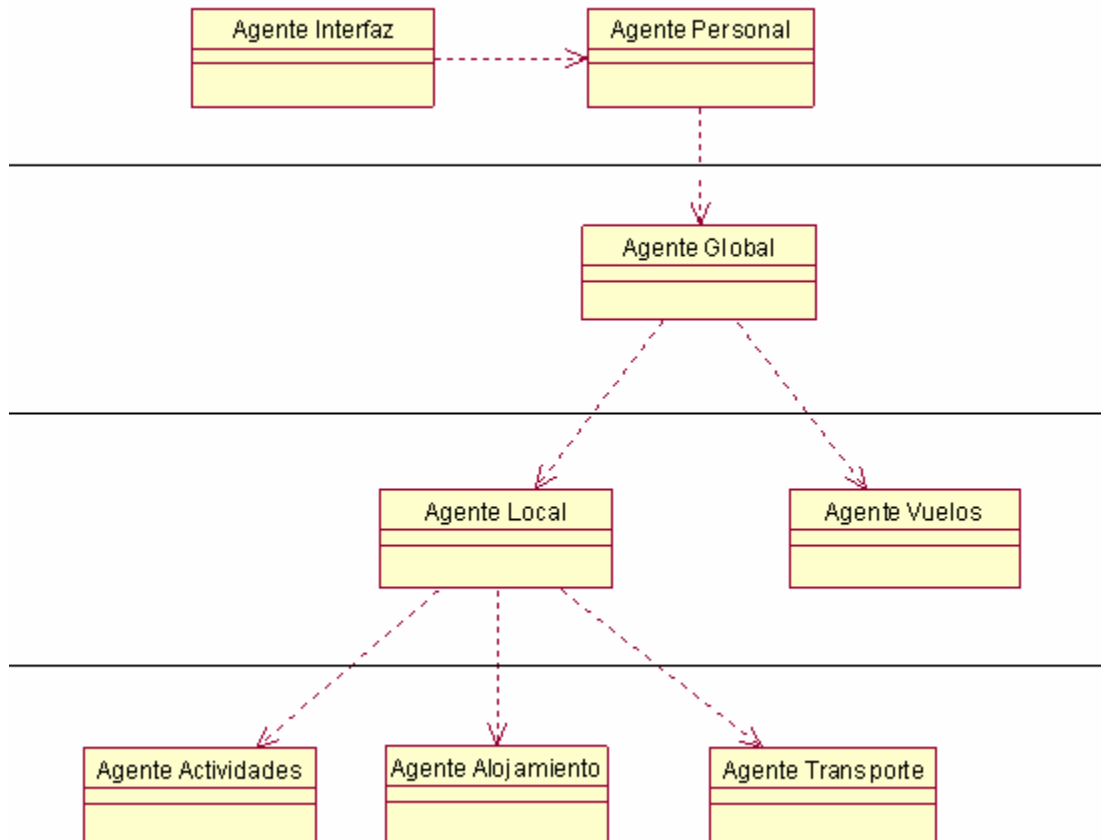
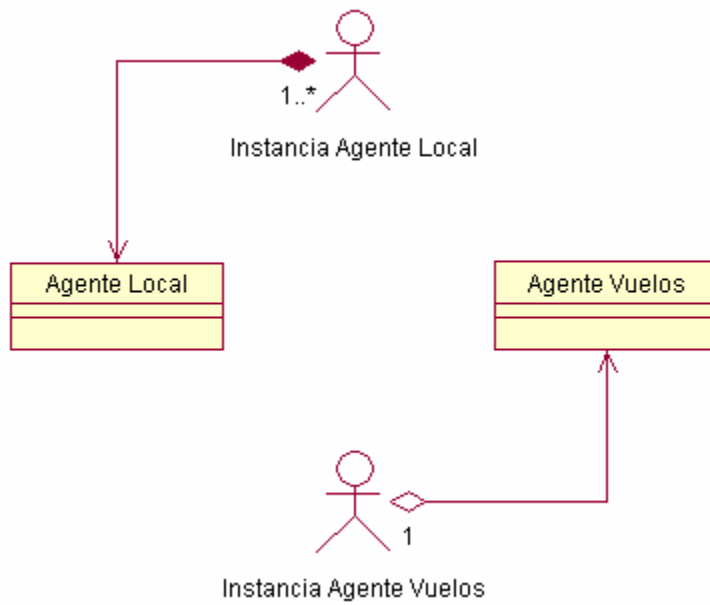
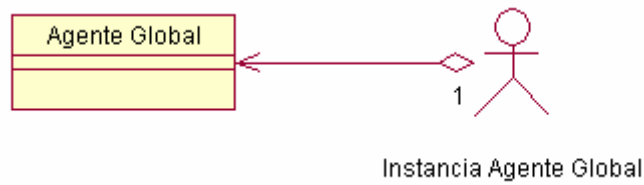
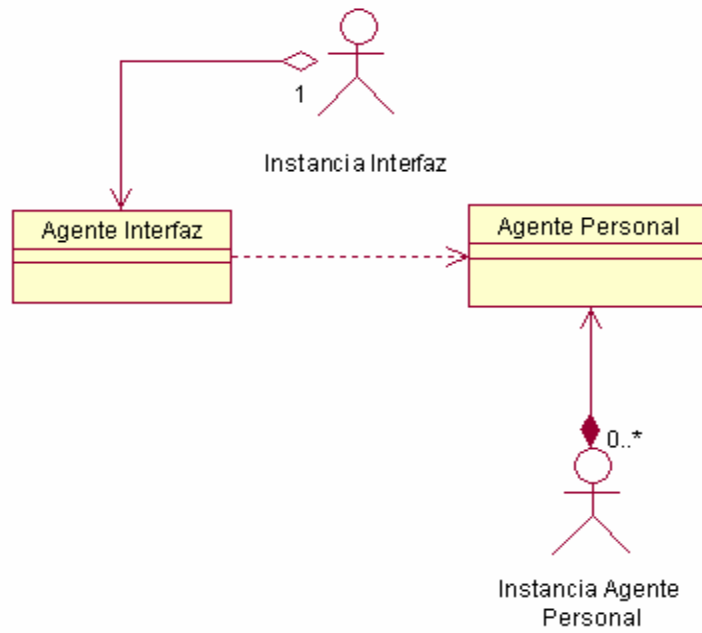


Figura 2. 1 : Modelo de Clases de Agentes

2.6.4 Modelo de Instancias de Agente

Este diagrama representa las Instancias de los agentes definidos en el modelo de Clases de Agentes, los cuales se instancian en tiempo de ejecución.



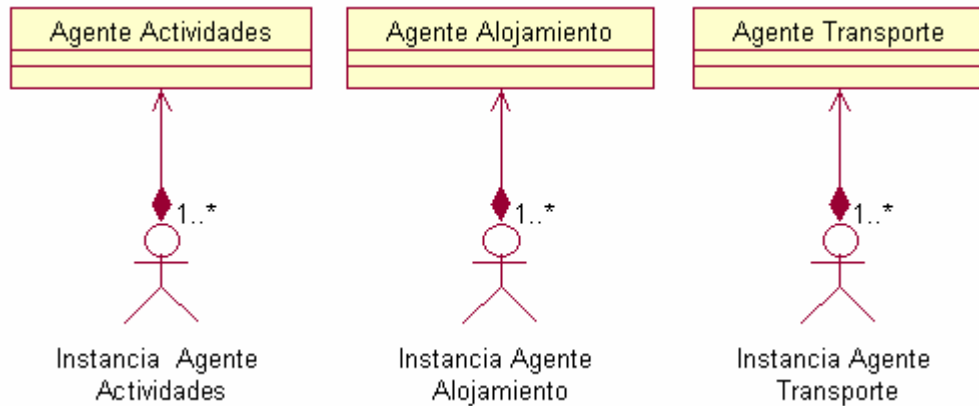


Figura 2. 2 : Modelo de Instancias de Agente

2.6.5 Modelo de Interacción

A través de este modelo se describen las responsabilidades de cada clase de agente del sistema “tour operador de viajes”, al igual que los servicios que provee.

Responsabilidad: Verificar la información entrada por el usuario, es decir, solicita el registro de un nuevo usuario o la autenticación del mismo si es habitual. También solicita un itinerario de viajes según las preferencias del usuario y la confirmación de uno ya aceptado por el cliente.

Servicios: Solicitar_Autenticar_Usuario

Solicitar_Registrar_Usuario

Solicitar_Itinerario_Viajes

Solicitar_Modificación_Itinerario_Viajes

Solicitar_Confirmación_Itinerario_Viajes

Modelo de Interacción del Agente Interfaz

Responsabilidad: Registrar a un nuevo usuario en la base de datos del sistema, y si ya es usuario habitual, lo busca y carga sus preferencias. Además se encarga de solicitar un conjunto de posibles itinerarios de viajes y escoger de ellos el más adecuado a las preferencias del usuario. Solicita también la confirmación de un itinerario aceptado por el usuario, y una vez recibida, almacena sus preferencias en la base de datos.

Servicios: Registrar_ Usuario

Autenticar_ Usuario

Buscar_ Itinerario_ Viajes

Modificar_ Itinerario_ Viajes

Confirmar Itinerario Viajes

Modelo de Interacción del Agente Personal

Responsabilidad: Distribuir las preferencias del usuario, a través de la solicitud de posibles itinerarios en cada uno de los destinos y posibles vuelos entre destinos. También es el responsable de conformar los itinerarios de viajes con toda la información recolectada, para cuando el usuario acepta uno de ellos, entonces solicitar la confirmación de los destinos y de vuelos entre destinos.

Servicios: Elaborar_ Itinerarios_ Globales

Confirmar_ Itinerario_ Global

Modelo de Interacción del Agente Global

Responsabilidad: Solicitar los posibles vuelos entre destinos, y una vez aceptado por el usuario uno de los itinerarios propuestos, solicita la confirmación de los vuelos.

Servicios: Buscar_ Vuelos
Reservar_ Vuelos

Modelo de Interacción del Agente Vuelos

Responsabilidad: Distribuir las preferencias del usuario para el destino, a través de la solicitud de posibles actividades, alojamiento, transporte e itinerarios de viajes en el destino, así como la confirmación de todo esto, una vez que el usuario acepte un itinerario propuesto.

Servicios: Elaborar_ Itinerarios _Locales
Confirmar Itinerario Local

Modelo de Interacción del Agente Local

Responsabilidad: Solicitar posibles actividades en el destino, y una vez que el usuario acepta uno de los itinerarios propuestos, solicita la confirmación de estas actividades.

Servicios: Buscar_ Actividades
Reservar_ Actividades

Modelo de Interacción del Agente Actividades

Responsabilidad: Solicitar posibles alojamientos en el destino, y una vez que el usuario acepta uno de los itinerarios propuestos, solicita la confirmación del alojamiento en el destino.

Servicios: Buscar_ Alojamiento
Reservar_ Alojamiento

Modelo de Interacción del Agente Alojamiento

Responsabilidad: Solicitar posibles transportes en el destino, y una vez que el usuario acepta uno de los itinerarios propuestos, solicita la confirmación del transporte en el destino.

Servicios: Buscar_ Transporte
Reservar_ Transporte

Modelo de Interacción del Agente Transporte

2.7 Diseño de la Vista Interna

El diseño de la Vista Interna que propone la metodología AAll, tiene como objetivo implementar interiormente los agentes que conforman el sistema. Esta se compone de tres modelos fundamentales, que especifican el estado mental inicial de cada agente.

- El Modelo de Objetivos especifica el dominio de eventos y objetivos para cada Clase de Agente.
- El Modelo de Creencias describe la información acerca del ambiente y el estado interno que un agente puede poseer, así como las acciones que puede realizar.
- El Modelo de Planes describe los planes que un agente puede emplear para lograr sus objetivos.

2.7.1 Modelos de Objetivos

Agente Interfaz

Autenticar Usuario (!)
Registrar Usuario (!)
Mostrar Itinerario de Viajes (!)
Solicitar Cambios en Itinerario () \$
Solicitar Confirmación de Itinerario () \$

Modelo de Objetivos del agente Interfaz

Agente Personal

Solicitar Preferencias del Usuario (!)
Enviar Preferencias del Usuario (!)
Enviar Itinerario de Viajes (!)
Buscar Itinerario de Viajes (!)
Modificar Itinerario de Viajes () \$
Confirmar Itinerario de Viajes () \$

Modelo de Objetivos del agente Personal

Agente Global

Distribuir preferencias del usuario (!)
Solicitar Itinerarios entre destinos (!)
Elaborar Itinerarios Globales (!)
Confirmar Itinerario Global (!)
Solicitar Vuelos (!)
Confirmar Vuelos (!)

Modelo de Objetivos del agente Global

Agente Vuelos

Buscar Vuelos ()!

Reservar Vuelos ()!

Confirmar Reservación ()!

Modelo de Objetivos del agente Vuelos

Agente Local

Recibir Solicitud Global ()!

Solicitar Actividades ()!

Solicitar Alojamiento ()!

Solicitar Transporte ()!

Solicitar Itinerarios en el Destino ()!

Confirmar Actividades ()!

Confirmar Alojamiento ()!

Confirmar Transporte ()!

Confirmar Itinerarios en el Destino ()!

Elaborar Itinerario Local ()!

Modelo de Objetivos del agente Local

Agente Actividades

Recibir Solicitud de Actividades ()!

Reservar Actividades ()!

Confirmar Actividades ()!

Modelo de Objetivos del agente Actividades

Agente Alojamiento

Recibir Solicitud de Alojamiento ()!

Reservar Alojamiento ()!

Confirmar Alojamiento ()!

Modelo de Objetivos del agente Alojamiento

Agente Transporte

Recibir Solicitud de Transporte ()!

Reservar Transporte ()!

Confirmar Transporte ()!

Modelo de Objetivos del agente Transporte

2.7.2 Modelos de Creencias

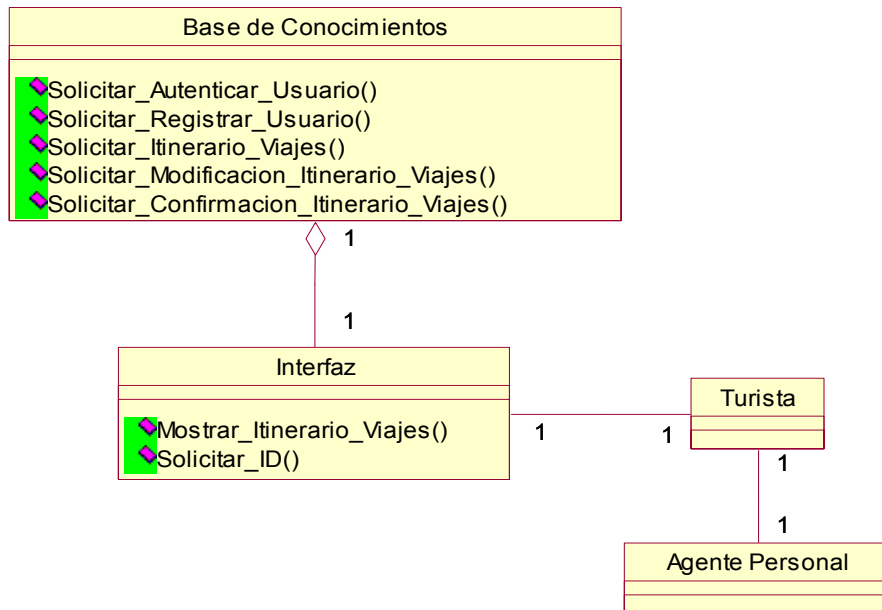


Figura 2. 3 : Modelo de creencias del agente Interfaz

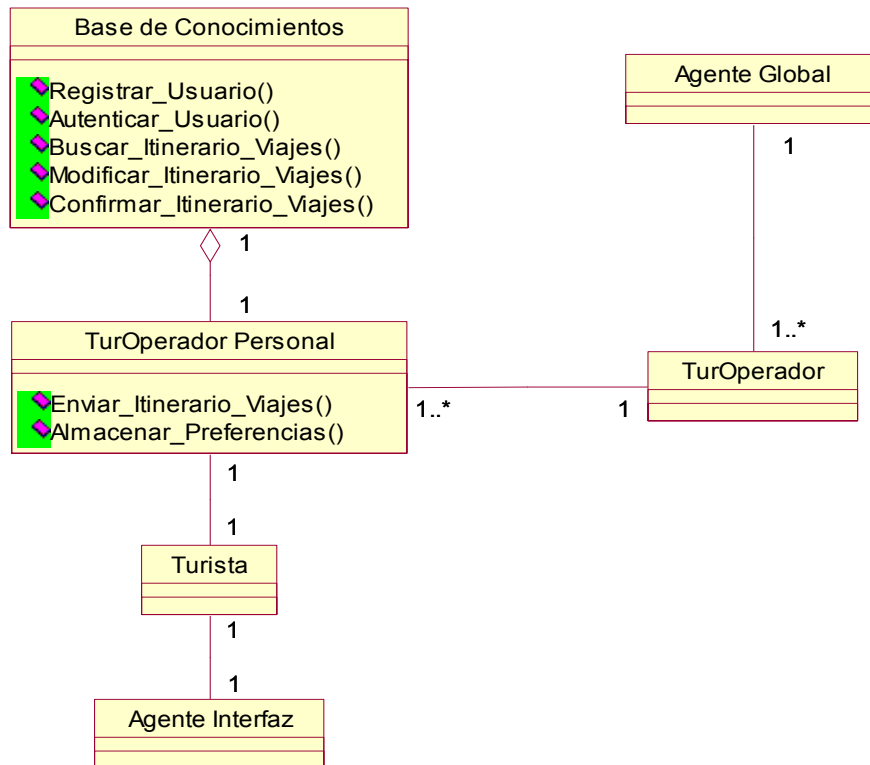


Figura 2. 4 : Modelo de creencias del agente Personal

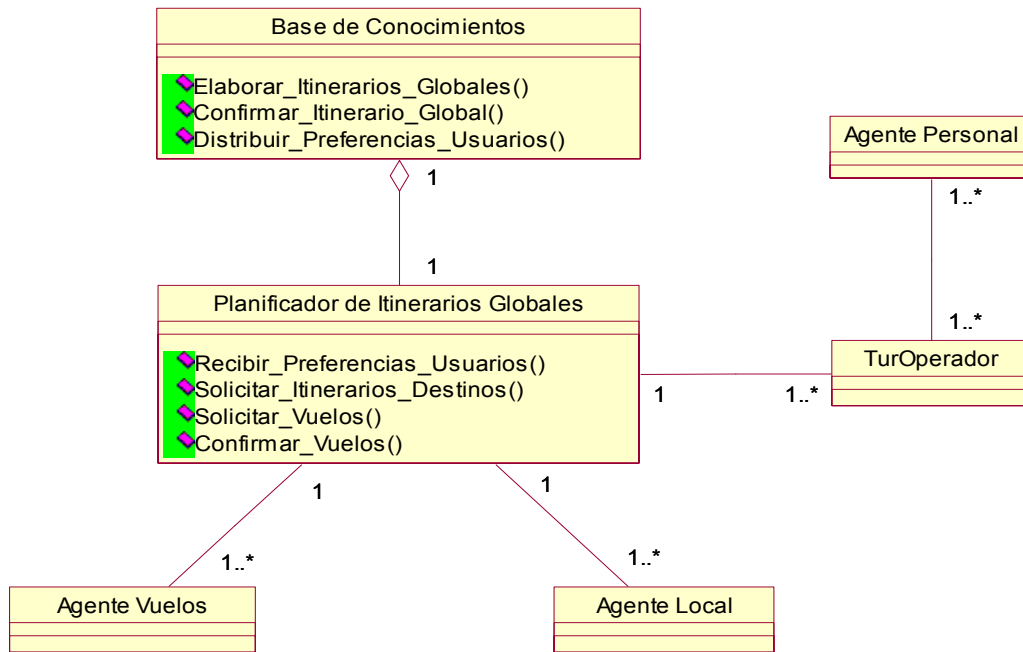


Figura 2. 5: Modelo de creencias del agente Global

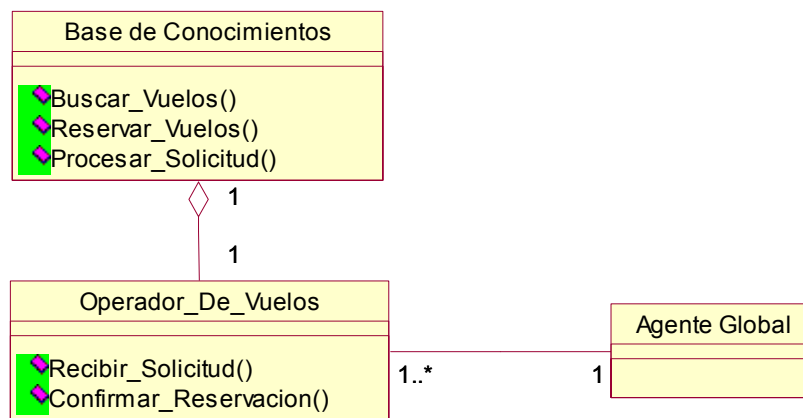


Figura 2. 6 : Modelo de creencias del agente Vuelos

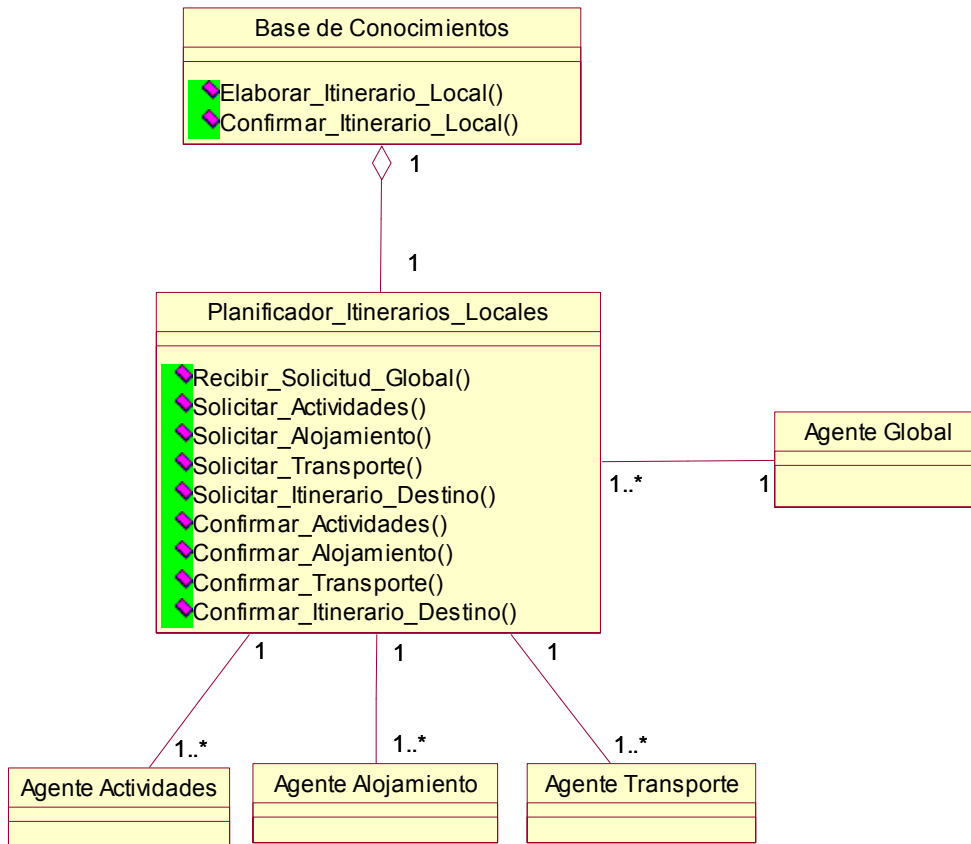


Figura 2. 7 : Modelo de creencias del agente Local

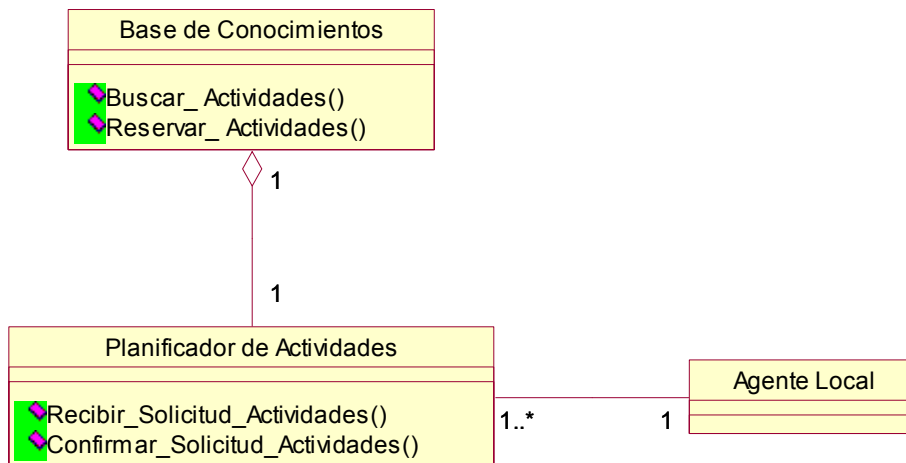


Figura 2. 8 : Modelo de creencias del agente Actividades

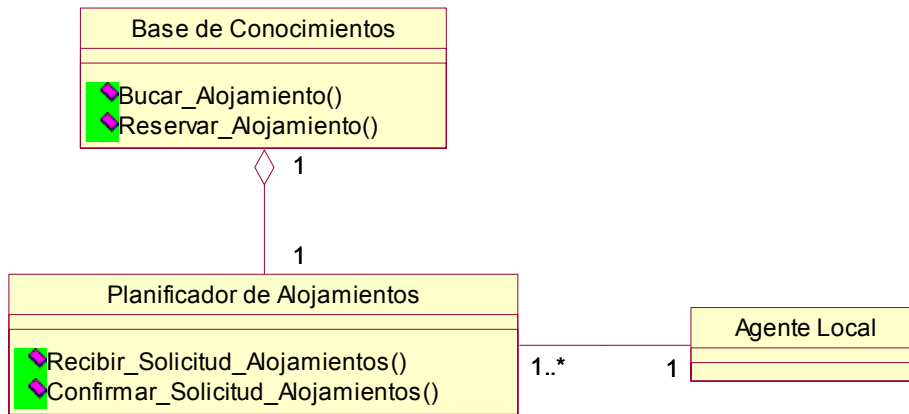


Figura 2. 9 : Modelo de creencias del agente Alojamiento

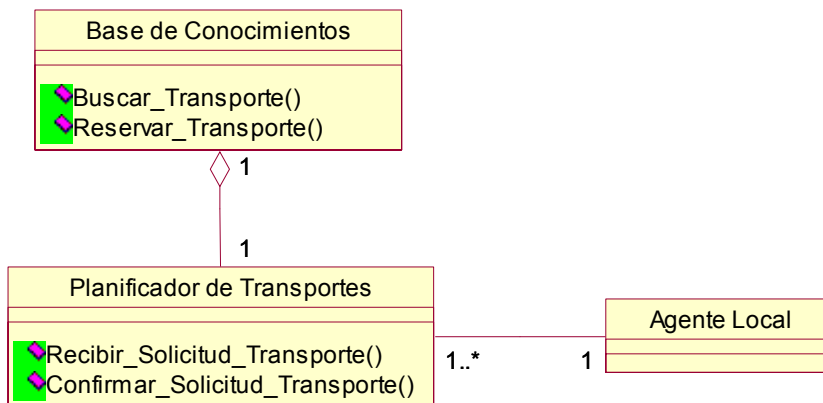


Figura 2. 10 : Modelo de creencias del agente Transporte

2.7.3 Modelos de Planes

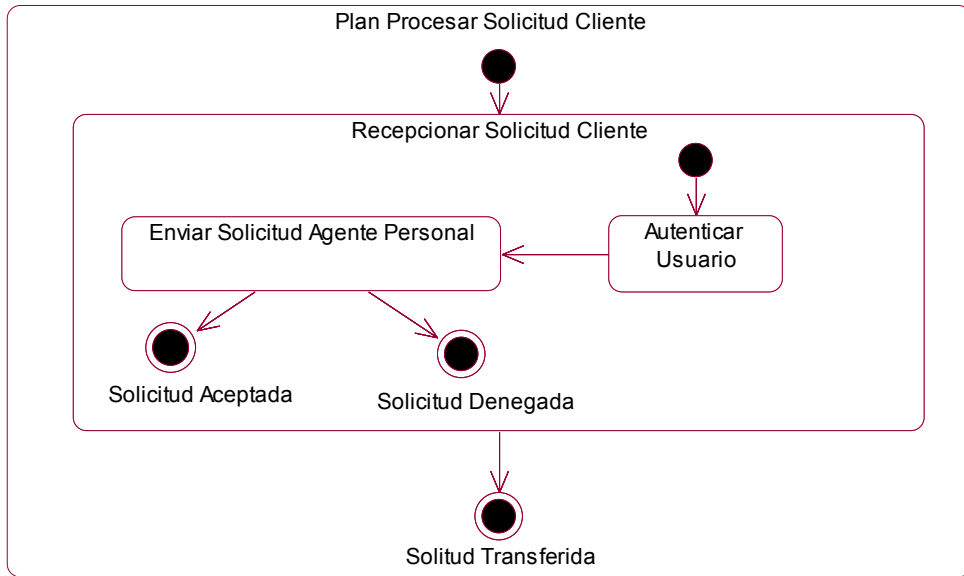


Figura 2. 11 : Modelo de planes del agente Interfaz

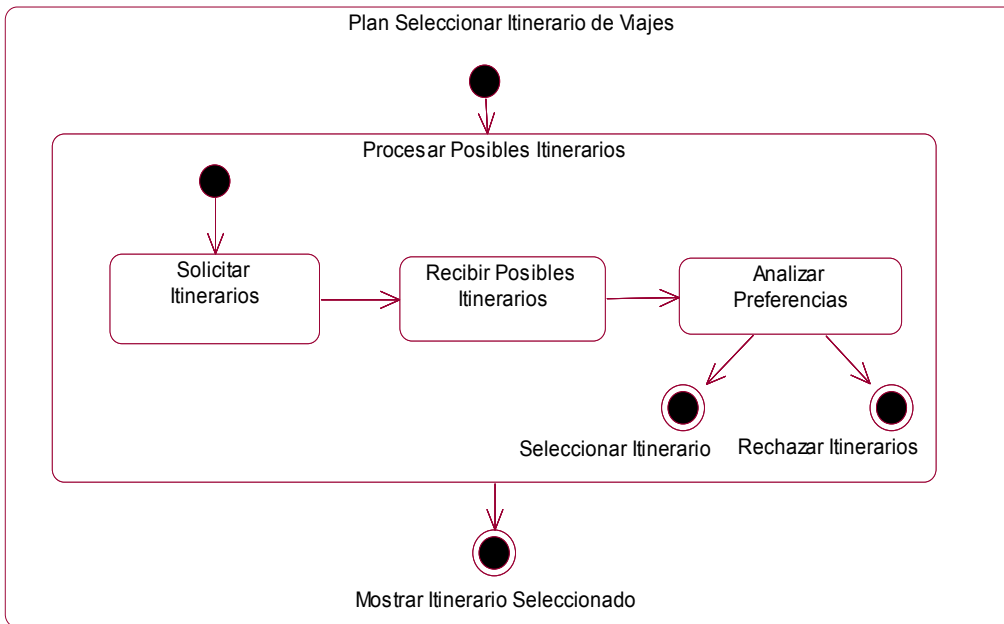


Figura 2. 12 : Modelo de planes del agente Personal

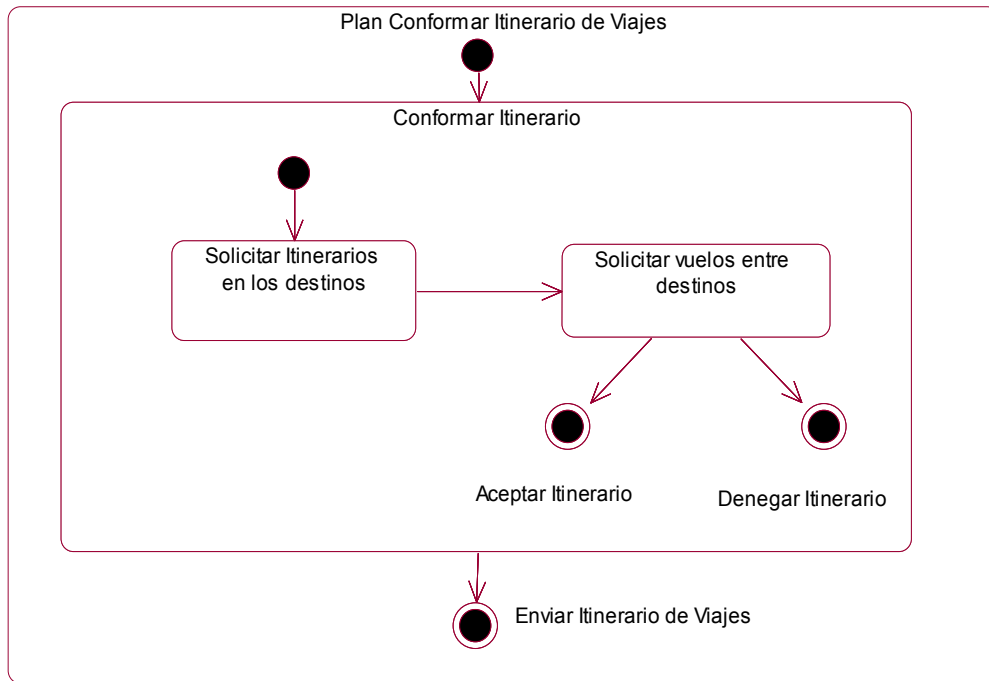


Figura 2. 13 : Modelo de planes del agente Personal

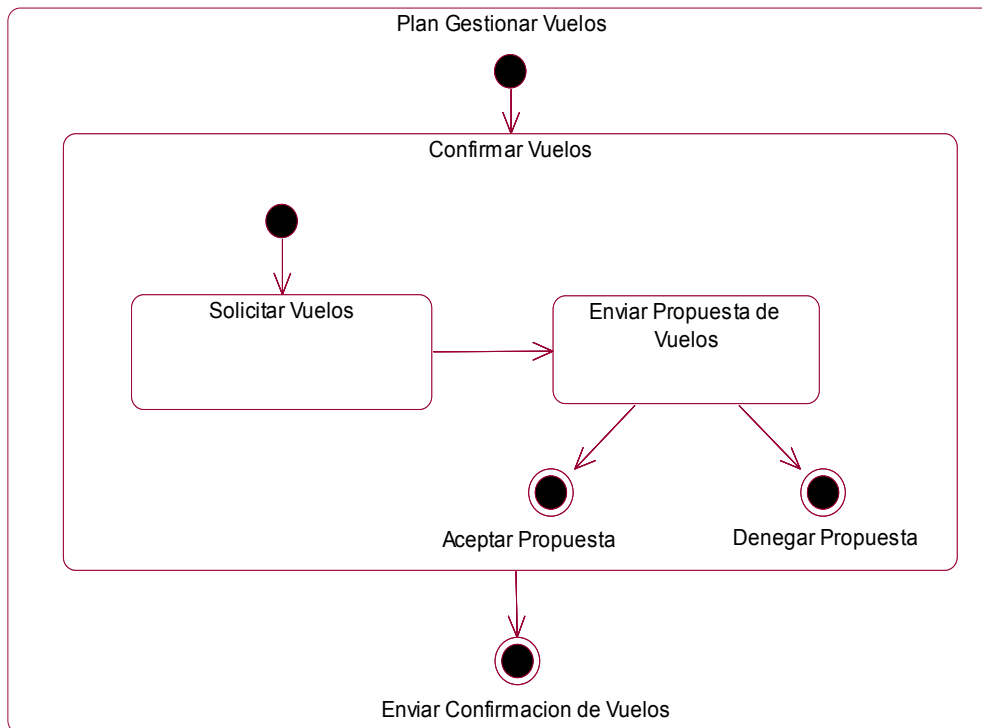


Figura 2. 14 : Modelo de planes del agente Vuelos

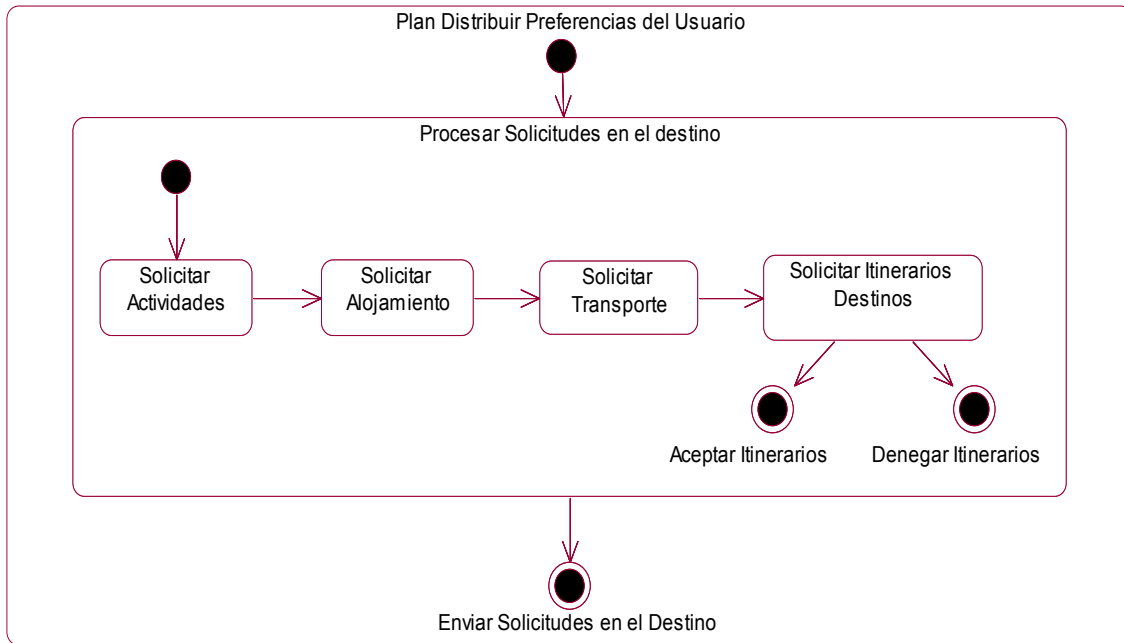


Figura 2. 15 : Modelo de planes del agente Local

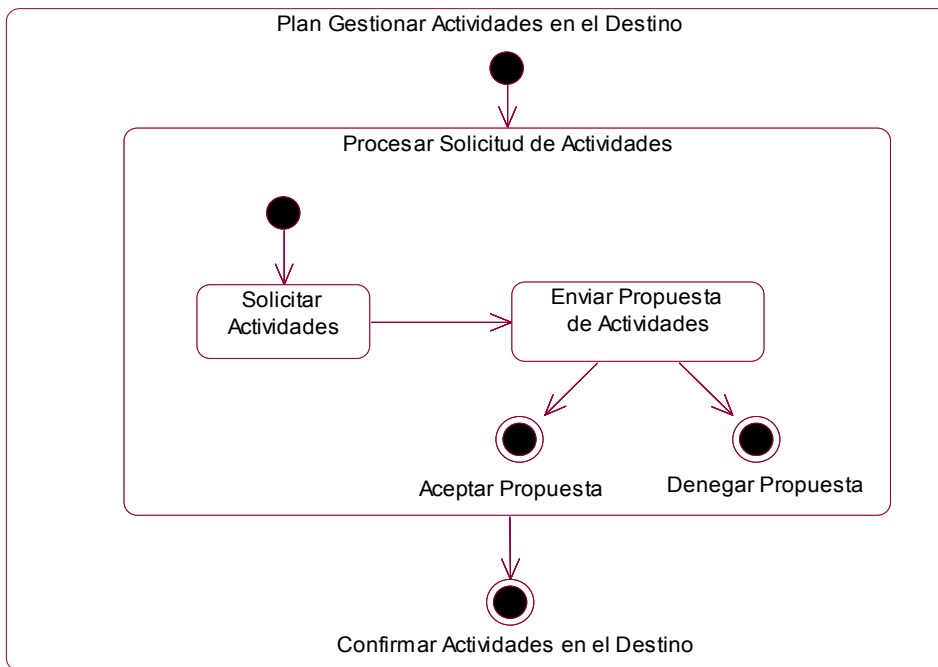


Figura 2. 16 : Modelo de planes del agente Actividades

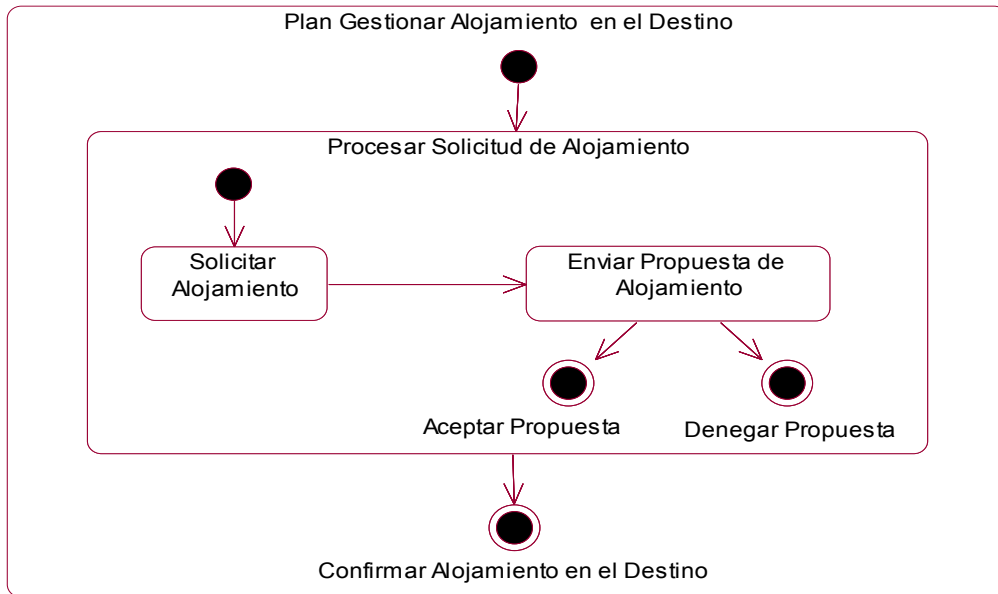


Figura 2. 17 : Modelo de planes del agente Alojamiento

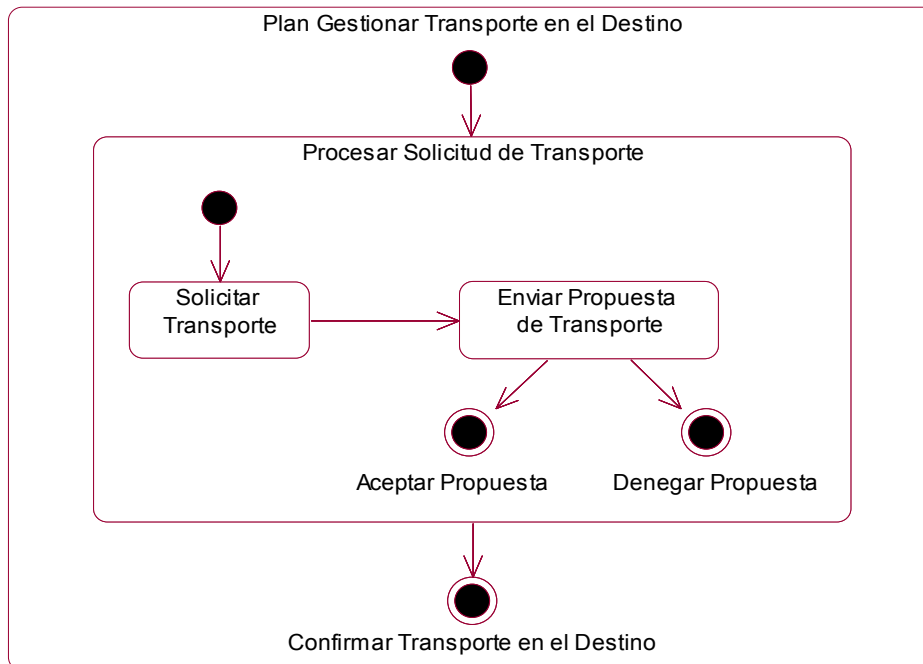


Figura 2. 18 : Modelo de planes del agente Transporte

2.8 Conclusiones

Después de haber desarrollado una profunda investigación de la metodología orientada a agentes AAIL, y de haber llevado a cabo el análisis y diseño del sistema Tour operador de viajes a través de la misma, se ha llegado a la conclusión de que resulta muy interesante para el desarrollo de sistemas multiagentes, ya que captura características claves de la definición de agentes como son las creencias, los deseos y las intenciones, además de hacer una clara separación entre los aspectos generales de la arquitectura del sistema de agentes y los aspectos particulares de la arquitectura de cada agente por separado.

Por otro lado, especifica la comunicación entre los agentes y describe los roles y responsabilidades de cada uno, como parte importante del funcionamiento del sistema en general. Sin embargo, es necesario mencionar que esta metodología posee ciertas limitaciones, por ejemplo, AAIL no abarca todas las fases del ciclo de vida de una aplicación basada en agentes. No obstante, por el conjunto de artefactos que propone, y por sus características en el modelado de sistemas multiagentes, se puede considerar como una metodología ágil y eficiente, lo cual le imprime una gran ventaja con respecto a otras metodologías orientadas a agentes.

Capítulo III: Evaluación de la metodología AAIL.

3.1 Introducción

RUP es el proceso de desarrollo de software más utilizado en la actualidad, debido a que ofrece un conjunto de etapas y flujos de trabajos con una serie de artefactos y abstracciones que permiten desarrollar con un nivel de detalle elevado todo el ciclo de vida de un sistema informático. Los sistemas basados en tecnología de agentes, como cualquier sistema inteligente, tienen varias características muy propias de ellos, como la representación y generación del conocimiento, las inferencias lógicas y los mecanismos de razonamiento, que hacen que su ciclo de vida se modele de manera diferente a los sistemas convencionales de gestión.

RUP está ampliamente validado y se encuentra en constante desarrollo y evolución, por lo que se ha considerado conveniente tomar esta metodología como base para realizar una valoración de AAIL.

3.2 El Proceso Unificado de Desarrollo (RUP)

RUP es el resultado de varios años de desarrollo y uso práctico, en el que se han unificado varias técnicas de desarrollo de software. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se le identifica. Más que un simple proceso, RUP constituye un marco de trabajo genérico, que puede especializarse para una gran variedad de sistemas, incluso para diferentes áreas de aplicación, tipos de organizaciones, niveles de aptitud y tamaños de proyectos, pero esencialmente se emplea en software de gestión de propósito general. Además cuenta con una serie de extensiones para aplicaciones más específicas, como son las multimedias, aplicaciones web, sistemas en tiempo real y algunas tecnologías de inteligencia artificial.

RUP reemplaza el tradicional esquema de desarrollo “en cascada”, donde cada etapa se iba desarrollando de forma secuencial. La aplicación de este enfoque durante años, ha demostrado que su uso genera serias dificultades, las cuales ponen en riesgo el éxito del proceso. Entre estas se pueden destacar:

- No se incorporan adecuadamente los cambios en los requerimientos que se producen durante el proceso de desarrollo, lo que la actual dinámica del mundo de los negocios hace resultar poco realista.
- No se enfrentan oportunamente los principales riesgos del proyecto, los cuales muchas veces son técnicos. Es frecuente, por tanto, constatar que proyectos que han llevado un gran esfuerzo, fracasan por problemas de performance que pudieron haber sido detectados prematuramente.

Para hacer frente a estos problemas RUP propone un esquema “iterativo” e “incremental”, donde el ciclo de desarrollo en vez de ser una única cascada, es una secuencia de cascadas más pequeñas; al término de cada iteración se efectúan evaluaciones que permiten definir el alcance de las próximas, y detectar oportunamente desviaciones en el alcance o la planificación del proyecto, tomando así las medidas necesarias.

En RUP se han agrupado las actividades en grupos lógicos, definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería (Modelamiento del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba e Instalación) y los 3 últimos como de apoyo (Administración del proyecto, Administración de configuración y cambios y Ambiente). Este proceso está estructurado de forma bidimensional, tal como se muestra en la figura 4, donde se representan los flujos de trabajo y la evolución en el tiempo a través de 4 fases (Inicio, Elaboración, Construcción, Transición), mostrándose la dinámica expresada en iteraciones y puntos de control.

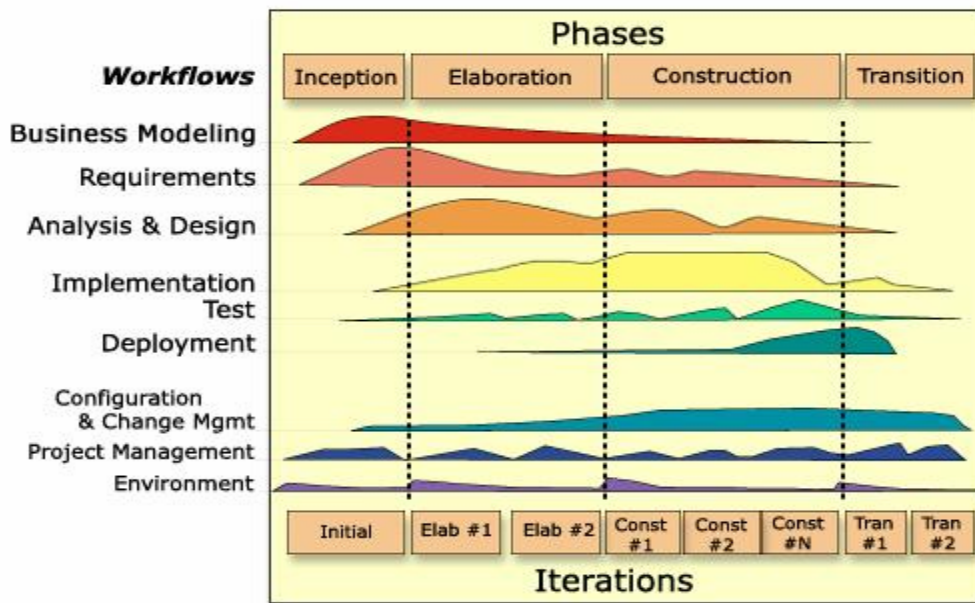


Figura 3. 1 : RUP en dos dimensiones

El ciclo de vida de RUP se caracteriza por ser:

- **Dirigido por Casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos funcionales. A partir de aquí los casos de uso guían el proceso de desarrollo, obteniéndose los distintos modelos como resultado de la transición por los diferentes flujos de trabajo.
- **Centrado en la Arquitectura:** La arquitectura muestra la visión común del sistema completo, en la que el equipo del proyecto y los usuarios deben estar de acuerdo, ya que describe los elementos del modelo que son más importantes para su construcción, y los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura. Tal como se aprecia en la figura 5, el modelo de

arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML (Lenguaje Unificado de Modelado).

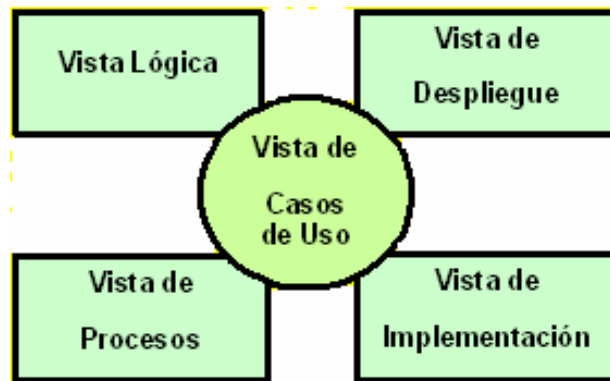


Figura 3. 2 : Vista del Modelo de Arquitectura

- **Iterativo e Incremental:** RUP propone que cada una de sus fases se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros, según el objetivo de la fase en que se encuentre el proyecto. Por ejemplo, una iteración de la fase de elaboración, centra su atención en el análisis y diseño, aunque también refina los requerimientos y obtiene un producto con un determinado nivel, que ira creciendo incrementalmente en cada iteración. Cada iteración tiene como resultado una versión del software, que se irá desarrollando incrementalmente durante todo el ciclo de vida del proyecto.

Según RUP, el tiempo necesario para llevar a cabo el proceso de desarrollo de un software (ciclo de vida del software), se debe distribuir en 4 fases. Como ya se dijo anteriormente, en cada fase hay una o más iteraciones, según sus objetivos específicos:

- La fase de **Concepción** o **Inicio** tiene por finalidad definir la visión, los objetivos y el alcance del proyecto, tanto desde el punto de vista funcional como del técnico, identificándose entre los principales resultados, los casos de uso del sistema, los factores de riesgo y el costo del proyecto. El principal esfuerzo radica en los flujos

de trabajo Modelamiento del negocio y Requerimientos. Es la única fase que no necesariamente culmina con una versión ejecutable, pero si debe exhibir y demostrar al menos una arquitectura candidata.

- La Fase de **Elaboración** tiene como meta definir la arquitectura del sistema y obtener una aplicación ejecutable que responda a los casos de uso que la comprometen, proporcionando una base sólida para el diseño y la implementación. Esta fase centra su atención en el flujo de trabajo de Análisis y diseño, y el equipo del proyecto establece los fundamentos de los elementos arquitectónicamente significativos de los modelos de diseño y despliegue.
- En la fase de **Construcción** se produce un cambio de enfoque. Mientras que las fases anteriores podrían ser consideradas como de investigación, la fase de Construcción es análoga al desarrollo. El propósito de la misma es obtener un producto software en su versión operativa inicial, documentado y con manual de usuario. Se obtienen además 1 o varios release del producto que han pasado las pruebas y se ponen a consideración de un subconjunto de usuarios. Los principales esfuerzos de esta fase se desarrollan en los flujos de Implementación y Prueba.
- En la fase de **Transición** el release ya está listo para su instalación en las condiciones reales, por lo que los principales esfuerzos radican en el flujo de Instalación. Esta fase puede implicar reparación de errores.

RUP brinda un proceso integrado que utiliza el estándar de notación UML (Lenguaje Unificado de Modelado), para representar todos los artefactos de modelado, tanto del negocio como del sistema, imprescindibles para el desarrollo satisfactorio de cualquier proyecto informático. (SOFTWARE, 2004-2005)

3.3 Tecnología orientada a agentes

Los sistemas basados en agentes inteligentes constituyen una de las más avanzadas tecnologías de inteligencia artificial en la actualidad. Los mismos permiten integrar bases de conocimiento, sistemas adaptativos, técnicas de inteligencia artificial subsimbólica (redes neuronales y redes bayesianas), y ontologías para la representación del conocimiento.

El proceso unificado de desarrollo (RUP) no cuenta con un soporte o extensión para el análisis y diseño de este tipo de sistemas, que además poseen características muy específicas que no pueden ser modeladas como en un sistema convencional de gestión, por lo que se han creado nuevas metodologías de análisis y diseño orientadas a agentes inteligentes (GAIA, AAIL, Zeus y Tropos), o se han extendido algunas de las ya existentes. Estas últimas intentan aplicar métodos ya probados al enfoque de agentes, como son PASSI, Mase, y MESSAGE/UML, entre otras.

3.4 Análisis comparativo entre la metodología AAIL y el Proceso Unificado de Desarrollo (RUP)

Para comparar la metodología AAIL como propuesta para la construcción de sistemas multiagentes, con el Proceso Unificado de Desarrollo (RUP), estándar actual de referencia del paradigma orientado a objetos, se han tenido en cuenta los siguientes aspectos, que serán detallados a continuación:

- Paradigma que respaldan
- Proceso que siguen.
- Fases y etapas que abarcan.
- Roles del equipo de proyecto que definen.
- Lenguaje de modelado que emplean.
- Herramienta CASE que utilizan.
- Trazabilidad entre los pasos que proponen.

El Proceso Unificado de Desarrollo (RUP) constituye el estándar actual de referencia del conocido paradigma de orientación a objetos y más actualmente, la llamada programación orientada a componentes. La metodología AAll, en otro sentido, establece una propuesta orientada al análisis y diseño de sistemas enmarcados en el novedoso paradigma de orientación a agentes, siguiendo el modelo BDI.

El proceso seguido por RUP es iterativo e incremental, es decir, opta por dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Por otra parte, la metodología AAll sigue un desarrollo secuencial, transitando de una etapa a la otra; análisis y diseño, en ese orden.

El Proceso Unificado de Desarrollo (RUP), identifica cuatro fases dentro de su ciclo de vida: inicio, elaboración, construcción y transición, las cuales se desarrollan a través de iteraciones que abarcan los nueve flujos de trabajo: Modelamiento del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, Instalación, Administración del proyecto, Administración de configuración y cambios y Ambiente. La metodología AAll, como se mencionara anteriormente, sigue un proceso secuencial y no define ninguna fase, pues su desarrollo consta de un ciclo de vida donde se suceden dos etapas: análisis y diseño. La etapa de análisis incluye: división de la organización, modelado del entorno, modelado de roles y protocolos de interacción preliminares, y la definición de la estructura organizacional del sistema. La etapa de diseño modela la vista externa y la vista interna del sistema multiagentes. En el diseño de la vista externa se definen los modelos de roles y protocolos de interacción finales, y se desarrollan los modelos de agentes y de interacción. Los detalles del diseño de la vista interna, se capturan para cada clase de agente, a través de los modelos de objetivos, creencias y planes.

RUP define claramente para cada iteración de una fase, los roles involucrados del equipo de proyecto, con sus respectivas actividades en cada flujo de trabajo. AAll no identifica claramente la existencia de roles dentro del grupo de trabajo de un proyecto, por lo cual se asume que cada especialista del equipo participa en cada una de las etapas del proceso de desarrollo del sistema multiagentes. La ausencia de definición de roles en AAll, puede traer como consecuencia que no se consigan establecer responsabilidades

en cuanto a la realización de actividades y modelos, donde la especialización o las habilidades de una persona, pueden no ser las mismas para desempeñarse en todas las etapas de la metodología, lo que repercutiría como ineficiencias en la entrega de los productos de software, provocando retrocesos y reestructuraciones innecesarias, entre otros inconvenientes.

RUP emplea como estándar de notación el Lenguaje Unificado de Modelado (UML), ya que permite visualizar, especificar, construir, documentar y comunicar los artefactos a lo largo del ciclo de vida del proyecto. La metodología AAIL utiliza elementos de UML para el modelado de algunos artefactos que la componen, por ejemplo los modelos de clases e instancias de agentes, que forman parte del diseño de la vista externa, y los modelos de creencias y de planes de agentes, lo cuales se desarrollan durante el diseño de la vista interna del sistema multiagentes. En estos momentos se está proponiendo AUML (Agent Unified Modeling Language), como lenguaje de modelado para todas las metodologías orientadas a agentes, pero aún no ha sido validado totalmente para AAIL; por este motivo todavía es aceptado emplear UML para modelar los sistemas desarrollados a partir de esta metodología. En el modelado de otros artefactos, tanto en la etapa de análisis como en la de diseño, AAIL no emplea ningún lenguaje en específico, sino que los representa de forma manual, utilizando esencialmente tablas y esquemas.

La empresa Rational propuso en el año 2000 una herramienta CASE para los proyectos que se fuesen a desarrollar con RUP, conocida como Rational Rose, que constituye la líder en el mundo de modelación visual para el proceso de modelado del negocio, análisis de requerimientos y diseño de arquitectura de componentes, brindando una serie de posibilidades que facilitan el trabajo de los miembros del equipo. Es importante destacar que Rational Rose no es la única herramienta CASE, que puede emplear RUP, existen otras como Selects Enterprise, Visual UML y Visual Paradigm que permiten desarrollar los modelos gráficos propuestos por UML, por lo tanto un software desarrollado a partir de RUP, pudiese emplear algunas de estas herramientas. La selección de la misma está en dependencia de las características del proyecto que se este realizando, y del personal de dicho proyecto.

Por otra parte, no existe ninguna herramienta CASE específica que sustente los modelos definidos por AAll durante sus etapas de desarrollo, por lo que se considera oportuno utilizar cualquiera que permita desarrollar los elementos propuestos por la misma y modelados con UML., por ejemplo Visual Paradigm, Mono UML, Argo UML , BOUML, entre otras. En el desarrollo del caso de estudio seleccionado para esta investigación, se utilizó el Rational Rose, por ser una herramienta debidamente estandarizada, líder en el mercado de herramientas de modelado y abarcadora de toda la simbología de UML. Además es fácil de emplear y es ampliamente estudiada y aceptada en el ámbito académico y empresarial en Cuba.

Se señala que si AAll contara con una herramienta CASE elaborada específicamente para ella, como Rational Rose lo es para RUP, el desarrollo de proyectos con tecnología basada en agentes inteligentes, se facilitaría y haría más eficiente.

El Proceso Unificado de Desarrollo (RUP) proporciona un conjunto claro de actividades a desarrollar durante el transcurso de todas sus fases y etapas, sugiriendo además cuales modelos, diagramas o artefactos generar primero y cuales más tarde; no obstante, el proceso no es rígido, pues puede realizarse a criterio del desarrollador, teniendo en cuenta las características y restricciones del sistema a implementar. La transición entre fases y etapas está bien definida, especificándose en cada caso los resultados esperados. La metodología AAll por su parte, cuenta con un conjunto de modelos y artefactos definidos durante el transcurso de sus etapas, las cuales si tienen una forma rígida de desarrollo, pero entre los artefactos de una misma etapa no existe relación de dependencia, permitiendo su construcción en el orden que los desarrolladores estimen conveniente.

Es oportuno señalar que, si bien el desarrollo con RUP es dirigido por casos de uso que proporcionan un medio sistémico e intuitivo para el recorrido por todas sus fases y flujos de trabajo; la identificación de los roles y los protocolos de interacción en AAll, guían el desarrollo de todo el proceso, ya que ambos son refinados y completados en el transcurso de las etapas de la metodología. Además estos están relacionados con los recursos del entorno, tributan a la elección de la estructura organizacional que les da soporte, y son determinantes en el momento de definir los agentes y los servicios que estos prestan.

A partir de análisis anterior se puede concluir:

- RUP es un proceso de desarrollo propuesto esencialmente para sistemas de gestión de cualquier tipo (empresariales, académicos, salud, etc.). Es sumamente robusto, profundo, y muy completo. Al ser un proceso iterativo e incremental permite ir corrigiendo desde el principio, los posibles errores e insuficiencias que pueda tener el sistema. No es apropiado para ser empleado en el desarrollo de sistemas inteligentes o de módulos de sistemas que requieran de técnicas de inteligencia artificial, ya que en estos sistemas debe ser modelado y representado el conocimiento y debe quedar bien especificado como emplearlo. Los sistemas inteligentes dan solución a problemas de índole heurística o donde los algoritmos que existen son de complejidad no polinomial. RUP no cuenta con ningún artefacto que permita considerar las condiciones heurísticas de estos sistemas.
- AAIL es una metodología de análisis y diseño orientada a agentes, por lo que solo es apropiada para el desarrollo de sistemas multiagentes, habiéndose demostrado esto al ser empleada en, al menos, el desarrollo de dos sistemas multiagentes. Es ágil y muy fácil de entender debido a la sencillez de la mayoría de los artefactos con que cuenta.

3.5 Evaluación de la metodología AAIL

Para evaluar la metodología AAIL se tomaron en cuenta algunos aspectos que se consideraron importantes, contenidos en las plantillas de evaluaciones de metodologías orientadas a agentes (*framework*). A estos se le agregaron otros que por su relevancia, se consideró necesario reflejarlos.

- Proceso de desarrollo
- Vista de modelos
- Agentes
- Modelación de características adicionales
- Documentación
- Otros aspectos

3.5.1 Proceso de desarrollo

La evaluación del proceso de desarrollo de cualquier metodología consiste en el análisis de su dominio y áreas de aplicación, capacidad para sustentar sistemas abiertos, tipo de ciclo de vida que presenta, completitud de sus fases, herramienta CASE que utiliza y plataforma de desarrollo que propone.

La metodología AAIL está orientada al análisis y diseño de sistemas multiagentes, por lo que su dominio de aplicación abarca todo lo relacionado con la ingeniería de software para sistemas multiagentes, definiendo artefactos y abstracciones que anteriormente no habían sido empleados en el desarrollo de sistemas de gestión convencionales orientados a objetos. Entre estos artefactos se encuentran el modelo de clases de agentes y el de instancias de agentes, como parte del diseño de la vista externa; y en el diseño de la vista interna se desarrollan los modelos de creencias y de planes para cada uno de los agentes que componen el sistema.

AAIL puede modelar cualquier sistema basado en agentes inteligentes, en cualquier escenario de desarrollo, tanto físico como virtual. Los resultados obtenidos en la aplicación de la misma a la construcción de sistemas multiagentes reales, han sido satisfactorios, por ejemplo el sistema Lucia, que tiene como objetivo la enseñanza a niños ciegos; y el sistema para el control de tráfico aéreo del aeropuerto de Sidney. Esto demuestra que el tránsito por las distintas etapas, y el desarrollo de los diferentes modelos y artefactos que posee la metodología, permite llevar a cabo exitosamente el desarrollo de un sistema basado en la tecnología de agentes.

Esta metodología intenta modelar sistemas multiagentes en entornos abiertos y distribuidos, en los que cada agente tiene que sacar provecho del uso de sus servicios, sus conocimientos y su capacidad para encontrar otros agentes diseminados a través de una red, con los cuales alcanzar un flujo de trabajo eficiente para el logro de sus objetivos. En el caso de los sistemas distribuidos, los agentes deben tener la capacidad de desarrollar simultáneamente varias tareas en diferentes entornos de trabajo mediante sus instancias.

Las aplicaciones multiagentes se caracterizan por encontrarse en un entorno sumamente dinámico, en el que por un lado, el contexto de los recursos del sistema puede cambiar de acuerdo con determinados patrones tanto internos como externos a la organización, y por otro, algunos de estos recursos pueden estar disponibles intermitentemente, así como también pueden incluir componentes activos (otros agentes). Todo esto es tomado en cuenta por AAll, a la hora de desarrollar un sistema con estas características.

La etapa de análisis en AAll comienza con la división de la organización, para posteriormente realizar el modelado del entorno, de los roles preliminares y de los protocolos de interacción preliminares, así como la definición de la estructura organizacional del sistema. Estos elementos permiten obtener un conocimiento preciso de la composición de la organización y de los procesos que en ella se desarrollan. Además constituyen la base del desarrollo de cualquier sistema. La etapa de diseño se divide en: diseño de la vista externa y diseño de la vista interna, modelando los elementos necesarios para desarrollar el sistema. AAll no cuenta con una etapa propia de implementación, ni de prueba, trayendo como consecuencia que puedan existir contradicciones entre analistas, diseñadores e implementadores. Ambas etapas (implementación y prueba) vienen embebidas dentro del diseño de la vista interna del sistema, específicamente en los modelos de creencias y planes de agentes. A partir de los resultados obtenidos en el desarrollo de sistemas multiagentes y del caso de estudio abordado en este trabajo, se considera que las etapas de AAll, completan el proceso de desarrollo de sistemas multiagentes y son aplicables a todos los entornos donde puedan existir sistemas con tecnología basada en agentes, es decir entornos cerrados, abiertos y distribuidos.

AAll no define roles específicos dentro del equipo de proyecto, ni tampoco fases en el proceso de desarrollo del sistema. Se considera que si contase con estos elementos, sería más completa la planificación, y se ganaría en tiempo y optimización de recursos. Por otro lado, al no definirse roles dentro del equipo de proyecto, un mismo grupo de personas asumiría todo el desarrollo de un módulo del sistema, impidiendo que se pueda hacer más eficiente y económico el proceso de desarrollo.

Esta metodología no cuenta con una herramienta CASE propia, sus modelos y abstracciones pueden representarse empleando cualquier versión de Rational Rose u otra herramienta CASE que permita emplear el lenguaje UML. Se considera que sería muy apropiado y conveniente desarrollar una herramienta CASE, que se ajuste a las características de cada uno de los modelos que propone AAll.

Para la implementación de sistemas multiagentes existen varias plataformas como JADE (Java Agent Develepment Enviroment), Javalog (que es una extensión de Prolog), 3APL (basado en BDI), entre otros. Cualquier sistema multiagentes desarrollado con AAll, se puede implementar con cualquiera de estas plataformas. Es bueno señalar que la mayoría de ellas, constituyen extensiones de lenguajes y técnicas de programación orientada a objetos o del paradigma lógico de la programación.

Se puede concluir que la metodología AAll, permite desarrollar el análisis y diseño de un sistema multiagentes de manera satisfactoria, siendo esto corroborado con los resultados de los sistemas multiagentes reales desarrollados con esta metodología y el caso de estudio abordado en este trabajo. Sin embargo cuenta con limitaciones que le impiden lograr una mayor eficiencia y optimización en los procesos que se van a desarrollar a través de ella.

3.5.2 Vista de modelos

AAll presenta una serie de abstracciones en sus artefactos que caracterizan la organización y permiten realizar el análisis y diseño del sistema. En la siguiente tabla se representan esas abstracciones.

Artefacto	Descripción	Etapa	Concepto
División de la Organización	Descompone la organización en suborganizaciones.	Análisis	Organización
Modelo del entorno	Representación computacional del entorno donde será ubicado el sistema.	Análisis	Entorno

Modelo de roles preliminares	Contiene los roles preliminares que son identificados, pero no necesariamente definidos.	Análisis	Roles
Modelo de protocolos de interacción preliminares	Contiene las interacciones básicas requeridas para garantizar las funcionalidades del sistema.	Análisis	Interacciones y protocolos
Modelo de clases de agentes	Identifica las distintas clases de agentes del sistema, y las relaciones entre ellas.	Diseño de la vista externa	Agentes Vista externa
Modelo de instancias de agentes	Identifica las instancias de cada clase de agente del sistema.	Diseño de la vista externa	Agentes Instancias
Modelo de interacciones	Define las responsabilidades y servicios de cada clase de agente.	Diseño de la vista externa	Agentes Interacciones
Modelo de objetivos	Define los objetivos (deseos) de cada agente.	Diseño de la vista interna	Deseos de agentes
Modelo de creencias	Define la composición interna de cada agente.	Diseño de la vista interna	Creencias de Agentes
Modelo de planes	Define los planes (intenciones) de cada agente, para lograr sus objetivos.	Diseño de la vista interna	Intenciones de agentes

Tabla 19 : Artefactos fundamentales de la metodología AAll

Es necesario destacar que el diseño de la vista interna se rige por el modelo BDI (Belive, Desire, Intention), por lo que cada artefacto componente del diseño de esta vista se corresponde con alguna de estas características.

AAll no define ninguna técnica de modelado para determinar como debe ser la relación entre el sistema y sus usuarios, y no deja clara la definición de las responsabilidades de los mismos. Tampoco define en ninguna etapa, ni modelo, los diferentes tipos de usuarios que harán uso del sistema multiagentes. Estos dos elementos constituyen otra de las limitantes que presenta la metodología.

El hecho de que esta metodología no utilice ninguna técnica de modelado específica, permite a los especialistas emplear la que ellos estimen más adecuada. En el caso de estudio desarrollado se empleó UML, para representar los modelos compuestos por diagramas y esquemas. A partir de este resultado se considera que aunque la metodología ofrece flexibilidad a la hora de seleccionar el lenguaje de modelado, debiera elaborarse alguno específico para AAIL, que permita unificar el modelado de los sistemas multiagentes que se desarrollen a partir de la misma.

Es meritorio señalar que en la actualidad se realizan investigaciones para fomentar un estándar de modelado para agentes basado en UML, denominado AUML, aunque todavía no se han logrado resultados lo suficientemente viables y robustos como para su difusión total. Los autores de AAIL tienen presente este aspecto y aseguran que cuando las condiciones estén dadas, y alguna técnica visual de modelado resulte lo suficientemente fuerte, la metodología podría intentar ajustar sus artefactos y modelos, asumiéndola como su lenguaje. Esto le proporcionaría un mayor atractivo para la industria de desarrollo de software, y le aportaría una forma visual mucho más manejable, entendible, y fácil de actualizar, lo que en sentido general puede contribuir de forma precisa a la comunicación de sus modelos, y no solo entre los miembros de un proyecto, sino también entre otras personas involucradas con él, y con el paso del tiempo, entre los desarrolladores de futuras generaciones.

Los artefactos que suministra esta metodología, aunque no son rígidos en su orden de aparición, si muestran una traza entre ellos que los conecta, proporcionando una mayor comprensión y facilitando la gestión de cambios.

Como primer paso de la etapa de análisis de AAIL, se divide la organización en pequeñas partes o sub-organizaciones, lo que evidencia la modularidad del sistema y facilita el trabajo. La labor consiste en trabajar por suborganización, identificando en cada una los recursos del entorno que puede censar, así como los roles y protocolos de interacción preliminares que permitirán su desempeño básico, para luego en sentido general, definir la estructura organizacional, que regirá el funcionamiento global de todas las suborganizaciones.

En el diseño de la vista externa, se definen los roles y protocolos de interacción del sistema, se modelan las clases de agentes y las instancias de agentes, proporcionando comprensión y sencillez al diseño del sistema. Esto facilita el posterior modelado de la vista interna, que puede considerarse como el diseño detallado de cada agente en particular.

Las limitantes que presenta AAll no son exclusivas de ella, el resto de las metodologías orientadas a agentes también las tienen, incluso muchas de ellas son más insuficientes que AAll. No obstante, lo señalado anteriormente, no se considera un obstáculo para el desarrollo de sistemas multiagentes a partir de esta metodología, por el contrario los resultados de la aplicación de la misma son satisfactorios comparados con los de otras metodologías orientadas a agentes.

3.5.3 Agentes

El concepto de agentes que propone la metodología AAll, es el de una entidad que posee funcionalidades y estado propio, lo cual le permite cumplir un determinado objetivo. Esta metodología se basa además en el modelo BDI (*belief, desire, intention*: creencia, deseo, intención), que ve a los agentes como sistemas intencionales, cuyo comportamiento puede predecirse y explicarse en términos de actitudes, de manera similar a cómo se predice el comportamiento de los sistemas inteligentes complejos (personas). Como se expresó anteriormente cada agente tiene objetivos que cumplir, haciéndolo de manera independiente, delegando responsabilidades y funcionalidades. No obstante, los agentes de manera general, no existen solos en un medio, sino que conviven con otros agentes en entornos donde cada uno de ellos tiene una serie de responsabilidades y funciones, que se integran en un sistema para darle solución a una problemática. Estos sistemas son los llamados multiagentes.

La definición de agente propuesta por AAll integra además, los conceptos de autonomía (descrita como la capacidad para decidir que acción debe tomar y en que momento hacerlo), reactividad (acción ante una situación particular dada), proactividad (capacidad de actuar de manera oportuna antes de que se sucedan los acontecimientos), y habilidad

social (la interacción con otros agentes como una necesidad o para lograr un objetivo común). Todo esto se basa, como se ha explicado anteriormente, en el modelo BDI.

La arquitectura multiagentes propuesta por la metodología pudiera ser considerada como dinámica y abierta, prevaleciendo una clara distinción entre entidades activas (agentes) y pasivas (modeladas en términos de recursos del entorno), donde la delegación del control de las acciones en componentes autónomos, puede ser enriquecida con habilidades sociales a la hora de tomar decisiones sobre el alcance, la naturalidad y la flexibilidad de las interacciones.

La metodología AAll, propone una representación de la inteligencia de los agentes a través del modelado de bases de conocimientos para cada uno de ellos. Esta cierta inteligencia es lo que regula su comportamiento y la base de conocimientos se puede modelar a través de ontologías, frame, redes semánticas o cualquier otra técnica subsimbólica, como redes neuronales o bayesianas, reglas de producción e incluso razonamiento basado en casos. Se puede afirmar que el conocimiento se puede representar con cualquier técnica, tanto simbólica como subsimbólica.

Otro aspecto importante que la metodología tiene en cuenta desde el principio, es el modelo de roles, que define esencialmente los agentes que van a conformar el sistema y las responsabilidades que cada uno de ellos va a tener, ya que un agente puede asumir uno o varios roles, dentro del sistema.

AAll modela la interacción entre los agentes y otras entidades del sistema y su entorno, en este caso, desde su etapa de análisis, identifica los recursos tanto externos como internos con los que los agentes tendrán que interactuar, y establece un modelo de protocolos de interacción, que durante el transcurso de sus etapas se va refinando.

Los sistemas que se construyen basados en esta metodología, se caracterizan por su gran dinamismo, y autonomía. Por lo que un seguimiento y perfeccionamiento de las interacciones contribuirá al éxito y solidez de los mismos. La facilidad para modelar las interacciones puede evaluarse desde la óptica que AAll propone, como sencilla, sin embargo, para implementar las interacciones puede ayudar muchísimo una

representación visual de las mismas, para esclarecer así el orden de los mensajes que se intercambian los distintos agentes.

3.5.4 Modelado de características adicionales

La evaluación respecto a este tema de la metodología AAll, consiste en valorar aspectos tales como el modelado del ambiente, la organización, la ontología y la movilidad de los agentes.

La metodología AAll emplea la modelación del ambiente como un elemento esencial, que va a servir de base para el análisis y diseño del sistema. Partiendo de esta premisa, los primeros artefactos de AAll son la división de la organización y el modelado del entorno, permitiendo esto una mejor comprensión de la organización y del entorno donde se desarrollará el sistema multiagentes. Un eficiente modelado del entorno permitirá determinar con precisión los procesos que serán objeto de automatización y los requerimientos que deberá cumplir el sistema, lo que garantizará posteriormente una representación más natural e intuitiva del mismo.

La metodología no hace mención a la ontología, ni a rasgos de movilidad de los agentes, estos temas no son abordados por los autores de AAll. Esto le aporta mayor flexibilidad a la hora de seleccionar la técnica a emplear para representar el conocimiento.

3.5.5 Documentación.

La evaluación respecto a la documentación de la metodología AAll consiste en valorar aspectos tales como: la bibliografía disponible y las aplicaciones terminadas realizadas con la misma.

La documentación disponible de la metodología AAll puede encontrarse de forma libre en Internet, preferiblemente referenciada por sus autores, aunque existen varios artículos donde es abordada. Pudiera calificarse la documentación como escasa, sin embargo describe de forma correcta todos los aspectos relacionados con la metodología.

Hasta la fecha se han reportado varios sistemas multiagentes que han empleado AAll en su análisis y diseño satisfactoriamente. Como criterio de la autora de este trabajo, el sistema más importante desarrollado hasta el momento, es el Oasis, para el control de tráfico aéreo en el aeropuerto de Sidney, Australia; publicado por Kinny en 1996. Este describe el sofisticado sistema sustentado por la tecnología de agentes. Otro reporte sobre el empleo de esta metodología proviene de Paraguay, donde se llevó a cabo el desarrollo de un sistema llamado Lucia, con la finalidad de ayudar en el proceso de fortalecimiento de las destrezas sensor- motriz de niños con discapacidad visual a través de hiperhistorias, y además ofrecer indicaciones a los docentes sobre el desempeño del alumno en el juego. Los resultados obtenidos con este sistema avalan la viabilidad y valor práctico del proceso de desarrollo para sistemas multiagentes propuesto por AAll, ya que se demostró que los niños estaban más motivados con el juego cuando recibían las sugerencias del agente; lo que sugiere que, a mayor interacción con el entorno de aprendizaje, mejor será el afianzamiento de sus destrezas.

Estos elementos demuestran que esta metodología es viable y válida para guiar el proceso de desarrollo de sistemas multiagentes. Esto último es corroborado también por los resultados del caso de estudio abordado en este trabajo.

3.5.6 Otros elementos

AAll es una metodología que se puede considerar ágil, por el número de artefactos que contempla, y además es factible de aplicar en el proceso de desarrollo de un sistema multiagentes, debido a que modela eficientemente las etapas de análisis y diseño de un sistema de este tipo. Un conjunto de sistemas desarrollados a partir de esta metodología así lo corroboran. Por otra parte, AAll es flexible en muchos aspectos, pero esta flexibilidad se hace más notable en dos aspectos fundamentales:

- Plataforma para el desarrollo de sistemas multiagentes: AAll no está vinculada a ningún entorno de desarrollo de agentes o sistemas multiagentes, esto le da la posibilidad al equipo de proyecto de seleccionar la que estimen conveniente, pero tiene la desventaja de que no de la posibilidad de generar código de manera automática.

- Modelación del conocimiento de los agentes: Este es uno de los elementos fundamentales de cualquier sistema inteligente, y en este sentido, AAIL también deja en manos de los especialistas la técnica de modelación del conocimiento a seleccionar.

Se pueden identificar como ventajas de la metodología AAIL, las siguientes:

- Es de rápido desarrollo, y cuenta con los artefactos necesarios para conducir eficientemente las etapas de análisis y diseño de un sistema multiagentes.
- Al estar basada en el modelo BDI, se centra fundamentalmente en los objetivos de cada agente, en lo que cada uno de ellos debe hacer para cumplirlos y en el funcionamiento inteligente de los mismos, a través del modelado de su comportamiento y características esenciales.
- Ha sido validada por el desarrollo de al menos dos sistemas multiagentes reales a partir de ella.
- Es fácil de entender y sus abstracciones fundamentales no son muy complejas, después de estudiar la metodología y tener conocimientos de ingeniería de software.
- Modela la inteligencia de los agentes y su forma de interactuar, lo que simula un cierto comportamiento “social” y controla la autonomía de los mismos.
- Modela el entorno donde el sistema va a ser implantado, como resultado del reconocimiento de todas las entidades y recursos que la aplicación multiagentes puede explorar, controlar o consumir cuando trabaja en torno al logro de los objetivos de su organización.

Se pueden mencionar como desventajas de AAll:

- Solamente incluye las etapas de análisis y diseño, no cuenta con implementación, prueba y despliegue.
- No cuenta con una etapa propia para la captura de los requerimientos del sistema, pues se considera la etapa de análisis como una forma un tanto tardía de este proceso.
- No cuenta con una herramienta CASE propia, que sirva de sostén al modelado de todos y cada uno de sus artefactos.
- No realiza iteraciones a lo largo de su proceso de desarrollo, que posibiliten en algún momento una reestructuración de sus actividades y artefactos, ante posibles errores, omisiones de información o nuevos requerimientos.

3.6 Conclusiones

En el capítulo se ha llevado a cabo un análisis comparativo entre las metodologías AAll y RUP, para posteriormente realizar una valoración de AAll, concluyéndose que la misma es una de las metodologías orientadas a agentes más robusta, completa y profunda, ya que posee como basamento el modelo BDI. Debido a esto, AAll tiene como esencia, detallar los elementos componentes de la inteligencia de los agentes y las relaciones entre ellos, haciendo que todo el proceso tenga una lógica más natural e intuitiva.

De la misma manera presenta ciertas limitaciones, como se mencionara anteriormente, no cuenta con las etapas de implementación, prueba y despliegue, como otras metodologías empleadas en el proceso de desarrollo de sistemas más convencionales, tal es el caso de RUP. Por otro lado, al estar validada AAll por la implementación de al menos dos sistemas multiagentes a partir de ella, le confiere a la misma un valor agregado, con el que no cuentan otras metodologías de análisis y diseño orientadas a agentes, que aún se encuentran en el marco teórico del desarrollo de sistemas multiagentes.

Conclusiones

Los objetivos propuestos al comienzo del presente trabajo con respecto al estudio, aplicación y evaluación de la metodología AAll han sido cumplidos, y partiendo de esto se puede arribar a las siguientes conclusiones:

- Los agentes inteligentes han revolucionado la forma que existía hasta ahora de desarrollar las aplicaciones. Esta revolución ha generado un nuevo campo de investigación, como es la ingeniería de software orientada a agentes.
- La tecnología basada en agentes inteligentes, por su nivel de integración y posibilidad de modelar las soluciones de manera más natural e intuitiva, permite desarrollar situaciones cada vez más complejas en el campo de la inteligencia artificial.
- La metodología AAll es apropiada para el desarrollo de sistemas multiagentes. La misma se encuentra validada por el caso de estudio abordado en este trabajo, así como por el desarrollo de al menos dos sistemas multiagentes reales.
- La carencia de etapas propias de requerimientos, implementación y prueba, así como la ausencia de un lenguaje de modelado y una herramienta CASE pueden influir negativamente en la aceptación internacional de la metodología AAll, sin embargo, el modelado de la inteligencia, del entorno, de las interacciones, y su visión organizacional de los sistemas, pueden hacerla evolucionar y tornarse más atractiva para el desarrollo de aplicaciones basadas en agentes.
- Este trabajo permitirá a la Universidad de las Ciencias Informáticas crear las bases para continuar investigando en la tecnología orientada a agentes. Y a partir de este momento desarrollar sistemas basados en agentes inteligentes.
- Es necesario unificar y estandarizar las diferentes metodologías orientadas a agentes que existen, así como definir un lenguaje de modelado que permita desarrollar todas las abstracciones necesarias para un sistema multiagentes.

- Es necesario desarrollar y estandarizar una herramienta CASE que permita representar toda la simbología de un lenguaje que modele las abstracciones necesarias para un sistema multiagentes.
- La comparación de AAll con el Proceso Unificado de Desarrollo (RUP), más que evaluar la metodología, marcó pautas entre el actual y el nuevo paradigma de ingeniería de software, lo cual contribuye al conocimiento general de esta nueva tendencia y sienta las bases para futuras investigaciones sobre el tema.

En sentido general, resulta muy temprano para hablar de perspectivas de uso y generalización de alguna metodología para el desarrollo de sistemas multiagentes, sin embargo, AAll abre un camino a la investigación en este campo, que a pesar de contar con un futuro inmediato incierto, fomenta las bases para lo que será la construcción de aplicaciones cada vez más eficientes, dinámicas e inteligentes.

Recomendaciones

- Continuar trabajando en la investigación del paradigma orientado a agentes, debido a las posibilidades que este ofrece y a lo novedoso que resulta actualmente.
- Preparar, empleando este material, un curso optativo sobre la tecnología basada en agentes inteligentes, que favorezca el estudio de la misma por estudiantes y profesores de la UCI.
- Incluir en la línea de investigación de inteligencia artificial de la facultad 3 de la UCI, un tema o un proyecto relacionado con la tecnología basada en agentes inteligentes y con las metodologías actuales para el desarrollo de sistemas multiagentes.
- Elaborar una propuesta de estándar metodológico para sistemas multiagentes, lo suficientemente robusto, confiable y eficiente, que unifique el conjunto de metodologías que existen hasta el momento.

Bibliografía Citada

- IVAR JACOBSON, G. B., JAMES RUMBAUGH. El Proceso Unificado de Desarrollo de Software. Madrid: Addison Wesley 2000. 464 p. ISBN 84-7829-036-2.
- MARTIN FOWLER, G. B., KENDALL SCOTT. UML Distilled second edition- A Brief Guide to the Standard Object Modeling Language. Massachusetts: Addison Wesley 2000. 415 p.
- VICENTE JULIAN, J. S. Estudio de métodos de desarrollo de sistemas multiagente. 2003, vol. 18, 65-80 p. Disponible en: <http://lacs.uned.es:8080/aepia/downloadArticulo.do?id=39>
- ESPINO, M. M. Estudio de Metodologías Orientadas a Agentes. Tesina de diplomado, Departamento de Informática Aplicada. Instituto Superior Politécnico José Antonio Echeverría, 2005.
- GARCIA-SERRANO, A., OSSOWSKI, S. Inteligencia Artificial Distribuida y Sistemas Multiagentes. 1998, vol. 2, 12-23 p. Disponible en: <http://sensei.ieec.uned.es/cgi-bin/aepia/contenidoNum.pl?numero=6>
- RUSSELL, S. Inteligencia Artificial: un enfoque moderno. Prentice Hall, 1996, Disponible en: <http://www.agapea.com/Inteligencia-Artificial-n112059i.html>. ISBN 842054003x.
- M. WOOLDRIDGE, N. R. J. Intelligent agents: theory and practice. Knowledge Engineering Review. 1995. vol. 10(2), 115–152 p.
- EURESCOM. Methodology for Engineering Systems of Software AGENTS EURESCOM - P907, Última actualización: 23 de mayo del 2005. [Consultado el: 20 de marzo de 2007]. Disponible en: <http://www.eurescom.de/~public-website/P900-series/P907/index.htm>.
- V. JULIAN, V. B. Agentes Inteligentes: el siguiente paso en la Inteligencia Artificial. 2000, 5 p. Disponible en: <http://www.ati.es/novatica/2000/145/vjulia-145.pdf>.
- GABRIEL HERNAN TOLOSA, F. R. A. B. Revisión: tecnología de agentes de software Brasilia: Instituto Brasileño de Información, Ciencia y Tecnología - IBICT, [Consultado el: 20 de abril de 2007].
- NWANA, H. S. Software Agents: An Overview. 1996, vol. 11(2), 205–244 p. Disponible en: <http://trevinca.ei.uvigo.es/~pcuesta/sm/bibliografia.html>.
- MÁRQUEZ, J. J. S. AGENTES INTELIGENTES ADc JJSM Última actualización: mayo del 2005. [Consultado el: 22 de marzo de 2007]. Disponible en: <http://www.neoyet.com/Trabajos.html>.

- NORES, J. E. G. Agentes Inteligentes. 2001, Disponible en: <http://www.estadistico.com/arts.html?20011105>.
- JAQUERO, V. M. L. INTERFACES DE USUARIO ADAPTATIVAS BASADAS EN MODELOS Y AGENTES SOFTWARE. Tutor: Dr. D. Pascual González López, D. D. A. F. C. Tesis de doctorado, Departamento de Sistemas Informáticos. Universidad de Castilla-La Mancha, 2005.
- ZULMA CATALDI, F. S., FERNANDO LAGE. Sistemas tutoriales multiagentes con modelado del estudiante y del autor. Revista Electrónica de Tecnología Educativa, Enero 2006, vol. 20, nº p. 22.
- ESPINO, M. M. Metodologías Orientadas a Agentes: un estudio comparativo. Tesis de maestría, CENTRO DE ESTUDIOS DE INGENIERÍA DE SISTEMAS. INSTITUTO SUPERIOR POLITÉCNICO JOSÉ ANTONIO ECHEVERRÍA, 2006.
- FERNÁNDEZ, C. Á. I. DEFINICIÓN DE UNA METODOLOGÍA PARA EL DESARROLLO DE SISTEMAS MULTIAGENTE. Tesis Doctoral, Departamento de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid, 1998.
- J.FERBER. Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence. London: Addison Wesley, 1999. 420 p.
- ZAMBONELLI, F., JENNINGS, N. R. Y WOOLDRIDGE,. Developing Multiagent Systems: The Gaia Methodology. ACM Transactions on Software Engineering and Methodology, 2003, vol.12 (3), Disponible en: <http://is.ls.fi.upm.es/udis/docencia/profundizacion/bibliografia.html>.
- SIMÓN, R. V., A. ROSETE, M. MORENO, E. LEYVA, J. PINA, R. SOCORRO. Las metodologías para el desarrollo de Sistemas Multi-Agentes y RUP. En XI Convención Internacional Informática 2005. La Habana, Cuba. 2005-05-09 2005. p. 17.
- MÁRQUEZ, J. J. S. Áreas de Aplicación de los Agentes Inteligentes Wikilearning.com última actualización: 22 de Septiembre del 2005 [Consultado el: 20 de marzo de 2007]. Disponible en: http://www.wikilearning.com/areas_de_aplicacion_de_los_agentes_inteligentes-wkccp-5095-5.html.
- PÉREZ, L. G. EVALUACIÓN DE LA METODOLOGÍA MESSAGE/UML EN UN CASO DE ESTUDIO. Tutor: Ing. Mailyn Moreno Espino, D. A. R. S. Tesis de

grado, Centro de Estudios de Ingeniería y Sistemas. Instituto Superior Politécnico “José Antonio Echeverría”, 2006.

- JORGE GÓMEZ SANZ, J. P. M. Desarrollo de Sistemas Multi-Agente: La metodología INGENIAS. 2004, 21 p. Disponible en: <http://grasia.fdi.ucm.es>.
- LUCA CERNUZZI, I. G., JAIME SÁNCHEZ. Uso de agentes para hiperhistorias con niños ciegos. En 8vo Taller Internacional de Software Educativo. Santiago de Chile. 24,25 y 26 de noviembre 2003. p. 11.
- SARASA, R. B. EVALUACIÓN DE LA METODOLOGÍA GAIA EN UN CASO DE ESTUDIO. Tutor: Ing. Mailyn Moreno Espino, D. A. R. S. Tesis de grado, Centro de Estudios de Ingeniería en Sistemas. Instituto Superior Politécnico José Antonio Echeverría 2006.
- SOFTWARE, D. D. I. D. Introducción a la Ingeniería de Software. En Conferencia I de Ingeniería de Software. UCI. 2004-2005.

Bibliografía Consultada

- ALIS-TECHNET-PROJECT. (2005). "Sitio Web de @lis TechNet Project." Retrieved 8 de abril, 2007, from <http://www.alis-tech.net.org>.
- Arnon Sturm, O. S. (2003). "A Framework for Evaluating Agent-Oriented Methodologies." Agent-Oriented Information Systems AOIS'03.
- B. BAUER, J. O. (2005). "UML 2.0 and Agents: How to Build Agent-based Systems with the new UML Standard." Journal of Engineering Applications of Artificial Intelligence 18(2) 141-157.
- Bernhard Bauer, J. P. M., James Odell (2001). "Agent UML: A Formalism for Specifying Multiagent Interaction." Agent-Oriented Software Engineering: 91-103.
- Crain, A. (2005). "Understanding RUP Roles. The Rational edge." Retrieved 8 de abril, 2007 from <http://www.therationaledge.com>.
- F. Zambonelli, N. J. y. M. W. (2003). Developing Multiagent System: The Gaia Methodology.
- FIPA. (2003). "FIPA ACL Message Structure Specification." Retrieved 15 de enero del 2007, from <http://www.fipa.org>.
- FIPA. (2003). "FIPA Methodology: Glossary of Terms." Retrieved 15 de enero, 2007, from <http://www.fipa.org>.
- Iglesias, C. A. (1998). Definición de una Metodología para el Desarrollo de Sistemas Multiagentes. Departamento de Ingeniería de Sistemas Telemáticos. Madrid, Universidad Politécnica 294.
- M. Moreno, A. R., A. Simón, R. Valdés, E. Leyva, R. Socorro, J. Pina, A. García (2005). Ingeniería de Software Orientada a Agente: roles y metodologías. Ciudad de la Habana, CEIS, CUJAE.

- M. Moreno, A. R., A. Simón, R. Valdés, E. Leyva, R. Socorro, J.Pina, A. García (2006). "Los roles y las metodologías en la Ingeniería de Software Orientada a Agentes." Revista Ingeniería al Día.
- Martínez, G. M. (2005). "Ingeniería de SoftwareUML." Retrieved 10 de marzo, 2007, from <http://www.monografias.com/trabajos5/insof/insof.shtml>.
- Peiró, K. (2000, 20 de marzo del 2006). ""Los agentes inteligentes entenderán el contenido de la web"." Retrieved 12 de marzo, 2007, from <http://karmapeiro.blogspot.com/2006/03/los-agentes-inteligentes-entendern-el.html>
- Sanz, J. J. G. (2003) "Metodologías para el desarrollo de sistemas multi-agente." Volumen, 13 DOI.
- Serrano, C. (2007). "Agentes inteligentes, la Contabilidad en la era del conocimiento." Retrieved 10 de marzo, 2007, from <http://www.5campus.org/leccion/introduc>.
- Stuart Russel, N. (2004). Inteligencia Artificial: Un enfoque moderno. Madrid, Pearson Educación, S. A.
- Vallejo, C. (2005). "¿Qué es UML y que es RUP?" Retrieved 10 de marzo 2007, from <http://www.solutecperu.com/blog/2005/10/qu-es-uml-y-qu-es-rup.html>
- Xavier Ferré Grau, M. I. S. S. (2004). "Desarrollo orientado a objetos con UML." Retrieved 10 de marzo 2007, from <http://www.clikear.com/manuales/uml/>.

Glosario de Términos

AAII: (Australian Artificial Intelligence Institute) Metodología de análisis y diseño para sistemas multiagentes.

Agente inteligente: Sistema informático capaz de tener un comportamiento autónomo y flexible en algunos ambientes.

BDI: (Believe, Desire, Intention) Modelo de arquitectura para sistemas multiagentes, centrado esencialmente en el modelado de los objetivos, creencias y planes del agente.

Herramientas CASE: (Computing Aided Software Engineering) Conjunto de herramientas computacionales que permiten automatizar los artefactos de las diferentes metodologías de ingeniería de software.

Ingeniería de software: La Ingeniería de Software es el establecimiento y uso de principios de ingeniería para obtener un software confiable y que funcione eficientemente en máquinas reales.

Inteligencia Artificial: Tecnología de hardware y software, que da solución a problemas para los cuales no existen algoritmos o los que existen son de complejidad no polinomial, simulación de los procesos lógicos del pensamiento humano.

JADE: (Java Agent Development Framework) Plataforma para el desarrollo de sistemas multiagentes, basada en el lenguaje de programación Java.

OASIS: Sistema multiagentes desarrollado en Australia para el control de tráfico aéreo en el aeropuerto de Sidney.

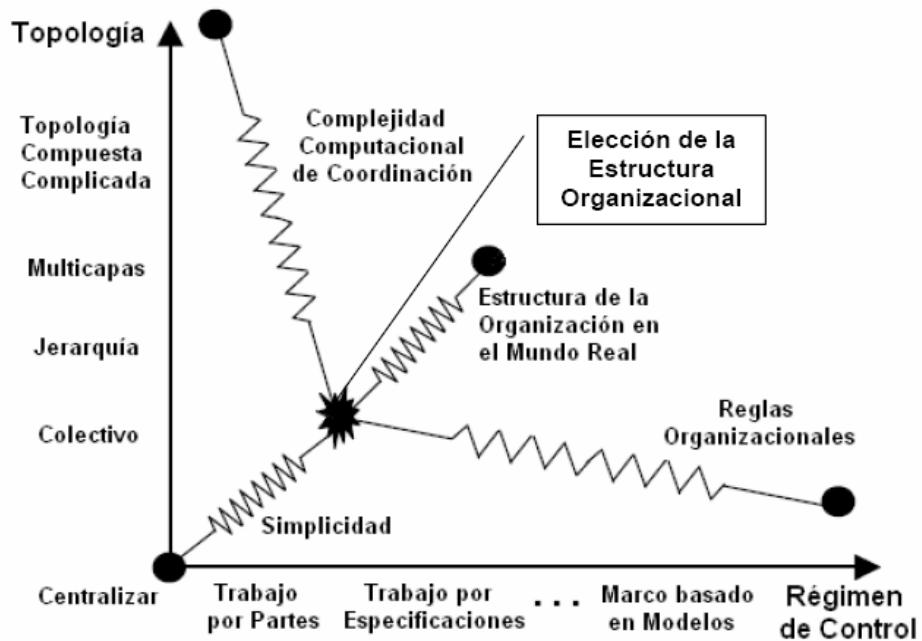
RUP: (Rational Unified Process) Es la metodología de IBM Rational para el desarrollo y construcción de software basado íntegramente en UML como soporte a la metodología.

UML: (Unified Modeling Language) Es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

Visual Paradigm: Herramienta CASE que da soporte al modelado visual con UML 2.0.

Anexos

A.1 Figura 1: Fuerzas involucradas en la identificación de la estructura organizacional de un sistema multiagentes



A.2 Plantilla 1: Esquema de un rol

Nombre del Rol
Nombre del Rol
Descripción
Pequeña descripción del rol
Protocolos y Actividades
Protocolos y actividades de los cuales un rol forma parte
Permisos
Permisos asociados con el rol.
Responsabilidades: Propiedades de existencia
Propiedades de existencia
Responsabilidades: Propiedades de seguridad
Propiedades de seguridad

A.3 Plantilla 2: Esquema de un protocolo de interacción

Nombre del Protocolo
Breve descripción textual que captura la naturaleza de la interacción
Descripción
Descripción textual explicando el propósito del protocolo y las actividades implicadas en su ejecución
Rol iniciador
El rol o los roles responsables de comenzar la interacción
Rol asociado
El rol o los roles que responden a la interacción iniciada
Entradas
Información usada por el rol iniciador mientras establece el protocolo
Salidas
Información suministrada por el asociado durante la interacción

A.4: Roles preliminares del sistema “tour operador de viajes”

A.4.1: Esquemas de roles preliminares de la suborganización “gestión personal”

Nombre del Rol
Autenticar
Descripción
Este rol busca a un usuario habitual en la base de datos del sistema y carga sus preferencias.
Protocolos y Actividades
Autenticar_ Usuario Cargar_ Preferencias
Permisos
Leer preferencias
Responsabilidades: Propiedades de existencia
1. Autenticar = Autenticar_ Usuario. Cargar_ Preferencias
Responsabilidades: Propiedades de seguridad

Nombre del Rol
Buscar_ Itinerario
Descripción
Este rol solicita un conjunto de posibles itinerarios de viajes atendiendo a las preferencias de un usuario, y escoge de ellos el itinerario más a fin con las mismas. Si el usuario realiza modificaciones a un itinerario de viajes propuesto, se escoge del conjunto de posibles itinerarios, alguno que cumpla con las nuevas preferencias, de no existir ninguno, se solicita un nuevo conjunto de posibles itinerarios de viajes y se escoge el que más se corresponda con las nuevas preferencias del usuario.
Protocolos y Actividades
Solicitar_ Itinerarios Escoger_ Itinerario
Permisos
Leer preferencias
Responsabilidades: Propiedades de existencia
1. Buscar_ Itinerario = Escoger_ Itinerario Solicitar_ Itinerarios
Responsabilidades: Propiedades de seguridad

A.4.2: Esquemas de roles preliminares de la suborganización “gestión local”

Nombre del Rol
Alojamiento
Descripción
Este rol solicita posibles alojamientos en el destino y una vez aceptado por el usuario un itinerario propuesto, solicita confirmación del alojamiento en el destino
Protocolos y Actividades
Solicitar_ Alojamientos Confirmar_ Alojamientos
Permisos
Leer alojamientos Consumir alojamientos
Responsabilidades: Propiedades de existencia
1. Alojamiento = Solicitar_ Alojamientos 2. Alojamiento = Confirmar_ Alojamientos
Responsabilidades: Propiedades de seguridad

Nombre del Rol
Transporte
Descripción
Este rol solicita posibles transportes en el destino y una vez aceptado por el usuario un itinerario propuesto, solicita confirmación del transporte en el destino
Protocolos y Actividades
Solicitar_ Transportes Confirmar_ Transportes
Permisos
Leer transportes Consumir transportes
Responsabilidades: Propiedades de existencia
1. Transporte = Solicitar_ Transportes 2. Transporte = Confirmar_ Transportes
Responsabilidades: Propiedades de seguridad

A.5: Interacciones preliminares del sistema “tour operador de viajes”

A.5.1: Esquemas de protocolos de interacción preliminares de la suborganización “gestión personal”

Nombre del Protocolo
Confirmar_ Itinerario
Descripción
Este protocolo confirma la aceptación por el usuario del itinerario propuesto.
Rol iniciador
Interfaz Ratificar_ Itinerario
Rol asociado
Ratificar_ Itinerario Itinerario_ Global
Entradas
Itinerario de viajes aceptado por el usuario
Salidas
Confirmación de que se ratificó el itinerario de viajes aceptado por el usuario.

Nombre del Protocolo
Solicitar_ Itinerarios
Descripción
Este protocolo solicita un conjunto de posibles itinerarios de viajes, señalando los destinos, tiempo, presupuesto y preferencias del usuario.
Rol iniciador
Buscar_ Itinerario
Rol asociado
Itinerario_ Global
Entradas
Destinos, tiempo, presupuesto y preferencias.
Salidas
Conjunto de posibles itinerarios de viajes, atendiendo a las características descritas por el usuario.

Nombre del Protocolo
Guardar_ Preferencias
Descripción
Este protocolo guarda en la base de datos del sistema las preferencias del usuario, luego de aceptar un itinerario de viajes propuesto.
Rol iniciador
Ratificar_ Itinerario
Rol asociado
(base de datos del sistema)
Entradas
Nombre de usuario, contraseña y nuevas preferencias.
Salidas
Confirmación de que se actualizaron las preferencias del usuario en la base de datos del sistema.

A.5.2: Esquemas de protocolos de interacción preliminares de la suborganización “gestión global”

Nombre del Protocolo
Confirmar_ Itinerario_ Destino
Descripción
Este protocolo solicita la confirmación del itinerario local aceptado por el usuario.
Rol iniciador
Itinerario_ Global
Rol asociado
Itinerario_ Local
Entradas
Actividades, alojamiento y transporte en el destino, tiempo de estancia, presupuesto.
Salidas
Confirmación de que se ratificó el itinerario de viajes local aceptado por el usuario.

Nombre del Protocolo
Confirmar_ Vuelos
Descripción
Este protocolo solicita la confirmación de los vuelos aceptados por el usuario.
Rol iniciador
Itinerario_ Global Vuelos
Rol asociado
Vuelos (base de datos del entorno)
Entradas
Aerolíneas, números de las aviones, días de los vuelos entre destinos, presupuesto, preferencias para cada vuelo.
Salidas
Confirmación de que se ratificaron los vuelos entre destinos aceptados por el usuario.

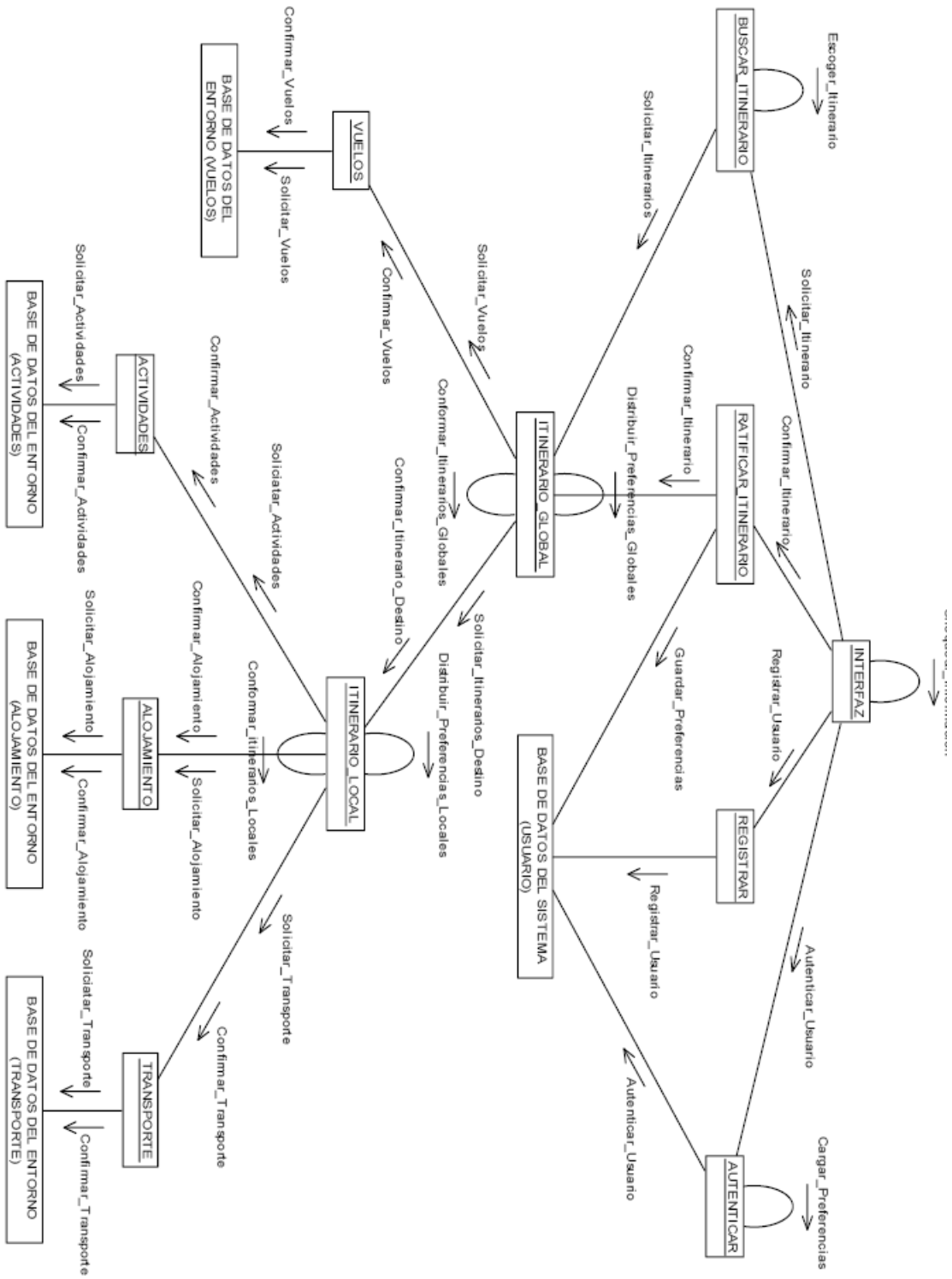
A.5.3: Esquemas de protocolos de interacción preliminares de la suborganización “gestión local”

Nombre del Protocolo
Solicitar_ Transporte
Descripción
Este protocolo solicita un conjunto de posibles transportes en el destino señalado, apuntando los días asignados para ello, el presupuesto y las preferencias descritas por el usuario.
Rol iniciador
Itinerario_Global Transporte
Rol asociado
Transporte (base de datos del entorno)
Entradas
Días para los transportes, presupuesto y preferencias descritas por el usuario.
Salidas
Conjunto de posibles transportes en el destino atendiendo a las características expresadas.

Nombre del Protocolo
Confirmar_Actividades
Descripción
Este protocolo solicita la confirmación de las actividades aceptadas por el usuario para el destino.
Rol iniciador
Itinerario_Local Actividades
Rol asociado
Actividades (base de datos del entorno)
Entradas
Días de las actividades, presupuesto y preferencias descritas por el usuario.
Salidas
Confirmación de que se ratificaron las actividades aceptadas por el usuario.

Nombre del Protocolo
Confirmar_Alojamiento
Descripción
Este protocolo solicita la confirmación de los alojamientos aceptados por el usuario para el destino.
Rol iniciador
Itinerario_Local Alojamiento
Rol asociado
Alojamiento (base de datos del entorno)
Entradas
Días de alojamientos, presupuesto y preferencias descritas por el usuario.
Salidas
Confirmación de que se ratificaron los alojamientos aceptados por el usuario.

A.6: Estructura organizacional del sistema “tour operador de viajes”



A.7: Framework de evaluación de la metodología GAIA

Metodología: AAll

Revisado por: Leanedy de la Caridad Tur Villazán

Fecha: junio 2007

1. Proceso de desarrollo

- **Dominio de aplicación:** Sistemas multiagentes o sistemas basados en tecnología orientada a agentes.

Evaluación:

- Orientado a dominio
 Independiente del dominio
 No se sabe

- **Áreas de aplicación:**

Comentario: Puede ser aplicada a cualquier área, aunque no se menciona ninguna en específico por los autores.

Áreas concretas:

_____	<input type="checkbox"/> Se menciona	<input type="checkbox"/> Hay documentación
_____	<input type="checkbox"/> Se menciona	<input type="checkbox"/> Hay documentación
_____	<input type="checkbox"/> Se menciona	<input type="checkbox"/> Hay documentación
_____	<input type="checkbox"/> Se menciona	<input type="checkbox"/> Hay documentación

- **Sistemas abiertos:**

Comentario: modela sistemas multiagentes de entornos abiertos y distribuidos.

Evaluación: Sí No No se sabe

- **Tipo de ciclo de vida:**

Comentario: No define fases, todas sus etapas se suceden en una sola fase no identificada. Tampoco hace iteraciones

Nombre del tipo de ciclo: secuencial

- **Fase:**

Nombre: no identificada.

Breve explicación: Los autores no definen explícitamente fases. Se definen las etapas de análisis y diseño, y en esta última se desarrollan el diseño de las vistas externa e interna.

Requisitos: Centrado Centrado parcialmente No centrado
 Análisis: Centrado Centrado parcialmente No centrado
 Diseño de la vista externa: Centrado Centrado parcialmente No centrado
 Diseño vista interna: Centrado Centrado parcialmente No centrado
 Implementación: Centrado Centrado parcialmente No centrado
 Pruebas: Centrado Centrado parcialmente No centrado
 Despliegue1: Centrado Centrado parcialmente No centrado
 Mantenimiento1: Centrado Centrado parcialmente No centrado
 Implicación del usuario: Centrado Centrado parcialmente No centrado

- **Herramientas de soportes:**

Nombre de las herramientas: No presenta herramienta.

Requisitos: Sí No Soportada parcialmente No se sabe
 Análisis: Sí No Soportada parcialmente No se sabe
 Diseño: Sí No Soportada parcialmente No se sabe
 Implementación:
 -Generación del código: Sí No Soportada parcialmente No se sabe
 -Orientada a una herramienta de desarrollo concreta Sí No Soportada parcialmente No se sabe
 Pruebas: Sí No Soportada parcialmente No se sabe
 Despliegue1: Sí No Soportada parcialmente No se sabe

2. Vista de modelos

- **Conceptos y representación:**

La metodología AAll implementa varias abstracciones básicas en sus artefactos que caracterizan una organización y que son apropiadamente explotadas durante sus etapas, estas son descritas a continuación:

Artefacto	Descripción	Etapas	Concepto
División de la Organización	Descompone la organización en suborganizaciones.	Análisis	Organización
Modelo del entorno	Representación computacional del entorno donde será ubicado el sistema.	Análisis	Entorno
Modelo de roles preliminares	Contiene los roles preliminares que son identificados, pero no necesariamente definidos.	Análisis	Roles

Modelo de protocolos de interacción preliminares	Contiene las interacciones básicas requeridas para garantizar las funcionalidades del sistema.	Análisis	Interacciones y protocolos
Modelo de clases de agentes	Identifica las distintas clases de agentes del sistema, y las relaciones entre ellas.	Diseño de la vista externa	Agentes Vista externa
Modelo de instancias de agentes	Identifica las instancias de cada clase de agente del sistema.	Diseño de la vista externa	Agentes Instancias
Modelo de interacciones	Define las responsabilidades y servicios de cada clase de agente.	Diseño de la vista externa	Agentes Interacciones
Modelo de objetivos	Define los objetivos (deseos) de cada agente.	Diseño de la vista interna	Deseos de agentes
Modelo de creencias	Define la composición interna de cada agente.	Diseño de la vista interna	Creencias de Agentes
Modelo de planes	Define los planes (intenciones) de cada agente, para lograr sus objetivos.	Diseño de la vista interna	Intenciones de agentes

- **Interacción con el usuario: ¿Se modela el usuario de algún modo? (especialmente si el usuario es modelado de forma diferente que los agentes)**

Comentario: Si se modela el usuario, este es un aspecto que en AAIL, aparece bien explícito.

- **Relaciones entre modelos:**

Salto entre los modelos relacionales: Se suministra un conjunto de modelos durante el transcurso de sus etapas que aunque no siguen una forma rígida de aparición, si presentan trazas entre ellos.

Tipo de relación: pasos o trazas.

- **Resultados:**

La metodología define o no los resultados a producir: Se definen claramente los resultados a obtener en cada etapa.

Nombre de los resultados a entregar:

Etapa de Análisis:

- División de la Organización.
- Modelo del Entorno
- Modelo de roles preliminares.
- Modelo de protocolos de interacción preliminares.
- Definición de la estructura organizacional

Etapa de Diseño de la Vista externa

- Definir los roles y protocolos de interacción.
- Modelo de Agentes.
- Modelo de Clases de agentes.
- Modelo de instancias de agentes
- Modelo de Interacciones.

Etapa de Diseño de la Vista interna:

- Modelo de Objetivos
- Modelo de Creencias
- Modelo de Planes

3. Agentes

- **Conceptos:**

Definición:

Restricciones en la arquitectura:

Atributos del agente:

Autonomía: Sí No No se sabe
Sociabilidad: Sí No No se sabe
Reactividad: Sí No No se sabe
Proactividad: Sí No No se sabe
Inteligencia: Sí No No se sabe
Veracidad: Sí No No se sabe
Benevolencia Sí No No se sabe
Racionalidad: Sí No No se sabe
Cooperación: Sí No No se sabe

- **Tipos de comunicación:**

Comentario: Se modela cualquier tipo de comunicación.

Tipos: A-A A-H A-O Cualquiera

- **Protocolos de comunicación:**

Comentario: No define protocolos de comunicación, más bien suministra una guía para identificarlos durante el transcurso de sus etapas de análisis y diseño, sugiriendo que para implementarlos se empleen estándares ya desarrollados de protocolos de comunicación, con las especificaciones establecidas en sus etapas.

Protocolos:

- Predefinido
- Definido por el usuario
- Definido por metodología
- No se sabe

- **Cooperación:**

Comentario: Al establecer una estructura organizacional, se dota a los roles que desempeñarán sus agentes de cierto nivel jerárquico que los ubica en una determinada posición dentro del sistema. En este sentido, las interacciones entre los agentes que se encuentran en un mismo nivel de la jerarquía pueden ser de negociación de información (o servicios), mientras que los que se encuentran en niveles superiores pueden delegar responsabilidades o funciones a los de niveles más inferiores.

Cooperación: Cualquiera No se sabe

- **Organización:**

Comentario: Centra su desarrollo sobre un eficiente modelado de la organización, en este sentido, su visión gira en torno al establecimiento de una estructura organizacional que sustente las aplicaciones lo más eficientemente posible.

Organización:

- Predefinida
- Definida por el usuario
- Definida por metodología
- No se sabe

4. Modelado de características adicionales

- **Ontologías:**

Comentario: No hace mención.

Evaluación: Sí No No se considera

- **Movilidad:**

Comentario: No hace mención.

Evaluación: Sí No No se considera

- **Otras:**

Comentario: no hace mención.

Evaluación: Sí No No se considera

5. Documentación

- **Documentación disponible:**

Explicación sobre el tipo de documentos: La documentación disponible puede encontrarse de forma libre en Internet, preferiblemente referenciada por sus autores, aunque existes varios trabajos que la tratan.

Evaluación: __Buena Suficiente __Pobre

- **Casos de estudios presentados:**

Comentario: Los casos de estudio realizados pueden considerarse como parciales, pues aunque cubren todos los flujos de trabajo de la metodología, no son lo suficientemente detallados ni se encuentra realizada completamente la documentación de ningún sistema.

Evaluación: __Trivial __Parcial Completo

6. Aspectos adicionales a evaluar

- **Proceso de desarrollo**

Plataforma de desarrollo:

Nombre: Independiente de la plataforma.

Evaluación: Sí __No __No se sabe

Facilidad de uso de la Herramienta CASE:

Comentario: No presenta herramienta CASE, aunque es posible emplear cualquiera que soporte UML.

Evaluación: __Bueno Regular __Mal

- **Vista de modelos**

Lenguaje de modelado:

Comentario: No asume ninguna técnica de modelado.

Lenguaje: Se puede emplear UML para el diseño de los modelos de clases e instancias de agentes.

Facilidad de uso del lenguaje de modelado:

Comentario: No asume ninguna técnica de modelado.

Evaluación: __Bueno Regular __Mal

Entendimiento del lenguaje de modelado:

Comentario: No asume ninguna técnica de modelado.

Evaluación: Bueno Regular Mal

Ajuste del lenguaje de modelado a la metodología:

Comentario: No asume ninguna técnica de modelado.

Evaluación: Bueno Regular Mal

- **Agentes**

Modelado de la inteligencia o conocimiento del agente:

Comentario: Utiliza la inteligencia para modelar el comportamiento de los agentes.

Evaluación: Si No Parcialmente No se sabe

Tiene pasos claros el modelado de la inteligencia:

Comentario:

Evaluación: Si No Parcialmente No se sabe

Facilidad del modelado de la interacción:

Comentario: Modelas las interacciones de manera ágil y sencilla, pero a su vez completa a través de un artefacto específico para ello

Evaluación: Bueno Regular Mal

- **Modelado de características adicionales**

Modelado del ambiente o entorno:

Comentario: Constituye una de las primeras abstracciones.

Evaluación: Si No Parcialmente No se sabe

Modelado de la organización:

Comentario: Constituye primera abstracción.

Evaluación: Si No Parcialmente No se sabe

- **Documentación**

Aplicaciones que han utilizado la metodología:

Comentario: Ha sido reportado el empleo de la metodología en al menos dos sistemas multiagentes.