



Universidad de Cienfuegos “Carlos Rafael Rodríguez”
Facultad de Informática

**Metodología de Pruebas integrada a PASSI para Sistemas
Multiagentes.**

Trabajo de Diploma para optar por el título de Ingeniería en Informática.

Autor(es):

José Ernesto Saura Fernández.

Tutor(es):

**MsC. Karina Leonor Fernández Sánchez. Dirección Territorial ETECSA
Cienfuegos.**

**MsC. Viviana Toledo Rivero. Universidad de Cienfuegos “Carlos Rafael
Rodríguez.”**

Cienfuegos Cuba

2011

“Año 53 de la Revolución.”

*DECLARACIÓN
DE
AUTORÍA*

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del presente trabajo de diploma que fue realizado en la Universidad de Cienfuegos “Carlos Rafael Rodríguez” y en colaboración con la Dirección Territorial ETECSA Cienfuegos, como parte de la culminación de estudios de la especialidad de Ingeniería Informática, autorizando a que el mismo sea utilizado por ambas instituciones para los fines que estime conveniente, tanto de forma parcial como total.

José Ernesto Saura Fernández.

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática desarrollada.

Para que así conste firmamos la presente a los días del mes de de.

MsC. Karina Leonor Fernández Sánchez.

MsC. Viviana Toledo Rivero.

Responsable de Información
Científico Técnica.

Vicedecana de Facultad.

PENSAMIENTO

PENSAMIENTO

*“ Quién tiene el derecho de criticar debe tener
el corazón para ayudar.”*

Abraham Lincoln.

*OPINIÓN
DEL
TUTOR*

OPINIÓN DEL TUTOR

DEDICATORIA

DEDICATORIA

A mi abuela Cristina, por su constante apoyo, amor y preocupación a lo largo de toda mi carrera, con todo el amor del mundo.

A mis padres por su amor y dedicación.

A mis hermanos, con el gran deseo de que algún día me dediquen también sus trabajos de diploma.

A Rocío, por su amor, su permanente apoyo y su persistencia para el logro de este trabajo.

En especial, A toda mi familia.

AGRADECIMIENTOS

AGRADECIMIENTOS

A mi mamá y mi papá por todo su apoyo y dedicación durante toda la carrera.

A mis hermanos, por quererme tanto y por su preocupación.

A Migue, por estar siempre presente y por toda su dedicación.

A mis abuelos por estar siempre presente y poder contar con su amor.

A mi novia Rocío, por permitirme todo este tiempo a su lado.

A mis tías, tíos y primos, por estar ahí cuando los necesité, en especial a Katy.

A Karina y Viviana, por brindarme sus conocimientos y porque sin ellas no hubiese sido posible la realización de este trabajo.

A Lien, Yasel, Sandy, Antuanet, Liset, y al resto de mis amigos que siempre estuvieron dispuestos a apoyarme.

A la familia de Rocío, por tenerme siempre presente y apoyarme en todo momento.

A Betsy y toda su familia, por todo el apoyo y la comprensión.

A Sarahy, por poder contar siempre con su sincera amistad.

A Carlitos y el Guaji, por soportarme durante todos estos años.

A todos mis profesores de la carrera, por su ayuda, dedicación y amor.

A mis compañeros de aula, con quienes compartí estos 5 años, los quiero mucho.

A toda mi familia.

A todos los que de una forma u otra, se sientan parte del equipo de realización de este trabajo.

RESUMEN

RESUMEN

Los Sistemas Multiagentes son una de las áreas de la Inteligencia Artificial, caracterizada por ofrecer soluciones a problemas complejos. Existen disímiles metodologías de desarrollo aunque hasta nuestros días ninguna complete el ciclo de vida del software. La metodología PASSI es una de las más desarrolladas en este aspecto, pero adolece de una guía para la ejecución de la fase de pruebas.

En este trabajo inicialmente se describe cual ha sido el comportamiento y la evolución de las metodologías de desarrollo de Sistemas Multiagentes, haciendo esencial énfasis en sus potencialidades para la modelación de sistemas de gran calidad, al adolecer en algunos aspectos. Se describe además una metodología de pruebas con el fin de integrarla a PASSI para el desarrollo de estos sistemas, realizando una integración de cinco nuevas fases, con las fases ya conocidas de PASSI. Este proceso trae como resultado una metodología de pruebas integrada a PASSI para el desarrollo de Sistemas Multiagentes. Por último, se hace un proceso de validación, con los resultados esperados, al ser la metodología presentada aquí, más completa y con un mayor nivel de detalle en sus procesos. Se concluye que su futura aplicación será de gran utilidad en la modelación de sistemas, al contener todas las fases de la ingeniería de software.

*TABLA
DE
CONTENIDOS*

Tabla de Contenidos

INTRODUCCIÓN	2
CAPÍTULO I: "Estado del arte de las metodologías orientadas al desarrollo de Sistemas Multiagentes."	
.....	11
1.1 Introducción.	11
1.2 Metodologías de desarrollo de sistemas orientadas a agentes.	11
1.2.1 MAS-CommonKADS.....	11
1.2.2 INGENIAS.....	14
1.2.3 MASE (Multi-agent systems Software Engineering).	17
1.2.4 PASSI (Process for Agent Societies Specification and Implementation).....	19
1.3 Comparación de las Metodologías.	21
1.4 Conclusiones del Capítulo.	28
CAPÍTULO II: "Diseño de la metodología de pruebas integrada a PASSI."	30
2.1 Introducción.	30
2.2 Fundamentos de la metodología de pruebas para SMA.	30
2.3 Integración de la Metodología de Pruebas con la Metodología PASSI.	32
2.4 Presentación de la Metodología de Pruebas.	34
Etapa 1(E1): Planeación de las Pruebas.	34
Etapa 2(E2): Validación de Objetivos, Tareas y Roles.	36
Etapa 2 / Sub-Fase 2.1 (E2 (SF2.1)): Validación de Objetivos, Agentes y Tareas.....	36
Etapa 2 / Sub-Fase 2.2 (E2 (SF 2.2)) Validación y Verificación de Roles.	38
Etapa 3 (E3): Validación de Vínculos y Comportamiento del Agente.	39
Etapa 4 (E4): Validación y Verificación de Agentes Finales.	42
2.5 Procesos de Control.	44
2.6 MEDSIM- PASSI: Metodología resultante de la integración de las fases ya conocidas de PASSI, con la etapa de pruebas anteriormente descrita.	45
Descripción de la Metodología " MEDSIM-PASSI ".....	47
2.7 Conclusiones del Capítulo.	61

METODOLOGÍA DE PRUEBAS INTEGRADA A PASSI PARA SISTEMAS MULTIAGENTES

CAPÍTULO III: "Validación de la Metodología MEDSIM-PASSI".	63
3.1 Introducción.	63
3.2 Validación de la metodología "MEDSIM-PASSI" fase por fase.	63
3.2.1 Planeación de las Pruebas.	63
3.2.2 Modelo de Requerimientos del Sistema.	64
3.2.3 Validación de Objetivos, Agentes y Tareas.	65
3.2.4 Modelo de Sociedades del Agente.....	69
3.2.5 Validación y Verificación de Roles.	70
3.2.6 Modelo de Implementación del Agente.	71
3.2.7 Validación de Vínculos y Comportamiento del Agente.	72
3.2.8 Modelo de Código.	73
3.2.9 Validación y Verificación de Agentes Finales.	74
3.2.10 Modelo de Despliegue.	76
3.3 Ventajas y desventajas de la aplicación de la metodología MEDSIM-PASSI.	77
3.4 Conclusiones del Capítulo.	77
CONCLUSIONES	80
RECOMENDACIONES	82
BIBLIOGRAFÍA	84

*ÍNDICE
DE
TABLAS*

ÍNDICE DE TABLAS

Tabla 1. Comparación en cuanto a aspectos relacionados con el proceso.....	22
Tabla 2. Aspectos que toma en cuenta el modelado.	23
Tabla 3. Comparación en cuanto a los conceptos considerados en los Modelos.....	24
Tabla 4. Supuestos y pilares de Metodología de Pruebas de SMA.....	31

*ÍNDICE
DE
FIGURAS*

ÍNDICE DE FIGURAS

Figura 1. Modelos de MAS-CommonKADS para la definición del SMA.....	12
Figura 2. Conjunto de Modelos que tratan con el SMA para Ingenias.	15
Figura 3. Modelos y fases en la metodología PASSI.	21
Figura 4. Fases de la Metodología "MEDSIM-PASSI".	46
Figura 5. Interfaz de Datos Generales de la Paciente.....	66
Figura 6. Interfaz de Síntomas y Signos.....	67
Figura 7. Interfaz de Antecedentes Obstétricos.	68
Figura 8. Interfaz de Ayuda.	69
Figura 9. Diagrama de Descripción de Roles.....	71

INTRODUCCIÓN

INTRODUCCIÓN

Las Tecnologías de la Informática y las Comunicaciones (TIC) se hacen cada día más importantes en diversas áreas del quehacer humano, debido a su rapidez para analizar, procesar y comunicar grandes cantidades de información. Su avance a lo largo de la historia ha logrado importantes transformaciones sociales, económicas y culturales en la humanidad, a tal punto que la informatización de la sociedad, a escala mundial, es algo que ya se ha hecho imprescindible y con lo que obligatoriamente vivimos.

Hacia la década de los sesenta este impetuoso desarrollo de las TIC experimenta el importante surgimiento de la Inteligencia Artificial (IA), acuñado inicialmente por John McCarthy para definir los métodos algorítmicos capaces de hacer pensar a los ordenadores. Su principal objetivo es hacer las computadoras más inteligentes creando software que imiten las funciones del cerebro en diferentes áreas de aplicación, mostrando un comportamiento similar al humano.¹

Hacia mediados de los sesenta la IA se convierte en un área en la que se interesan e interactúan especialistas de diversas disciplinas: ingenieros, psicólogos, matemáticos, lingüistas, médicos, entre otros.

A principios de los años ochenta la IA empieza a aplicarse en la solución de problemas reales con la aparición de los sistemas basados en el conocimiento; complejos programas de computador en los que se codifica el conocimiento de expertos en una materia muy concreta, los cuales se declaran como una línea de investigación prioritaria dentro de la Informática. Este conocimiento puede ser representado de diversas maneras, siendo las más difundidas mediante reglas de producción y conocimiento basado en casos.

¹ Díaz Gómez, "Sistema Multiagente para MPEG."

INTRODUCCIÓN

Recientemente se han desarrollado variadas técnicas de la Inteligencia Artificial, las cuales han probado que se caracterizan por ser soportes de ayuda poderosos para tomar decisiones en diversas actividades humanas. Estos soportes deben ser capaces de considerar la información cualitativa y a partir de ella diseñar e implementar modelos estadísticos y computacionales que asistan a los decisores en la resolución de los diversos problemas a ser resueltos.

En la actualidad, la IA se está aplicando a numerosas acciones realizadas por los seres humanos y se destacan entre otras, líneas de investigación científicas como la robótica, la visión artificial, técnicas de aprendizaje y la gestión del conocimiento. Estas dos últimas aplicaciones son dirigidas fundamentalmente al campo de las ciencias médicas, debido a que existe una fuerte motivación orientada a la construcción de sistemas de información que incorporen conocimiento, y que permitan a los médicos tomar decisiones eficientes y oportunas en el ámbito de la gestión de la información médica.

Resulta fundamental la integración de diversas técnicas surgidas dentro de la IA para la representación y uso del conocimiento, como son las redes neuronales, la lógica difusa, algoritmos genéticos, los agentes inteligentes en la programación distribuida, sistemas expertos, entre otros.

El proceso hasta lograr el desarrollo de una inteligencia artificial de nivel comparable a la de un ser humano es un largo camino que se prolonga durante muchos años.

La fase de pruebas ha sido una etapa particular en todo proceso de Software empezando por su enfoque que es totalmente opuesto al de las demás etapas del Ciclo de Vida. Mientras que fases como el análisis, el diseño y la implementación se enfocan en la construcción del sistema, la fase de pruebas se enfoca en la “destrucción” del mismo, mediante la detección de la mayor cantidad posible de defectos dentro del sistema, con el fin de garantizar el

INTRODUCCIÓN

cumplimiento de los requerimientos Funcionales y No Funcionales definidos al inicio del proyecto y por tanto contribuir a la generación de un Software de calidad.²

Las pruebas en sí son un proceso, en el cual se busca tener la mayor cantidad de pruebas para así detectar la mayor cantidad de defectos, así mismo estas pruebas no necesariamente deben ejecutarse en una sola fase. El concepto fase se refiere a un intervalo de tiempo en el cual una actividad se realiza con mayor frecuencia que otras sin que esta actividad tenga por qué destinarse solo en este intervalo de tiempo. Visto concretamente, *el hecho de que se tenga una fase de pruebas no quiere decir que las pruebas deban ejecutarse solo en dicho período de tiempo*. Se considera como otro de los supuestos principales del enfoque de pruebas, que las pruebas más que una fase son un proceso iterativo paralelo al ciclo de vida, en el cual estas deben ser de naturaleza variada para abarcar diferentes artefactos generados en el proceso de desarrollo, así mismo estas pruebas deben ejecutarse tan pronto como los defectos puedan aparecer con el fin de corregirlos con mayor rapidez, y disminuir así el 50% del presupuesto de todo proyecto de Software, que estadísticamente siempre se invierte en el crítico período de pruebas.

El enfoque de pruebas como proceso iterativo es con seguridad el enfoque más adecuado para sistemas en los que la construcción y refinación sea también un proceso iterativo y en los que esta construcción relacione elementos críticos y variados. Los Sistemas Multiagentes (SMA) son el ejemplo perfecto de estos sistemas. En los Agentes la construcción parte de una definición de requerimientos a partir de los cuales se definen las metas/objetivos a cumplir a los cuales se asocian las habilidades/capacidades, recursos y tareas necesarios para su cumplimiento, posteriormente estos objetivos deben descomponerse, clasificarse y asociarse a entidades que tendrán una arquitectura interna definida, así mismo estas entidades deberán cooperar y relacionarse de algún modo con el fin de cumplir los objetivos

² Díaz Gómez, "Sistema Multiagente para DPEG."

INTRODUCCIÓN

generales del sistema evitando al máximo la presencia de conflictos y haciendo un uso eficiente de los canales y protocolos de comunicación definidos para el SMA.³

Como se ha mencionado este proceso es iterativo y el sistema se va construyendo por etapas donde cada etapa relaciona conceptos específicos propios del paradigma de Agentes. Ante este hecho se propone una metodología de pruebas que se ejecute paralelamente a este proceso y que contemple diferentes tipos de pruebas asociados a la verificación y validación de todos los elementos ya descritos. Un proceso de este tipo garantiza la ejecución temprana de pruebas, y la detección y corrección temprana de defectos, así mismo se contará con puntos de verificación más frecuentes los cuales permitirán llegar a un proceso de pruebas libre de traumatismos, en el que la corrección del sistema se irá evaluando por iteraciones y en puntos de fallo que garanticen mayor rapidez, y disminuyan el rango de posibilidades en las que el error se localice por la cercanía entre puntos de verificación. Así mismo se garantiza la trazabilidad y consistencia que debe tener este proceso de pruebas.

En la realización de Software de calidad donde se hace necesario la utilización de una metodología de desarrollo de Agentes, es indispensable contar con pasos correctamente definidos para cada una de las etapas.

PASSI (Process for Agent Societies Specification and Implementation), que en español significa Proceso para la Especificación de Sociedades del Agente y la Implementación, es una metodología de desarrollo de sistemas multiagentes, donde podemos identificar varias fases con las cuales se asegura llegar de la especificación del sistema a una versión implementable.

Dentro de la etapa de análisis, tenemos las siguientes fases: descripción del dominio: utilizando los diagramas de casos de uso de UML se describe el dominio de la aplicación,

³ Sistemas Multi Agentes.

INTRODUCCIÓN

identificación de agentes: a partir del diagrama anterior y utilizando Rational Rose identificamos los agentes que componen el sistema, identificación de roles: los roles de los agentes los representamos mediante diagramas de secuencia, e identificación de tareas: se dibuja un diagrama de actividad para cada agente y se deciden que tareas son necesarias para realizar las funcionalidades descritas anteriormente.

En la segunda etapa, la etapa de diseño, podemos identificar otras fases entre las cuales se encuentran: descripción de ontologías: se describe la sociedad desde el punto de vista ontológico, descripción de roles: se modela la vida de los agentes observando sus roles, descripción de protocolos: se define cuales protocolos se utilizarán y si es necesario, se definirán los nuevos, definición de la estructura y del comportamiento de los agentes: se diseña la estructura de la sociedad y de los agentes en particular.

Finalmente tenemos las fases de implementación y configuración. La fase de producción de código no está completamente implementada, pero está en desarrollo la búsqueda de una solución en un metalenguaje basado en XML. En cuanto a la configuración, se acentúa su importancia si se trabaja con agentes móviles y con problemas significativos en la diseminación de los agentes en el sistema.⁴

En los SMA integrados a PASSI no se ha definido hasta el momento un diseño metodológico para la realización de pasos, en la difícil etapa de pruebas. Se han realizado aplicaciones de este tipo a las que no se ha podido realizar pruebas.

Por todo esto se plantea el siguiente **Problema a resolver**:

Ausencia de una Metodología de Pruebas integrada a PASSI, para el desarrollo de Sistemas Multiagentes.

⁴ Marchetti Tulio José, García Alejandro Javier, "Metodologías de desarrollo de sistemas multiagentes: un análisis comparativo."

INTRODUCCIÓN

El **Objeto de Estudio**:

Los Sistemas Multiagentes basados en el conocimiento.

El **Campo de Acción**:

Etapas de pruebas en los Sistemas Multiagentes integrados a PASSI.

Como **Hipótesis** se plantea que:

Si se diseña una metodología para realizar pruebas a un sistema multiagente, **entonces** se logra el completamiento del ciclo de vida del software, contribuyendo a la modelación de un sistema estándar.

Como **Objetivo General** se planteó:

Diseñar una metodología de prueba integrada a PASSI para Sistemas Multiagentes.

Del mismo se derivan los siguientes **Objetivos Específicos**:

- Estudiar el estado del arte de las metodologías de desarrollo de SMA, en especial de PASSI.
- Diseñar la metodología de pruebas.
- Aplicar la metodología diseñada a Diagnóstico Presuntivo de Enfermedades Ginecológicas (DPEG).

Para lograr esta investigación se desarrollarán las siguientes **Tareas Científicas**:

1. Estudio del estado del arte de las Metodologías de Pruebas de Sistemas Multiagentes.

INTRODUCCIÓN

2. Estudio del proceso de prueba en el ciclo de vida del software.
3. Recopilación del conocimiento y sugerencias a través de entrevistas, encuestas y la literatura científico-técnica sobre el dominio que se aborda.
4. Revisión de las principales metodologías de la ingeniería del software y selección de una apropiada.
5. Aplicación de la metodología propuesta en el diseño de la estructura interna del sistema.
6. Validación de los resultados obtenidos.

Se aplicarán los siguientes **Métodos Teóricos**:

1. Inducción – deducción, con el objetivo de estructurar el conocimiento científico a partir de la revisión bibliográfica.
2. Histórico – lógico, para conocer el problema estudiado en su origen y desarrollo; desde el punto de vista de la informática.
3. Análisis y síntesis, para poder establecer nexos, comparar resultados, determinar enfoques comunes y aspectos distintivos de los diferentes enfoques estudiados, lo que permite arribar a conclusiones.

Se aplicarán los siguientes **Métodos Empíricos**:

1. Análisis de documentos: estudio bibliográfico y documental, acerca del estado actual de la Ingeniería Orientada a Agentes. Se revisarán diferentes trabajos científicos y se realizará una breve comparación.

INTRODUCCIÓN

2. Criterios de especialistas: Se consultará la opinión de diferentes expertos, que por sus conocimientos y experiencias aporten ideas a la implementación propuesta, y ofrezcan valoraciones y recomendaciones al respecto.

Estructura del Documento

Para desarrollar cada uno de estos objetivos se estructuró este documento escrito en 3 capítulos, más Conclusiones y Recomendaciones.

En el **Capítulo 1**, "Estado del arte de las metodologías orientadas al desarrollo de Sistemas Multiagentes", se expondrá todo lo relacionado al estado del arte de las metodologías de desarrollo de sistemas multiagentes, haciendo énfasis en PASSI. En el **Capítulo 2**, "Diseño de la metodología propuesta integrada a PASSI", se describirá la propuesta de la solución, con pasos correctamente definidos para la realización de la etapa de pruebas, así como "**MEDSIM-PASSI**": metodología resultante de la integración de las fases conocidas de PASSI con dicha metodología de pruebas. En el **Capítulo 3**, "Validación de la Metodología", se ofrecerá un estudio de la validación de la metodología propuesta "MEDSIM-PASSI", así como una breve descripción de algunas ventajas y desventajas de su aplicación a SMA.

CAPÍTULO I:

*“Estado del arte de las
metodologías orientadas al
desarrollo de Sistemas
Multiagentes.”*

CAPÍTULO I: "Estado del arte de las metodologías orientadas al desarrollo de Sistemas Multiagentes."

1.1 Introducción.

La ingeniería de software ha ido desarrollándose y transita por diferentes etapas como son el paradigma orientado a objeto, el estructurado y ha ido evolucionando hasta centrarse en los Agentes Inteligentes. Surge así la Ingeniería de Software Orientada a Agentes que utiliza el enfoque de agentes para la implementación de sistemas de software complejos (distribuidos). Los agentes no dejan de ser software, y como todo proceso para su desarrollo, requiere de Análisis de requerimientos, Especificación de diseño, Pruebas, etc.; por lo que surgen las metodologías para su desarrollo, ya que las tradicionales no tienen en cuenta las necesidades de diseño de estas arquitecturas, como pueden ser la especificación de planificación de tareas, la comunicación mediante lenguajes específicos, la movilidad del código o la motivación de los componentes del sistema, o el tratamiento de las ontologías. Como consecuencia de esto existe un gran auge de trabajos metodológicos en este sentido, no contándose aún con una estandarización de ellas. A continuación se exponen las principales características de las más difundidas en la actualidad, principalmente en cuanto a su aplicabilidad, desarrollo del ciclo de vida, herramienta de soporte y principales críticas realizadas.

1.2 Metodologías de desarrollo de sistemas orientadas a agentes.

1.2.1 MAS-CommonKADS.

Esta metodología extiende CommonKADS aplicando ideas de metodologías orientadas a objetos para su trabajo en la producción de SMA. De hecho considera sólo dos agentes básicos: usuario y sistema. MAS-CommonKADS⁵ propone siete modelos (Figura 1) para la

⁵ Pavón Mestres, "Modelos y Arquitecturas de Agentes."

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

definición del sistema: modelo de la organización, modelo de la experiencia, modelo de tareas, modelo de coordinación, modelo de agente, modelo de comunicación y modelo de diseño.

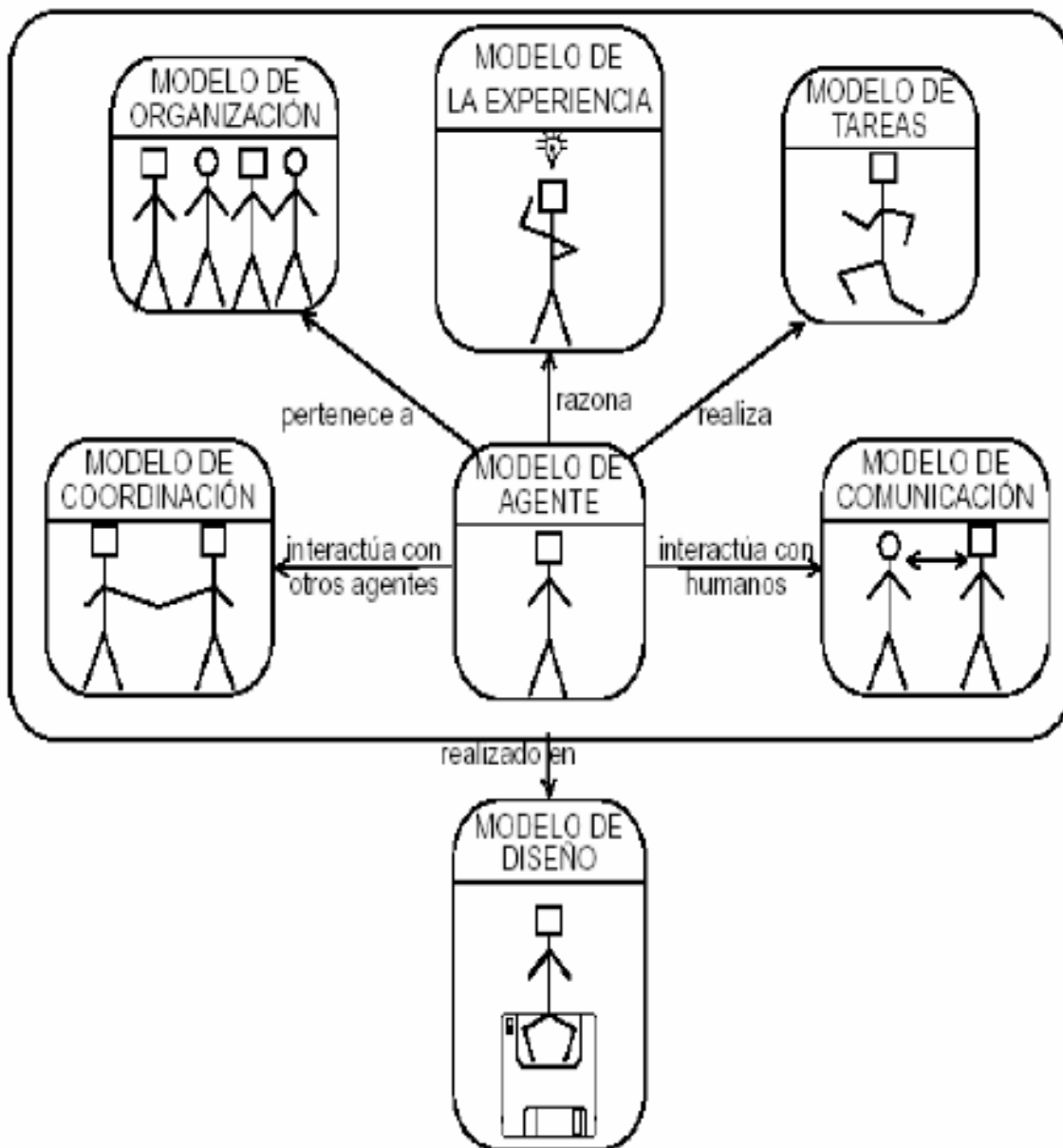


Figura 1. Modelos de MAS-CommonKADS para la definición del SMA.

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

MAS-CommonKADS ha sido aplicada con éxito en la optimización de sistemas industriales, la automatización de asistentes de reservas de vuelos y el desarrollo de sistemas de comercio electrónico.⁶ Cuenta con un modelo completo e independiente para la fase de diseño, sumado a una clara descripción de las relaciones de entrada-salida entre sus siete modelos y a la definición de un ciclo de vida específico, lo cual la hace robusta y coherente. Al igual que RUP (Rational Unified Process), es iterativa, lo cual facilita el refinamiento tanto del análisis como del diseño. La herramienta de desarrollo propuesta es AgentEditor. En el modelado incluye la modelación del ambiente, de la Inteligencia, la Interacción, la Organización y las Ontologías.⁷

Utiliza UML (Unified Model Language) como lenguaje de modelado.⁸ Dentro del ciclo de vida desarrolla la fase de Conceptualización donde se debe comprender el sistema que desea el cliente, los objetivos que debe satisfacer y los actores que interactúan con el sistema. En el Análisis, se determinan los requerimientos funcionales del sistema y en la fase de Diseño se determina la arquitectura interna de los agentes y la del sistema.⁹

La división que realiza en siete modelos distintos parece un poco secuencial y artificial en tanto no se observa una forma intuitiva y razonable de ir obteniendo estos modelos. El proceso no parece funcionar de forma iterativa, lo que resulta ilógico, puesto que todas las creaciones necesitan añadir detalles de forma progresiva.

Partir de la orientación a objetos y extender para crear orientación a agentes como hace MAS-CommonKADS, así como otras muchas metodologías, tiene tintes de ser un camino erróneo, el paradigma de agentes engloba al de objetos, pero es algo más que una simple extensión del mismo. Diseñar mediante agentes implica un gran salto cualitativo en la forma

⁶ Gómez Sanz, “Metodologías para el desarrollo de sistemas multiagentes.”

⁷ Moreno Espino, “Metodologías Orientadas a Agentes: un estudio comparativo.”

⁸ Cuesta Morales Pedro, “ISOA.”

⁹ Henderson Sellers y Giorgini, “Appropriate methodologies for constructing agent-oriented systems.”

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

de pensar, pudiendo asimilarse como proporcional al salto que hay desde la programación estructurada a la orientación a objetos, y eso debería verse también reflejado en el modelo.¹⁰

En cuanto a su alcance es una metodología de conceptualización, análisis y diseño; no abarca completamente etapas anteriores y posteriores a ambas, por lo que en las etapas del ciclo de vida está muy débil. Su principal inconveniente es que el nivel de detalle alcanzado en la descripción no es realizable sin el apoyo de herramientas de soporte, el proceso es demasiado costoso y dado a errores.

A pesar de estos inconvenientes constituye un ejemplo a seguir en lo que a metodologías se refiere. Es exhaustiva como pocas a la hora de detallar el sistema y además es consecuente con que el proceso de desarrollo en la mayoría de los casos es más complejo que un conjunto de pasos.¹¹

1.2.2 INGENIAS.

INGENIAS fue desarrollada por GRASIA, un grupo de investigación del Departamento de Sistemas Informáticos y Programación de la Universidad Complutense de Madrid. INGENIAS parte del paradigma OO, asume a AUML como lenguaje de modelado, y en ella los agentes deben tener metas y estados.

El método de desarrollo de SMA los concibe como la representación computacional de un conjunto de modelos (Figura 2), donde cada uno de estos modelos muestra una visión parcial del SMA.

¹⁰ Gallego Durán, Llorens Lar, y Rizo Aldeguer, *Breve análisis de algunas metodologías de diseño de SMA*.

¹¹ Gómez Sanz, “Metodologías para el desarrollo de sistemas multiagentes.”

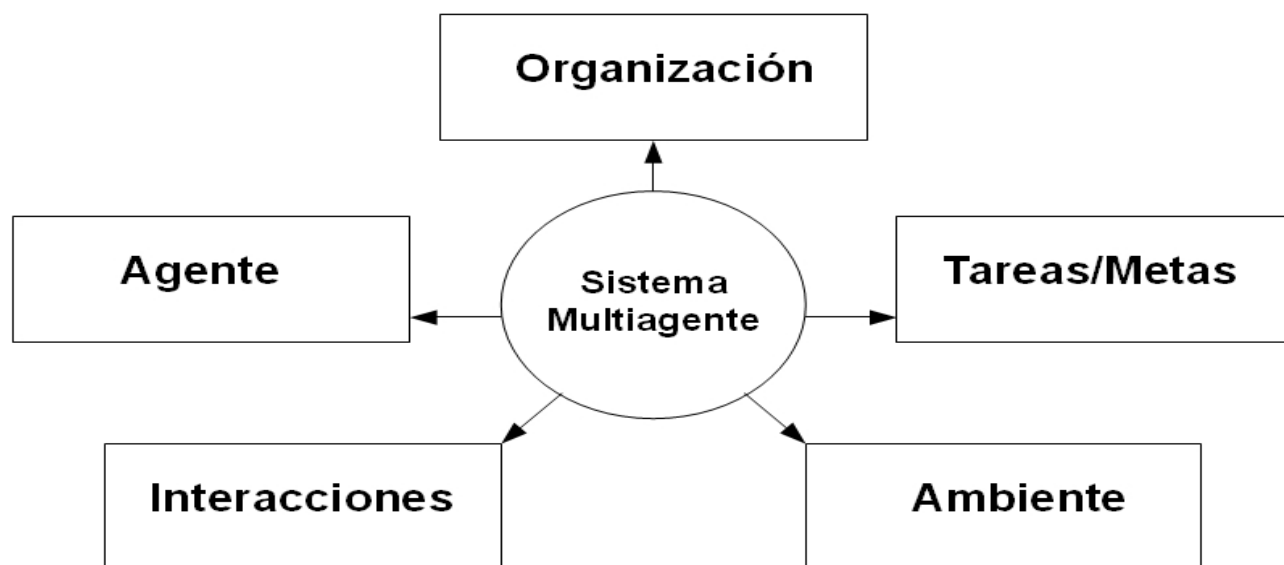


Figura 2. Conjunto de Modelos que tratan con el SMA para Ingenias.

Un agente aquí es un programa que sigue el principio de racionalidad y eso procesa el conocimiento. El principio de racionalidad dice que un agente ejecutará esas acciones que lo hacen lograr sus metas.¹² El punto de vista del agente se preocupa por la funcionalidad de cada uno: el propósito (qué metas un agente se compromete a seguir), responsabilidades (qué tareas tiene que ejecutar), y capacidades (qué roles juega).

INGENIAS modela algunas partes de la inteligencia, pero los pasos no quedan claros; el entorno, la interacción y la organización quedan claramente modelados en ella. Cubre en su ciclo de vida las etapas de Análisis, Diseño e Implementación, aunque esta última no la desarrolla del todo. La ejecución de actividades para producir modelos se basa en INGENIAS IDK (Ingenias Development Kit), una herramienta para modelado visual, que almacena la especificación del sistema utilizando XML, además de generar código, editor de modelos, generación de código y de documentación.

¹² Henderson Sellers y Giorgini, "Appropriate methodologies for constructing agent-oriented systems."

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

Luego de realizar un estudio práctico se ve que su primera fase está centrada a la captura de requisitos y que tiene una fase centrada en la generación de código.

INGENIAS dispone de una cantidad ingente de entidades y relaciones. Su uso mediante la herramienta de soporte INGENIAS IDE, se facilita pero se sigue requiriendo que el desarrollador revise la documentación de la tesis para entender qué hace cada entidad y cuál es el propósito de cada relación. El proceso de generación de código es un poco más flexible, los desarrolladores pueden configurarlo a voluntad y adaptarlo a sus necesidades sin tener que modificar la herramienta de análisis/diseño.

Es independiente de la plataforma de trabajo y el protocolo de comunicación. También es un poco tentativa en cuanto a la madurez de la herramienta CASE, la facilidad de uso de la misma y además la buena valoración que tiene en los aspectos de la vista de modelado.¹³

Las fases de pruebas e implementación no se han incluido. La fase de pruebas según el autor, no tiene por qué ser diferente de la del software convencional, sin embargo, esto no parece ser así porque el comportamiento de un SMA es emergente y por tanto difícil de pronosticar.

INGENIAS obvia el modelado de las ontologías y le falta fortalecer las etapas del ciclo de vida, pero tiene fortalezas en los demás aspectos. Ha sido aplicada en interfaces inteligentes, procesos de negocios, entre otros. En cuanto a la valoración subjetiva de la completitud de los flujos de trabajos, en todos los casos fue valorado de regular, ya que de una forma u otra faltan artefactos para completar los flujos de trabajo, estas características diferentes se pueden determinar por el desarrollo del caso de estudio.¹⁴

¹³ Gallego Durán, Llorens Lar, y Rizo Aldeguer, *Breve análisis de algunas metodologías de diseño de SMA*.

¹⁴ Moreno Espino, “Metodologías Orientadas a Agentes: un estudio comparativo.”

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

El proceso de desarrollo que propone es excesivo cuando se trata de desarrollos reducidos, da la impresión de dedicarse exclusivamente a desarrollos de gran tamaño.¹⁵ En el momento, no soporta el despliegue o el diagrama de componentes en su versión actual. En cambio, delega estas tareas a las existentes herramientas de desarrollo UML. A modo de conclusión, INGENIAS es una metodología que no es ágil y robusta, no soporta estructuras dinámicas ni sistemas abiertos, agentes móviles y ontologías.¹⁶

1.2.3 MASE (Multi-agent systems Software Engineering).

La metodología fue desarrollada en el Air Force Institute of Technology y trata de cubrir todas las etapas en el proceso de construcción de un SMA, partiendo de la especificación del mismo hasta su implementación. Dispone de un lenguaje de especificación basado en UML+OCL (Object Constraint Language), lo que evidencia mucho acercamiento a los conceptos orientados a objetos.¹⁷

MASE es iterativa, el diseñador se mueve libremente entre pasos y etapas. MASE parte del paradigma orientado a objetos y asume que un agente es sólo una especialización de un objeto. Los agentes no son considerados autónomos, proactivos y sociales, sino simples procesos que se comunican para conseguir el objetivo global del sistema. La especialización consiste en que se coordinan unos con otros vía conversaciones, y actúan proactivamente para alcanzar metas individuales y del sistema. En MASE los agentes son sólo una abstracción conveniente, que puede o no poseer inteligencia.¹⁸

¹⁵ Gómez Sanz, “Metodologías para el desarrollo de sistemas multiagentes.”

¹⁶ Henderson Sellers y Giorgini, “Appropriate methodologies for constructing agent-oriented systems.”

¹⁷ Moreno Espino, “Metodologías Orientadas a Agentes: un estudio comparativo.”

¹⁸ Gallego Durán, Llorens Lar, y Rizo Aldeguez, *Breve análisis de algunas metodologías de diseño de SMA*.

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

Como producto de sus etapas se esperan diagramas de clases de agentes y roles; se identifican conversaciones entre los mismos; se descompone el sistema (agente) en subsistemas (componentes del agente) e interconexión de los mismos, y diagramas UML de despliegue para indicar cuántos agentes habrá en el sistema.

El proceso de desarrollo en MASE es un conjunto de pasos, la mayoría de los cuales se ejecutan dentro de la herramienta que soporta MASE, AgentTool. La misma permite generar código automáticamente a partir de la especificación del sistema.¹⁹ Al igual que PASSI, es una metodología que propone una etapa de despliegue.²⁰

MASE sólo considera sistemas estáticos. Todo el proceso de desarrollo está bien descrito y detallado, los modelos gráficos tienen una expresividad adecuada y están bien coordinados, todo en su conjunto está basado en ideas de agentes muy semejantes a la arquitectura BDI (*Believes, Desieres, Intentions*). Su herramienta permite trabajar con todos los modelos de forma gráfica e interactiva y convierte modelos de análisis en modelos de diseño.

Luego de un estudio práctico se comprobó que MASE propone la etapa de implementación pero no la desarrolla, aunque sus herramientas de desarrollo generan código fuente y no cubre a plenitud las etapas del ciclo de vida de Requisitos y Prueba. El ciclo de vida de MASE es iterativo por todas las fases, pero no permite ir incrementando funcionalidades. El protocolo de comunicación entre los agentes es sólo definido por el usuario. De forma general puede ser valorada de buena, aunque adolece en el modelado de la inteligencia, las ontologías, la organización y el ambiente que es algo crucial, además de sus deficiencias en la fase de pruebas.

¹⁹ Gómez Sanz, “Metodologías para el desarrollo de sistemas multiagentes.”

²⁰ Moreno Espino, “Metodologías Orientadas a Agentes: un estudio comparativo.”

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

Como la metodología es resultado de una investigación realizada por la fuerza aérea de los Estados Unidos, como se apuntó anteriormente, parte de la información relevante del mismo podría no estar disponible públicamente.²¹

1.2.4 PASSI (Process for Agent Societies Specification and Implementation).

PASSI es una metodología que va desde los requerimientos hasta el código, paso por paso, para el diseño y desarrollo de un SMA. Integra el diseño de modelos y conceptos que vienen de la ingeniería de software orientada a objeto y un acercamiento a la IA. Usa UML, aunque para soportar algunos problemas específicos relacionados a la naturaleza del SMA, también usa AUML (Agent Unified Model Language). Ha evolucionado de un largo período de construcción teórica y experimentos para convertirse en un acercamiento más amplio a dicha disciplina que abarca una arquitectura de visión flexible y un modelado extenso del dominio y de la ontología. Tiene cinco modelos y doce pasos (Figura 3), concernientes a diferentes niveles de diseño, para el proceso de construcción de un SMA.

Sus modelos y fases abarcan una representación de los requerimientos, punto de vista social, solución de arquitectura, reutilización y producción de código y configuración del despliegue del sistema. El proceso de diseño es soportado por el PTK (PASSI ToolKit), el cual agrega una extensión al Rational Rose y una herramienta para la reutilización de patrones de agentes.

En PASSI, un agente es una instancia de una clase agente y cada uno tiene una representación del mundo en términos de una ontología. Además, cada uno puede emprender varios roles funcionales durante la interacción con otros agentes para lograr sus

²¹ Henderson Sellers y Giorgini, “Appropriate methodologies for constructing agent-oriented systems.”

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

metas. Un rol es una colección de tareas realizadas siguiendo sub metas u ofreciendo algunos servicios a otros miembros de la sociedad.

PASSI se caracteriza por un doble nivel de iteración.²² Es general y puede ser aplicada en diferentes ambientes de implementación de agente, su herramienta de desarrollo cubre todas las etapas de desarrollo. Es válido destacar que en sus modelos se refleja la ontología, las interacciones y el ambiente; y a su vez la interacción humana es modelada a través de un actor.

Hay que destacar que PASSI incluye todas las etapas del ciclo de vida del software aunque la de requisitos y prueba, no son desarrolladas del todo. Como plataforma de desarrollo propone a JADE y FIPA-OS. El protocolo de comunicación es predefinido y sustenta cualquier tipo de cooperación.

De forma general desde un punto de vista teórico, es una metodología bastante acabada, aunque adolece de algo muy importante, como lo es el modelado de la inteligencia y la organización. PASSI es tentativa en cuanto a la madurez de la herramienta CASE, la facilidad de uso de la misma y además por la buena valoración que tiene en los aspectos de la vista de modelado.²³

Entre los rasgos más apreciados, podemos listar: la facilidad de transición para los diseñadores que vienen del mundo Orientado a Objetos (OO), desde que los conceptos iniciales de PASSI fueron adoptados para el análisis de los requisitos, que es muy común en ese contexto; las diversas vistas que permiten un fácil análisis de los sistemas complejos de muchos aspectos diferentes; el apoyo de una herramienta de diseño específica (PTK, un

²² Díaz Gómez, "Sistema Multiagente para DPEG."

²³ Moreno Espino, "Metodologías Orientadas a Agentes: un estudio comparativo."

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

complemento para Rational Rose), y los modelos reutilizables que permiten un desarrollo rápido de los SMA.

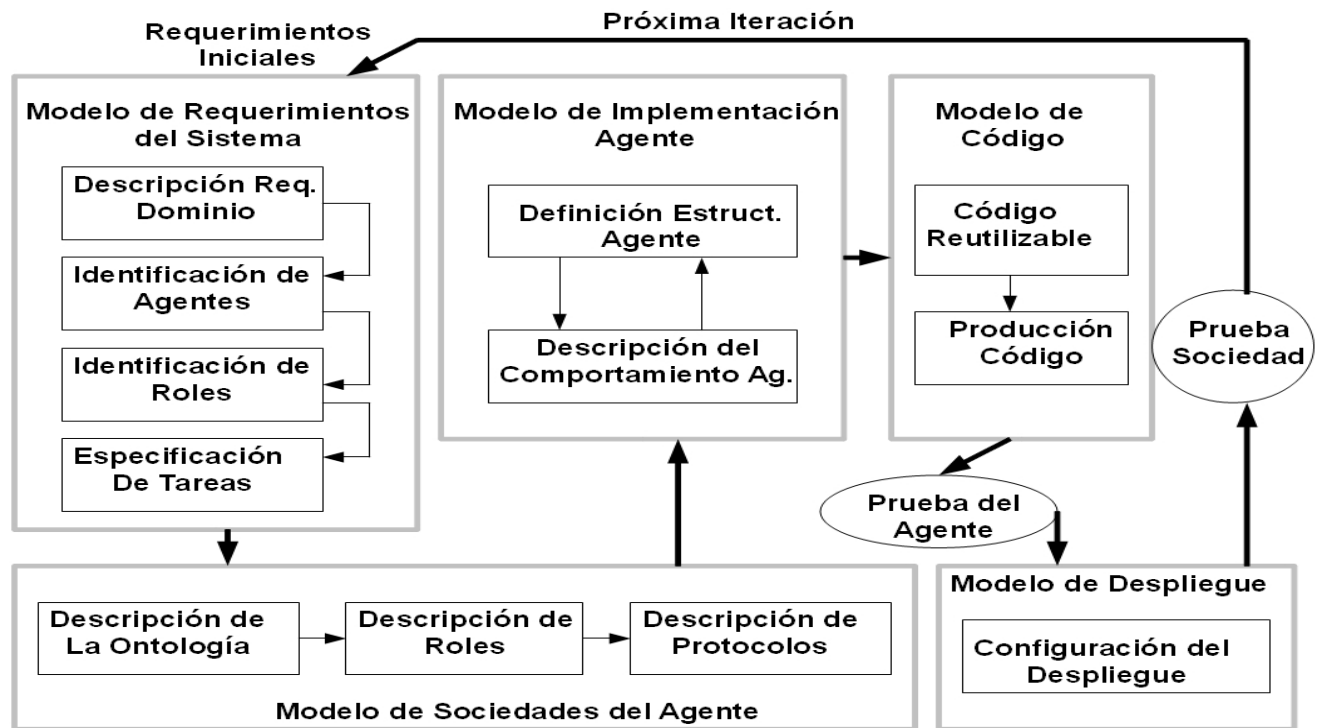


Figura 3. Modelos y fases en la metodología PASSI.

Excluyendo PASSI, ninguna de las otras metodologías tratadas aquí especifican el modelado de la Ontología. También se propone realizar una prueba de agente luego del modelo de código y una prueba de sociedad luego del modelo de despliegue, destacando en ambos casos la necesidad de estas pruebas pero no cómo hacerlas, ni que artefactos obtener.

1.3 Comparación de las Metodologías.

Las cuatro metodologías descritas son importantes contribuciones al desarrollo de los SMA, no obstante poseen fortalezas y debilidades. A continuación se presenta una tabla comparativa en cuanto a los aspectos relacionados con el proceso.

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

Tabla 1. Comparación en cuanto a aspectos relacionados con el proceso.

Aspectos Relacionados	MAS-CommonKADS	INGENIAS	MASE	PASSI
Naturaleza del Agente	Heterogénea.	Agentes con metas y estados.	No especificada pero posible Heterogénea.	Heterogénea.
Ciclo de Vida	Cíclico orientado a riesgos.	Iterativo e Incremental.	Iterativo.	Iterativo e Incremental.
Etapas del Ciclo de Vida	Conceptualización, Análisis y Diseño.	Requisitos, Análisis, Diseño e Implementación.	Requisitos, Análisis y Diseño.	Requisitos, Análisis, Diseño, Despliegue e Implementación.
Sistemas Abiertos	No.	No.	No.	No.
Tamaño del SMA	No especificado.	No especificado.	≤ 10 .	No especificado.
Usabilidad	Medio.	Alto.	Alto.	Alto.
Facilidad de Entender	Alto.	Alto.	Alto.	Alto.
Antecedentes	Ingeniería del conocimiento + conceptos OO.	OO.	OO.	OO.
Fase de Pruebas	Realizada en forma individual al agente.	No está incluida.	Métodos de verificación de la corrección.	No está desarrollada completamente.

A continuación se presenta una tabla que recoge una comparación entre los aspectos a tener en cuenta por cada metodología para el modelado.

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

Tabla 2. Aspectos que toma en cuenta el modelado.

Aspectos Relacionados	MAS-CommonKADS	INGENIAS	MASE	PASSI
Ontologías	Sí	No	No	Sí
Organización	Sí	Sí	No	No
Interacción	Sí	Sí	Sí	Sí
Inteligencia	Sí	Poco	No	No
Ambiente	Sí	Sí	No	Sí

Según la Tabla 1 ninguna de estas cuatro metodologías modela sistemas abiertos, lo cual constituye una deficiencia en todas ellas.

Hasta el momento podemos ver que MASE no tiene un ciclo de vida tan sólido como las restantes metodologías. La etapa de Requisitos es pobre y termina en el diseño aunque su herramienta de desarrollo permite generar código automáticamente. En sus modelos no se detallan ni el ambiente, la inteligencia, organización u ontologías, por lo que necesita enriquecerse en este sentido.

MAS-CommonKADS posee igualmente un ciclo de vida que deja mucho por desear en comparación con las potencialidades de la metodología, pues se considera la más robusta de todas. En sus modelos podemos encontrar la mayoría de los conceptos más importantes dentro del paradigma de agentes. Ella junto con PASSI e Ingenias serán comparadas en cuanto a los conceptos que tratan sus modelos detalladamente como en el libro.²⁴

²⁴ Díaz Gómez, "Sistema Multiagente para DPEG."

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

A continuación una tabla comparativa con los diferentes conceptos considerados en los distintos modelos.

Tabla 3. Comparación en cuanto a los conceptos considerados en los Modelos.

CONCEPTOS	MAS-CommonKADS.	INGENIAS.	PASSI.
Sistemas de Metas	Casos de Metas.	Modelo de Tareas y Metas.	-
Sistemas de Tareas/ Comportamiento	Modelo de Tareas.	Modelo de Tareas y Metas.	Modelo de Requerimientos del Sistema.
Escenario de CU	Casos de Uso.	Diagrama de Casos de Uso.	Modelo de Requerimientos del Sistema.
Roles	-	Modelo de Agentes, Modelo de Organización y Modelo de Interacción.	Modelo de Requerimientos del Sistema, Modelo de la Sociedad de Agentes.
Conceptualización del Dominio	Modelo de Experiencia.	-	Modelo de la Sociedad de Agentes.
Asignación de Roles al Agente	Modelo de Agentes.	Modelo de Agentes.	Modelo de la Sociedad de Agentes
Metas/ Tareas del Agente	Modelo de Agentes.	Modelo de Agentes.	Modelo de Requerimientos del Sistema.
Creencias/ Conocimientos del Agente	Modelo de Experiencia.	-	Modelo de Implementación del Agente.
Capacidades/	Modelo de Agentes,	-	Modelo de la Sociedad

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

Servicios del Agente	Modelo de la Organización.		de Agentes.
Razonamiento del Agente para la Solución de Métodos	Modelo de Experiencia.	Modelo de Agentes.	Modelo de Implementación del Agente.
Percepción del Agente	Como eventos del Diagrama de Transición de Estados en el Modelo de Coordinación.	-	-
Arquitectura del Agente	Modelo de Diseño.	-	Modelo de Implementación del Agente.
Conocimiento del Agente	Modelo de Coordinación.	Modelo de Interacción.	Modelo de Requerimientos del Sistema.
Protocolos de Interacción	Modelo de Coordinación.	Modelo de Interacción.	Modelo de la Sociedad de Agentes.
Contenido del Intercambio de Mensajes.	Modelo de Coordinación.	Modelo de Interacción.	Modelo de Implementación del Agente.
Arquitectura del Sistema	Modelo de Coordinación.	Modelo de Organización.	Modelo de Implementación del Agente.
Recursos del Ambiente/ Facilidad	Modelo de Coordinación, Modelo de Diseño.	Modelo del Ambiente.	-
Relaciones de Agregación de	Modelo de Organización.	-	-

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

Agentes			
Despliegue de Instancias de Agentes	-	-	Modelo de Despliegue.

Se han tratado aquí aspectos esenciales en lo que se refiere a modelado de agentes. Primeramente en MAS-CommonKADS, sus siete modelos conceptualizan detalladamente el sistema como se muestra en la tabla anterior, pero no logran una relación detallada y lógica entre ellos. Por otra parte las principales debilidades son sus limitaciones soportando las etapas de diseño, implementación y prueba.

PASSI e INGENIAS parecen ser ambas las mejores propuestas, aunque se ha demostrado que poseen aspectos negativos. Ninguna de las dos modela sistemas abiertos, ni son robustas y ágiles, en cambio PASSI soporta agentes móviles y las ontologías, al contrario de INGENIAS.

En la Tesis de Maestría²⁵ se realiza una comparación desde el punto de vista práctico de ambas metodologías. Entre sus principales conclusiones estaban que:

En cuanto a **INGENIAS**:

- ✚ La herramienta CASE cubre todas las etapas de desarrollo que desarrolla la metodología, incluso se comprobó que posee una etapa de implementación.
- ✚ Los agentes deben tener metas y estados, y el concepto de autonomía es bastante amplio, permitiendo una serie de atributos como son autonomía, sociabilidad, reactividad, proactividad, inteligencia, entre otros.
- ✚ La apreciación de la completitud de los modelos son evaluados de regular.
- ✚ No modela la interacción humana, ni las ontologías.

²⁵ Moreno Espino, “Metodologías Orientadas a Agentes: un estudio comparativo.”

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

En cuanto a **PASSI**:

- ✚ Su herramienta cubre todas las etapas y es de fácil manejabilidad.
- ✚ Propone a JADE y FIPA-OS como plataformas de desarrollo
- ✚ Se modela la interacción humana como un actor
- ✚ Sustenta cualquier tipo de comunicación
- ✚ Es la única que considera el modelado de las Ontologías.

Sturm, Shehory y Dori²⁶ realizan una evaluación donde están ambas metodologías. Se puede ver que son realmente muy similares y que adolecen en los mismos aspectos generalmente. INGENIAS no posee buen soporte para las etapas de prueba y despliegue, mientras que PASSI adolece en el grado de implicación del usuario.

Generalmente ambas son buenas propuestas; pero considerando que PASSI fue evaluada de Bien en la evaluación práctica de Moreno Espino²⁷, tiene como plataforma de desarrollo a JADE, utiliza al Rational Rose, herramienta ampliamente conocida por sus potencialidades en el mundo de la Ingeniería del Software, fue seleccionada como la metodología para realizar el diseño metodológico para la realización de la fase de pruebas.

PASSI posee dos etapas anteriores a la etapa de identificación de los roles que son la descripción del dominio y la identificación de agentes, todo dentro del modelo de requerimientos del sistema. Identifica los agentes, describe las ontologías y finalmente describe los protocolos de comunicación. Además, en la última etapa del diseño, involucra el modelado individual y social de los agentes. Ya en cuanto a la etapa de implementación y generación de código, PASSI no posee en la actualidad la posibilidad de generar código,

²⁶ Sturm, Shehory, y Dori, "Methodologies evaluation."

²⁷ Moreno Espino, "Metodologías Orientadas a Agentes: un estudio comparativo."

CAPÍTULO I: ESTADO DEL ARTE DE LAS METODOLOGÍAS ORIENTADAS AL DESARROLLO DE SISTEMAS MULTIAGENTES

aunque esta característica está en desarrollo. Finalmente en la etapa de pruebas se adolece de pasos correctamente definidos para la realización de la misma.

1.4 Conclusiones del Capítulo.

Luego de realizar un minucioso estudio de las principales metodologías en el desarrollo de sistemas multiagentes se ha seleccionado la metodología PASSI para realizar el diseño, pues aún con los inconvenientes señalados, es la que más se ajusta por sus características a la posible solución que necesitamos.

Entre sus principales características podemos señalar que posee una herramienta de soporte bien aceptada por los desarrolladores y es compatible con la Plataforma de Desarrollo JADE. Además sus modelos utilizan las potencialidades de UML y AUML para modelar sus diagramas. Realiza el proceso de inferencia de la base de conocimientos con Amzi Prolog, al tener facilidades de comunicación con aplicaciones Java, y la disponibilidad de una clase con las principales funcionalidades implementadas.

CAPÍTULO II:

“Diseño de la metodología de pruebas integrada a PASSI.”

CAPÍTULO II: "Diseño de la metodología de pruebas integrada a PASSI."

2.1 Introducción.

PASSI ha evolucionado de un largo período de construcción teórica y experimentos en el desarrollo de aplicaciones orientadas a agentes; posee importantes características como una arquitectura de visión flexible y un modelado extenso del dominio y de la ontología. En su largo periplo, ha adolecido de la completitud de las etapas del ciclo de vida del software, al sufrir la ausencia de pasos correctamente definidos para la realización de la etapa de pruebas. En este capítulo se diseña la metodología para la realización de dicha etapa, de gran importancia para el desarrollo de sistemas multiagentes de gran calidad.

Habiendo terminado con la descripción de los fundamentos, etapas y actividades de las metodologías de desarrollo de SMA, en la siguiente sección se presenta la documentación detallada de la metodología de Pruebas para SMA diseñada dentro de esta investigación y que se integró con PASSI. Se da inicio a esta documentación describiendo en principio los pilares y supuestos de este nuevo enfoque, posteriormente se detallará la forma en que esta metodología se integra con el ciclo de vida de PASSI y finalmente se documentarán las etapas, Artefactos y Pruebas contempladas en el ciclo de vida de la metodología de Pruebas planteada.

2.2 Fundamentos de la metodología de pruebas para SMA.

Como se había explicado antes en la introducción de este trabajo, la fase de pruebas ha sido una etapa particular en todo proceso de Software empezando por su enfoque que es totalmente opuesto al de las demás etapas del Ciclo de Vida.

Las pruebas en sí son un proceso, en el cual se busca tener la mayor cantidad de pruebas

CAPÍTULO II: DISEÑO DE LA METODOLOGÍA DE PRUEBAS INTEGRADA A PASSI

para así detectar la mayor cantidad de defectos, así mismo estas pruebas no necesariamente deben ejecutarse en una sola fase.

En los SMA, las pruebas son un proceso iterativo, en los que el sistema se va construyendo por etapas, y en cada una de ellas se relacionan conceptos propios del paradigma de agentes.

Ante este hecho se propone una metodología de pruebas que se ejecute paralelamente a este proceso y que contemple diferentes tipos de pruebas asociados a la verificación y validación de todos los elementos anteriormente descritos.

Tabla 4. Supuestos y pilares de Metodología de Pruebas de SMA.

Las pruebas como proceso iterativo y no como una sola fase.	Total integración con PASSI, las pruebas como avance interno y no como un trabajo externo.
No basta con probar los Agentes finales y el SMA implementado para garantizar su calidad.	Iteraciones continuas y puntos de verificación más frecuentes.
Técnicas de verificación y validación que permitan un control constante.	El mejor control que puede ejecutarse es el que se lleva durante el proceso y no posterior a él.
Planes de Prueba como mecanismo para estandarizar el proceso de pruebas en los proyectos de Agentes.	Diferentes tipos de pruebas para diferentes tipos de defectos.
Trazabilidad como elemento indispensable para garantizar la lógica y la consistencia de los procesos de pruebas.	Las pruebas deben ejecutarse prontamente por el crecimiento exponencial en la complejidad de los defectos de Software.

2.3 Integración de la Metodología de Pruebas con la Metodología PASSI.

En atención a los supuestos descritos en la sección anterior se ha generado una metodología basada en un proceso iterativo de pruebas totalmente integrado a la metodología PASSI ya explicada, de modo que la metodología dedica pruebas exclusivas para cada uno de los elementos y factores principales en el proceso metodológico llevado por PASSI.

El proceso de pruebas contemplado considera que: en la ejecución de actividades como detección y refinación de Objetivos, Habilidades, en la identificación de Tareas y Roles, así como en la descripción de Ontologías, Roles y Protocolos, pueden existir puntos de fallo que deben ser contemplados mediante alguna prueba específica, con el fin de identificar y corregir la mayor variedad de defectos posibles y de este modo garantizar la robustez y corrección del SMA respecto a los requerimientos definidos para el mismo.²⁸

En virtud de lo anterior la Metodología de Pruebas contempla estrategias de validación exclusivas para cada punto de fallo que pueda presentarse en el proceso metodológico de PASSI, para este fin. La metodología define pruebas específicas situadas dentro de las etapas del proceso metodológico de PASSI. Estas pruebas pueden ser de carácter Opcional u Obligatorio dependiendo la configuración de la metodología de pruebas que se seleccione en la fase de “Requerimientos Iniciales” etapa en la cual según, la documentación de esta metodología, se ejecutan tareas como la Recopilación de los Requerimientos del SMA: Prerrequisito para entrar al ciclo de vida propuesto por PASSI.

Dentro de este contexto, se define la primera etapa de la metodología de Pruebas llamada Planeación de Pruebas dentro de la cual como tarea principal se definen, aspectos como: Qué pruebas van a ejecutarse y con Qué nivel de detalle se aplicarán estas pruebas. Existen pruebas como la de Validación de Roles, Consistencia de Vínculos y Validación del

²⁸ Garavito Oscar, Ocampo Juan, Torres Miguel, “Metodología de Pruebas para Sistemas Multiagentes integrada a AOPOA.”

CAPÍTULO II: DISEÑO DE LA METODOLOGÍA DE PRUEBAS INTEGRADA A PASSI

Comportamiento de los Agentes, que por tratarse de pilares fundamentales de PASSI, son obligatorias independientemente de la configuración que se seleccione de la metodología mientras que otras pruebas aplicadas a requerimientos u objetivos son opcionales y puede optarse por ellas si se cuenta con la disponibilidad de tiempo y recursos para este fin.

Una vez que se ha seleccionado la configuración de Metodología que se aplicará y se ha elaborado el Plan de Pruebas para la aplicación de este proceso al proyecto de Agentes, se ingresa a la fase de Análisis de PASSI en la cual se contemplan dos pruebas orientadas a las dos sub-etapas en las que se divide esta fase. La prueba 1 en la sub-etapa de caracterización busca validar y verificar la relación existente entre los Objetivos y Requerimientos Generales del Sistema identificados en esta etapa, mientras que la segunda prueba de carácter obligatorio es iterativa y se ejecuta dentro de cada iteración del ciclo de descomposición organizacional, validando en cada n-prueba diversos puntos de fallo heredados de las actividades internas realizadas en el ciclo de descomposición para generar roles especializados a partir de roles globales.

Pasando a la etapa de diseño de PASSI en ella pueden también encontrarse dos pruebas. La primera de ellas es de carácter obligatorio y busca validar la definición de ontologías y roles establecido en la etapa anterior y la consistencia de la estructura seleccionada en base a los Requerimientos Funcionales y No Funcionales del SMA. Para este fin la Metodología define pruebas a nivel general y específico sobre ontologías y roles, y la aplicación de criterios predefinidos para validar que la estructura contemple todos los aspectos necesarios para garantizar un comportamiento normal y consistente en el SMA. Posterior a esto en la metodología de pruebas se encuentra una prueba orientada a validar el agrupamiento de roles y la definición de los estados y eventos de roles. Ya en las fases finales, se realiza una prueba de Agente posterior al modelo de código y una última prueba, esta de Sociedad luego del modelo de despliegue.²⁹

²⁹ Marchetti Tulio José, García Alejandro Javier, "Metodologías de desarrollo de sistemas multiagentes: un análisis comparativo."

De este modo se espera que los ciclos de vida de PASSI y de pruebas puedan ejecutarse en paralelo, siguiendo una estrategia de Definición y Validación continua que garantice que las pruebas de los elementos claves se encuentren y se ejecuten tan pronto se termine la definición del elemento a probar. Así se mantendrán los puntos de verificación frecuentes y la detección temprana de defectos que garantizarán un menor “esfuerzo” en la aplicación de las pruebas a SMA permitiendo así, la contemplación del elemento de pruebas de la forma más íntegra y transparente posible.

2.4 Presentación de la Metodología de Pruebas.

En esta sección se presentan las etapas de las cuales se compone la metodología de pruebas para SMA obtenida en el proyecto generador del presente trabajo. Estas etapas, como pudo verse en la sección anterior están completamente integradas con las etapas de la metodología PASSI, de modo que además de la primera etapa de Planeación de Pruebas las siguientes tres etapas se acoplan con las etapas de análisis, diseño y despliegue de PASSI respectivamente. Cada una de estas etapas considera una o más pruebas que a su vez generan artefactos y reportes de resultados. A continuación se describen una a una las etapas de esta metodología revisando las pruebas de las que se compone cada etapa, los artefactos generados y por último el concepto de proceso de control enmarcado dentro del ciclo de vida de la metodología explicada.

Etapas 1(E1): Planeación de las Pruebas.

En el contexto de PASSI, esta etapa puede considerarse como prerrequisito de entrada a la metodología, el objetivo de este proceso es definir con claridad aspectos como: ¿Qué distribución de recursos y tiempos se destinará para la fase de pruebas?, ¿Qué plan de actualización se llevará a cabo para los artefactos de prueba documentados a lo largo del proyecto, incluyendo el Plan de Pruebas?, ¿Qué configuración de la metodología de pruebas sugerida se aplicará para el proyecto en curso?, es decir ¿Qué etapas se tendrán en

CAPÍTULO II: DISEÑO DE LA METODOLOGÍA DE PRUEBAS INTEGRADA A PASSI

cuenta?, ¿Qué pruebas se aplicarán?, ¿Se aplicarán o no los procesos de control definidos en la metodología?. Así mismo, el objetivo de esta etapa es concluir con la generación del artefacto: Plan de Pruebas del Sistema, en el cual se consignará toda la información recogida en esta etapa, con el fin de estandarizar y formalizar el proceso de pruebas del proyecto que va iniciar. Los aspectos que deben tenerse en cuenta en este artefacto son:

1. Objetivos del proyecto y de las pruebas a efectuar.
2. Configuración de la metodología de pruebas que será utilizada en el proyecto a iniciar, teniendo en cuenta que no todas las pruebas son obligatorias.
3. Calendario y distribución en la ejecución de las etapas y pruebas seleccionadas.
4. Descripción de los hitos y artefactos de prueba dentro del proyecto.
5. Distribución de recursos del proyecto y asignación al seguimiento de las etapas de prueba.
6. Herramientas de prueba.

Este proceso de pruebas es personalizado de acuerdo a las condiciones de complejidad, presupuesto y tiempo asociadas a cada proyecto, por esta causa la metodología sugiere diversas configuraciones de ejecución con el fin de acoplarse a diferentes tamaños de proyecto, de modo que el tiempo invertido en la actividad de pruebas sea proporcional al tiempo global exigido por el proyecto. En vista de que esta etapa está directamente relacionada con uno de los pilares de la metodología de pruebas y que es en este período de tiempo en el que se define el plan de estandarización del proceso de pruebas en el proyecto, la ejecución de esta etapa es de carácter obligatorio y debe realizarse en común acuerdo por todos los integrantes del equipo de proyecto, de forma paralela a actividades de PASSI como el estudio de factibilidad y el levantamiento de requerimientos del Sistema, actividades en las que también deben participar todos los asociados al proyecto de Agentes.

Terminando con la etapa 1 de la metodología se prosigue con la etapa 2, la cual se ejecuta en paralelo con la fase de análisis de PASSI.

Etapas 2(E2): Validación de Objetivos, Tareas y Roles.

Esta etapa se enfoca en garantizar la total trazabilidad entre los elementos identificados en la sub-fase de caracterización del sistema de PASSI (tales como requerimientos, objetivos, habilidades, recursos y tareas) así como en la validación de la correcta descomposición organizacional del rol sistema en los roles atómicos que después serán detallados y agrupados para constituir los agentes finales. Para este fin la etapa contempla las siguientes dos sub-fases que se ocupan respectivamente de las tareas anteriormente descritas.

Etapas 2 / Sub-Fase 2.1 (E2 (SF2.1)): Validación de Objetivos, Agentes y Tareas.

En esta sub-fase la metodología se enfoca en garantizar la trazabilidad y consistencia entre todos los elementos identificados y relacionados en la fase de caracterización del sistema de PASSI. Esta trazabilidad se define como: “Las relaciones establecidas entre los elementos tales como: requerimientos, objetivos, agentes y tareas deben permitir mapear en su totalidad los resultados esperados del SMA, respetando condiciones como la consistencia entre los objetivos y las habilidades necesarias para su cumplimiento, la correcta definición de tareas y la relación lógica entre las tareas asociadas con las habilidades de cada entidad”. En vista de que es complejo evaluar la total consistencia entre todos estos elementos en una prueba, en esta sub-fase se sugiere la realización de las siguientes 2 pruebas:

1. Prueba de Objetivos y Requerimientos Generales del Sistema.

En esta prueba se busca garantizar en primer lugar la consistencia entre los requerimientos y los objetivos identificados para el SMA, para este fin se propone validar que cada uno de los requerimientos definidos haya sido considerado de forma explícita o implícita en al menos uno de los objetivos definidos para el SMA, esto con el fin de garantizar que ningún requerimiento se descarte en esta etapa de PASSI.

CAPÍTULO II: DISEÑO DE LA METODOLOGÍA DE PRUEBAS INTEGRADA A PASSI

Para este fin la metodología propone un formato mediante el cual es posible diligenciar y validar la consistencia entre estos dos elementos sin efectuar el mismo proceso de documentación dos veces. Este consta de una tabla en la cual se enumeran los requerimientos y objetivos identificados para el Sistema, luego se establecen las relaciones directas o indirectas entre ambos para encontrar relaciones redundantes o elementos sobrantes.

A través de esta plantilla es posible detectar los casos en los que un requerimiento no sea cubierto por algún objetivo (Punto de Fallo Principal), así como los casos en los que un objetivo surja de forma espontánea sin ningún requerimiento asociado, hecho que puede reflejar o una definición incompleta de requerimientos u objetivos innecesarios que no son relevantes para el problema tratado. Esta plantilla se convierte en el artefacto de salida de esta prueba y es de carácter opcional. Su ejecución depende de la disponibilidad de tiempo y recursos demandados por este proceso.

2. Prueba de Agentes y Tareas.

En esta prueba se parte de los requerimientos y objetivos que ya fueron validados a través de la prueba anterior, respecto a estos objetivos se busca validar que para cada uno se haya definido la configuración adecuada de agentes, roles y tareas necesarias para su cumplimiento.

Esta prueba se realiza con el fin de garantizar que en el futuro no se presenten problemas relacionados con roles insuficientes para un requerimiento dado, o tareas que requieran agentes no definidos, entre otras situaciones. Al garantizar la consistencia entre estos cuatro elementos en el principio del ciclo de vida del proyecto se está logrando que cada requerimiento sea una unidad atómica que cuente en sí con todos los elementos necesarios para su total cumplimiento sin la dependencia de otras unidades. Para este fin la metodología define un solo formato en el cual es posible documentar y validar la consistencia entre estos cuatro elementos, de modo que en una tabla se identifican para cada requerimiento la información total de la configuración de agentes, roles y tareas necesarios para su lograr su

cumplimiento. Esta plantilla no solo constituye el artefacto de salida de esta prueba, sino también se convierte en una evidencia clara del punto de verificación logrado en esta iteración.

Etapas 2 / Sub-Fase 2.2 (E2 (SF 2.2)) Validación y Verificación de Roles.

El seguimiento de esta etapa así como la ejecución de la prueba básica que la compone son de carácter obligatorio, debido a que la prueba sugerida apunta a la verificación y validación de uno de los pilares de la metodología PASSI: El Modelo de Sociedades del Agente. En vista de que este proceso es iterativo y en cada iteración se busca descomponer un rol general en roles específicos y atómicos a partir de la descomposición de los objetivos del rol original, la prueba paralela a este proceso también debe ser de ejecución iterativa y su validación se enfoca a los puntos de fallos detectados en el interior de cada iteración. A continuación se describe la ejecución y lógica de esta prueba.

1. Prueba de Validación de Roles.

Esta prueba se ejecuta de forma iterativa, de modo que en cada iteración del proceso de descomposición debe realizarse la prueba que se documenta en la presente sección. Allí la Tabla de Roles es un artefacto de PASSI el cual contiene los roles que se van obteniendo en cada ciclo del modelo de sociedades.

Internamente la prueba revisa aspectos tales como la correcta interpretación de la complejidad del rol a descomponer, así como criterios discrecionales orientados a determinar el número de requerimientos, agentes y tareas implicados en cada rol y de este modo concluir si es necesario o no que sea descompuesto. Así mismo la prueba valida que la división de objetivos sea consistente de modo que la sumatoria del cumplimiento de los objetivos específicos definidos constituya el cumplimiento del objetivo padre y que para estos nuevos objetivos se tenga además la configuración adecuada de requerimientos, agentes y tareas necesarios para su cumplimiento; nótese que esta actividad se relaciona con la prueba b de la sub-fase anterior. Finalmente, la prueba valida que el agrupamiento de

objetivos en roles sea óptimo de modo que se concluya con roles atómicos, especializados, independientes y de complejidad menor al rol descompuesto.

Una vez se ha probado y garantizado que el modelo de sociedades ha conducido a roles atómicos y consistentes con definiciones claras de objetivos, tareas y roles con respecto a los requerimientos del sistema y que en alguna medida se ha empezado con el proceso de identificación de la estructura y el comportamiento de los agentes, en la siguiente etapa se busca garantizar que los vínculos identificados a lo largo del modelo de sociedades y detallados en la fase de diseño de PASSI, son consistentes respecto a la configuración de roles existente en el sistema y que los agentes obtenidos representan un agrupamiento correcto de los roles que lo componen, y su estado y estructura internos permitirán el cumplimiento de los requerimientos Funcionales y No Funcionales predeterminados para el desarrollo.

Etapas 3 (E3): Validación de Vínculos y Comportamiento del Agente.

En esta etapa se busca validar los vínculos identificados en la etapa de diseño de PASSI. Esta validación se lleva a cabo a través de dos pruebas, la primera valida los vínculos y la segunda prueba valida el comportamiento del agente, basándose en los requerimientos del sistema y en la estructura que actualmente se lleva del mismo.

1. Prueba de Consistencia de Vínculos.

En esta prueba se realizan tres actividades.

En la primera se toman los vínculos definidos para el SMA para validar su naturaleza (Simple colaboración, colaboración coordinada, independencia), analizando que roles relaciona dicho vínculo para estudiar si la relación que deben tener es acorde con base a la configuración de requerimientos y roles, luego se elabora una lista de las implicaciones de ese vínculo tanto en procesamiento como en comportamiento esperado. Debido a que resulta complicado predecir todas las implicaciones se intenta detallar la mayor cantidad y de mayor importancia,

CAPÍTULO II: DISEÑO DE LA METODOLOGÍA DE PRUEBAS INTEGRADA A PASSI

encontrando de esta manera si los vínculos interfieren entre sí, y si son necesarios en el sistema.

En la segunda actividad se listan todos los recursos que por los vínculos hallados pueden implicar conflictos futuros o sugieren una necesidad de coordinación para su acceso, respecto a esta lista se revisa que estos recursos ya existan en la tabla de recursos del Sistema y que su descripción haya contemplado las necesidades de coordinación y conflictos que evidenciaron los vínculos encontrados, por ejemplo si se determina que un recurso no es renovable, a partir de ello se puede escoger un mecanismo para garantizar que el recurso esté disponible en el Sistema. Una vez identificada la naturaleza de los vínculos, estos resultados ayudarán para la solución de conflictos que tendrá el Sistema y para la selección de los adaptadores (elementos mediante los cuales los roles manipulan y acceden a los recursos) que tendrán cada uno de los recursos problemáticos listados.

La tercera actividad consiste en revisar el diagrama de roles (generado al ejecutar PASSI) y vínculos del SMA como un todo, intentando encontrar patrones anormales como interbloqueos, ciclos innecesarios, roles incomunicados. El análisis sobre este diagrama se debe hacer desde diferentes puntos de vista, desde los objetivos, tomando cada objetivo general del sistema y analizando si en el diagrama los roles encargados de dar cumplimiento a este objetivo se comunican y se acoplan correctamente, desde el punto de vista de los recursos, analizando si existen interbloqueos dados, desde el punto de vista No Funcional, intentando encontrar fuentes de bajo desempeño, relaciones redundantes, baja eficiencia, cuellos de botella.

El estudio global sobre los vínculos debe realizarse detalladamente, manteniendo consistencia en los resultados y buscando que el sistema en cuanto a sus vínculos sea lo más sencillo posible, pero sin descuidar ninguna de las funcionalidades y objetivos definidos y validados en pruebas anteriores.

2. Prueba y Validación de Comportamiento del Agente.

En esta prueba se valida que el comportamiento favorece el cumplimiento de los objetivos y es acorde con la estructura definida, así mismo se busca determinar que este comportamiento considere todas las posibles variaciones y aspectos que se deben tener en cuenta y así mismo tome una alternativa óptima para cada aspecto, en términos de la arquitectura del SMA que se ha definido y del cumplimiento de los objetivos definidos y validados en etapas anteriores.

Para ello se toman cuatro criterios (Compromiso entre agrupamiento y distribución, aprovechamiento de la comunicación, resolución de conflictos y aprovechamiento de sinergias y mecanismos de colaboración y asignación de tareas), los cuales evalúan el comportamiento de acuerdo a requerimientos Funcionales que deben tenerse en cuenta en un SMA, a su vez se crearon tres criterios nuevos (robustez, flexibilidad, y adaptabilidad y eficiencia) los cuales se encargan de evaluar requerimientos No Funcionales, analizando si logran mantener una alta disponibilidad en cualquier instante de tiempo, que logren estimar comportamientos futuros, o si tienen un manejo eficiente de los recursos.

La descripción de los objetivos de esta prueba, la realización de las mismas a través de los diferentes criterios, y los resultados se documentan en el reporte Validación del Comportamiento, los resultados se exponen en forma general y contrastando los diferentes requerimientos que se evaluaron en los criterios y manteniendo un compromiso de equilibrio entre ellos, teniendo en cuenta que algunos criterios son opuestos entre sí (Compromiso entre agrupamiento y distribución por ejemplo). Los resultados se valoran en términos del margen de error que manejan, que tan alejados están del cumplimiento de los objetivos del sistema y si este margen de error es solucionable en esta etapa o si es necesario devolverse a fases anteriores hasta encontrar el motivo de distorsión.

Etapas 4 (E4): Validación y Verificación de Agentes Finales.

Esta fase se ejecuta al final de la fase de implementación de PASSI; habiendo garantizado y probado los vínculos y garantizado el comportamiento de los agente para el Sistema diseñado, y habiendo realizado las actividades de la segunda etapa correspondiente al diseño de Agentes. El objetivo de esta etapa es garantizar que los agentes obtenidos como resultado de las tareas de detalle de roles e identificación de vínculos y estados, permitan el cumplimiento de los requerimientos funcionales y no funcionales definidos al inicio del proyecto, y que estos agentes representen de forma transparente todo el proceso de creación y depuración de los roles que lo componen sin que en estos se presente ninguna desviación respecto a lo establecido previamente. Para esto dentro de esta fase se ejecutan dos pruebas:

1. Prueba de Agentes.

Para esta prueba se llevan a cabo tres actividades, en la primera se debe garantizar que el rol cuente con eventos consistentes respecto a sus requerimientos y a los objetivos que debe cumplir. Así mismo, la respuesta definida para cada uno de los eventos del rol debe estar dentro de sus posibilidades, teniendo en cuenta las habilidades que posee y que además dichas respuestas sean acordes con los eventos definidos para los roles que se vinculan con el rol que se está analizando.

En la segunda actividad se debe analizar si para cada recurso se han definido los adaptadores necesarios para su correcta manipulación y si estos adaptadores son suficientes teniendo en cuenta la forma en la que estos recursos serán accedidos por los demás roles del sistema. Con esta información se tendrá claridad sobre que tan crítico es cada recurso, y como debe ser accedido para favorecer condiciones de baja carga de conflictos y coordinación adecuada.

En la tercera actividad se toman los agentes obtenidos como resultado del agrupamiento de

roles, revisando para cada uno de ellos que su complejidad total no sea ni alta ni tampoco baja, al igual que se debe garantizar que los roles agrupados no sean dispares entre sí en la medida en que todos tengan similares configuraciones de recursos, habilidades y objetivos. Esta evaluación se realiza con el fin de que los agentes finales sean específicos, detallados y no se presenten situaciones de excesiva dependencia de agentes.

2. Prueba de Sociedad.

Luego de haber llevado a cabo las etapas anteriores, se llega a la última prueba de la metodología. Esta se debe desarrollar luego de la etapa de despliegue de PASSI.

El objetivo principal de esta prueba es garantizar que las actividades realizadas en esta fase de PASSI sean consistentes con la arquitectura SMA que se ha construido y validado hasta el momento, así como que permitan el cumplimiento de los requerimientos y objetivos del SMA.

Se llevan a cabo tres tipos de pruebas, las cuales van creciendo en nivel de abstracción, de esta manera las pruebas empiezan por un nivel unitario, donde se prueban los agentes individualmente, probando que puedan cumplir las tareas que les han sido asignadas por medio de sus habilidades y recursos disponibles, luego se integran grupos de agentes para probar los vínculos de comunicación establecidos entre ellos, así como la estructura, y finalmente, se prueba el funcionamiento del sistema en conjunto, para probar que se cumplen los objetivos del Sistema a través de la configuración establecida.

Estas pruebas buscan asegurar que se cumplan los objetivos y requerimientos que han sido planteados para el SMA en etapas previas, a través de la ejecución correcta de las funcionalidades correspondientes al comportamiento de los agentes sobre el despliegue que ha sido realizado en la etapa final de PASSI.

La descripción de los objetivos de esta prueba, así como su realización y conclusiones son registradas en el artefacto Pruebas de Agentes Finales.

2.5 Procesos de Control.

Se crearon cuatro procesos de control que se ejecutan de forma iterativa y en paralelo con el seguimiento de las etapas descritas. Estos se fundamentaron con base en conceptos, en los que se define que toda metodología de pruebas es un proceso dinámico y paralelo al avance del proyecto en el cual no es suficiente contar con una serie de etapas secuenciales para garantizar que el proceso sea exitoso.

Por esto, además de las etapas secuenciales, hitos y artefactos que se han definido en esta metodología, es necesario tener procesos dinámicos que se ejecuten en paralelo a estas etapas y permitan no solo la actualización de la metodología de pruebas, sino el seguimiento de tareas que por su naturaleza no podrían encapsularse solamente en una fase.

Para este fin se identificaron cuatro procesos de control en la metodología de pruebas diseñada:

- ✓ Seguimiento al Plan de Pruebas.
- ✓ Trazabilidad de Objetivos vs. Habilidades/Recursos/Tareas.
- ✓ Validación de Vínculos y Comportamiento.
- ✓ Validación de Agentes Finales.

El objetivo principal del primer proceso es garantizar que el plan que ha sido definido para la metodología de pruebas sea llevado a cabo en su totalidad y sea seguido durante las diferentes etapas de la metodología. El objetivo del segundo proceso de control se basa en

su orientación a reforzar la validación de las nuevas tareas y roles identificados para el sistema en el fin de cada iteración de la Prueba de Validación de Roles, los cuales se esperan sigan garantizando la configuración conseguida para alcanzar el cumplimiento de los objetivos planteados. El tercer proceso nace en la fase de descomposición organizacional debido a que en esta fase se empiezan a identificar los primeros vínculos entre los roles que van surgiendo en cada una de las iteraciones de este proceso de descomposición, por lo tanto está orientado a empezar de forma temprana con el proceso de identificación de defectos en los vínculos que van surgiendo en esta fase entre los roles justo en el momento en que se definen. De este modo su corrección será más rápida, esto no excluye la obligatoria ejecución de la primera prueba en la tercera etapa de la metodología de pruebas, debido a que es aquí donde se realiza el proceso completo de validación de los vínculos. Finalmente el último proceso se crea con el objetivo de reforzar la validación de los agentes finales mediante cada una de las iteraciones realizadas a las pruebas propuestas para esta fase, dentro del modelo de código y del modelo de despliegue.

De esta forma el objetivo de estos procesos es mantener el control durante las diferentes etapas que se ejecutan, enfocado en el aseguramiento de la correcta recolección de los datos en las diferentes pruebas que han sido llevadas a cabo, verificando la no aparición de errores en los mismos, o encontrando discrepancias en ellos, para poder tomar medidas de acuerdo a los problemas encontrados.

2.6 MEDSIM- PASSI: Metodología resultante de la integración de las fases ya conocidas de PASSI, con la etapa de pruebas anteriormente descrita.

PASSI ha evolucionado de un largo período de construcción teórica y experimentos en el desarrollo de aplicaciones orientadas a agentes; posee importantes características como una arquitectura de visión flexible y un modelado extenso del dominio y de la ontología. En este epígrafe se describe la metodología que se obtuvo como resultado de la integración de las fases ya conocidas de PASSI, junto a la fase de pruebas propuesta en este trabajo. Es una

CAPÍTULO II: DISEÑO DE LA METODOLOGÍA DE PRUEBAS INTEGRADA A PASSI

metodología, paso por paso, que incluye todas las fases de la ingeniería del software, un aspecto vital y que antes de la realización de este trabajo, se adolecía de un diseño para la realización de la fase de pruebas. A continuación se describe "MEDSIM-PASSI".

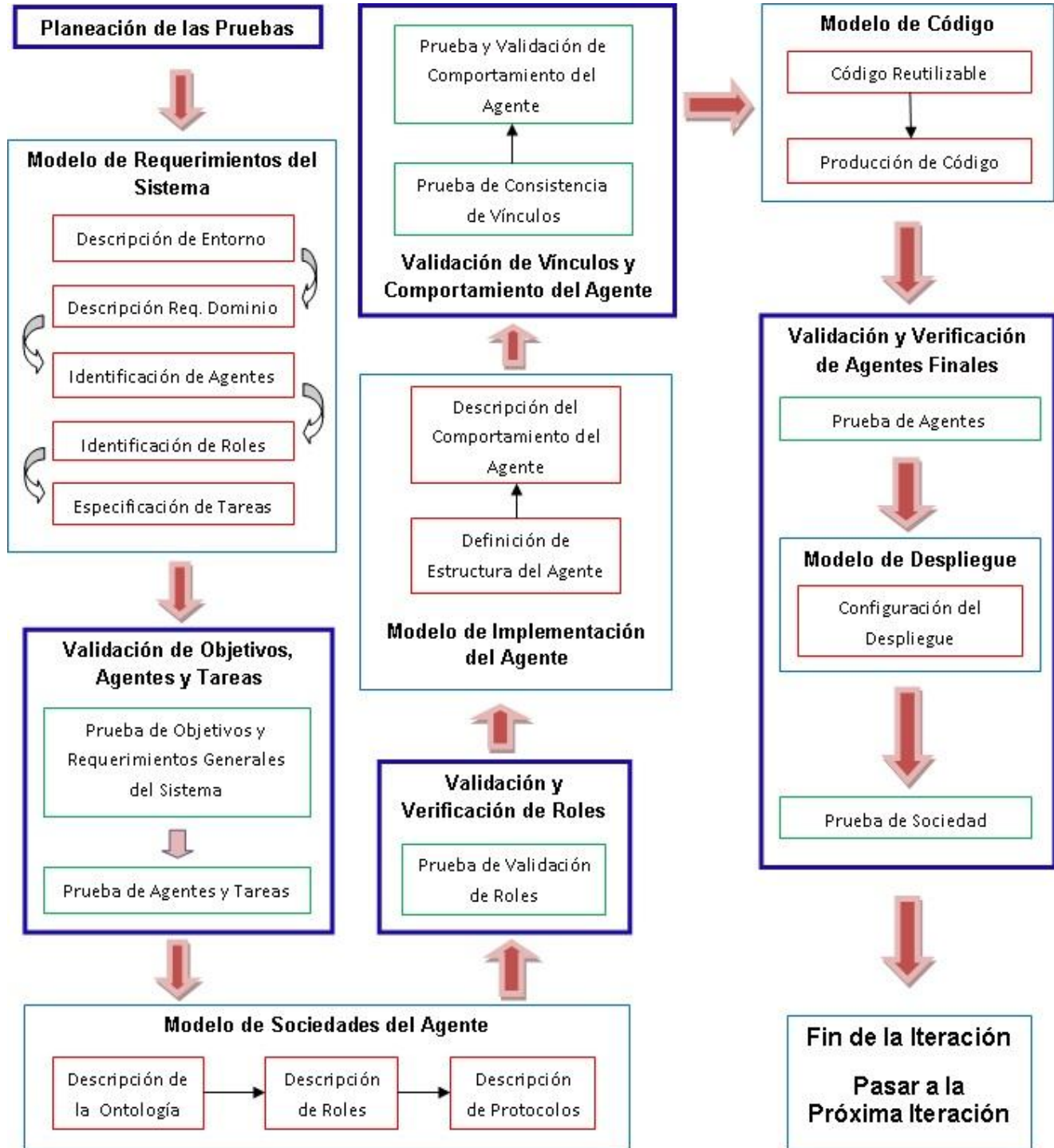


Figura 4. Fases de la Metodología "MEDSIM-PASSI".

Descripción de la Metodología "MEDSIM-PASSI".

Fase 1 Planeación de las Pruebas

Esta etapa puede considerarse como prerrequisito de entrada a la metodología, el objetivo de este proceso es definir con claridad aspectos como: ¿Qué distribución de recursos y tiempos se destinará para la fase de pruebas?, ¿Qué plan de actualización se llevará a cabo para los artefactos de prueba documentados a lo largo del proyecto, incluyendo el Plan de Pruebas?, ¿Qué configuración de la metodología de pruebas sugerida se aplicará para el proyecto en curso?, es decir ¿Qué etapas se tendrán en cuenta?, ¿Qué pruebas se aplicarán?, ¿Se aplicarán o no los procesos de control definidos en la metodología?. Así mismo, el objetivo de esta etapa es concluir con la generación del artefacto: Plan de Pruebas del Sistema, en el cual se consignará toda la información recogida en esta etapa, con el fin de estandarizar y formalizar el proceso de pruebas del proyecto que va iniciar. Los aspectos que deben tenerse en cuenta en este artefacto son:

1. Objetivos del proyecto y de las pruebas a efectuar.
2. Configuración de la metodología de pruebas que será utilizada en el proyecto a iniciar, teniendo en cuenta que no todas las pruebas son obligatorias.
3. Calendario y distribución en la ejecución de las etapas y pruebas seleccionadas.
4. Descripción de los hitos y artefactos de prueba dentro del proyecto.
5. Distribución de recursos del proyecto y asignación al seguimiento de las etapas de prueba.
6. Herramientas de prueba.

Este proceso de planeación es personalizado de acuerdo a las condiciones de complejidad, presupuesto y tiempo asociadas a cada proyecto, por esta causa se sugieren diversas configuraciones de ejecución con el fin de acoplarse a diferentes tamaños de proyecto, de modo que el tiempo invertido en la actividad de pruebas sea proporcional al tiempo global

exigido por el proyecto. En vista de que esta etapa está directamente relacionada con uno de los pilares de la metodología de pruebas y que es en este período de tiempo en el que se define el plan de estandarización del proceso de pruebas en el proyecto, la ejecución de esta etapa es de carácter obligatorio y debe realizarse en común acuerdo por todos los integrantes del equipo de proyecto, de forma paralela a actividades de PASSI como el estudio de factibilidad y el levantamiento de requerimientos del Sistema, actividades en las que también deben participar todos los asociados al proyecto de Agentes.

Fase 2 Modelo de Requerimientos del Sistema

2.1 Descripción de Entorno

La Descripción del Entorno constituye el primer modelo de PASSI, y a su vez proporciona una visión del sistema al máximo nivel de abstracción mediante un Diagrama de Casos de Uso utilizando la notación UML. Constituye el punto de partida de la descripción funcional del sistema. En el mismo se identifican los actores internos y externos del mismo, considerándolo como un todo único.

Amzi Prolog, uno de los actores identificados, constituye la interacción del sistema con un recurso externo, en este caso el Servidor Lógico de Prolog, que se encarga de las actividades de inferencia, o cualquier otra manipulación de las reglas y hechos de Prolog.³⁰

2.2 Descripción de Dominio

Esta sub fase es una descripción funcional de Casos de Uso del sistema usando los diagramas y estereotipos del Unified Model Language (UML). Aquí son descritos los requerimientos funcionales del sistema mediante una serie jerárquica de Diagramas de Caso

³⁰ Díaz Gómez, "Sistema Multiagente para MPEG."

de Uso. El diagrama obtenido en esta sub fase constituye el punto de partida de desarrollo del sistema en la identificación de los agentes involucrados en la solución. La herramienta PTK supone, en esta etapa, identificar a cada Caso de Uso con el nombre del agente responsable. Como consecuencia el siguiente diagrama (Diagrama Identificación de Agente) es generado de manera automática.

2.3 Identificación de Agentes

Según PASSI, es posible ver a un Agente como un Caso de Uso o un Paquete de Casos de Uso provenientes del diagrama generado en la sub fase anterior. Las funcionalidades son agrupadas según diferentes criterios que dan lugar a los agentes. De esta forma, cada paquete define las funcionalidades de un agente específico. Las relaciones entre los CU del mismo agente siguen la sintaxis usual y estereotipos de UML; pero las relaciones entre CU de diferentes agentes son estereotipadas como comunicación, dirigidas del iniciador hacia el participante. Este diagrama, organizado por paquetes, es automáticamente generado por el PTK, como se dijo anteriormente.

2.4 Identificación de Roles

En la Identificación de Roles se exploran las responsabilidades de cada agente y se identifican todos los roles posibles que pueden jugar. Esto se realiza a través de escenarios, que no son más que caminos, que definen las diferentes conductas del sistema. Los escenarios son descritos a través de Diagramas de Secuencia, donde un agente puede jugar uno o varios roles, y cada interacción representará una comunicación entre los agentes. A cada escenario identificado del SMA corresponde un diagrama, los cuales se corresponden al flujo normal de eventos del camino que se trate; suponiendo con ello la realización satisfactoria de las tareas involucradas. El alcance de la misma es modelar el ciclo de vida de

cada agente, identificando los roles que él puede jugar, las colaboraciones que necesita, y las comunicaciones en que participa.³¹

2.5 Especificación de Tareas

Esta sub fase especifica a través de Diagramas de Actividad las capacidades de cada agente. Su alcance es modelar el ciclo de vida de cada agente, reflejando las colaboraciones que necesita, y a las comunicaciones en que participa. Este diagrama se obtiene a partir del Diagrama Identificación de Roles explorando todos los escenarios en los que participa el agente. A cada agente le pertenece un diagrama con todas sus tareas desglosadas y relacionadas. El mismo contiene dos calles: la primera con una colección de actividades que simbolizan las tareas del propio agente, mientras que la segunda contiene actividades que representan a los otros agentes entrelazados.

Fase 3 Validación de Objetivos, Agentes y Tareas

En esta fase la metodología se enfoca en garantizar la trazabilidad y consistencia entre todos los elementos identificados y relacionados en la fase de caracterización del sistema de PASSI. Esta trazabilidad se define como: “Las relaciones establecidas entre los elementos tales como: requerimientos, objetivos, agentes y tareas deben permitir mapear en su totalidad los resultados esperados del SMA, respetando condiciones como la consistencia entre los objetivos y las habilidades necesarias para su cumplimiento, la correcta definición de tareas y la relación lógica entre las tareas asociadas con las habilidades de cada entidad”. En vista de que es complejo evaluar la total consistencia entre todos estos elementos en una prueba, en esta fase se sugiere la realización de las siguientes 2 pruebas:

3.1 Prueba de Objetivos y Requerimientos Generales del Sistema

³¹ Díaz Gómez, “Sistema Multiagente para DPEG.”

En esta prueba se busca garantizar en primer lugar la consistencia entre los requerimientos y los objetivos identificados para el SMA, para este fin se propone validar que cada uno de los requerimientos definidos haya sido considerado de forma explícita o implícita en al menos uno de los objetivos definidos para el SMA, esto con el fin de garantizar que ningún requerimiento se descarte en esta sub fase.

3.2 Prueba de Agentes y Tareas

En esta prueba se parte de los requerimientos y objetivos que ya fueron validados a través de la prueba anterior, respecto a estos objetivos se busca validar que para cada uno se haya definido la configuración adecuada de agentes, roles y tareas necesarias para su cumplimiento.

Esta prueba se realiza con el fin de garantizar que en el futuro no se presenten problemas relacionados con roles insuficientes para un requerimiento dado, o tareas que requieran agentes no definidos, entre otras situaciones.

Fase 4 Modelo de Sociedades del Agente

Esta fase modela las interacciones sociales y dependencias entre los agentes involucrados en la solución. Las sub fases de esta etapa son las siguientes:

4.1 Descripción de la Ontología

Esta sub fase describe el conocimiento atribuido a agentes individuales y a sus interacciones mediante el uso de Diagramas de Clase. Un sistema basado en agentes puede archivar sus reglas semánticas a través de ontologías explícitas, o terminologías y teorías específicas del

dominio. Para detallar la ontología resultante de la solución se introduce esta sub fase que posee dos sub-etapas.

4.1.1 Descripción de la Ontología de Dominio

Esta sub etapa describe la ontología del dominio que representa a las entidades a través de las clases. Del nivel de descripción obtenido en el Modelo de Requerimientos del Sistema se obtiene una serie de clases clasificadas en términos de conceptos, acciones y predicados, que representan todos los diversos conceptos con que interactúan los agentes identificados. Las acciones representan, a grandes rasgos, las principales funcionalidades de los conceptos; mientras que los predicados hacen referencia a expresiones alternativas, que definen posibles cursos y que tienen un papel relevante para la arquitectura.³²

4.1.2 Descripción de la Ontología de Comunicación

Esta sub etapa se enfoca en el conocimiento de los agentes y sus relaciones comunicativas. Para cada comunicación es imprescindible especificar tres elementos: ontología, lenguaje y protocolo de interacción. La ontología, al relacionarse estrictamente al problema, será definida como consecuencia de la propia aplicación. Permite la definición simultánea de dos elementos relacionados: el conocimiento del agente y su especificación de comunicaciones.

4.2 Descripción de Roles

Un Diagrama de Paquetes es el resultado de esta sub fase. En estas clases correspondientes a cada rol se introducen las tareas que el rol en sí debe efectuar.

³² Díaz Gómez, "Sistema Multiagente para DPEG."

De igual forma se especifican las dependencias entre los diversos roles, y se establecen las comunicaciones entre roles de diferentes agentes como se especifica en la Descripción de la Ontología de Comunicación. Parte del diagrama de esa sub etapa, es generado automáticamente por PTK, o sea, los paquetes que representan a los agentes, y los diversos roles que se identificaron en la fase Identificación de Roles.

4.3 Descripción de Protocolos

Esta sub fase especifica la gramática de cada protocolo de comunicación en términos de “actos del habla” mediante el uso de Diagramas de Secuencia.³³

Fase 5 Validación y Verificación de Roles

El seguimiento de esta fase así como la ejecución de la prueba básica que la compone son de carácter obligatorio, debido a que la prueba sugerida apunta a la verificación y validación de uno de los pilares de la metodología PASSI: El Modelo de Sociedades del Agente. En vista de que este proceso es iterativo y en cada iteración se busca descomponer un rol general en roles específicos y atómicos a partir de la descomposición de los objetivos del rol original, la prueba paralela a este proceso también debe ser de ejecución iterativa y su validación se enfoca a los puntos de fallos detectados en el interior de cada iteración. A continuación se describe la ejecución y lógica de esta prueba.

5.1 Prueba de Validación de Roles

Esta prueba se ejecuta de forma iterativa, de modo que en cada iteración del proceso de descomposición debe realizarse la prueba que se documenta en la presente sección. Allí la Tabla de Roles es un artefacto de PASSI el cual contiene los roles que se van obteniendo en

³³ Díaz Gómez, “Sistema Multiagente para MPEG.”

cada ciclo del modelo de sociedades.

Internamente la prueba revisa aspectos tales como la correcta interpretación de la complejidad del rol a descomponer, así como criterios discrecionales orientados a determinar el número de requerimientos, agentes y tareas implicados en cada rol y de este modo concluir si es necesario o no que sea descompuesto. Así mismo la prueba valida que la división de objetivos sea consistente de modo que la sumatoria del cumplimiento de los objetivos específicos definidos constituya el cumplimiento del objetivo padre y que para estos nuevos objetivos se tenga además la configuración adecuada de requerimientos, agentes y tareas necesarios para su cumplimiento.

Finalmente, la prueba valida que el agrupamiento de objetivos en roles sea óptimo de modo que se concluya con roles atómicos, especializados, independientes y de complejidad menor al rol descompuesto.

Fase 6 Modelo de Implementación del Agente

El modelo de implementación del agente de PASSI es un modelo clásico en términos de clases y métodos, pero a diferencia de los enfoques orientados a objetos, existen dos niveles de abstracción: el social y el individual. Se utilizan diagramas de clases y diagramas de transición de estados para modelar los agentes. Consta de dos sub fases, en las que se define la estructura del agente y se describe su comportamiento.

6.1 Definición de la Estructura del Agente

En esta sub fase se describe la estructura de las clases de agente mediante el uso de diagramas de clases convencionales generando dos diagramas:

6.1.1 Definición de Estructura del SMA

Esta sub etapa consta de un diagrama, el cual representa la estructura del SMA mediante un diagrama de Clases, donde cada una simboliza un agente. El conocimiento de cada agente puede representarse mediante compartimientos de atributos (como se vio en la Descripción de la Ontología), considerando el uso de compartimientos de operaciones para la identificación de las tareas del agente.

6.1.2 Definición de Estructura del Agente Simple

En esta sub etapa se ilustra la estructura interior del agente a través de las clases que lo constituyen, o sea, la clase principal del agente y las clases internas que identifican sus tareas. Se utiliza un Diagrama de Clase para cada agente, declarando los métodos y atributos del mismo, y las clases responsables de las tareas.³⁴

6.2 Descripción del Comportamiento del Agente

Esta sub fase consta de dos diagramas de clases subdivididos lógicamente, donde el correspondiente al SMA diseña el flujo de eventos por invocación de los métodos e intercambio de mensajes, mientras que el referente al agente en particular, especifica los métodos anteriores.

6.2.1 Descripción de Comportamiento del SMA

Se muestra el flujo de eventos entre y dentro de las clases de los agentes principales y sus clases internas (representando sus tareas) mediante uno o más Diagramas de Actividad. Se debe diseñar una calle para cada agente y para cada tarea. Las actividades dentro de la calle

³⁴ Díaz Gómez, "Sistema Multiagente para DPEG."

indican los métodos de la clase relacionada. Se establecen transiciones usuales de la norma de UML para significar cualquier evento o invocación de métodos.

6.2.2 Descripción de Comportamiento del Agente Simple

En esta sub etapa se incorpora la implementación de los métodos introducidos en los Diagramas de Definición de Estructura del Agente. La manera de describirlo es libre; o sea, la más apropiada a consideración del diseñador.

Fase 7 Validación de Vínculos y Comportamiento del Agente

En esta fase se busca validar los vínculos identificados en la etapa de diseño de PASSI. Esta validación se lleva a cabo a través de dos pruebas, la primera valida los vínculos y la segunda prueba valida el comportamiento del agente, basándose en los requerimientos del sistema y en la estructura que actualmente se lleva del mismo.

7.1 Prueba de Consistencia de Vínculos

En esta prueba se realizan tres actividades.

En la primera se toman los vínculos definidos para el SMA para validar su naturaleza (Simple colaboración, colaboración coordinada, independencia), analizando que roles relaciona dicho vínculo para estudiar si la relación que deben tener es acorde con base a la configuración de requerimientos y roles, luego se elabora una lista de las implicaciones de ese vínculo tanto en procesamiento como en comportamiento esperado.

En la segunda actividad se listan todos los recursos que por los vínculos hallados pueden

implicar conflictos futuros o sugieren una necesidad de coordinación para su acceso, respecto a esta lista se revisa que estos recursos ya existan en la tabla de recursos del Sistema y que su descripción haya contemplado las necesidades de coordinación y conflictos que evidenciaron los vínculos encontrados.

La tercera actividad consiste en revisar el diagrama de roles (generado al ejecutar PASSI) y vínculos del SMA como un todo, intentando encontrar patrones anormales como interbloqueos, ciclos innecesarios, roles incomunicados.

7.2 Prueba y Validación de Comportamiento del Agente

En esta prueba se valida que el comportamiento favorece el cumplimiento de los objetivos y es acorde con la estructura definida. Así mismo se busca determinar que este comportamiento considere todas las posibles variaciones y aspectos que se deben tener en cuenta y así mismo tome una alternativa óptima para cada aspecto, en términos de la arquitectura del SMA que se ha definido y del cumplimiento de los objetivos definidos y validados en fases anteriores.

Para ello se toman cuatro criterios (Compromiso entre agrupamiento y distribución, aprovechamiento de la comunicación, resolución de conflictos y aprovechamiento de sinergias y mecanismos de colaboración y asignación de tareas), los cuales evalúan el comportamiento de acuerdo a requerimientos funcionales que deben tenerse en cuenta en un SMA, a su vez se crearon tres criterios nuevos (robustez, flexibilidad, y adaptabilidad y eficiencia) los cuales se encargan de evaluar requerimientos no funcionales, analizando si logran mantener una alta disponibilidad en cualquier instante de tiempo, que logren estimar comportamientos futuros, o si tienen un manejo eficiente de los recursos.

Fase 8 Modelo de Código

El modelo de código de PASSI es un modelo de la solución a nivel de código. Involucra generación de código a partir de modelos y completa manualmente el código fuente. Consta de dos sub fases, la primera es una biblioteca de código reutilizable y ya en la segunda se perfecciona el código básico.

8.1 Biblioteca de Código Reutilizable

En esta sub fase se trata de rehusar patrones existentes de agentes y tareas. Aquí se realiza la implementación de las interfaces de las clases de cada uno de los agentes identificados. La herramienta realiza la generación del código, permitiendo redefinir los métodos necesarios.

8.2 Perfeccionamiento del Código Básico

En esta sub fase, más bien convencional, se realiza la modificación del código fuente.³⁵

Fase 9 Validación y Verificación de Agentes Finales

Esta fase se ejecuta al final de la fase de implementación de PASSI; habiendo garantizado y probado los vínculos y garantizado el comportamiento de los agentes para el sistema diseñado, y habiendo realizado las actividades de la segunda etapa correspondiente al diseño de agentes. El objetivo de esta etapa es garantizar que los agentes obtenidos como resultado de las tareas de detalle de roles e identificación de vínculos y estados, permitan el cumplimiento de los requerimientos funcionales y no funcionales definidos al inicio del proyecto, y que estos agentes representen de forma transparente todo el proceso de creación y depuración de los roles que lo componen sin que en estos se presente ninguna

³⁵ Díaz Gómez, "Sistema Multiagente para DPEG."

desviación respecto a lo establecido previamente. Para esto dentro de esta fase se ejecutan dos pruebas. La primera es una Prueba de Agentes y la segunda es una Prueba de Sociedad. Esta última se realiza después de haber realizado el Modelo de Despliegue y su configuración, por lo que será detallada más adelante.

9.1 Prueba de Agentes

Para esta prueba se llevan a cabo tres actividades, en la primera se debe garantizar que el rol cuente con eventos consistentes respecto a sus requerimientos y a los objetivos que debe cumplir. Así mismo, la respuesta definida para cada uno de los eventos del rol debe estar dentro de sus posibilidades, teniendo en cuenta las habilidades que posee y que además dichas respuestas sean acordes con los eventos definidos para los roles que se vinculan con el rol que se está analizando.

En la segunda actividad se debe analizar si para cada recurso se han definido los adaptadores necesarios para su correcta manipulación y si estos adaptadores son suficientes teniendo en cuenta la forma en la que estos recursos serán accedidos por los demás roles del sistema.

En la tercera actividad se toman los agentes obtenidos como resultado del agrupamiento de roles, revisando para cada uno de ellos que su complejidad total no sea ni alta ni tampoco baja, al igual que se debe garantizar que los roles agrupados no sean dispares entre sí en la medida en que todos tengan similares configuraciones de recursos, habilidades y objetivos. Esta evaluación se realiza con el fin de que los agentes finales sean específicos, detallados y no se presenten situaciones de excesiva dependencia de agentes.

Fase 10 Modelo de Despliegue

Esta fase constituye un modelo de la distribución de las partes del sistema, a través de unidades de procesamiento de hardware; y su migración entre las mismas. Consta de una única sub fase.

10.1 Configuración del Despliegue

Esta sub fase describe la asignación de agentes a las unidades físicas de procesamiento disponibles y cualquier restricción en la migración y movilidad mediante el uso de Diagramas de Despliegue. Esta sub fase es la respuesta a la necesidad de detallar la posición de los agentes en sistemas distribuidos o en contextos de agentes móviles. Aquí se describen donde se localizan los agentes y qué unidades de procesamiento necesitan para la comunicación entre sí. En este diagrama es posible especificar los dispositivos de hardware usados por los agentes (sensores y efectores), y los modos de comunicación entre agentes en unidades de procesamiento diferentes (por ejemplo, redes por cable o inalámbricas).³⁶

9.2 Prueba de Sociedad

Luego de haber llevado a cabo las fases anteriores, se llega a la última prueba de la metodología. El objetivo principal de esta prueba es garantizar que las actividades realizadas en esta fase de PASSI sean consistentes con la arquitectura SMA que se ha construido y validado hasta el momento, así como que permitan el cumplimiento de los requerimientos y objetivos del SMA.

Se llevan a cabo tres tipos de pruebas, las cuales van creciendo en nivel de abstracción, de

³⁶ Díaz Gómez, "Sistema Multiagente para DPEG."

esta manera las pruebas empiezan por un nivel unitario, donde se prueban los agentes individualmente, probando que puedan cumplir las tareas que les han sido asignadas por medio de sus habilidades y recursos disponibles, luego se integran grupos de agentes para probar los vínculos de comunicación establecidos entre ellos, así como la estructura, y finalmente, se prueba el funcionamiento del sistema en conjunto, para probar que se cumplen los objetivos del sistema a través de la configuración establecida.

Estas pruebas buscan asegurar que se cumplan los objetivos y requerimientos que han sido planteados para el SMA en etapas previas, a través de la ejecución correcta de las funcionalidades correspondientes al comportamiento de los agentes sobre el despliegue que ha sido realizado en la fase final de PASSI.

2.7 Conclusiones del Capítulo.

En el desarrollo de este capítulo se realizó el diseño de la metodología de pruebas para sistemas multiagentes integrados a PASSI. Se obtuvo una detallada descripción en las fases de análisis, diseño e implementación a un significativo nivel de detalle, con la especificación de las pruebas propuestas para cada etapa. Se propuso la realización de siete pruebas, todas de carácter iterativo, siendo realizadas paralelamente al desarrollo del sistema, y además se propuso la realización de cuatro procesos de control, para ir verificando en cada etapa la correcta realización de estas pruebas. Se realizó una integración de la metodología de pruebas con las fases ya conocidas de PASSI y se obtuvo como resultado "**MEDSIM-PASSI**", una metodología más completa ya que abarca las cuatro fases de la ingeniería de software, contribuyendo así a la modelación de sistemas con un mayor nivel de calidad y corrección de sus procesos. Finalmente la metodología propuesta ha cumplido con los objetivos esperados satisfactoriamente.

CAPÍTULO III:
"Validación de la Metodología
MEDSIM-PASSI. "

CAPÍTULO III: "Validación de la Metodología MEDSIM-PASSI".

3.1 Introducción.

En el presente capítulo describiremos aspectos relacionados a la validación de la metodología que se obtuvo como resultado de la integración de fases ya conocidas de PASSI, con la metodología de pruebas propuesta en el capítulo anterior. Para realizar dicho estudio, aplicamos la metodología a un Sistema Multiagente integrado a PASSI, titulado "Diagnóstico Presuntivo de Enfermedades Ginecológicas (DPEG) ".

Con el objetivo principal de detectar la mayor cantidad de defectos, expondremos los resultados de la aplicación, mediante un minucioso análisis fase por fase de la metodología diseñada. Además de esto, describiremos algunas de las ventajas y desventajas que surgieron después de la aplicación de la misma, y que nos muestran una visión más amplia de cuánto se puede mejorar la calidad de los sistemas, que en un futuro utilicen la metodología "MEDSIM-PASSI".

3.2 Validación de la metodología "MEDSIM-PASSI" fase por fase.

3.2.1 Planeación de las Pruebas.

La metodología cuenta inicialmente con una fase de planeación de las pruebas, considerada como requisito de entrada, donde quedan definidos entre otros aspectos, la configuración de las pruebas a utilizar, distribución de recursos, etapas que se tendrán en cuenta, y la realización o no de los procesos de control. Se considera por ser la etapa inicial, de gran importancia su ejecución, pues de ella depende en gran medida el éxito en los resultados finales de un proyecto. Decir como dato adicional que en la realización de esta etapa, deben

participar todos los integrantes asociados al proyecto de agentes, en común acuerdo entre ellos.

Como resultado de esta etapa, queda generado el artefacto "Plan de Pruebas del Sistema". Para el caso de DPEG, se destinaron quince días a razón de ocho horas diarias, que dio como resultado un total de 120 horas para la realización de la fase de pruebas, un tiempo acorde con las necesidades del proyecto, con el fin de realizar un profundo análisis en cada una de las etapas. Respecto al plan de actualización, decir que no se utilizó ninguno pues se utilizó la metodología anteriormente propuesta, siendo la primera vez que se le aplica a un sistema de este tipo.

Pasando a la configuración de las pruebas utilizadas, decir que en DPEG se utilizó una configuración que abarca todas las etapas de la metodología anteriormente propuesta, aplicando además los procesos de control, puesto que al ser la primera vez que se utiliza, se necesitaba de una detallada aplicación de la misma, con el fin de obtener los resultados correspondientes en cada una de las etapas, contribuyendo de esta forma a una excelente estandarización del proceso de pruebas del proyecto en cuestión.

3.2.2 Modelo de Requerimientos del Sistema.

Posterior a la etapa de planeación, se entra en la fase de requerimientos, la cual consta de cinco sub fases:

1. Descripción de Entorno.
2. Descripción de Dominio.
3. Identificación de Agentes.
4. Identificación de Roles.

5. Especificación de Tareas.

En cada una de estas sub fases, se generan diferentes diagramas, los cuales nos proporcionan una visión más detallada del sistema.

Posterior a la realización de esta etapa, se pasa a la etapa de Validación de Objetivos, Agentes y Tareas. Es en esta etapa donde se verifica la correcta realización de las sub fases realizadas en la fase de requerimientos.

3.2.3 Validación de Objetivos, Agentes y Tareas.

Como se explicó anteriormente, es en esta etapa donde se verifica si las sub fases que integran el modelo de requerimientos del sistema fueron realizadas correctamente. Se verifican aspectos tales como la interfaz de inicio con todos los rasgos de entrada de DPEG, agentes con sus roles correspondientes y capacidad de los roles a realizar las tareas que le fueron asignadas, entre otros.

Dicha verificación se realiza mediante la ejecución de dos pruebas. La primera de ellas es la Prueba de Objetivos y Requerimientos Generales del Sistema y la segunda es la Prueba de Agentes y Tareas.

Después de haber realizado la validación de la etapa de requerimientos mediante la ejecución de las pruebas definidas anteriormente, se constató que la mayoría de los errores se encuentran en la interfaz de inicio y rasgos de entrada de DPEG, no siendo así en la identificación de agentes y roles, al contar cada uno de los agentes con los roles necesarios, y estos a su vez con sus tareas correspondientes correctamente definidas. Además se verificó que se cumpliera con los requerimientos iniciales del sistema, y esto no tuvo dificultad alguna puesto que el principal objetivo de DPEG es realizar el diagnóstico de las

CAPÍTULO III: VALIDACIÓN DE LA METODOLOGÍA "MEDSIM-PASSI"

patologías ginecológicas, y al ser probado con datos reales, muestra el diagnóstico de la paciente, diagnóstico presuntivo y exámenes complementarios que se debe realizar.

A continuación enumeramos los principales problemas encontrados en la interfaz de inicio y en los rasgos de entrada de DPEG:

- ✓ En "Datos de la Paciente", los campos de Edad y Número de Historia Clínica, debieron estar diseñados para aceptar solo números, no siendo así pues los mismos, permiten la entrada de letras.
- ✓ También en "Datos de la Paciente", pero en Dirección Particular, el campo destinado a la Calle, se debió diseñar para que aceptara tanto letras como números, no siendo así al no permitir la entrada de estos últimos.

The screenshot displays the 'DPEG' application window. On the left is a sidebar menu with categories like 'Datos de la Paciente', 'Motivos de la Consulta', and 'Examen Ginecológico'. The main area is titled 'Datos Generales De La Paciente' and contains several input sections. The 'Datos de la Paciente' section includes fields for 'Nombre(s)', 'Apellidos', 'Edad' (with 'aa' entered), 'Estado Civil', and 'Raza'. The 'Fecha de la Consulta' is set to '22 de junio de 2011'. The 'No Historia Clínica' section has a field with 'bbbbbb' entered. A 'Dirección Particular' section includes fields for 'Calle', '#', 'Reparto', and 'Ciudad'. A circular icon of a person is visible on the right side of the form.

Figura 5. Interfaz de Datos Generales de la Paciente.

- ✓ En "Síntomas y Signos", específicamente en los Generales, constatamos que al tratar de escribir datos en los campos de Presión Arterial y Frecuencia Cardíaca, se permite la entrada de letras, algo que no debe ser, puesto que esos valores solo pueden ser números.

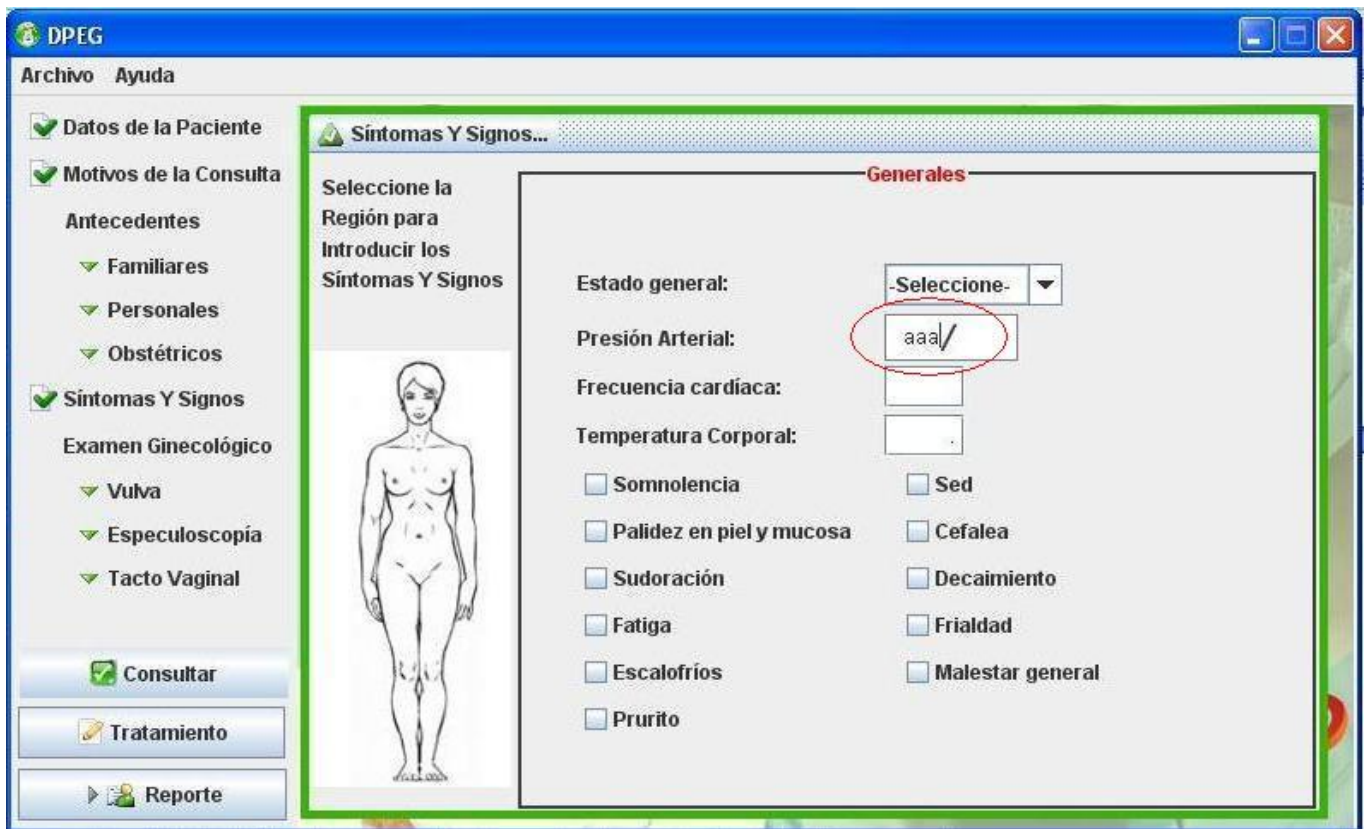


Figura 6. Interfaz de Síntomas y Signos.

- ✓ En los "Antecedentes Obstétricos", específicamente en los Partos, en el campo Cantidad de Partos, cuando la paciente no ha tenido ninguno y hay que poner un cero, no debe activarse el campo de Fecha del último Parto, puesto que si no ha parido, no hay un último parto. En este caso no sucede así, permitiendo además en caso de

CAPÍTULO III: VALIDACIÓN DE LA METODOLOGÍA "MEDSIM-PASSI"

tener partos, la entrada de un número mayor en el campo de Cantidad de Partos Eutócicos, que en el destinado a la Cantidad de Partos en general.

- ✓ Igualmente en los "Antecedentes Obstétricos", pero ahora pasando a los Abortos, en el campo Cantidad de Abortos, cuando la paciente no ha tenido ninguno y hay que poner un cero, no debe activarse el campo de Fecha del último Aborto, puesto que si no ha tenido nunca ningún aborto, no hay un último aborto. En este caso no sucede así, permitiendo además en caso de tener algún aborto, la entrada de un número mayor en el campo de Cantidad de Abortos Espontáneos, que en el destinado a la Cantidad de Abortos en general.

The screenshot shows the 'Antecedentes Obstétricos' window in the DPEG application. The left sidebar contains a tree view with categories like 'Datos de la Paciente', 'Motivos de la Consulta', 'Antecedentes' (with sub-items: Familiares, Personales, Obstétricos), 'Síntomas Y Signos', and 'Examen Ginecológico' (with sub-items: Vulva, Especuloscopia, Tacto Vaginal). At the bottom of the sidebar are buttons for 'Consultar', 'Tratamiento', and 'Reporte'. The main area of the window is titled 'Antecedentes Obstétricos...' and contains two sections: 'PARTOS' and 'ABORTOS'. The 'PARTOS' section has fields for 'Cantidad de Partos' (0), 'Cantidad de Partos Eutócicos' (5), and 'Fecha del último Parto' (22 de junio de 2011). Below these are checkboxes for 'Técnicas usadas en los partos distócicos': 'Fórceps', 'Cesarea', and 'Espátulas'. The 'ABORTOS' section has fields for 'Cantidad de Abortos' (1), 'Cantidad de Abortos espontáneos' (3), and 'Fecha de último Aborto' (22 de junio de 2011). Below these are checkboxes for 'Técnicas Usadas en los Abortos': 'Legrado', 'Regulación menstrual', 'Complicaciones post interrupción de embarazo', 'Medicamentoso', and 'Método de Rivanol'. A red circle highlights the 'Cantidad de Partos' field, and another red circle highlights the 'Cantidad de Abortos' field. A red arrow points from the 'Cantidad de Partos Eutócicos' field to the 'Fecha del último Parto' field.

Figura 7. Interfaz de Antecedentes Obstétricos.

- ✓ La aplicación carece de una ayuda y un glosario de términos, con todos los aspectos correspondientes, que les brinde facilidades a los usuarios. Al hacer clic en el apartado

Ayuda que existe en la barra de Menú del sistema, solo aparece una ventana con el título del sistema multiagente diseñado.



Figura 8. Interfaz de Ayuda.

Como se dijo anteriormente en las sub fases de identificación de agentes y roles, así como de especificación de tareas no hubo grandes contratiempos. Respecto a la fase de requerimientos, se deben de resolver las deficiencias encontradas en la misma, antes de pasar a la realización de la siguiente fase, para evitar arrastrar errores a lo largo del desarrollo del proyecto.

Después de haber corregido los errores encontrados, se pasa a la siguiente fase de la metodología que consiste en el Modelo de Sociedades del Agente.

3.2.4 Modelo de Sociedades del Agente.

En la presente etapa se realiza la descripción de ontología, de roles y de protocolos. Una descripción de conocimientos del agente y sus interacciones, descripción de clases

correspondientes a cada rol con sus tareas y la gramática de los protocolos de comunicación, son los aspectos que se tienen en cuenta para la realización de los diferentes diagramas.

Posterior a la realización de estas sub fases, se pasa a la realización de la siguiente etapa, de suma importancia, al apuntar la Prueba de Validación de Roles a la verificación de uno de los pilares de la metodología.

3.2.5 Validación y Verificación de Roles.

Es en la presente etapa, donde se realiza el proceso de descomposición de roles en roles específicos y atómicos. Como se dijo en el capítulo anterior, la ejecución de la prueba debe ser iterativa, y se enfoca a los puntos de fallo detectados en el interior de cada una de las iteraciones.

Se agruparon los objetivos del sistema en roles, siendo los roles especializados, menos complejos que el rol que fue descompuesto. En el Diagrama de Descripción de Roles (Figura 9), se definen los agentes, se detallan explícitamente todos los roles generales, así como los roles que surgieron del resultado de la descomposición en roles especializados e independientes, siendo estos consistentes al incluir todas las tareas que deben efectuar.

La presente etapa debe concluirse con un óptimo agrupamiento de objetivos en roles, siendo estos últimos, atómicos, especializados, independientes y con menor complejidad que el que fuera descompuesto, como se describió en el capítulo anterior de este trabajo.

Una vez probado y garantizado que el modelo de sociedades del agente ha conducido a roles atómicos y consistentes con definiciones claras de objetivos y tareas con respecto a los requerimientos del sistema, se pasa a la etapa de implementación.

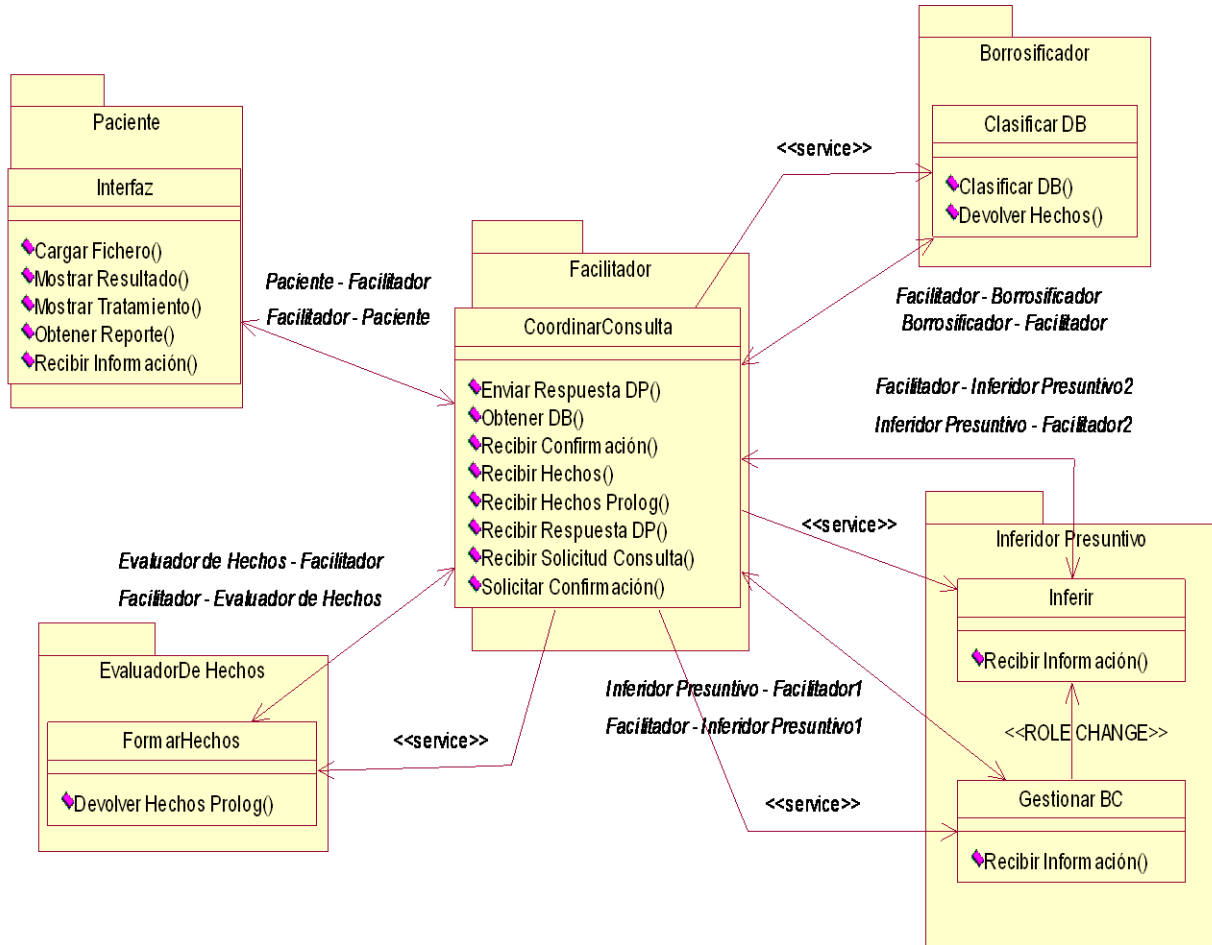


Figura 9. Diagrama de Descripción de Roles.

3.2.6 Modelo de Implementación del Agente.

La presente etapa constituye el modelo de implementación del agente, que consta de dos sub fases. En la primera de estas, es donde se define la estructura del agente y en la segunda, se realiza la descripción de comportamiento del agente. Para ambos casos, se realizan diagramas de descripción de forma independiente, uno para el agente y otro para el SMA.

Como fue explicado con mayor detalle anteriormente, en los diagramas correspondientes a la estructura, mediante diagramas de clases se realiza la representación del SMA como un todo y por otro lado se representa el agente simple, ilustrando su estructura interior a través de las clases que lo constituyen. Ya en los diagramas correspondientes al comportamiento, en el correspondiente al SMA se muestra el flujo de eventos entre y dentro de las clases de los agentes principales y sus clases internas, y en el que corresponde al agente simple se incorpora la implementación de los métodos introducidos en los Diagramas de Definición de Estructura del Agente, siendo libre su descripción.

Después de concluir la realización de estos diagramas, se pasa a la siguiente etapa que es donde se verifica la excelencia de dichos diagramas. Mediante la Validación de Vínculos y Comportamiento del Agente, y a través de la realización de dos pruebas, se realiza una detallada verificación y se asegura con esto el correcto desarrollo de la metodología.

3.2.7 Validación de Vínculos y Comportamiento del Agente.

Como fuera descrito en el capítulo anterior, la realización de esta fase se realiza mediante la ejecución de dos pruebas, después de haber realizado el modelo de implementación. Apoyándose en los requerimientos y la estructura del sistema, se realiza la Prueba de Consistencia de Vínculos y posteriormente la Prueba y Validación de Comportamiento del Agente.

Mediante la realización de estas dos pruebas, se realiza la verificación de la etapa anterior. Fue analizado el modelo de implementación de DPEG, y se pudo constatar que los vínculos definidos para el SMA son de colaboración coordinada. No se encontraron patrones anormales que puedan bloquear el desarrollo del sistema. Los roles que fueron analizados (Interfaz, Coordinar Consulta, Clasificar DB, Formar Hechos, Inferir, Gestionar BC), son suficientes para dar cumplimiento a los objetivos definidos para el SMA, con un correcto

acoplamiento entre ellos. Decir también que el comportamiento de los agentes (Paciente, Facilitador, Borrosificador, Evaluador de Hechos, Inferidor Presuntivo), que involucran los roles antes mencionados, favorece el cumplimiento del objetivo principal del sistema (Emitir un Diagnóstico Presuntivo).

Podemos concluir que para esta etapa en específico, según se pudo notar mediante la validación de la etapa, se realizó un excelente trabajo en lo que a hechos se refiere, con un explícito desarrollo interno de los procesos, con un manejo eficiente de recursos, que no permite la detección de patrones anormales, de interbloqueos, o de otros aspectos que no contribuyan a un correcto funcionamiento del sistema.

Posterior a la realización de la validación del modelo de implementación, se pasa al modelo de código. Constituye un requisito importante para el paso a la etapa siguiente, la realización de las pruebas definidas en esta etapa, y que ambas proporcionen los resultados que de ellas se esperan.

3.2.8 Modelo de Código.

Después de haber hecho el modelo de implementación y realizado su validación, entramos en el modelo de código. Como fuera detallado anteriormente, este modelo involucra la generación de código y completa el código fuente de forma manual. Una biblioteca de código reutilizable donde se implementan las interfaces de las clases de cada uno de los agentes identificados, permitiendo además la redefinición de los métodos que se necesiten, y una modificación de código fuente, esto último de manera convencional, son las dos sub fases que se realizan en esta etapa.

Después de la realización de dicha etapa, se pasa a realizar su validación mediante la Validación y Verificación de Agentes Finales.

3.2.9 Validación y Verificación de Agentes Finales.

Esta fase fue ampliamente detallada en el capítulo anterior. Se ejecuta posterior a la realización del modelo de código y su objetivo es garantizar que los agentes obtenidos como resultado de las tareas de detalle de roles e identificación de vínculos y estados, permitan el cumplimiento de los requerimientos funcionales y no funcionales definidos al inicio del proyecto. Estos agentes a su vez, deben representar la creación y depuración de roles sin presentar ninguna desviación respecto a lo anteriormente establecido. Consta de la ejecución de dos pruebas, realizando la segunda de ellas, posterior a la realización del Modelo de Despliegue.

Analizando el sistema como un todo, ya debe llegarse a esta etapa con la mayor cantidad de deficiencias resueltas, puesto que la metodología cuenta con procesos de control que permiten la ejecución de pruebas para cada una de las etapas, y como requisito de avance, es que cada una de esas pruebas, den solución local a cada uno de los problemas que se van encontrando. Así y después del modelo de requerimientos, se encontraron deficiencias y hasta no quedar completamente resueltas, no se procede a la realización del modelo de sociedades. Ya al haber concluido este, se procede a la ejecución de las pruebas para la etapa, y tal y como en las pruebas anteriores, es preciso corregirlas antes de continuar. Concluyendo con dicha etapa de validación, se pasa a la realización del modelo de implementación, y con este a la realización de las pruebas correspondientes. Posteriormente se entra en el modelo de código y concluyendo el mismo, se pasa a la ejecución de las pruebas correspondientes para la etapa. Dichas pruebas constituyen en primera instancia, la presente prueba, Prueba de Agentes.

Ya al llegar a dicha Prueba de Agentes, la metodología ha pasado por ocho fases con la ejecución de cinco pruebas, las cuales en su momento resolvieron los problemas que se fueron encontrando. Aquí se procede ahora a la validación de los agentes del sistema como

CAPÍTULO III: VALIDACIÓN DE LA METODOLOGÍA "MEDSIM-PASSI"

un todo. Para un mejor entendimiento del proceso, describimos un resumen del mismo a continuación:

1. El Rol RInterfaz (Agte. Interfaz Paciente), proporciona la comunicación entre usuario y sistema. Sus principales actividades son mostrar, recepcionar y validar información.
2. El Rol RCoordinar Consulta (Agte. Facilitador), coordina la consulta dentro del sistema, solicitando las tareas necesarias para completar un Diagnóstico Presuntivo. Sus principales actividades son solicitar la clasificación de los datos borrosos, la evaluación de los hechos prolog, la gestión de la base de conocimiento, así como ordenar la inferencia de los datos.
3. El Rol RFormar Hechos (Agte. Evaluador De Hechos), tiene como objetivo obtener los hechos en prolog a través de la evaluación de cada uno de los rasgos de entrada Su actividad principal es formar hechos prolog.
4. El Rol RClasificar DB (Agte. Borrosificador), clasifica los datos borrosos que existan como consecuencia de la entrada de datos iniciales del sistema, utilizando redes neuronales. Sus principales actividades son entrenar las redes y clasificar dichos datos borrosos.
5. El Rol RGestionar BC (Agte. Inferidor Presuntivo), tiene como objetivo principal la gestión de la base de conocimientos almacenada, realizando actividades como inserción de datos, actualización, entre otras.
6. El Rol RInferir (Agte. Inferidor Presuntivo), constituye uno de los de mayor importancia, pues es el que se encarga del proceso de inferencia del sistema, para devolver a la paciente un diagnóstico presuntivo. Sus principales actividades son inferir y devolver los datos a la paciente.

Después de haber analizado en profundidad cada uno de estos procesos, y el funcionamiento del sistema como un todo, no se detectaron grandes problemas. Esto sucede debido a que en cada una de las etapas anteriores en las que se ejecutan pruebas, se

detectaron y corrigieron problemas, y no se pasó a la etapa siguiente hasta dichos problemas no se les diera una solución. Así mismo se verificó que los agentes finales fuesen específicos y detallados, con el fin de evitar una excesiva dependencia entre ellos.

Habiendo concluido con la Prueba de Agentes, se pasa a la realización del Modelo de Despliegue. Dicha etapa constituye la última en la metodología, aunque después de la realización de la misma, se procede a su verificación mediante una Prueba de Sociedad, prueba que como ya se había aclarado, pertenece a la etapa anterior.

3.2.10 Modelo de Despliegue.

Esta fase constituye un modelo de la distribución de las partes del sistema, a través de unidades de procesamiento de hardware; y su migración entre las mismas. La configuración del despliegue constituye la única sub fase de esta etapa, donde de forma general se describe la asignación de agentes a las unidades físicas de procesamiento disponibles y cualquier restricción en la migración. Esta fase fue detallada en el capítulo anterior.

Luego de haber llevado a cabo las fases anteriores, se llega a la última prueba de la metodología. El objetivo principal de esta prueba es garantizar que las actividades realizadas en esta fase sean consistentes con la arquitectura SMA que se ha construido y validado hasta el momento, así como que permitan el cumplimiento de los requerimientos y objetivos del SMA.

Después de la ejecución de esta prueba, donde se vio a DPEG como un sistema de forma general, se culmina con la realización de las diferentes etapas de la metodología.

3.3 Ventajas y desventajas de la aplicación de la metodología MEDSIM-PASSI.

En este apéndice, trataremos brevemente algunas de las ventajas y desventajas que surgen con la aplicación de la metodología en los sistemas multiagentes.

Constituye la principal ventaja de la metodología, que se le haya incorporado a sus procesos, una guía metodológica correctamente definida para la ejecución de la fase de pruebas. Esto permitirá a partir de ahora, con un mayor nivel de detalle en sus procesos, el diseño de sistemas multiagentes de gran calidad, al completar las cuatro fases de la ingeniería del software.

Otra de las ventajas es que aún cuando la aplicación de esta metodología en el diseño de estos sistemas, aumente el tiempo de realización de los mismos, se obtendrán sistemas con mejor calidad y excelencia. Esto estará dado por los grandes beneficios de contar con una fase de pruebas detallada de forma explícita.

Ya pasando al tema de las desventajas, la principal desventaja para esta metodología, constituye que debido a que el tiempo con el que se contaba para su diseño y aplicación fuese poco para lo que se necesitaba, no se pudo lograr para DPEG un análisis con una mayor profundidad en las clases definidas en el modelo de implementación.

3.4 Conclusiones del Capítulo.

En el desarrollo de este capítulo se realizó el proceso de validación de la metodología de desarrollo de sistemas multiagentes que fuese diseñada en el capítulo anterior. Se obtuvo una detallada descripción de los errores que fueron detectados en cada una de las fases de análisis, diseño e implementación con excelencia en el desarrollo de las pruebas. Finalmente se explicaron de forma general, ventajas y desventajas de la aplicación de dicha metodología

CAPÍTULO III: VALIDACIÓN DE LA METODOLOGÍA "MEDSIM-PASSI"

en el SMA titulado "Diagnóstico Presuntivo de Enfermedades Ginecológicas". Es preciso aclarar que si se encontraron pocas deficiencias en dicho sistema, no es porque no fuese necesaria la ejecución de la fase de pruebas, pues aún cuando no se contaba con pruebas definidas para cada una de las etapas, se pudo lograr un excelente sistema, que cumple brillantemente con los objetivos y requerimientos que se esperaban del mismo. Solo resta esperar que su aplicación de ahora en adelante mejore la calidad en el proceso de desarrollo de SMA.

CONCLUSIONES

CONCLUSIONES

CONCLUSIONES

Durante el diseño de la metodología de pruebas integrada a PASSI para el desarrollo de sistemas multiagentes, y después de haber realizado su integración con fases conocidas de PASSI obteniendo la metodología "MEDSIM-PASSI", se han arribado a las siguientes conclusiones:

- ✓ Se realizó el diseño de la metodología de pruebas integrada a PASSI para sistemas multiagentes.
- ✓ Se obtuvo como resultado una metodología con 10 fases y la ejecución de 7 pruebas a lo largo del proceso.
- ✓ Se aplicó la metodología obtenida al SMA "Diagnóstico Presuntivo de Enfermedades Ginecológicas" (DPEG).
- ✓ En la validación realizada se comprobó que hubiese sido necesaria dicha metodología para un mejor producto final luego de la realización de DPEG.

RECOMENDACIONES

RECOMENDACIONES

RECOMENDACIONES

Una vez concluido el diseño de la metodología "MEDSIM-PASSI", para el desarrollo de sistemas multiagentes integrados a PASSI, nos permitimos recomendar:

- ✓ Poner en uso la metodología "MEDSIM-PASSI" en el futuro diseño de sistemas multiagentes integrados a PASSI.
- ✓ Generalizar el contenido de este trabajo, específicamente su resultado principal, utilizando la metodología resultante del mismo con fines para los que fue diseñada en cualquier lugar de nuestro país y/o fuera de él.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- [1] J. J. A. Arias S. Francisco J., Ovalle Demetrio A. "Una Aproximación Metodológica para la Construcción de Sistemas Tutoriales Adaptativos Multiagente con Énfasis en el Modelo Pedagógico.," Revista Avances en Sistemas e Informática., vol. 4., 2007.

- [2] M. M. Espino., "Metodologías Orientadas a Agentes: un estudio comparativo.," Tesis de Maestría., Centro de Estudios de Ingeniería de Sistemas., Instituto Superior Politécnico José Antonio Echeverría., Ciudad de la Habana., 2006.

- [3] F. J. Gallego Durán, Faraón Llorens Lar, y Ramón Rizo Aldeguer., "Breve análisis de algunas metodologías de diseño de SMA.," 2008.

- [4] O. J. Garavito Oscar, Torres Miguel. , "Metodología de Pruebas para Sistemas MultiAgentes (SMA) integrada a AOPOA.," Grupo de Investigación SIDre Facultad de Ingeniería Universidad Javeriana Colombia., 2008.

- [5] T. J. M. y. A. J. García., "Metodologías de desarrollo de sistemas multiagentes: un análisis comparativo.," 2002.

- [6] P. M. J. Gómez Sanz Jorge "Desarrollo de Sistemas Multi-Agente. La metodología INGENIAS.," Dpto. de Sistemas Informáticos y Programación., 2004.

- [7] D. D. Gómez., "Sistema Multiagente para el Diagnóstico Presuntivo de Enfermedades Ginecológicas.," Departamento de Informática., Universidad de Cienfuegos., Cienfuegos., 2010.

BIBLIOGRAFÍA

- [8] B. y. P. G. Henderson Sellers. (2008, Appropriate methodologies for constructing agent-oriented systems. Available: www.open.org.au/Conferences/oopsla2004/AO.html
- [9] C. P. M. Cossentino, "PASSI: a Process for Specifying and Implementing Multi-Agent Systems Using UML." 2002.
- [10] P. M. J. L. Mamani Aliaga Jaime Ysaac, "Agentes Inteligentes.," 2006.
- [11] C. Massimo., "PASSI (Process for Agent Societies Specification and Implementation)." ICAR-Italian National Research Council., 2003.
- [12] J. P. Mestres. (2008., Modelos y Arquitecturas de Agentes. Available: <http://grasia.fdi.ucm.es>
- [13] P. C. Morales., "Ingeniería de Software Orientada a Agentes (ISOA)." Grupo Web de Agentes Inteligentes., 2008.
- [14] R. A. Moreno Espino Mailyn, Garrido Maray, García Lynette, Pérez Rolando, Bardají Yasser L., Britos Raycos., "Comparación teórica y en un caso de estudio de Metodologías Orientadas a Agentes.," Centro de Estudios de Ingeniería de Sistemas (CEIS). ISPJAE., 2008.
- [15] J. J. G. Sanz., "Metodologías para el desarrollo de sistemas multiagente.," Revista Iberoamericana de Inteligencia Artificial., vol. 18., 2003.
- [16] A. Sturm, O. Shehory, y D. Dori. "Methodologies evaluation." 2005.